Computer Vision and Image Understanding 221 (2022) 103446

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

SIFNet: Free-form image inpainting using color split-inpaint-fuse approach

S.M. Nadim Uddin, Yong Ju Jung *

School of Computing, Gachon University, South Korea

ARTICLE INFO

Communicated by Nikos Paragios

Keywords:
Image inpainting
Convolutional neural network
Generative adversarial networks
Attention mechanisms
Color space decomposition

ABSTRACT

Recent deep learning-based approaches have shown outstanding performance in generating visually plausible and refined contents for the missing regions in free-form image inpainting tasks. However, most of the existing methods employ a coarse-to-refine approach where the refinement process depends on a single coarse estimation, often leading to texture and structure inconsistencies. Though several existing methods focus on incorporating additional inputs to mitigate this problem, no learning-based studies have investigated the effects of decomposing input corrupted image into luma and chroma images and performing decoupled inpainting of the decomposed components. To this end, we propose a Split-Inpaint-Fuse Network (SIFNet), an end-to-end two-stage inpainting approach that uses a split-inpaint sub-network for separately inpainting the corrupted luma and chroma images using two decoupled branches in the coarse stage and a fusion sub-network for fusing the inpainted luma and chroma images into a refined image in the refinement stage. Additionally, we propose two attention mechanisms for the coarse stage – a progressive context module to find the patch-level feature similarity for the luma image reconstruction and a spatial-channel context module to find important spatial and channel features for the chroma image reconstruction. Experimental results reveal that our Split-Inpaint-Fuse approach outperforms the existing inpainting methods by comparative margins. In addition, extensive ablation studies confirm the effectiveness of the proposed approach, constituting modules and architectural choices.

1. Introduction

Since the emergence of the first digital image inpainting or hole-filling approach (Bertalmio et al., 2000), it has been regarded as one of the most researched yet intriguing ill-posed problems of the computational photography domain. In the case of rectangular masks, the inpainting methods learn to propagate information from surrounding regions into the missing regions in a deterministic manner and it is intuitive to model information propagation based on the mask characteristics (e.g., spatial discounting–providing progressively more weights to the mask borders (Yu et al., 2018)). Although rectangular holes are difficult to inpaint due to large missing areas, free-form or irregular masks possess more challenges because of the non-deterministic characteristics in terms of shape, size, or location, making it difficult to devise a generalized mask model. So, free-form image inpainting methods face three main challenges to tackle – (a) generate realistic contents for the missing regions, (b) maintain texture and structure consistencies with non-corrupted regions, and (c) handle non-deterministic nature of the damaged area.

Traditional image inpainting methods (Weickert, 1999; Bertalmio et al., 2000; Efros and Freeman, 2001; Ballester et al., 2001; Esedoglu and Shen, 2002; Bertalmio et al., 2003; Drori et al., 2003; Criminisi et al., 2004) are generally inefficient for generating novel contents or fill in larger holes with background-consistent contents. As a result, most of the early inpainting methods adopted learning-based approaches, e.g., convolutional neural network (CNN) based approaches, as the de-facto technique for the inpainting tasks. Despite the ability to generate novel contents for large holes, CNN-based image inpainting techniques tend to generate blurry contents, boundary artifacts, and unrealistic contents. To mitigate this, recent inpainting researches have leaned towards additional adversarial supervision, i.e., generative adversarial networks (GANs) (Goodfellow et al., 2014) coupled with CNN models to generate more refined and visually aesthetic results (Pathak et al., 2016; Iizuka et al., 2017a; Yang et al., 2017; Yan et al., 2018; Song et al., 2018a; Yu et al., 2018, 2019; Zheng et al., 2019; Hong et al., 2019; Ren et al., 2019; Yu et al., 2020; Li et al., 2020; Uddin and Jung, 2020; Wadhwa et al., 2021).

Recent free-form image inpainting approaches employ different strategies such as explicit losses (Wang et al., 2018), additional inputs (Nazeri et al., 2019; Ren et al., 2019), attention mechanisms to provide the feature similarity information for feature-level reconstructions (Yu et al., 2019; Zheng et al., 2019; Uddin and Jung, 2020; Wadhwa et al., 2021), explicit normalization (Yu et al., 2020), mask update mechanisms (Yu et al., 2019; Liu et al., 2018a; Uddin and Jung, 2020) etc. for generating missing pixel values. Most prominently, Yu et al. (2019, 2018) and Uddin and Jung (2020) proposed a coarse-to-refine network where a coarse stage predicts a "coarse" inpainted image,

* Corresponding author.
E-mail address: yjung@gachon.ac.kr (Y.J. Jung).

https://doi.org/10.1016/j.cviu.2022.103446
Received 9 September 2021; Received in revised form 18 February 2022; Accepted 10 May 2022
Available online 26 May 2022

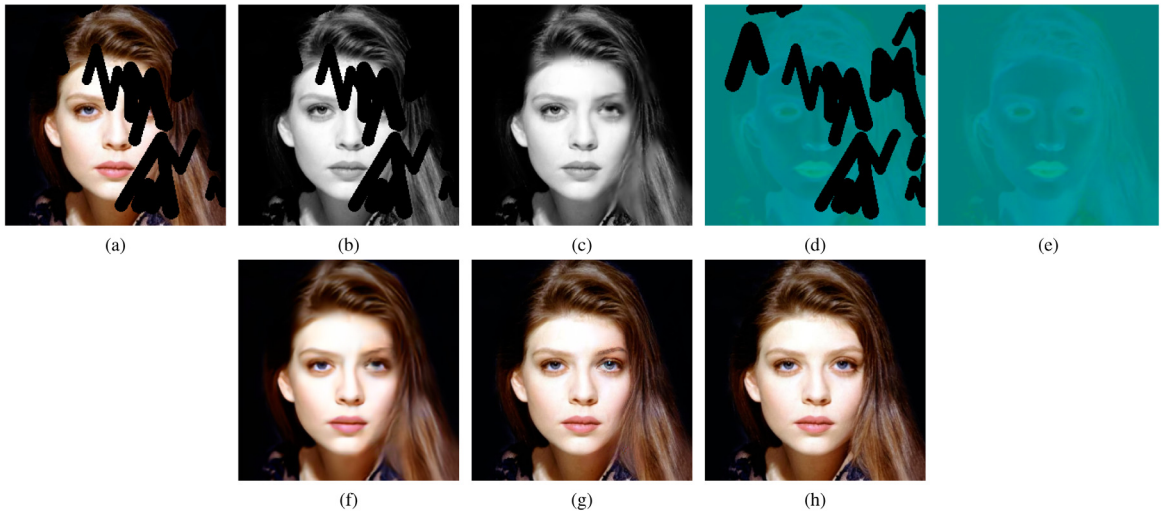

Fig. 1. Our Split-Inpaint-Fuse network (SIFNet) takes a corrupted image as input and produces a realistic inpainted image by utilizing a two-stage inpainting strategy. In the first stage, the SIFNet splits the input into a luma and a chroma component and inpaints them separately to produce inpainted luma and chroma images using a split-inpaint sub-network. The inpainted luma and chroma images are concatenated to produce a coarse inpainted image. In the second stage, coarse inpainted image is fed into a fusion sub-network to produce a fused and refined inpainted image. From left, (a) input corrupted image, (b) input luma component, (c) inpainted luma component (d) input chroma component, (e) inpainted chroma component, (f) coarse inpainted image, (g) refined inpainted image, and (h) ground truth.

which is refined in a refinement network. Also, Nazeri et al. (2019) proposed a two-stage network where the first stage hallucinates missing edges and the second stage hallucinates the texture. These approaches work plausibly well with most free-form cases.

In the coarse-to-refine approaches, a coarse image is first estimated and then a refinement network refines the coarse image into the final inpainted image. The coarse image acts as the basis for the refinement process. Hence, the coarse texture and structure will be propagated and refined in the refinement process e.g., using explicit attention mechanisms. Most of the attention mechanisms either depend on utilizing local feature-level similarities (e.g., Yu et al. (2018, 2019), Uddin and Jung (2020)) or global feature-level similarities (e.g., Zheng et al. (2019), Uddin and Jung (2020)) to find the most contributing feature values. However, in both cases, the feature values come from the coarse estimation. If the coarse reconstruction fails to correctly recover texture and/or structure, the refined results will have visual artifacts in terms of structure, texture, and color. This is owing to depending only on a *single* coarse color image for the refinement process can lead to inconsistent textures and structures among the inpainted and surrounding regions.

Our intuition is that an inpainting network can generalize better if provided with explicit prior information (e.g., decouple images into their constituting aspects such as luma-chroma, texture-structure, etc.) during the coarse estimation and re-use the decoupled information for the refinement process. To elaborate this intuition, let us consider two cases:

1. **De-correlated information processing:** There are many cases of natural scene images where the image structures and textures do not exactly match between the luma and chroma information. For example, luma can vary a lot as the lighting environment changes, while chroma will remain relatively more constant as lighting changes (disregarding colored lights and color bleeding), but chroma varies for other causal reasons such as varying pigmentation in the skin, vegetation, and building materials, etc. Hence, one possible strategy is that the luma and chroma inputs are separately recovered for better structure, texture, and color in the coarse network. Instead of generating a single coarse output, the coarse network outputs two coarse images (i.e., one for luma image and another for chroma image).
2. **Correlated information processing:** However, it is also true that, in many cases, the structure and texture information exactly matches between the luma and chroma images (e.g., same luma and chroma edges of an object). For this reason, a second stage can be further used to fuse the two coarse luma, and chroma results recovered in the first stage (i.e., in the coarse network). In summary, a two-stage inpainting approach can be used to inpaint a luma (i.e., intensity) image and a chroma image separately using two branches in the first stage network and fuse the recovered luma and chroma images using the second stage network. In this way, the model can learn to hallucinate the luma and chroma information separately and then learn to fuse and refine the structure, texture, and color for a better inpainted image.

Hence, we hypothesize that, by using an explicit Split-Inpaint-Fuse (SIF) mechanism, a deep learning-based inpainting network can learn individual de-correlated semantics in the initial stages and correlated semantics in the later stages, providing the network comparatively greater generalization ability. This intuition has been well explored in other computer vision tasks (e.g., intrinsic image decomposition, image enhancement, image-to-image translation, etc.) which have leveraged such decoupled information as priors and obtained significant performance improvements. From our initial experiments with color spaces (e.g., Fig. 9, Table 3), we have found that the SIF mechanism can generalize comparatively faster and better than those without the SIF mechanism, with the same amount of training time, which validates our initial hypothesis.

In this paper, we propose a SIFNet (i.e, Split-Inpaint-Fuse Network) for image inpainting that employs a two-stage network, namely a split-inpaint sub-network and a fusion sub-network. The network first decomposes the corrupted input image into a luma and a chroma image. The split-inpaint sub-network separately inpaints the corrupted luma and chroma images using two decoupled branches (i.e., a Luma inpainting branch and a Chroma inpainting branch) in the first stage. Then, the fusion sub-network takes the coarse inpainted luma image & chroma image and fuses them to produce the final inpainted image. Fig. 1 shows the outputs from the different sub-networks in the proposed SIFNet. Additionally, we propose two attention mechanisms to facilitate the luma and chroma predictions in a separate manner. Specifically, we propose a patch-level attention mechanism, namely "Progressive Context Module (PCM)" for finding the missing structures and coarse textures in the Luma branch and "Spatial-Channel Context Module (SCCM)" for incorporating both spatial and channel attention in the Chroma branch. During the training, the inpainted images are

evaluated using two discriminators, namely a pixel discriminator and a patch discriminator, to supervise the content generation process in both patch-level and pixel-level. To the best of our knowledge, this

The summary of our contributions is listed as follows:

- We propose SIFNet - a novel free-form image inpainting approach that decomposes a corrupted image into luma and chroma images, inpaints them separately (i.e., using a Luma branch and a Chroma branch), and fuses the coarsely inpainted luma and chroma into a refined output.
- We propose a progressive context module (PCM) for the Luma branch that performs the patch-level attention mechanism to find the most contributing and similar patches in two stages, ensuring a robust attention mechanism to find correspondences among similar patches for the feature reconstruction.
- We also propose a spatial-channel context module (SCCM) for the Chroma branch that performs both spatial and channel attentions and incorporates both attended information via learnable parameters.

The proposed SIFNet has been evaluated with several state-of-the-art methods for irregular-sized holes on two popular datasets used for image inpainting tasks, namely Places365 (Zhou et al., 2017) and CelebA-HQ (Karras et al., 2018). In the experiments, the qualitative and quantitative results reveal that our model outperforms the existing models. Additionally, extensive ablation studies have been performed to show the effectiveness of the proposed modules and the performance gain of the proposed approach.

This paper is organized into six sections. Section 2 discusses related studies on both traditional and learning-based image inpainting methods. Section 3 describes the proposed inpainting model along with the detailed descriptions of the two proposed attention modules. Section 4 outlines the experimental setups and provides the comparison results with the existing methods and a description of the feasibility of the proposed modules. Section 5 discusses the summary and possible future directions of the proposed approach. Section 6 provides the concluding remarks.

2. Related work

2.1. Traditional image inpainting

Traditional non-learning based methods focus on information propagation from hole boundaries, copying similar patches from the background, or using exemplar images to find similar patches (Weickert, 1999; Bertalmio et al., 2000; Efros and Freeman, 2001; Ballester et al., 2001; Esedoglu and Shen, 2002; Bertalmio et al., 2003; Drori et al., 2003; Criminisi et al., 2004; Levin et al., 2003; Sun et al., 2005; Simakov et al., 2008; Barnes et al., 2009; Xu and Sun, 2010; Darabi et al., 2012; Huang et al., 2014). These methods work plausibly well with smaller holes, texts, or scratches. However, these methods do not work well with non-stationary textures such as natural scenes and complex images requiring the generation of novel contents. Additionally, traditional methods are slower in inference, making them difficult for real-time inpainting tasks. With the emergence of CNNs for content generation (LeCun et al., 1998; Krizhevsky et al., 2012) and GANs (Goodfellow et al., 2014) for adversarial supervision, a major shift of interest has been directed toward learning-based image inpainting methods.

2.2. Learning-based image inpainting

Early deep learning-based approaches (Pathak et al., 2016; Iizuka et al., 2017a; Yang et al., 2017; Yan et al., 2018; Song et al., 2018a; Yu et al., 2018; Yeh et al., 2017; Li et al., 2017; Song et al., 2018b; Yang et al., 2018; Zhang et al., 2018a; Zhao et al., 2019; Dolhansky and Canton Ferrer, 2018; Wang et al., 2018; Yu et al., 2018) mostly focus on rectangular holes for image inpainting tasks. These methods generate plausible inpainting results by learning the semantics of the data based on extensive reference images for the learning process. However, image inpainting tasks in real applications generally contain free-form or irregular holes that require different optimization processes or 'attention mechanisms' compared with rectangular holes. As a result, research interests have shifted more toward free-form image inpainting.

Recent approaches (Yu et al., 2019; Liu et al., 2018a; Zheng et al., 2019; Nazeri et al., 2019; Hong et al., 2019; Ren et al., 2019; Liu et al., 2020; Yu et al., 2020; Li et al., 2020; Uddin and Jung, 2020; Wadhwa et al., 2021) have focused more on irregular-shaped holes because they are more frequent in real cases and often prone to texture inconsistencies. Most of the newer methods incorporate mask update processes (Yu et al., 2019; Liu et al., 2018a; Uddin and Jung, 2020), attention mechanisms (Yu et al., 2019; Zheng et al., 2019; Uddin and Jung, 2020) or additional inputs (Ren et al., 2019; Liu et al., 2020; Nazeri et al., 2019). Wang et al. (2018) proposed a three-branch encoder (with varying convolutional kernel sizes) for feature extraction. Liu et al. (2018a) proposed the partial convolution that performs heuristic mask updates based on the total number of mask/non-mask values in a convolution filter grid. Zeng et al. (2019) proposed a pyramid context encoder that incorporates a multi-layer attention transfer mechanism. Yu et al. (2019) proposed a soft gating mechanism-based mask update method for irregular masks. Nazeri et al. (2019) predicted an edge map and used it as an additional input for the texture generation. Yu et al. (2020) used different normalization techniques for background and mask regions. Li et al. (2020) proposed a progressive content generation process. Uddin and Jung (2020) proposed an explicit mask value pruning mechanism and a global-local attention mechanism. Wadhwa et al. (2021) utilized hyper-graphs to find the most contributing features for feature reconstruction.

Though most of the methods work well with certain free-form masks, they fail to maintain structure or texture if the characteristics of the masks are different or uncertain (Ntavelis et al., 2020). Moreover, most of the methods fail to maintain color consistencies in most of the complex scenarios.

2.3. Color space decomposition-based image inpainting

Though most of the traditional approaches performed inpainting tasks by decomposing images into constituting segments (e.g. texture and structure images, low-frequency and high-frequency images, etc.), no learning-based approaches have examined the effect of luma-chroma decomposition in image inpainting. It has been shown in other computer vision tasks that color space decomposition approach is highly effective and generalizes comparatively faster, e.g., hint-based colorization can be regarded as predicting mission color values (Leung et al., 2011; Zhang et al., 2017; Sharif and Jung, 2019; Jang and Jung, 2020; Zhang et al., 2016; Iizuka et al., 2016; Vitoria et al., 2020; Su et al., 2020). The closest work to our approach is of Liu et al. (2020). Specifically, Liu et al. (2020) uses a single encoder–decoder architecture where the input is fed to a series of convolutions to obtain both shallow and deep features which are denoted as 'texture features' and 'structure features', respectively. This implicit decoupling mechanism is inherently different from our proposed method. Instead of the *implicit* decoupling of features, our proposed method explicitly decomposes the input into luma and chroma images, which are inpainted separately using two decoupled branches in the split-inpaint sub-network. By doing so, we can use different attention modules for the two branches that can be beneficial for a better inpainted image. To the best of our knowledge, no method has explicitly leveraged the luma-chroma decomposition and a split-inpaint-fuse approach for the inpainting tasks.

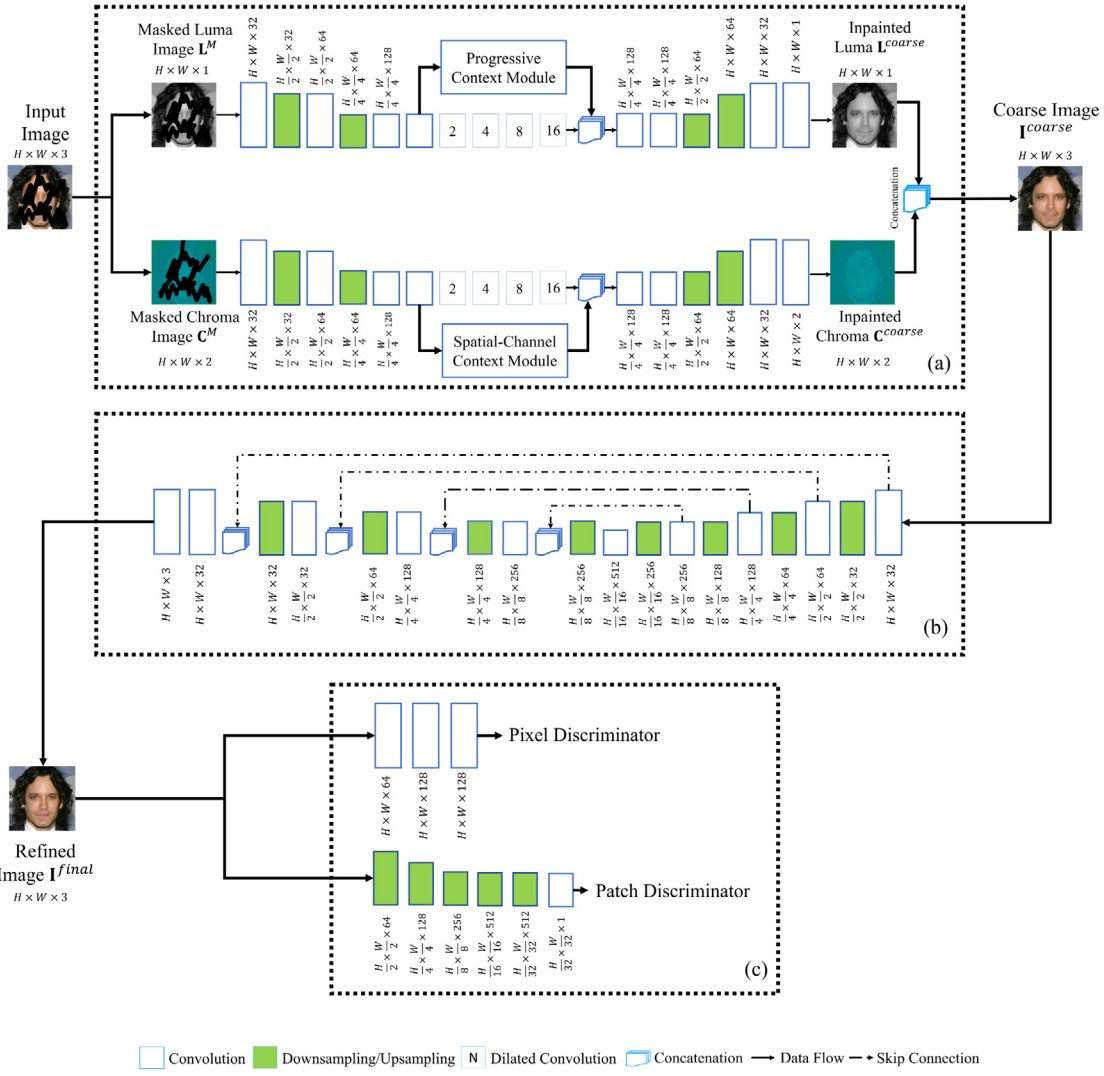

Fig. 2. Overview of the proposed SIFNet framework. The SIFNet consists of three components - (a) split-inpaint sub-network, (b) fusion sub-network, and (c) discriminator network. The split-inpaint sub-network decomposes the corrupted input image into a luma image and a chroma image and inpaints them concurrently using two branches. The resultant coarse luma and chroma images are then concatenated and fed to a fusion sub-network that fuses the coarse images and refines them into the final inpainted image. During training, the final inpainted image is evaluated using two discriminators, namely a pixel discriminator and a patch discriminator. The numbers up/below each layer represent the dimensions of the output feature maps.

3. Proposed framework

3.1. Overview of the SIFNet

The proposed SIFNet follows a GAN framework (Goodfellow et al., 2014), as shown in Fig. 2. Specifically, the SIFNet has a generator network that reconstructs the corrupted image and a discriminator network with two discriminators that validate the consistency of the reconstructed image.

We first convert the input RGB image $\mathbf{I}^{RGB}$ to a CIELAB image $\mathbf{I}^{LAB}$. Note that we have used the CIELAB color space owing to its perceptually uniform characteristics and similarity to the human visual system (Robertson et al., 1977; Cohen et al., 1968; Schwarz et al., 1987). However, we also show that any perceptually uniform color space (e.g. LUV, YCbCr, etc.) can be used for the split-inpaint sub-network (see ablation studies–Section 4.4).

The $\mathbf{I}^{LAB}$ is decomposed into a luma image $\mathbf{I}^{luma}$ and a chroma image $\mathbf{I}^{chroma}$. Here, $\mathbf{I}^{luma}$ contains L channel and $\mathbf{I}^{chroma}$ contains two channels–A and B channels. Then we multiply the input binary mask $\mathbf{M}$ with the luma image to obtain a corrupted luma image $\mathbf{L}^M$ (i.e., $\mathbf{L}^M = \mathbf{I}^{luma} \odot \mathbf{M}$). Similarly, we multiply the input binary mask $\mathbf{M}$ with the chroma image to obtain a corrupted chroma image $\mathbf{C}^M$ (i.e., $\mathbf{C}^M = \mathbf{I}^{chroma} \odot \mathbf{M}$). The $\mathbf{L}^M$ and $\mathbf{C}^M$ images are normalized to $[-1, 1]$. The inputs are then fed to a generator network to construct inpainted luma and chroma images.

The generator consists of a split-inpaint sub-network and a fusion sub-network. The split-inpaint sub-network consists of two branches - a Luma inpainting branch and a Chroma inpainting branch. The Luma branch inpaints the $\mathbf{L}^M$ to obtain an inpainted luma image $\mathbf{L}^{coarse}$ by utilizing a progressive context module that leverages inner product similarities to find similar regions in the luma image in a progressive manner. Similarly, the Chroma branch inpaints the $\mathbf{C}^M$ to obtain an inpainted chroma image $\mathbf{C}^{coarse}$ by utilizing a spatial-channel context module that seeks long-range feature dependencies for a consistent chroma image reconstruction. We construct the coarse inpainted image $\mathbf{I}^{coarse}$ by concatenating $\mathbf{L}^{coarse}$ and $\mathbf{C}^{coarse}$ in a channel-wise manner. Please refer to Section 3.2 for the detailed description of the split-inpaint sub-network. The $\mathbf{I}^{coarse}$ from the split-inpaint sub-network is then fed to the fusion sub-network, which fuses the information from $\mathbf{L}^{coarse}$ and $\mathbf{C}^{coarse}$ to produce a refined inpainted image $\mathbf{I}^{final}$.

The fusion sub-network utilizes a U-Net-based encoder–decoder architecture with symmetric skip connections between the encoder and

decoder layers to ensure the propagation of both low-level and high-level information (Ronneberger et al., 2015). The refined inpainted CIELAB image $\mathbf{I}^{final}$ from the fusion sub-network is then converted back to an RGB image to ensure the color space consistency between the input and output image. Please refer to Section 3.3 for the detailed description of the fusion sub-network.

For both sub-networks in the generator, we use Leaky ReLU activation (Maas et al., 2013b) between the convolutions, except the last layer where we use hyperbolic tangent function to map the activation values in $[-1, 1]$.

Discriminator network. During the training, the refined inpainted image is evaluated using two discriminators – a pixel discriminator and a patch discriminator – to ensure global and local consistencies among the inpainted regions and background regions of the image. During the inference, only the trained generator is used to inpaint the missing regions. Please refer to Section 3.4 for the detailed description of the discriminator network.

3.2. Split-inpaint sub-network

The split-inpaint sub-network consists of two branches, namely the Luma branch and the Chroma branch. Both the Luma branch and Chroma branch follow an identical encoder–decoder architecture with a corresponding attention module suited for each architecture.

As seen in Fig. 2, the Luma branch and Chroma branch take the corrupted luma image $\mathbf{L}^M$ and the corrupted chroma image $\mathbf{C}^M$, with the corresponding binary mask $\mathbf{M}$, as inputs. Following a series of convolution layers and down-sampling operations, the obtained feature maps are fed concurrently to two segments within the encoder – a block of dilated convolutions and the respective attention modules (i.e. PCM in the Luma branch and SCCM in the Chroma branch)– to obtain two sets of feature maps that are concatenated and fed to the decoder. The dilated convolutions (i.e. rate of 2, 4, 8, and 16) are shown to work plausibly for capturing larger receptive fields, which are crucial for a better inpainted image (Iizuka et al., 2017b; Yu et al., 2019; Uddin and Jung, 2020).

Both feature maps (i.e. from the dilated convolutions and the attention modules) act as complementary feature maps to each other (e.g. in terms of texture and structure) and provide rich feature values for the inpainting task. The decoders reconstruct the $\mathbf{L}^{coarse}$ and $\mathbf{C}^{coarse}$ concurrently. The inpainted images are then concatenated to obtain a coarse inpainted image $\mathbf{I}^{coarse}$.

3.2.1. Progressive Context Module - PCM

Reconstructing the luma image (i.e. intensity image) requires both structure and gray texture information. So, we propose a Progressive Context Module (PCM) in the Luma branch that incorporates a progressive feature reconstruction mechanism using local patch-wise similarities. Instead of relying on finding the most similar information in one stage (Yu et al., 2018) or explicitly pruning mask values (Uddin and Jung, 2020), we propose to find the most similar information in two stages–(i) stage 1–first reconstruct the masked region of the down-sampled feature map, and (ii) stage 2–use the reconstructed feature as a guidance for reconstructing the original feature map, utilizing a progressive coarse-to-refine approach to find the most contributing context information for a better inpainted image.

Stage 1. As seen in Fig. 3, the PCM takes an intermediate feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ as input, where H, W, and C denote the height, width, and channel, respectively. The PCM first down-samples $\mathbf{X}$ to $\hat{\mathbf{X}} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$ and produces a masked feature map $\hat{\mathbf{X}}^M$ and a non-masked feature map $\hat{\mathbf{X}}^{NM}$ using the binary mask $\mathbf{M}$, as follows,

$$\hat{\mathbf{X}}^M = \hat{\mathbf{X}} \odot \hat{\mathbf{M}}, \tag{1}$$

$$\hat{\mathbf{X}}^{NM} = \hat{\mathbf{X}} \odot (1 - \hat{\mathbf{M}}), \tag{2}$$

where $\odot$ denotes a Hadamard product and $\hat{\mathbf{M}}$ is obtained by resizing $\mathbf{M}$ to match the spatial size of $\hat{\mathbf{X}}$ using the nearest neighbor interpolation.

The PCM performs the attention-based feature reconstruction in two steps – (a) finding a softmax-normalized inner product similarity map $\mathbf{S}$ between the masked and non-masked feature, and (b) using the similarity score map to find the most similar non-masked features for reconstructing the masked features.

To do so, PCM first extracts N number of $k \times k \times C$ overlapping patches from the non-masked region. These patches are used as convolution kernels to efficiently calculate the inner product with the masked region (Yu et al., 2018, 2019; Uddin and Jung, 2020). Let $\mathbf{P}_{i,j,i',j'}$ be the output of the inner product between a patch centered at (i, j) spatial location in $\hat{\mathbf{X}}^M$ and a patch centered at (i', j') spatial location in $\hat{\mathbf{X}}^{NM}$,

$$\mathbf{P}_{i,j,i',j'} = \langle \frac{\hat{\mathbf{X}}^M_{i,j}}{\|\hat{\mathbf{X}}^M_{i,j}\|}, \frac{\hat{\mathbf{X}}^{NM}_{i',j'}}{\|\hat{\mathbf{X}}^{NM}_{i',j'}\|} \rangle. \tag{3}$$

It is then normalized into a similarity score map $\mathbf{S} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times N}$ using a softmax operation. This similarity score map $\mathbf{S}$ contains the similarity weights for each pixel location i.e., if a $k\times k$ non-masked patch is similar to a $k \times k$ masked region, the patch will have higher weights for that masked region. $\mathbf{S}$ is obtained by

$$\mathbf{S}_i = \frac{e^{\mathbf{P}_i}}{\sum_{j=1}^{N} e^{\mathbf{P}_j}}. \tag{4}$$

The PCM then samples the most similar patches based on $\mathbf{S}$ and reconstructs an attended feature map a feature map $\hat{\mathbf{X}}^R \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$. This reconstruction operation can be efficiently performed using a transposed convolution operation. Specifically, in a transposed convolution, each $k\times k$ convolution kernel is multiplied with each pixel value from the input (Noh et al., 2015) . The final output from a transposed convolution is obtained by summing all the outputs for each pixel location, denoting the contribution of all kernels for each location. If a kernel is multiplied with a higher pixel value, the pixel location in the resultant feature map will have higher contribution from that kernel.

Utilizing this principle, we employ a transpose convolution to efficiently reconstruct the masked region. During the transposed convolution, all patches will by multiplied with the similarity weights for each pixel location in $\mathbf{S}$. As the convolution sums the resultant output maps, each location in the resultant feature map in $\hat{\mathbf{X}}^R$ will have weighted feature values according to the similarity score map $\mathbf{S}$, meaning that the each location in the masked regions will be reconstructed using the similar non-masked region values.

Stage 2. The PCM up-samples the reconstructed feature map $\hat{\mathbf{X}}^R$ to the original size $\mathbf{X}^U$ (i.e. $\mathbb{R}^{H\times W\times C}$). Similar to the operation performed on the $\hat{\mathbf{X}}$, the PCM samples the most similar patches among $\mathbf{X}^{UM} = \mathbf{X}^U \odot \mathbf{M}$ and $\mathbf{X}^{NM} = \mathbf{X} \odot \mathbf{M}$ to reconstruct another feature map $\mathbf{X}^R \in \mathbb{R}^{H\times W\times C}$. Note that, in this case, the PCM does not decomposes $\hat{\mathbf{X}}^R$ into a masked and non-masked feature map. Rather, it extracts a non-masked feature map $\mathbf{X}^{NM}$ from the input feature $\mathbf{X}$. Both $\mathbf{X}^U$ and $\mathbf{X}^R$ are then fed to a convolution layer to obtain a final reconstructed feature map $\mathbf{X}^A$.

The PCM calculates the contextual information from two scales and utilizes it to progressively reconstruct features from the most similar non-masked patches. The intuition behind such design is to maximize the chances of finding the most similar patches in both down-sampled and original scales. This learnable strategy enables the module capable of handling masks with different characteristics and minimizing boundary inconsistency problems during the reconstruction of feature maps. As seen in the attention visualization from Fig. 4, the PCM focuses on finding the most similar patches to reconstruct structurally consistent features in both scales, leading to a better inpainted image.

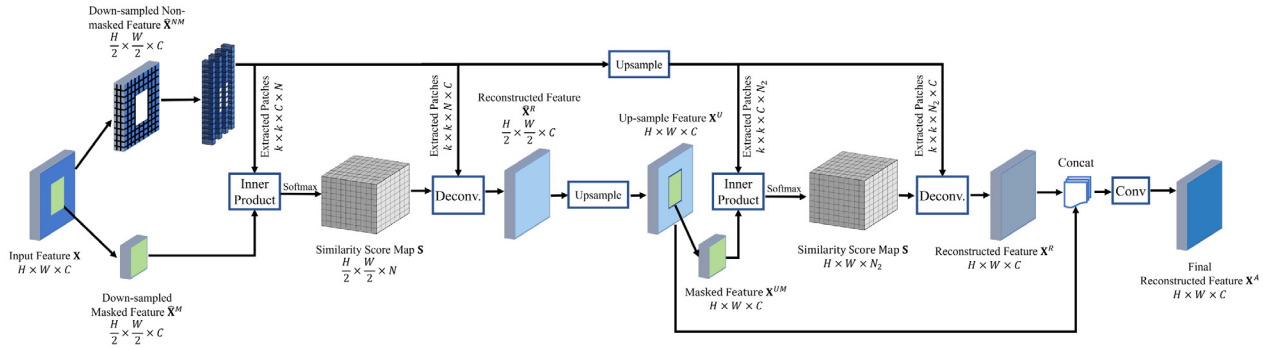

Fig. 3. Overview of the proposed Progressive Context Module (PCM). The PCM first searches for similar patches in the down-sampled feature map and reconstructs a feature map based on the similarity score. Then it up-samples the reconstructed feature map and again searches for similar patches to reconstruct another feature map. Both of the reconstructed feature maps are then fed to a convolution layer to obtain the final attended feature map. The two-stage strategy ensures the coverage of larger receptive areas in a coarse-to-refine manner to find the most contributing features for the reconstruction of the final attended feature.

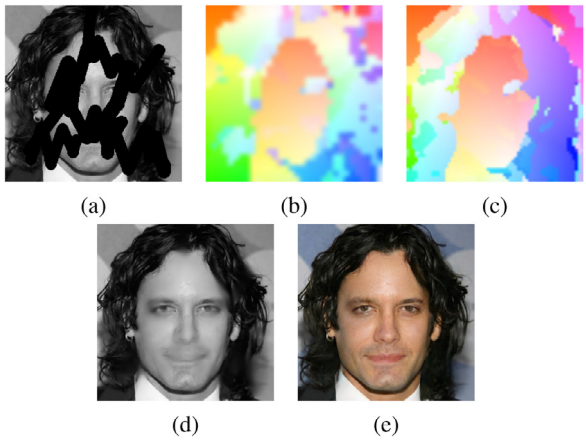

Fig. 4. Attention visualization of the proposed Progressive Context Module. (a) corrupted luma image, (b) attention flow of the first stage (up-scaled for visualization), (c) attention flow of the second stage (up-scaled for visualization), (d) predicted luma image, and (e) final inpainted image. Owing to the progressive contextual feature searching mechanism, the predicted luma image contains better coarse structure and texture that contribute to a better inpainted image.

3.2.2. Spatial-Channel Context Module - SCCM

The Chroma branch reconstructs the color information for the missing regions. However, reconstructing the color information is difficult comparing with reconstructing structures, as missing regions can have different colors in a small segment. Hence, it is important to consider both spatial and channel contributions in a feature map to ensure structure and texture consistencies. So, we propose a Spatial-Channel Context module (SCCM), integrated into the Chroma branch, that calculates both spatial and channel contributions among the values from a feature map and integrates the contributions in a learnable manner. Fig. 5 shows the overview of the proposed SCCM.

Specifically, let $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ be an intermediate feature map obtained from a convolution layer where H, W, and C denote height, width and channels, respectively. For calculating spatial attention on the input feature map $\mathbf{X}$, the SCCM first performs three 1×1 convolutions to learn the transformation of the feature $\mathbf{X}$. Then the SCCM calculates a global correlation map, ρ, among the feature values in $\mathbf{X}$ as,

$$\rho = \omega_1(\mathbf{X}) \otimes \omega_2(\mathbf{X})^T, \tag{5}$$

where $\otimes$ denotes a matrix multiplication and $\omega(\cdot)$ is a 1×1 convolution. The global correlation map, $\rho \in \mathbb{R}^{HW\times HW}$, contains the correlation information among all possible pixel pairs in $\mathbf{X}$. The SCCM performs a softmax operation on ρ to highlight the most contributing correlations among the pixel pairs and performs another matrix multiplication between ρ and $\omega_3(\mathbf{X})$ to produce a spatially weighted feature map $\mathbf{X}^S$,

$$\mathbf{X}^S = \rho \otimes \omega_3(\mathbf{X}). \tag{6}$$

The spatially attended feature $\mathbf{X}^{SA}$ is then obtained by adding $\mathbf{X}$ with the $\mathbf{X}^S$ through a learnable parameter β as,

$$\mathbf{X}^{SA} = \mathbf{X} + \beta \odot \mathbf{X}^S. \tag{7}$$

Here, the learnable parameter β controls the extend of the attended feature $\mathbf{X}^S$ to be added to $\mathbf{X}$.

For the channel re-calibration operation on $\mathbf{X}^{SA}$, let us consider $\mathbf{X}^{SA}$ as a combination of channels $\mathbf{X}^{SA} = [\mathbf{X}_1^{SA}, \mathbf{X}_2^{SA}, \dots \mathbf{X}_C^{SA}]$, where $\mathbf{X}_k^{SA} \in \mathbb{R}^{H\times W}$. First, we perform a global average pooling operation with 1×1 kernel to produce an embedded vector $\mathbf{z} \in \mathbb{R}^{1\times 1\times C}$ which acts as a channel descriptor containing the global spatial information. The kth element of $\mathbf{z}$ is defined as

$$\mathbf{z}_k = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}(\mathbf{X}_k^{SA}(i,j)). \tag{8}$$

Here, $\mathbf{z}_k$ is a scalar value denoting the spatially averaged value of the feature map from kth channel. Following Hu et al. (2018), we perform a transformation operation on $\mathbf{z}$ as

$$\hat{\mathbf{z}} = \sigma(FC_2(\delta(FC_1(\mathbf{z})))), \tag{9}$$

where $FC_1 \in \mathbb{R}^{C\times\frac{C}{r}}$ and $FC_2 \in \mathbb{R}^{\frac{C}{r}\times C}$ are two fully connected layers. Here, r, δ and σ denote the channel bottleneck (i.e., $r = 16$ in SIFNet), ReLU, and Sigmoid operation, respectively. The channel descriptor $\hat{\mathbf{z}}$ is then used to re-calibrate $\mathbf{X}^{SA}$ to obtain channel re-calibrated feature $\mathbf{X}^{CR}$ as,

$$\mathbf{X}^{CR} = [\hat{\mathbf{z}}_1 \odot \mathbf{X}_1^{SA}, \hat{\mathbf{z}}_2 \odot \mathbf{X}_2^{SA}, \dots \hat{\mathbf{z}}_C \odot \mathbf{X}_C^{SA}]. \tag{10}$$

The spatial-channel attended feature $\mathbf{X}^{SC}$ is then obtained by adding $\mathbf{X}$ with $\mathbf{X}^{CR}$ through a learnable parameter γ as

$$\mathbf{X}^{SC} = \mathbf{X} + \gamma \odot \mathbf{X}^{CR}. \tag{11}$$

The SCCM calculates the global contributions in both spatial and channel contexts. The spatial context attention ensures that the color consistency is maintained in the spatial locations among the channels while the channel re-calibration ensures the most contributing color information are propagated throughout the feature map, as seen in Fig. 6. Owing to the integration of both context information in a learnable manner (i.e. β for the spatial contribution and γ for the channel contribution), the network does not heuristically add the information. Rather, it progressively weighs the contributions and adds the most contributing context information to the output, making the network capable of handling the difficult color information.

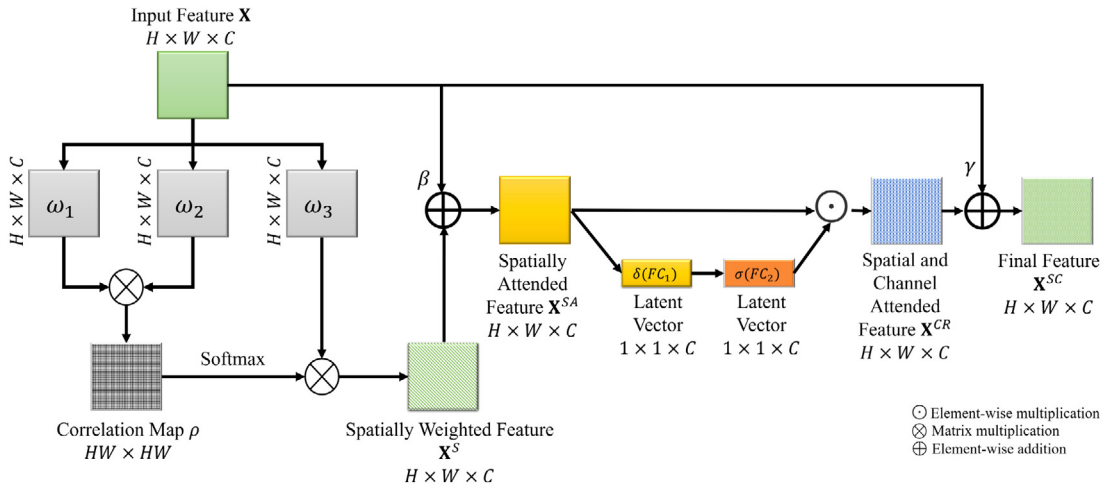

Fig. 5. Overview of the proposed Spatial-Channel Context Module (SCCM). The SCCM first calculates a global correlation map consisting of per-pixel-wise spatially important feature values and generates a correlated feature. This correlated feature is added to the input feature map via a learnable parameter to obtain a spatially attended feature map, ensuring that the model learns to incorporate spatial attention. This attended feature map is then used to calculate the important channel contribution, which is then added to the input feature via another learnable parameter to obtain a spatial-channel attended feature map. With the SCCM, the model learns to incorporate the important spatial and channel information necessary for a texture and structure consistent inpainted image.

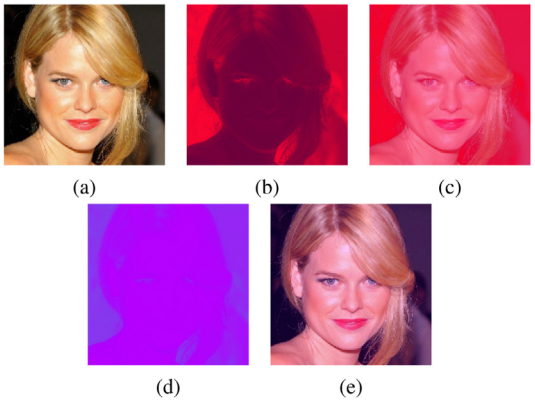

Fig. 6. Visualization of the spatial and spatial-channel contributions in the SCCM. (a) Ground truth image, (b) contribution of the spatial attention mechanism (i.e., correlated feature in Fig. 5), and (c) image obtained from the spatial attention mechanism (i.e. spatially attended feature in Fig. 5), (d) contribution of the spatial-channel context module (i.e., spatial and channel attended feature in Fig. 5), and (e) image obtained from spatial-channel context module (i.e., final feature in Fig. 5). The spatial attention focuses on finding the most contributing attributes in terms of spatial correlations and reconstructs a feature map containing spatially important values. The spatial-channel attention focuses on finding the most contributing attributes in both spatial & channel features and outputs a spatial-channel attended feature map, ensuring a better inpainting process in terms of structure and texture.

3.3. Fusion sub-network

We adopt a U-Net-based symmetric encoder–decoder architecture with skip connections (Ronneberger et al., 2015). The fusion sub-network takes the coarse inpainted image $\mathbf{I}^{coarse}$ and the corresponding binary mask $\mathbf{M}$ as inputs and then outputs a refined inpainted image $\mathbf{I}^{final}$.

The fusion sub-network consists of symmetric down-sampling and up-sampling layers with skip connections between the encoder and decoder. The input coarse image is gradually down-sampled (i.e., $\mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 32}$, $\mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 64}$, and so on) to a bottleneck feature map ($\mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times 512}$) that contains high-level features. The bottleneck feature is then gradually up-sampled with the information fusion from the encoder layers (i.e. via symmetric skip connections among the encoder–decoder layers) to generate a refined output.

The intuition behind using such architecture for fusion is that, as the Luma and Chroma branches produce a coarse inpainted luma image $\mathbf{L}^{coarse}$ and chroma image $\mathbf{C}^{coarse}$ separately, the $\mathbf{I}^{coarse}$ (obtained by a simple concatenation) does not have the mutual structure and color correspondence compared with $\mathbf{I}^{gt}$. Hence, to fuse the coarse luma image and chroma image into a refined inpainted image $\mathbf{I}^{final}$, it is essential to fuse both low-level features and high-level features. Though Yu et al. (2019) points out that the skip connections in U-Net-based architecture can propagate the mask values into the decoder (i.e., causing no available information for the inpainting task), we argue that this is only valid in the coarse stage where the corrupted input image contains mask values. In the case of using the coarse image as input, there are no mask values in the input image and hence, the skip connections in the architecture help to propagate both low-level and high-level features for a refined output.

3.4. Pixel and patch discriminators

In a GAN-based inpainting framework, the generator reconstructs an inpainted image and the discriminator supervises the generation process by outputting the difference between the distributions of real images and generated images. In the case of free-form inpainting, the mask can appear in any shape and location. Employing only a global discriminator or local discriminator cannot guarantee supervision on plausible content generation. The global discriminator examines the generated image as a whole and cannot provide supervision of the content generation at the local level. Though patch-based discriminators or pixel-based discriminators can provide local supervision due to the inherent local characteristics, they cannot supervise the content generation in the case of irregularly distributed masks or masks with different characteristics (e.g., free-form cellural automata (CA) masks used in Ntavelis et al. (2020)). Hence, we propose to utilize two discriminators, namely a pixel discriminator and a patch discriminator, concurrently, to ensure both patch-level and pixel-level consistencies between the inpainted regions and the background regions.

Specifically, both discriminators (i.e. patch discriminator and pixel discriminator) take a generated image $\mathbf{I}^{final} \in \mathbb{R}^{H\times W\times 3}$ and its corresponding ground-truth image $\mathbf{I}^{gt} \in \mathbb{R}^{H\times W\times 3}$ as inputs. In the case of the patch discriminator, the input is fed through a series of strided

convolutions (i.e. stride = 2) to obtain latent feature maps. The patch discriminator evaluates the latent feature maps to supervise the patch-level content generation. In the case of the pixel discriminator, the input is also fed through a series of convolutions to obtain feature maps of the same sizes as $\mathbf{I}^{gt}$ that are used to supervise pixel-level content generation. For both discriminators, we use the leaky ReLU (Maas et al., 2013a) as activation between the convolutions, except the last convolution where we do not employ any activation.

3.5. Objective function

The proposed generator network (i.e., the split and fusion sub-networks) first constructs a coarse image $\mathbf{I}^{coarse}$ and then refines it to reconstruct the final inpainted image $\mathbf{I}^{final}$. As the $\mathbf{L}^{coarse}$ serves as the structural basis for the inpainted image $\mathbf{I}^{final}$, it is required that the $\mathbf{L}^{coarse}$ contains less pixel difference with the ground truth luma image $\mathbf{L}^{gt}$. Again, the reconstructed chroma image $\mathbf{C}^{coarse}$ provides the color information to be propagated to $\mathbf{I}^{final}$, and hence it is also imperative that $\mathbf{C}^{coarse}$ is similar to the characteristics of the ground truth chroma image $\mathbf{C}^{gt}$. So, the objective function for the split-inpaint sub-network, $\mathcal{L}_{SN}$, is as follows:

$$\mathcal{L}_{SN} = \mathcal{L}_1(\mathbf{L}^{coarse}, \mathbf{L}^{gt}) + \mathcal{L}_H(\mathbf{C}^{coarse}, \mathbf{C}^{gt}) + \mathcal{L}_1(\mathbf{I}^{coarse}, \mathbf{I}^{gt}), \tag{12}$$

where $\mathcal{L}_1(\cdot)$ and $\mathcal{L}_H(\cdot)$, denote mean absolute difference and mean Huber loss (Huber, 1992). Following Liu et al. (2018a), we employ the hole loss and valid loss for the reconstruction of $\mathbf{L}^{coarse}$, $\mathbf{C}^{coarse}$ and $\mathbf{I}^{coarse}$.

As the fusion sub-network is responsible for fusing the information from both $\mathbf{L}^{coarse}$ and $\mathbf{C}^{coarse}$, the final inpainted image $\mathbf{I}^{final}$ must ensure both texture and structure consistencies among the inpainted and background regions. Hence, the objective function for the fusion sub-network, $\mathcal{L}_{FN}$, is as follows:

$$\mathcal{L}_{FN} = \mathcal{L}_1(\mathbf{I}^{final}, \mathbf{I}^{gt}) + \mathcal{L}_P(\mathbf{I}^{final}, \mathbf{I}^{gt}) + \mathcal{L}_S(\mathbf{I}^{final}, \mathbf{I}^{gt}) + \mathcal{L}_{TV}(\mathbf{I}^{final}), \tag{13}$$

where $\mathcal{L}_P(\cdot)$, $\mathcal{L}_S(\cdot)$ and $\mathcal{L}_{TV}(\cdot)$ denote the content loss, style loss and total variation loss, respectively (Johnson et al., 2016). For content loss and style loss, we use VGG-19 (Simonyan and Zisserman, 2014) as the feature extractor. Though the hyper-parameter tuning for pixel-level and feature-level losses plays a vital role for a better inpainted image (Liu et al., 2018a), we employed the same hyper-parameters (i.e., weight value of 1) for each loss in $\mathcal{L}_{SN}$ and $\mathcal{L}_{FN}$ while obtaining satisfactory results without tuning any hyper-parameters.

For the GAN framework, we adopt the Relativistic Least Square GAN (Jolicoeur-Martineau, 2018). The supervision process of relativistic discriminator depends on both real and generated data. Let $\mathcal{L}_G^{GAN}$, $\mathcal{L}_{D_{pixel}}^{GAN}$ and $\mathcal{L}_{D_{patch}}^{GAN}$ be the generator GAN loss, pixel discriminator GAN loss, and patch discriminator GAN loss, respectively. Then, the objective function for the GAN losses for the generator and both discriminators are as follow:

$$\begin{aligned}\mathcal{L}_G^{GAN} = {} & E_{x_f \sim P}[(D(x_f) - E_{x_r \sim P} D(x_r) - 1)^2] + \\ & E_{x_r \sim P}[(D(x_r) - E_{x_f \sim Q} D(x_f) + 1)^2],\end{aligned} \tag{14}$$

$$\begin{aligned}\mathcal{L}_{D_{pixel}}^{GAN} = {} & E_{x_r \sim P}[(D(x_r) - E_{x_f \sim Q} D(x_f) - 1)^2] + \\ & E_{x_f \sim Q}[(D(x_f) - E_{x_r \sim P} D(x_r) + 1)^2],\end{aligned} \tag{15}$$

$$\begin{aligned}\mathcal{L}_{D_{patch}}^{GAN} = {} & E_{x_r \sim P}[(D(x_r) - E_{x_f \sim Q} D(x_f) - 1)^2] + \\ & E_{x_f \sim Q}[(D(x_f) - E_{x_r \sim P} D(x_r) + 1)^2],\end{aligned} \tag{16}$$

where $D(\cdot)$ is the non-transformed discriminator output i.e., indicating how realistic the input data are compared with the generated data (Goodfellow et al., 2014; Jolicoeur-Martineau, 2018; Martin Arjovsky and Bottou, 2017). P, Q, x_r, and x_f are the distribution of real data, distribution of generated data, real data, and generated data, respectively.

Therefore, the objective functions for the generator and the discriminator network are given by

$$\mathcal{L}_G = \mathcal{L}_{SN} + \mathcal{L}_{FN} + \mathcal{L}_G^{GAN}, \tag{17}$$

$$\mathcal{L}_D = \mathcal{L}_{D_{patch}}^{GAN} + \mathcal{L}_{D_{pixel}}^{GAN}. \tag{18}$$

4. Experiments and results

4.1. Experimental setup

We evaluate our proposed method on two popular datasets used for the inpainting task: Places365 (Zhou et al., 2017) and CelebA-HQ (Karras et al., 2018). We use the original training, testing, and validation splits for Places365. For CelebA-HQ, we use the last 3,000 images as the testing images and the rest for training as CelebA-HQ does not have a predefined training-testing split. We compared our proposed method with ten existing state-of-the-art methods:

1. Partial convolution - PC (Liu et al., 2018a),[1]
2. Multi-column inpainting - MC (Wang et al., 2018),[2]
3. Pyramid context - PN (Zeng et al., 2019),[3]
4. Gated convolution - GC (Yu et al., 2019),[4]
5. EdgeConnect - EC (Nazeri et al., 2019),[5]
6. Mutual encoder–decoder - MD (Liu et al., 2020),[6]
7. Global–local attention - GLA (Uddin and Jung, 2020),[7]
8. Recurrent feature reasoning - RFR (Li et al., 2020),[8]
9. Region normalization - RN (Yu et al., 2020),[9]
10. Hyper-graph - HG (Wadhwa et al., 2021)[10]

Note that we refer to the official/available implementations along with the pre-trained weights from the respective sources and conduct an evaluation without any modifications of the original setups. For brevity, we will use the abbreviated names for each methods for easier referencing throughout the section. For the $LPIPS$ comparison in quantitative evaluation, we have used Inception-V2 (Szegedy et al., 2016) as the feature extractor.

Our model[11] is trained using PyTorch framework (Paszke et al., 2019) The model is optimized using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $\alpha_G = 1 \times 10^{-4}$ for the generator and $\alpha_D = 1 \times 10^{-5}$ for both discriminators. We set β_1 and β_2 as 0.5 and 0.9. We train each model on a single NVIDIA TITAN XP GPU with a batch size of 4 and image size of 256×256. We trained the Places365 model for 3 epochs and CelebA-HQ for 100 epochs. To generate different free-form masks (i.e. free-form and cellural) for training and testing, we follow the mask generation method of Ntavelis et al. (2020) (excluding rectangular mask). The size of the testing images is set to 512×512 for Places365 and 256×256 for CelebA-HQ. Our method is trained in an end-to-end training manner, without any post-processing steps.

[1] https://github.com/naoto0804/pytorch-inpainting-with-partial-conv.
[2] https://github.com/shepnerd/inpainting_gmcnn.
[3] https://github.com/researchmm/PEN-Net-for-Inpainting.
[4] https://github.com/JiahuiYu/generative_inpainting.
[5] https://github.com/knazeri/edge-connect.
[6] https://github.com/KumapowerLIU/Rethinking-Inpainting-MEDFE.
[7] https://github.com/SayedNadim/Global-and-Local-Attention-Based-Free-Form-Image-Inpainting.
[8] https://github.com/jingyuanli001/RFR-Inpainting.
[9] https://github.com/geekyutao/RN.
[10] https://github.com/GouravWadhwa/Hypergraphs-Image-Inpainting.
[11] Our code will be publicly available at: https://github.com/SayedNadim/SIFNet.

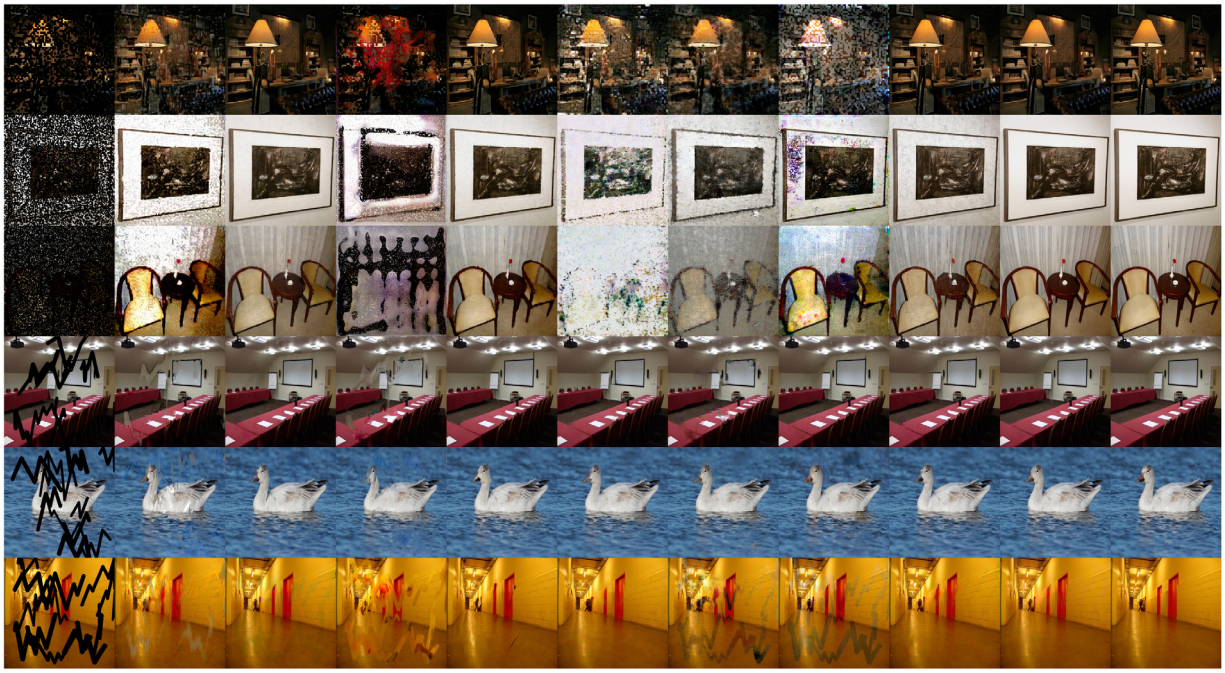

Fig. 7. Visual comparison for Places365 in different free-form masks (e.g., stroke masks, CA masks). From left, (a) input corrupted image, (b) PC (Liu et al., 2018a), (c) MC (Wang et al., 2018), (d) PN (Zeng et al., 2019), (e) GC (Yu et al., 2019), (f) EC (Nazeri et al., 2019), (g) MD (Liu et al., 2020), (h) RN (Yu et al., 2020), (i) GLA (Uddin and Jung, 2020), (j) proposed model, and (k) respective ground truths. Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Qualitative evaluation

For qualitative evaluations, we provide visual comparisons in Fig. 7 for Places365 consisting of complex natural scenes, and Fig. 8 for the CelebA-HQ dataset consisting of human faces for different irregular masks (i.e., free-form and CA mask).

It can be seen from the figures that the PC method fails to maintain texture consistencies in both cases due to the heuristic mask update mechanism. The PC validates all spatial pixel locations to be either valid or invalid based on a heuristic rule i.e. 1 for valid mask pixel and 0 for invalid mask pixel if the total number of mask values are greater than a threshold in a convolutional kernel. The MC method performs comparatively better in maintaining global structural consistencies due to the multi-scale feature extraction and regularization mechanism. However, it performs poorly in texture consistency and local structure preservation. The PN method uses a pyramid context encoder mechanism to transfer context information from different layers for feature reconstruction. However, it fails to generate plausible textures for CA masks due to transferring incorrectly attended features from different layers. The GC method produces visually plausible results in both structure and texture due to the patch-based inner product similarity mechanism and soft layer gating-based mask update mechanism. However, because of the non-learnable patch-based contextual similarity mechanism, it produces unwanted repetitions of structures and inconsistent textures. The EC method generates plausible structures in free-form masks. However, it fails in CA masks as it tries to hallucinate edges for the mask area and eventually generates inconsistent structures and textures. Additionally, the EC method makes use of the Canny edge detection mechanism (Canny, 1986), which is sensitive to noise and thresholds. The RN method uses explicit normalization methods for non-mask and mask areas and performs plausibly well with free-form masks. However, it fails to generate plausible structures or textures in CA masks. This is because in CA masks, the mask area is comparatively larger than the content area and mismatch in normalization leads to a poor hallucination of the missing area.

The RFR method uses a progressive content generation approach for inpainting the missing area and performs comparatively better in hallucinating global structures and textures for both masks. However, it fails to preserve the local structures and textures due to the shared feature fusion mechanism. The MD method extracts structure and texture features from different layers of the encoder. These features are fed to the different layers of the decoder to hallucinate the missing regions. However, such implicit decoupling cannot deal with different free-form masks that require both structure and texture consistencies, leading to a poor generalization. The GLA method uses an explicit mask pruning mechanism and global-local attention for feature reconstruction and generates visually plausible contents. However, it fails to maintain local textures due to the self-attention mechanisms (Zhang et al., 2019) as the self-attention mechanisms prioritize globally important features. The HG method uses the hyper-graph-based feature-level attention mechanism and generates better global structures while preserving texture consistencies. However, it shows inconsistencies in preserving local structures.

Our proposed SIFNet shows better inpainted results in most of the cases owing to the Split-Inpaint-Fuse method. Moreover, as PCM in the Luma branch performs progressive two-stage feature similarity mechanism and SCCM in the Chroma branch performs the spatial-channel attention mechanism, the SIFNet learns to produce better structure and texture information and fuses the information for the refinement of the inpainted image.

4.3. Quantitative evaluation

We performed the objective evaluation using four commonly used image quality metrics, i.e., L_1 error, $SSIM$ (Wang et al., 2004), $PSNR$ and $LPIPS$ (Zhang et al., 2018b). Tables 1 and 2 show the quantitative evaluation of the proposed method and the compared models for free-form masks (e.g., stroke masks and CA mask) in the Places365, and CelebA-HQ datasets, respectively.

From the tables, it can be seen that the PC method shows comparatively worse results in both masks as its proposed mask update

Fig. 8. Visual comparison for CelebA-HQ in different free-form masks (e.g., stroke masks, CA masks). From left, (a) input corrupted image, (b) MC (Wang et al., 2018), (c) PN (Zeng et al., 2019), (d) GC (Yu et al., 2019), (e) EC (Nazeri et al., 2019), (f) RFR (Li et al., 2020), (g) GLA (Uddin and Jung, 2020), (h) HG (Wadhwa et al., 2021), (i) proposed model, and (j) the respective ground truth image. Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Performance evaluation on irregular masks for Places365 (Zhou et al., 2017). Best values are in bold and the second best values are underlined.

Metrics ‖ Models	Mask (%)	PC	MC	PN	GC	EC	MD	RN	GLA	Proposed
$SSIM$ ↑	10–20	0.8718	0.9323	0.8852	0.9296	0.9259	0.9025	0.9346	0.9324	**0.9413**
	20–30	0.8327	0.9164	0.8441	0.9129	0.9067	0.8760	0.9184	0.9172	**0.9280**
	30–40	0.7688	0.8749	0.7756	0.8693	0.8626	0.8223	0.8787	0.8743	**0.8892**
	40–50	0.6779	0.8272	0.6963	0.8230	0.8113	0.7650	0.8316	0.8273	**0.8470**
$PSNR$ ↑	10–20	24.7896	27.7809	24.5678	27.2997	27.5725	25.5108	28.3124	27.4946	**29.0563**
	20–30	23.8279	27.0863	23.2276	26.6432	26.7338	24.5942	27.6207	26.9109	**28.4126**
	30–40	21.9923	24.9774	21.2606	24.4994	24.7864	22.5979	25.4878	24.6956	**26.2579**
	40–50	19.6273	23.1717	19.5417	22.8561	23.1350	20.9593	23.7580	22.9217	**24.5886**
L_1 ↓	10–20	0.0196	0.0142	0.0190	0.0146	0.0143	0.0148	0.0116	0.0144	**0.0101**
	20–30	0.0242	0.0165	0.0249	0.0169	0.0170	0.0184	0.0141	0.0166	**0.0122**
	30–40	0.0347	0.0226	0.0363	0.0234	0.0232	0.0267	0.0203	0.0229	**0.0178**
	40–50	0.0523	0.0302	0.0502	0.0308	0.0308	0.0367	0.0281	0.0304	**0.0244**
$LPIPS$ ↓	10–20	1.0697	0.7769	1.0478	0.7649	0.8167	0.8985	0.8083	0.8030	**0.6340**
	20–30	1.2702	0.8974	1.2705	0.8799	0.9475	1.0949	0.9550	0.9221	**0.7523**
	30–40	1.6172	1.1191	1.6512	1.1104	1.1736	1.4234	1.2157	1.1625	**0.9989**
	40–50	1.9694	1.3637	1.9666	1.3442	1.4319	1.7372	1.4916	1.4147	**1.2538**

mechanism is heuristic and cannot maintain structure or texture consistencies when the mask area is large or contains random missing pixels. The MC method shows a comparatively higher $SSIM$ and $PSNR$ due to the multi-scale feature extraction and regularization. However, it cannot handle larger free-form masks or CA masks and shows texture inconsistencies. The PN method shows worse results in free-form masks though it propagates attention values from different layers of the encoder and fuses them in the decoder. Though this approach works well with free-form masks, it introduces texture artifacts in CA masks. The GC method can handle larger holes and produces better inpainting results as it depends on inner-product similarity among the patches and employs a soft layer gating mechanism. However, it fails to preserve texture consistencies and produces artifacts in non-repetitive structures. The EC method performs comparatively better with free-form masks. However, as it tries to hallucinate edges for the missing regions, in the case of CA masks, it tries to hallucinate local missing edges and eventually produces structure and texture artifacts. The RN method tries to normalize non-hole and hole regions differently. However, as the CA masks contain more mask regions than the content/background regions, the different normalization technique leads to poor texture synthesis.

Though the MD method uses an implicit decoupling to extract structure and texture features, it fails to generate plausible contents. This is because inpainting large free-form masks generally require explicit attention to both structure and texture information, which cannot be obtained with implicit feature decoupling. The GLA method produces plausible synthesized contents for free-form masks due to the explicit mask value pruning mechanisms. However, in the case

Table 2
Performance evaluation on irregular masks for CelebA-HQ (Karras et al., 2018). Best values are in bold and the second best values are underlined.

Metrics ‖ Models	Mask (%)	MC	PN	Gated	EC	RFR	GLA	HG	Proposed
$SSIM$ ↑	10–20	0.8649	0.9155	0.9155	0.9202	0.9268	0.9147	0.9297	**0.9455**
	20–30	0.7654	0.8585	0.8846	0.7407	0.8988	0.8864	0.9109	**0.9251**
	30–40	0.7398	0.8344	0.8564	0.8622	0.8735	0.8585	0.8845	**0.8956**
	40–50	0.6404	0.7466	0.8104	0.8191	0.8396	0.8086	0.8536	**0.8634**
$PSNR$ ↑	10–20	24.1315	27.7069	28.9938	29.2072	29.9394	28.4372	30.0230	**31.1502**
	20–30	19.7710	24.2655	27.3491	27.3430	27.8801	27.2024	28.8481	**29.8063**
	30–40	19.6885	24.0393	25.8930	26.0602	26.7100	25.6783	27.2243	**27.6998**
	40–50	16.3597	19.7338	24.1355	24.5018	25.4998	23.6537	25.6350	**26.2770**
L_1 ↓	10–20	0.0231	0.0128	0.0142	0.0136	0.0124	0.0146	0.0122	**0.0077**
	20–30	0.0414	0.0225	0.0181	0.0175	0.0160	0.0182	0.0145	**0.0105**
	30–40	0.0462	0.0251	0.0219	0.0209	0.0191	0.0220	0.0179	**0.0146**
	40–50	0.0751	0.0483	0.0288	0.0274	0.0239	0.0299	0.0229	**0.0197**
$LPIPS$ ↓	10–20	1.1254	0.7813	0.8290	0.7866	0.7558	0.8074	0.7501	**0.4829**
	20–30	1.6577	1.1931	1.0397	0.9775	0.9489	1.0080	0.9011	**0.6735**
	30–40	1.6800	1.2421	1.0884	1.0268	0.9916	1.0530	0.9675	**0.7659**
	40–50	2.3269	1.9259	1.4502	1.3423	1.2442	1.4081	1.2378	**1.0871**

of CA masks, pruning mask values leads to comparatively fewer content values for content synthesis and eventually leads to poor local textures. The HG method works plausibly for both masks due to the hypergraph-based attention mechanism to find contributing features for content synthesis. However, though the global structure and texture are preserved, it shows inconsistencies in the local structures and corresponding textures.

Our proposed model achieves significantly lower values in terms of the L_1 and $LPIPS$ differences while attaining higher $SSIM$ and $PSNR$ values for both masks. This is due to the explicit luma-chroma inpainting with the intuitive context modules (i.e., PCM and SCCM) and an effective fusion sub-network that ensures both structure and texture consistencies for a better inpainted image.

4.4. Ablation studies

To validate the efficiency and effectiveness of the proposed SIFNet and its constituting modules, we have performed an extensive ablation studies. Specifically, we have performed the ablation experiments on the following cases:

1. Validation of color space split and fusion (Section 4.4.1)
2. Effects of the Split-Inpaint-Fuse method on other SOTA methods (Section 4.4.2)
3. Effectiveness of the proposed attention modules (Section 4.4.3)
4. Effectiveness of the discriminators (Section 4.4.4)
5. Effects of individual feature-level losses (Section 4.4.5)
6. Generalization ability on other free-form masks (Section 4.4.6)
7. Effects of training dataset size (Section 4.4.7)

Except the ablation of training dataset size, all ablation experiments are done using CelebA-HQ dataset and follows the same hyper-parameters of the proposed model (i.e., learning rate, loss weights) while keeping the epoch limited to 50. For the ablation experiment of training dataset size, the models are trained with Places365 dataset for 2 epochs.

4.4.1. Validation of color space split and fusion

To show the effectiveness of this SIF mechanism, we perform additional experiments with variants of the proposed model. Specifically, we train three variants — without any SIF-based method, a SIF-based method using LUV color space, and a SIF-based method using CIELAB color space. LUV color space is a perceptually uniform color space and bears significant resemblance with CIELAB color space in characteristics and hence, we choose LUV to perform the ablation experiment. Note that, all the variants are trained without any attention mechanism (i.e. PCM and SCCM) to examine the effects of the Split-Inpaint-Fuse mechanism only. Fig. 9 shows the visual comparison for the variants of the proposed model.

Table 3
Ablation studies on the effect of Split-Inpaint-Fuse mechanism and fusion mechanism using 60–70% CA mask in CelebA-HQ. Note that all models have been trained without any attention mechanism to validate the effects of Split-Inpaint-Fuse mechanism.

Model/Metrics	No SIF	SIF (LUV)	SIF (LAB)
$SSIM$ ↑	0.9499	0.9698	**0.9719**
$PSNR$ ↑	28.9243	29.4269	**30.6282**
L_1 ↓	0.0208	0.0163	**0.0147**

Table 4
Performance gain by using the Split-Inpaint-Fuse mechanism on the GLA method.

Model/Metrics	GLA	GLA-SIF
$SSIM$ ↑	0.8528	**0.8676**
$PSNR$ ↑	26.4413	**27.6094**
L_1 ↓	0.0216	**0.0202**

As seen in Fig. 9, the No-SIF-based method generates good structures but fails to produce texture consistency comparing with the LUV model or CIELAB model, though all the models are trained for the same epoch. Moreover, even though all the variants do not have any attention mechanisms, it can be seen that the SIF-based method alone can ensure better inpainted results. Table 3 shows that SIF-based methods achieve better $SSIM$ and $PSNR$ values while having comparatively lower L_1 error. Hence, it can be concluded that the SIF-based inpainting methods generalize faster and capture more semantic relationships among pixels for better structure and texture consistencies.

4.4.2. Effect of the Split-Inpaint-Fuse method on other SOTA methods

Additionally, we evaluated the effect of the Split-Inpaint-Fuse method on the existing methods. Specifically, we re-trained the GLA method (Uddin and Jung, 2020) by replacing the coarse network with the split-inpaint sub-network (retaining the attention mechanism used in the original method) and kept the refinement network unchanged. We trained the original GLA method and GLA-SIF-based method for 50 epochs while keeping all the original settings unchanged (i.e., GAN framework, losses, and hyper-parameters).

It can be seen from Table 4 that, integrating the Split-Inpaint-Fuse mechanism in the GLA method has improved the $SSIM$ and $PSNR$ while attaining a lower L_1 error. This is because the GLA method uses mask-value-pruning-based global attention in the coarse stage and this attention mechanism removes the less important feature values before calculating the global correlation map among feature values. As we introduced the separate luma and chroma information in the

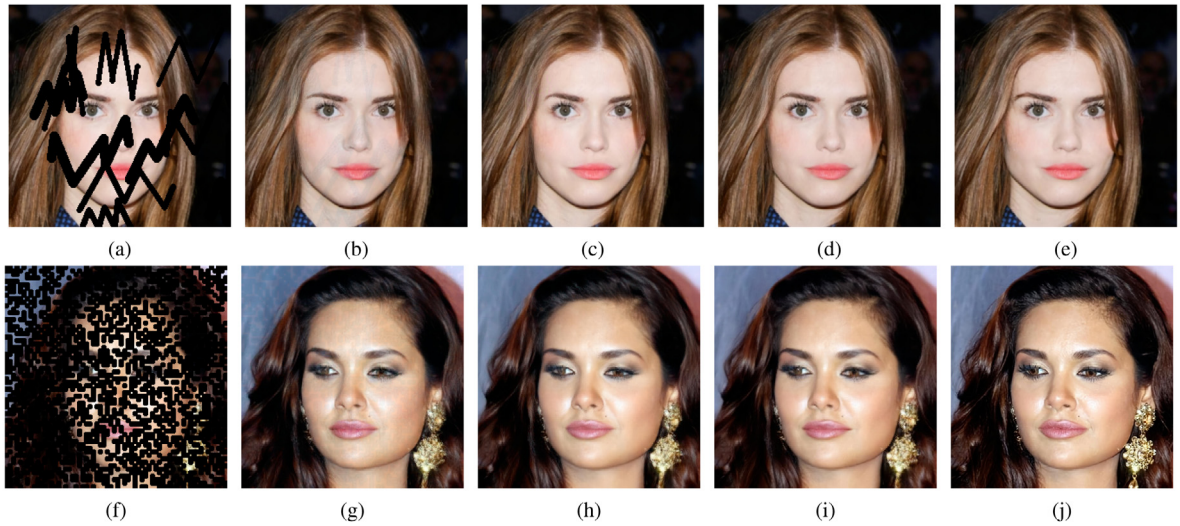

Fig. 9. Effect of the Split-Inpaint-Fuse mechanism. It can be seen that No-SIF-based method performs worse than the SIF-based methods in terms of color consistency. For each row, (a, f) input image, (b, g) No-SIF-based method, (c, h) SIF-based method using LUV color space, (d, i) SIF-based method using CIELAB color space, and (e, j) corresponding ground truth. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Ablation studies on the effectiveness of the proposed attention modules (20%–30% free-form mask on CelebA-HQ dataset). Best values are in bold and the second best values are underlined.

Metrics	CA + SCCM	PCM + SA	PCM + SE	PCM only	SCCM only	No-attention	Proposed
$SSIM \uparrow$	0.9209	0.9247	<u>0.9248</u>	0.9187	0.9199	0.9100	**0.9251**
$PSNR \uparrow$	29.4915	29.7428	<u>29.7494</u>	28.7336	28.8918	28.3439	**29.8063**
$L_1 \downarrow$	0.0117	0.0111	<u>0.0110</u>	0.0132	0.0127	0.0121	**0.0105**
$LPIPS \downarrow$	0.7517	0.7092	<u>0.6946</u>	0.7384	0.7078	0.7644	**0.6735**

coarse stage of the GLA method, the method learned to prune less important feature values from both luma and chroma images, leading to comparatively better structure and texture consistent inpainted image. Again, the global-local attention mechanism in the refinement stage of GLA depends on the feature map similarities. As the SIF mechanism provides more information (i.e., explicit luma and chroma information), the refinement process of the GLA method has more texture and structure information, that eventually facilitates more realistic and refined content generation.

4.4.3. Effectiveness of the proposed attention modules

The proposed PCM and SCCM facilitate the inpainting process by guiding the model to find the most contributing features. To evaluate the effect of the proposed modules, we perform additional experiments with different variants of the proposed model. Specifically, we train six variants of the proposed modules,

1. CA + SCCM: Replaced PCM with Contextual Attention from Yu et al. (2018) and SCCM
2. PCM + SA: Replaced SCCM with Self-attention from Zhang et al. (2019)
3. PCM + SE: Replaced SCCM with Squeeze-and-Excitation from Hu et al. (2018)
4. PCM: Trained with PCM only (i.e. attention in the Luma branch only)
5. SCCM: Model trained with SCCM only (i.e. attention in the Chroma branch only)
6. No-attention : Model trained without any modules (i.e. no attention in the Luma and Chroma branches)

Fig. 10 and Table 5 show the visual and objective comparisons for the variants. It can be seen from Fig. 10 that the SCCM contributes to the color values but fails to maintain local structure, the PCM preserves structure but fails in texture consistencies, the no-attention model fails to preserve both structure and texture consistencies, and the proposed

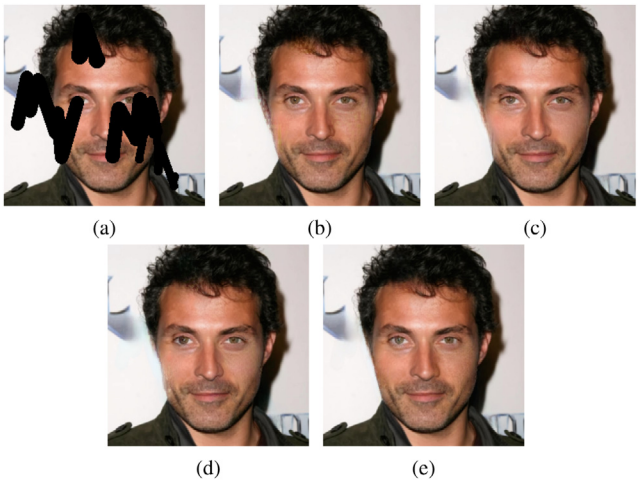

Fig. 10. Ablation studies on the effects of individual attention modules. (a) input image (b) No-attention (c) SCCM only (d) PCM only, and (e) proposed (SCCM + PCM). Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

SIFNet with both modules is effective in maintaining both structure and texture consistencies.

The visual observation on the effects of the attention modules is reflected in Table 5. It can be seen in the table that the model without any module (i.e., PCM and SCCM) attains lower values in all metrics (except L_1). It is interesting to observe that the mean per-pixel error of the no-attention model is smaller compared with PCM or SCCM. However, as there is no explicit attention for structure and texture reconstruction, this method fails in other metrics. It is another interesting observation that PCM combined with SE achieves comparative results

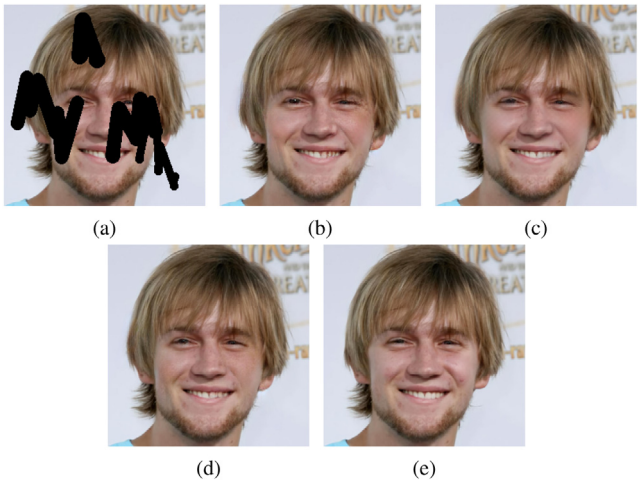

Fig. 11. Ablation studies on the effects of the discriminators. (a) input image (b) with the pixel discriminator only (c) with the patch discriminator only, (d) proposed (pixel + patch) and (e) ground truth. Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and the variants without any channel attention mechanism attain lower metrics; that proves the essentiality of integrating channel information in the inpainting process. Nevertheless, the proposed SIFNet obtains better results in all metrics, revealing the efficiency of the proposed PCM and SCCM.

Both the PCM and SCCM deal with finding the most contributing features to inpaint the image. However, the goal of the PCM is to ensure local structural similarities between the masked and non-masked region and so, the PCM tries to find local similarities among patches using the inner product. To do so, the PCM extracts patches from the non-masked regions and uses the extracted patches as convolution filters to obtain the inner product similarity with the masked region. Then it finds the most contributing patches using a softmax operation. The PCM uses these patches as filters for transpose convolution to reconstruct the masked region. As the operation is local in nature, the reconstructed regions will have consistencies with the non-masked region.

In the case of SCCM, it uses a non-local block to find the long-range dependencies among features while utilizing a channel re-weighting mechanism. The non-local block finds the most contributing feature values where the response is high if the similarity is high. Again, the channels of chroma features are re-weighted using an MLP to find the most contributing channel values. Both of these operations are global in nature. This is to ensure the long-range texture and structure consistencies among the chroma features. So, owing to the difference in modalities, the PCM works in a local manner to inpaint the luma image while the SCCM works in a global manner to inpaint the chroma image.

4.4.4. Effectiveness of the discriminators

To verify the effect of the both discriminators, we performed additional experiments by training two variants of the SIFNet -

1. Trained with the pixel discriminator only
2. Trained with the patch discriminator only

As seen in Fig. 11, as the pixel discriminator focuses on pixel-level supervision, it fails to maintain texture consistencies with surrounding pixels. The patch discriminator focuses on patch-level supervision and maintains local texture consistency. However, due to a comparatively larger receptive area (i.e., patches), the patch discriminator cannot efficiently ensure pixel-level texture consistencies. The SIFNet incorporates both of the discriminators for the supervision of the content generation process, ensuring both pixel-level and patch-level supervision.

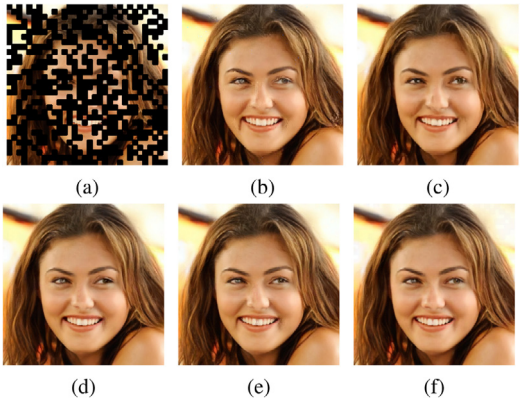

Fig. 12. Visualization of the effect of the individual reconstruction losses. (a) input image, (b) without feature losses (i.e., content loss, style loss and TV loss), (c) without content loss, (d) without style loss, (e) without TV loss, and (f) proposed method. Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.4.5. Effects of individual feature-level losses

The SIFNet uses feature-level losses in $\mathcal{L}_{FN}$ for the fusion sub-network (i.e., Eq. 8). To validate the effects of the individual feature-level losses, we performed additional experiments with various variants of the SIFNet with/without the feature-level losses. Specifically, we trained four variants -

1. Without the feature-level losses (i.e., content loss, style loss, and TV loss)
2. Without the content loss
3. Without the style loss
4. Without the TV loss.

Fig. 12 shows the visual comparisons among the loss variants. As it can be seen from the figures that, without the feature level losses, the model cannot produce plausible and realistic textures and structures. Hence, it can be concluded that the combination of pixel-level losses and feature-level losses complement each other to produce a better inpainted image in the refinement stage.

4.4.6. Generalization ability on other free-form masks

We perform additional experiments to verify the generalization ability of the SIFNet on other unseen irregular masks. For the experiments, we used the test masks from the NVIDIA irregular mask dataset (Liu et al., 2018b) and QuickDraw mask dataset (Ha and Eck, 2017). Note that, we did not use the NVIDIA irregular or QuickDraw mask datasets for training or validation.

It can be seen in Fig. 13 that the SIFNet can handle unseen irregular masks with different characteristics while maintaining both structure and texture.

4.4.7. Effects of training dataset size

We performed additional experiments on the effects of the dataset sizes in the training phase. Specifically, we experimented on the effectiveness of SIFNet in a finite data constraint i.e., whether, in the limit of infinite data, luma-chroma split may not help because the neural network can learn the decorrelated information, but with finite training data, the decorrelated information can be a helpful prior. To validate this intuition, we trained four variants of SIFNet based on the number of training images from the Places365 dataset, i.e.,

1. SIFNet trained with 1/10 of the total training images
2. SIFNet trained with 1/5 of the total training images
3. SIFNet trained with 1/2 of the total training images
4. No-SIF model trained with 1/2 of the total training images

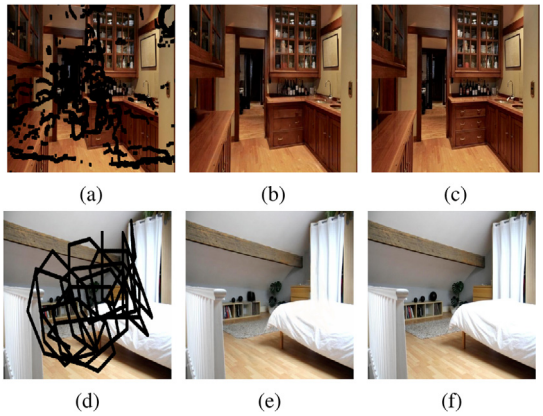

Fig. 13. Ablation studies on the generalization ability of the SIFNet on the NVIDIA irregular mask and QuickDraw mask datasets. Here, the first row shows the results for the NVIDIA irregular mask and the second row shows the results for the QuickDraw mask. Note that we did not use the datasets for training or validating the SIFNet. From left, input corrupted image, reconstructed image, and corresponding ground truth. Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

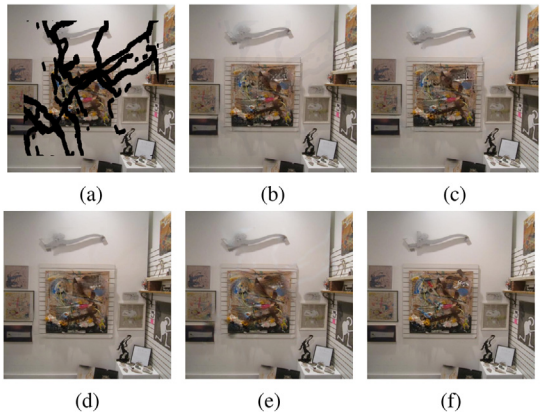

Fig. 14. Ablation studies on the effect of dataset size during training using Places365 dataset. From left, (a) input image, (b) trained with 1/10 of the training samples, (c) trained with 1/5 of the training samples (d) trained with 1/2 of the training samples (e) No-SIF model trained with 1/2 of the training samples, and (f) ground truth. Each model was trained for 2 epochs while keeping all the settings the same. It can be seen that SIF-based methods generalize faster in terms of training samples when compared with the No-SIF-based method, for structure and texture recovery. Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

It can be seen from Fig. 14 that though the color recovery gets better with more training images, structure recovery is consistent and provides a strong base for color recovery. This observation is consistent with Nazeri et al. (2019). Nazeri et al. (2019) depends on edge information for structure consistency and with more training, as the structure recovery becomes more robust, the texture recovery progressively improves.

The No-SIF-based method fails to recover meaningful structure or texture while trained for the same amount of time and training data, which emphasizes the difficulty in generalization. Compared with the No-SIF-based method, the SIF-based method generalizes faster with a smaller number of the training sample, in terms of structure and color recovery, hence providing a stronger base for exploring the Split-Inpaint-Fuse mechanism in future extensions.

5. Discussion and limitations

5.1. Discussion

Image inpainting is an ill-posed problem and there is no exact solution for the task. Hence, one corrupted image can have many candidate contents for the missing regions. Learning-based inpainting models or generative inpainting models leverage this property and try to generate novel content for the missing regions. As it is almost impossible to generate content as exact as the ground truth, there is no valid way to differentiate among the generated novel contents, except visual plausibility. An inpainted image is considered visually plausible and realistic if the inpainted contents are consistent with surrounding regions, i.e., in structure and texture. Most of the existing free-form inpainting methods excel in generating plausible contents for the inpainting tasks in most cases. For that, most of the methods depend on a single coarse estimation, additional structure inputs, or explicit normalization methods for maintaining the structural and textural consistencies among the generated regions and surrounding regions. However, we argue that a single coarse estimation or binary edge structure alone cannot provide the necessary texture and structure information for the refinement process.

Instead of using a single input color image for reconstructing the coarse inpainted image, the SIFNet decomposes the input image into a luma image and a chroma image and inpaints them separately using two identical branches in split-inpaint sub-network, generating two coarse images - a coarse luma image and a coarse chroma image. Though the branches of the split-inpaint network consist of identical architecture, they handle different modalities (i.e., luma and chroma information) and hence, weight sharing cannot be a good architectural choice. A common convolution kernel (i.e., shared weight) will learn features that are present in both luma image and chroma image. As the luma information differs from the chroma information, a shared weight will not be beneficial as it will learn only the correlated information in both modalities.

The fusion sub-network leverages the de-correlated coarse luma and chroma information for the fusion and produces a refined inpainted image. This Split-Inpaint-Fuse-based inpainting approach brings two benefits - (1) the availability of explicit luma and chroma information and (2) additional information for the refinement process for structure and texture consistencies. Moreover, the SIFNet addresses two important issues regarding the luma and chroma data relationship - (1) it considers the possible case scenarios where luma and chroma data are decorrelated through the split-inpaint sub-network, and (2) another possible case scenarios where luma and chroma data are correlated through the fusion sub-network. Hence, as seen in the qualitative and quantitative comparison sections, the SIFNet outperforms existing methods by significant margins.

5.2. Limitations

Though the SIFNet shows impressive results for the inpainting task with irregular masks, it is also bound by some limitations posed by several crucial factors.

First, though free-form inpainting is more practical in terms of real-world applications, rectangular or box mask inpainting is still considered as one of the most crucial tasks in the inpainting domain. SIFNet is designed to adapt for free-form masks, which limits its ability to inpainting rectangular masks.

Second, the SCCM in the chroma branch contains a self-attention mechanism for calculating spatial feature contributions. Self-attention is computationally expensive and limits the resolution of the test images owing to quadratic memory requirement (Tang et al., 2018; Shen et al., 2021). Hence, the SIFNet cannot inpaint higher resolution images.

Third, hyper-parameters such as loss weights play a vital role in the quality of the inpainted contents. However, though the common approach of hyper-parameter tuning is to use some validation images for searching the best values (e.g., Liu et al. (2018a)), different datasets

Fig. 15. Some failure cases of the SIFNet. Best viewed in color and zoomed in. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

contain different image characteristics (e.g., Places365 contains mostly outdoor images while CelebA-HQ contains human faces), and finding generalized hyper-parameters is difficult. Hence, we choose to use the same hyper-parameters for the loss weights and left hyper-parameter tuning as one of the future extensions of the SIFNet.

Fourth, as the SIFNet incorporates a GAN framework, it inherits the advantages as well as the demerits of the GAN architectures. GAN architectures are prone to mode collapse (i.e., generating the same outputs values for different inputs) and non-convergence problem (Barnett, 2018; Saxena and Cao, 2021; Jolicoeur-Martineau, 2018). The generator can produce meaningful content as long as the discriminator can provide enough supervision on the content generation. However, for a larger mask area, the discriminator cannot provide sufficient supervision, and eventually, the generator falls into a mode collapse problem. Hence, though the SIFNet can generalize for almost all kinds of irregular masks, it outputs visual artifacts if the mask area is comparatively larger (i.e., more than the training mask area). Larger mask area inpainting is left as another future extension for the SIFNet. Fig. 15 shows some failure cases of SIFNet.

6. Conclusion

In this paper, we proposed SIFNet – a two-stage image inpainting framework using a color space Split-Inpaint-Fuse approach for irregular masks – that leverages the decorrelated color space for refined inpainted images. For this, we proposed to incorporate a split-inpaint sub-network that leverages color space decomposition to obtain the luma and chroma images and inpaints them concurrently with two separate branches (i.e., luma branch and chroma branch) in the coarse stage. We also proposed to integrate a fusion sub-network that handles the fusion of the coarse luma and chroma images for the refinement stage. Additionally, we proposed two attention mechanisms — the progressive context module for the luma branch that performs the patch-level similarity calculation and the spatial-channel context module for the chroma branch that calculates spatial and channel feature contributions. The proposed SIFNet has been evaluated using different datasets and compared with the existing state-of-the-art methods. Experimental results show that the SIFNet can provide visually plausible contents for diverse irregular masks and outperforms the existing methods with significant margins. Moreover, the extensive ablation studies verify the effectiveness of the proposed Split-Inpaint-Fuse-based inpainting method and the efficiency of the proposed modules.

CRediT authorship contribution statement

S.M. Nadim Uddin: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization. **Yong Ju Jung:** Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded in part by the National Research Foundation of Korea (grant no. NRF-2020R1A2C1008753) and the Gachon University research fund of 2021 (GCU-202106460001). All authors have read and agreed to the published version of the manuscript.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cviu.2022.103446.

References

Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J., 2001. Filling-in by joint interpolation of vector fields and gray levels. IEEE Trans. Image Process. 10 (8), 1200–1211.

Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B., 2009. Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. (ToG) 28 (3), 24.

Barnett, S.A., 2018. Convergence problems with generative adversarial networks (gans). arXiv preprint arXiv:1806.11382.

Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C., 2000. Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 417–424.

Bertalmio, M., Vese, L., Sapiro, G., Osher, S., 2003. Simultaneous structure and texture image inpainting. IEEE Trans. Image Process. 12 (8), 882–889.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6), 679–698.

Cohen, J., Wyszecki, G., Stiles, W.S., 1968. Color science: Concepts and methods, quantitative data and formulas. Am. J. Psychol. 81 (1), http://dx.doi.org/10.2307/1420820.

Criminisi, A., Pérez, P., Toyama, K., 2004. Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. 13 (9), 1200–1212.

Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P., 2012. Image melding: combining inconsistent images using patch-based synthesis. ACM Trans. Graph. (ToG) 31 (4), 1–10.

Dolhansky, B., Canton Ferrer, C., 2018. Eye in-painting with exemplar generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7902–7911.

Drori, I., Cohen-Or, D., Yeshurun, H., 2003. Fragment-based image completion. In: ACM SIGGRAPH 2003 Papers. ACM New York, NY, USA, pp. 303–312.

Efros, A.A., Freeman, W.T., 2001. Image quilting for texture synthesis and transfer. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 341–346.

Esedoglu, S., Shen, J., 2002. Digital inpainting based on the Mumford–Shah–Euler image model. European J. Appl. Math. 13 (4), 353–370.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680.

Ha, D., Eck, D., 2017. A neural representation of sketch drawings. arXiv preprint arXiv:1704.03477.

Hong, X., Xiong, P., Ji, R., Fan, H., 2019. Deep fusion network for image completion. In: Proceedings of the 27th ACM International Conference on Multimedia. In: MM '19, Association for Computing Machinery, New York, NY, USA, pp. 2033–2042. http://dx.doi.org/10.1145/3343031.3351002.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141.

Huang, J.-B., Kang, S.B., Ahuja, N., Kopf, J., 2014. Image completion using planar structure guidance. ACM Trans. Graph. 33 (4), 1–10.

Huber, P.J., 1992. Robust estimation of a location parameter. In: Breakthroughs in Statistics. Springer, pp. 492–518.

Iizuka, S., Simo-Serra, E., Ishikawa, H., 2016. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Trans. Graph. (ToG) 35 (4), 1–11.

Iizuka, S., Simo-Serra, E., Ishikawa, H., 2017a. Globally and locally consistent image completion. ACM Trans. Graph. (ToG) 36 (4), 1–14.

Iizuka, S., Simo-Serra, E., Ishikawa, H., 2017b. Globally and locally consistent image completion. ACM Trans. Graph. (ToG) 36 (4), 107.

Jang, H.W., Jung, Y.J., 2020. Deep color transfer for color-plus-mono dual cameras. Sensors 20 (9), 2743.

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

Jolicoeur-Martineau, A., 2018. The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734.

Karras, T., Aila, T., Laine, S., Lehtinen, J., 2018. Progressive growing of GANs for improved quality, stability, and variation. In: 6th International Conference on Learning Representations.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Leung, B., Jeon, G., Dubois, E., 2011. Least-squares luma–chroma demultiplexing algorithm for bayer demosaicking. IEEE Trans. Image Process. 20 (7), 1885–1894.

Levin, Zomet, Weiss, 2003. Learning how to inpaint from global image statistics. In: Proceedings Ninth IEEE International Conference on Computer Vision, Vol. 1, pp. 305–312.

Li, Y., Liu, S., Yang, J., Yang, M.-H., 2017. Generative face completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3911–3919.

Li, J., Wang, N., Zhang, L., Du, B., Tao, D., 2020. Recurrent feature reasoning for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7760–7768.

Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C., 2020. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: European Conference on Computer Vision. Springer, pp. 725–741.

Liu, G., Reda, F.A., Shih, K.J., Wang, T.-C., Tao, A., Catanzaro, B., 2018a. Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 85–100.

Liu, G., Reda, F.A., Shih, K.J., Wang, T.-C., Tao, A., Catanzaro, B., 2018b. Nvidia irregular mask dataset. https://Nv-Adlr.Github.Io/Publication/Partialconv-Inpainting.

Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013a. Rectifier nonlinearities improve neural network acoustic models. In: Proc. Icml. Vol. 30, (1), Citeseer, p. 3.

Maas, A.L., Hannun, A.Y., Ng, A.Y., et al., 2013b. Rectifier nonlinearities improve neural network acoustic models. In: Proc. Icml. 30, (1), Citeseer, p. 3.

Martin Arjovsky, S., Bottou, L., 2017. Wasserstein generative adversarial networks. In: Proceedings of the 34 Th International Conference on Machine Learning. Sydney, Australia.

Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M., 2019. EdgeConnect: Structure guided image inpainting using edge prediction. In: Proceedings of the IEEE International Conference on Computer Vision Workshops.

Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528.

Ntavelis, E., Romero, A., Bigdeli, S., Timofte, R., Hui, Z., Wang, X., Gao, X., Shin, C., Kim, T., Son, H., et al., 2020. AIM 2020 challenge on image extreme inpainting. In: European Conference on Computer Vision. Springer, pp. 716–741.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A., 2016. Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2536–2544.

Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G., 2019. StructureFlow: Image inpainting via structure-aware appearance flow. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 181–190.

Robertson, A., Lozano, R., Alman, D., Orchard, S., Keitch, J., Connely, R., Graham, L., Acree, W., John, R., Hoban, R., et al., 1977. CIE recommendations on uniform color spaces, color-difference equations, and metric color terms. Color Res. Appl. 2, 5–6.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Saxena, D., Cao, J., 2021. Generative adversarial networks (GANs) challenges, solutions, and future directions. ACM Comput. Surv. 54 (3), 1–42.

Schwarz, M.W., Cowan, W.B., Beatty, J.C., 1987. An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models. ACM Trans. Graph. 6 (2), 123–158.

Sharif, S., Jung, Y.J., 2019. Deep color reconstruction for a sparse color sensor. Opt. Express 27 (17), 23661–23681.

Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H., 2021. Efficient attention: Attention with linear complexities. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3531–3539.

Simakov, D., Caspi, Y., Shechtman, E., Irani, M., 2008. Summarizing visual data using bidirectional similarity. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–8.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., Jay Kuo, C.-C., 2018a. Contextual-based image inpainting: Infer, match, and translate. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19.

Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., Kuo, C.-C.J., 2018b. Spg-net: Segmentation prediction and guidance network for image inpainting. arXiv preprint arXiv:1805.03356.

Su, J.-W., Chu, H.-K., Huang, J.-B., 2020. Instance-aware image colorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7968–7977.

Sun, J., Yuan, L., Jia, J., Shum, H.-Y., 2005. Image completion with structure propagation. In: ACM SIGGRAPH 2005 Papers. ACM New York, NY, USA, pp. 861–868.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826.

Tang, G., Müller, M., Gonzales, A.R., Sennrich, R., 2018. Why self-attention? A targeted evaluation of neural machine translation architectures. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4263–4272.

Uddin, S., Jung, Y.J., 2020. Global and local attention-based free-form image inpainting. Sensors 20 (11), 3204.

Vitoria, P., Raad, L., Ballester, C., 2020. ChromaGAN: adversarial picture colorization with semantic class distribution. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2445–2454.

Wadhwa, G., Dhall, A., Murala, S., Tariq, U., 2021. Hyperrealistic image inpainting with hypergraphs. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3912–3921.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.

Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J., 2018. Image inpainting via generative multi-column convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 331–340.

Weickert, J., 1999. Coherence-enhancing diffusion filtering. Int. J. Comput. Vis. 31 (2–3), 111–127.

Xu, Z., Sun, J., 2010. Image inpainting by patch propagation using patch sparsity. IEEE Trans. Image Process. 19 (5), 1153–1165.

Yan, Z., Li, X., Li, M., Zuo, W., Shan, S., 2018. Shift-net: Image inpainting via deep feature rearrangement. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 1–17.

Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H., 2017. High-resolution image inpainting using multi-scale neural patch synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6721–6729.

Yang, C., Song, Y., Liu, X., Tang, Q., Kuo, C.-C.J., 2018. Image inpainting using block-wise procedural training with annealed adversarial counterpart. arXiv preprint arXiv:1803.08943.

Yeh, R.A., Chen, C., Yian Lim, T., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N., 2017. Semantic image inpainting with deep generative models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5485–5493.

Yu, T., Guo, Z., Jin, X., Wu, S., Chen, Z., Li, W., Zhang, Z., Liu, S., 2020. Region normalization for image inpainting. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, pp. 12733–12740.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S., 2018. Generative image inpainting with contextual attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5505–5514.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S., 2019. Free-form image inpainting with gated convolution. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4471–4480.

Zeng, Y., Fu, J., Chao, H., Guo, B., 2019. Learning pyramid-context encoder network for high-quality image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1486–1494.

Zhang, H., Goodfellow, I., Metaxas, D., Odena, A., 2019. Self-attention generative adversarial networks. In: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, Vol. 97, PMLR, Long Beach, California, USA, pp. 7354–7363, URL http://proceedings.mlr.press/v97/zhang19d.html.

Zhang, H., Hu, Z., Luo, C., Zuo, W., Wang, M., 2018. Semantic image inpainting with progressive generative networks. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 1939–1947.

Zhang, R., Isola, P., Efros, A.A., 2016. Colorful image colorization. In: European Conference on Computer Vision. Springer, pp. 649–666.

Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595.

Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A., 2017. Real-time user-guided image colorization with learned deep priors. arXiv preprint arXiv:1705.02999.

Zhao, Y., Price, B., Cohen, S., Gurari, D., 2019. Guided image inpainting: Replacing an image region by pulling content from another image. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp. 1514–1523.

Zheng, C., Cham, T.-J., Cai, J., 2019. Pluralistic image completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1438–1447.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A., 2017. Places: A 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. 40 (6), 1452–1464.