Image Completion with Heterogeneously Filtered Spectral Hints

Xingqian Xu$^{1,3}$, Shant Navasardyan$^3$, Vahram Tadevosyan$^3$, Andranik Sargsyan$^3$, Yadong Mu$^2$, and Humphrey Shi$^{1,3}$

$^1$SHI Lab @ UIUC & UO, $^2$Peking University, $^3$Picsart AI Research (PAIR)

Abstract

Image completion with large-scale free-form missing regions is one of the most challenging tasks for the computer vision community. While researchers pursue better solutions, drawbacks such as pattern unawareness, blurry textures, and structure distortion remain noticeable, and thus leave space for improvement. To overcome these challenges, we propose a new StyleGAN-based image completion network, Spectral Hint GAN (SH-GAN), inside which a carefully designed spectral processing module, Spectral Hint Unit, is introduced. We also propose two novel 2D spectral processing strategies, Heterogeneous Filtering and Gaussian Split that well-fit modern deep learning models and may further be extended to other tasks. From our inclusive experiments, we demonstrate that our model can reach FID scores of 3.4134 and 7.0277 on the benchmark datasets FFHQ and Places2, and therefore outperforms prior works and reaches a new state-of-the-art. We also prove the effectiveness of our design via ablation studies, from which one may notice that the aforementioned challenges, i.e. pattern unawareness, blurry textures, and structure distortion, can be noticeably resolved. Our code will be open-sourced at: https://github.com/SHI-Labs/SH-GAN.

1. Introduction

Spectral analysis is a well-established research topic and has been intensely studied for decades [7, 8, 32, 35]. Its corresponding downstream techniques in remote sensing, telecommunication, and healthcare significantly affect our modern life. Earlier computer vision techniques largely adopted algorithms such as the Fourier transform [3, 38], wavelet transform [4, 15, 33, 50] and curvelet transform [47] for image denoising, anti-aliasing, restoration, and compression. Ever since the rapid growth of deep learning, spectral analysis on images has fallen from popularity largely due to the fact that the intriguing properties of the 2D frequency domain complicated many solutions for the content-based tasks. Nevertheless, as the progress of deep learning research goes deeper and wider, researchers start to re-focus on the image spectral analysis and its potential applications. Recent works, such as [9, 10, 40, 48, 54, 55, 57, 64], have shown that a network structure with spectral priors can be favorable in many tasks, including classification, segmentation, image synthesis, and super-resolution. These works will undoubtedly guide the future computer vision research in spectral analysis.
Despite the fast development of spectral-related deep learning methods, few works [9, 25, 48] have explored the potential of image completion with spectral priors, in which the major purpose of [9, 25] is still image synthesis. In the past few years, image completion heavily relied on feature extraction using CNNs and similarity-based patch matching techniques [36, 56, 58, 59, 61]. While these strategies have been proved useful in some scenarios, it remains a case-dependent approach due to its frequently shown structural distortion and texture artifacts. Meanwhile, the success of the StyleGAN series [24, 25, 26, 27] on generation tasks has established a robust baseline for many downstream tasks such as style transfer [1, 16, 39], GAN-inverse [37], latent space editing [17, 45, 49], and inpainting [63]. Among them, the recent image completion work CoModGAN [63] introduced the concept of co-modulation and pushed the performance to the next level.

Through our research, we compared several image completion works such as DeepFill [58, 59], LaMa [48], CoModGAN [63], etc. We noticed that these methods produce promising results in some cases but struggle in others. For instance, the patch-based approaches keep a good texture consistency for irregular textures with small textures (e.g., grass, wood, asphalt, etc.), but create large structural distortions, especially when the unknown masks are large. LaMa, on the other hand, utilizes a spectral transform module, namely FCC [10], and stands out in cases with strong pattern-like signals. However, it faces challenges in complex scenes and tends to create fast-repeating artifacts that are conceptually blurry. CoModGAN generates more natural-looking image content. However, CoModGAN is less bounded by image known regions; thus, it may ignore global patterns and create faulty objects.

Motivated by the prior works using spectral transform to handle low-level patterns and modulated generative block to handle high-level semantic, we introduce a novel approach: Spectral Hint GAN (SH-GAN) along with a new module: Spectral Hint Unit (SHU). Our goal is to minimize the summarized problems, i.e., pattern unawareness, blurry textures, and structural distortions and to maintain the natural balance between pattern and semantic consistency. Moreover, we also propose two new spectral transform strategies: Heterogeneous Filtering, and Gaussian Split. Heterogeneous Filtering aims to manipulate the 2D spectrum using a learnable smooth-varying function correlated with its frequency. Meanwhile, Gaussian Split is a spectral split algorithm that distributes information to different resolution scales for image synthesis. As a result, our FID performances are 3.4134 on FFHQ [26] and 7.0277 on Places2 [65], outperforming CoModGAN and other prior works and reaching a new state-of-the-art. We also demonstrate that our model outperforms LaMa and the other works on the narrow and wide masks using a LaMa-style evaluation scheme. Lastly, we perform ablation studies on SH-GAN, from which we clearly see the performance gain using SHU and the proposed spectral transform strategies, i.e. Heterogeneous Filtering and Gaussian Split.

In summary, the main contributions of our work are the following:

- We propose a novel spectral-aware StyleGAN-based image completion network, Spectral Hint GAN (SH-GAN), in which a new module, Spectral Hint Unit (SHU), is introduced.
- We also bring out two new spectral processing strategies: Heterogeneous Filtering and Gaussian Split. These strategies aim to enhance the 2D spectral signals and hierarchically fuse them inside the synthesis network.
- The FID scores of SH-GAN outperform the state-of-the-art on two popular benchmark datasets: FFHQ and Places2. Meanwhile, we perform inclusive studies, through which we demonstrate the effectiveness of our new designs.

2. Related Works

2.1. Spectral Research in Computer Vision

Early image spectral research largely focused on low-level vision such as compression [2, 3], restoration [50], and denoising [4, 15, 47, 51]. In 1971, Tsai and Huang proposed Transform Coding [3] using the discrete cosine transform, which was later extended into the well-known JPEG format [2]. Huang [50] also initiated the pioneering work for image enhancement and restoration using multi-frame discrete Fourier transform (DFT) and inverse-DFT. Popular approaches of image denoising utilize fast Fourier transform (FFT) [51] or wavelet transform [4, 15]. In [47], Jean et al. proposed two new frequency-domain tools: ridgelet transform and curvelet transform, by which they could recover images from noise with higher perceptual quality than prior works. In recent years, researchers have shown increasing enthusiasm for spectral neural networks. [43] was one of the first works that combined spectral layers with CNN, in which it proposed spectral pooling for dimension reduction. Another work [55] proposed SyncSpecCNN in which a set of 3D features were eigen-decomposed and passed through a CNN for 3D part segmentation. For super-resolution, [64] decomposed tensors with the wavelet basis and transformed them with fully connected layers. [40] explored the inductive bias of CNNs towards low-frequency signals. A similar discovery was mentioned in [9], in which the author performed analysis on the frequency domain of a GAN network. Recently, Chi et al. proposed Fast Fourier Convolution (FFC) [10], in which tensors were converted between spatial and frequency domain using FFT and inverse-FFT. Chi et al. also showed that FFC could substitute regular
Figure 2: This figure shows the structure of FFC [10] (left) and SHU (right). Different from FFC, our SHU does not use any external convolution layers. In the spectral transform, SHU utilizes HeFilter to transform spectral tensors after ReLU, while FFC directly connects ReLU outputs to iFFT. To make things compacted, we do not include Gaussian Split in this figure.

Residual Blocks [18] and reach better performance in classification.

2.2. Image Completion

The goal of image completion is to synthesize image content for missing regions. Traditional approaches performed gray-level gradient extension [5], image quilting [13], and patch-based methods [6, 12, 14]. Notwithstanding the success of these approaches in the cases of simple and highly-textured backgrounds, they fail to recover missing semantics and complex structures. Since the popularity of deep learning, [29, 42, 53] were the first groups to design deep network architectures for inpainting. Satoshi et al. [21] utilized dilated convolutions and adversarial training [30] inpainted face images using semantic maps as guidance. To address the vanilla convolutions’ drawback of treating all pixels equally, Liu et al. [31] introduced partial convolutions in which unknown elements in the input tensor are excluded from the calculations. Yu et al. further improved the performance with contextual attention [58] and gated convolutions [59]. Navasardyan & Ohanyan [36] proposed the onion convolutions, in which neighboring patches could be searched and relocated simultaneously with the convolution operation. Sharing a similar spirit with [6, 58], HiFill [56] generated contextual residual to fill in higher resolution textures. Another work, CR-Fill [61] proposed the CR loss, a patch similarity loss designed to reinforce contextual consistency. Zhu et al. [66] introduced the MADF module and cascaded refinement decoders. Zeng et al. [60] introduced the AOT block and utilized soft mask-guided PatchGAN [22] for the network training. Suvorov et al. introduced LaMa, which is a U-shape structure with FFCs [10]. Zhao et al. [63] proposed CoModGAN along with the co-modulation idea on top of the StyleGAN [24, 26, 27]. All these works have achieved plausible results on faces and natural images using free-form masks. The concurrent works in diffusion models [20, 46] can also be extended to inpainting tasks in which LDM [44] and DALL-E2 [41] shows promising results at a higher computational cost during inference time.

3. Method

In this section, we illustrate the key designs of our work, including Spectral Hint Unit (SHU), Heterogeneous Filtering Layer (HeFilter), Gaussian Split, and the overall GAN architecture. The definition of image completion is to restore an RGB image \( I_{fake} \) from a masked color image \( I = I_{real} \odot M \), where \( I_{real} \) is the ground-truth image and \( M \) is the mask, in which the known pixels have value 1 and the unknown pixels are represented with zeros.

3.1. Spectral Transform with SHU

Spectral Hint Unit (SHU) transforms tensors using a neat FFT−Network−iFFT pipeline (see Fig. 2). The recent work FFC [10] also suggested a similar structure in which the author blended spectral transform inside a densely connected convolutional network. Different from FFC, SHU is light-weighted because it has no extra convolutions outside the spectral transform. More precisely, let \( x \in \mathbb{R}^{N \times H \times W} \) be an \( N \)-channeled tensor with height and width equal to \( H \) and \( W \) respectively. Then the output of SHU is the tensor \( x' \) with the same dimensionality formed by the following way:

\[
x' = \text{concat}(x_{[\ldots N-K]}, x_{[N-K\ldots N]} + f(x_{[N-K\ldots N]}))
\]

\[
f = \text{iFFT} \circ g \circ \text{FFT}
\]

\[
g = \text{HeFilter} \circ \text{ReLU} \circ \text{Conv}1 \times 1
\]

Our design fits GAN training for the following reasons:

a) local operations such as a convolution should be well-
Figure 3: Graphic explanations on manipulating spectral tensors using different types of layers. Conv1 \( \times 1 \) is a homogeneous operation that treats the complex vector-space equally over the 2D frequency domain. ReLU works as a band-pass filter by changing negative components to zeros. Our HeFilter applies a smooth varying mapping function on the frequency domain and manipulates complex vectors based on their spectral location.

Figure 4: A graphic illustration on the Vanilla Split. Our Gaussian Split is an extended version adding a Gaussian mask on each split level for better anti-aliasing. Due to the limited space, we only show the Vanilla Split here.

handled by the synthesis network with modulated convolutions; b) GAN training should maintain a subtle balance between the spectral and spatial transforms, and it should not overwhelm tensors with spectral information.

Like the spectral transform of FFC [10], Conv1 \( \times 1 \) maps between two complex vector-spaces uniformly in the frequency domain. ReLU is the non-linearity operation that filters out all negative components in the vector-space. Lastly, HeFilter performs heterogeneous filtering in which the mapping is a smooth-varying function over the 2D spectral location. We will describe HeFilter with more details in the next subsection. In summary, the spectral transform of SHU is as follows:

\[
\begin{align*}
\text{a) } & \mathbb{R}^{K \times H \times W} \rightarrow \mathbb{C}^{K \times H \times W} \quad \text{(FFT)} \\
\text{b) } & \mathbb{C}^{K \times H \times \frac{W}{2}} \rightarrow \mathbb{C}^{K \times H \times \frac{W}{2}} \quad \text{(HeFilter } \circ \text{ ReLU } \circ \text{ Conv)} \\
\text{c) } & \mathbb{C}^{K \times H \times \frac{W}{2}} \rightarrow \mathbb{R}^{K \times H \times W} \quad \text{(iFFT)}
\end{align*}
\]

3.2. Heterogeneous Filtering and Gaussian Split

As mentioned earlier, one of the contributions of this work is to introduce two novel spectral processing strategies: Heterogeneous Filtering and Gaussian Split, that well-fit the deep learning training scheme.

Heterogeneous Filtering. Recall that FFC [10] transforms spectral tensors with Conv1 \( \times 1 \) and ReLU, which can also be viewed as a homogeneous operation and a band-pass filter. In many cases, ReLU is a necessary step but not a recommended end operation for spectral transform with the following reasons: a) it deactivates a frequency band with no recovery; b) it introduces aliasing effects due to non-smoothness; and c) it responds according to magnitude instead of location (i.e. frequency band). Therefore, we create HeFilter, inside of which a heterogeneous filtering strategy is introduced transforming the complex vector-space via a smooth-varying function over the frequency domain. More precisely, HeFilter learns several weight matrices scattered on an even-spaced 2D frequency domain. During propagation, HeFilter linearly interpolates these weights and multiplies them with the corresponding spectral vector. Figure 3 explains the characteristic of Conv1 \( \times 1 \), ReLU, and HeFilter in the spectral transform. We prepare a total of \( 3 \times 2 \) weights on the 2D spectral domain. The asymmetry along each dimension is because the FFT of \( \mathbb{R}^{K \times H \times W} \) is \( \mathbb{C}^{K \times H \times \frac{W}{2}} \), and the skipped half is the reflected complex conjugate. We do not impose constraints on learning these weights, thus HeFilter can be low-pass, band-pass or high-pass depend-
A fundamental property of the Fourier Transform $\mathcal{F}$ is linearity, in which the FT of the addition of the functions $f_1$ and $f_2$ equals the addition of the individual FTs of $f_1$ and $f_2$ (see Eq. 1).

$$f(x) \leftrightarrow \tilde{f}(\omega)$$
$$\alpha f_1(x) + \beta f_2(x) \leftrightarrow \alpha \tilde{f}_1(\omega) + \beta \tilde{f}_2(\omega)$$  \hspace{1cm} (1)

Utilizing the property mentioned above, we can split any spectral signal $\hat{x} = \mathcal{F}(x)$ into several sub-signals $\hat{x}_i, i \in \{1 \ldots n\}$. As long as $\sum_i \hat{x}_i = \hat{x}$, we expect no information loss on $x = \sum_i \mathcal{F}^{-1}(\hat{x}_i)$, and this property holds for FFT on discrete tensors. For convenience, we use the same set of symbols $\hat{x}, \hat{x}_i$ representing spectral tensors. The graphic explanation of the split is highlighted in Figure 4, from which one may notice that our decomposition, like the Wavelet Transform, automatically segregates signals based on their frequency bands. For example, a two-level Vanilla Split is formulated in Eq. 2 and 3, in which we migrate all low frequency values from the large tensor to the small one.

$$\hat{x} \in \mathbb{C}^{H \times W} \xrightarrow{\text{split}} \left( \hat{x}_1 \in \mathbb{C}^{H \times \frac{W}{4}}, \hat{x}_2 \in \mathbb{C}^{\frac{H}{4} \times \frac{W}{4}} \right)$$ \hspace{1cm} (2)

$$\hat{x}_1[i,j] = \begin{cases} \hat{x}[i,j] & (i, j) \notin \left( \frac{H}{4}, \frac{3H}{4}, 0 \ldots \frac{W}{4} \right) \\ 0 & (i, j) \in \left( \frac{H}{4}, \frac{3H}{4}, 0 \ldots \frac{W}{4} \right) \end{cases}$$ \hspace{1cm} (Vanilla)  \hspace{1cm} (3)

$$\hat{x}_2 = \hat{x}[\frac{H}{4}, \frac{3H}{4}, 0 \ldots \frac{W}{4}] \times \mathcal{N}(\frac{H}{4}, \frac{3H}{4}, 0 \ldots \frac{W}{4})$$ \hspace{1cm} (Gaussian) \hspace{1cm} (4)

3.3. Network Architecture

Similar to CoModGAN [63], our model is a U-shape architecture containing an encoder and a synthesis network. As Figure 5 shows, we first pass the masked input image $I$ into the encoder, inside which $I$ is encoded into a set of feature maps $x_i[i]$ with resolution $i$, and a global vector $w_0$. We then split $K$ channels from $x_i[i]$ ($K = 32, i = 64$), and pass it to SHU for spectral transform. Inside SHU, we create a series of wavelet-like feature maps $x_i, i \in \{4, \ldots, 64\}$ using Gaussian Split, in which the low-frequency information is encoded in the low-resolution feature maps (e.g. $x_4$).
<table>
<thead>
<tr>
<th>Method</th>
<th>FFHQ 256</th>
<th>Places2 256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID(↓)</td>
<td>LPIPS(↓)</td>
</tr>
<tr>
<td>CoModGan (small)</td>
<td>5.0184</td>
<td>0.2579</td>
</tr>
<tr>
<td>CoModGan (official)</td>
<td>4.7755</td>
<td>0.2568</td>
</tr>
<tr>
<td>LaMa</td>
<td>32.7035</td>
<td>0.5900</td>
</tr>
<tr>
<td>DeepFillV2</td>
<td>50.9323</td>
<td>0.3204</td>
</tr>
<tr>
<td>CR-Fill</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Onion-Conv</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MADF</td>
<td>33.6207</td>
<td>0.2800</td>
</tr>
<tr>
<td>AOT-GAN</td>
<td>73.7962</td>
<td>0.4270</td>
</tr>
<tr>
<td>(ours - small)</td>
<td>4.8225</td>
<td>0.2558</td>
</tr>
<tr>
<td>(ours - regular)</td>
<td>4.3459</td>
<td>0.2542</td>
</tr>
</tbody>
</table>

Table 1: This table compares the performance of prior models with our SH-GAN on datasets FFHQ and Places2 under resolution 256.

<table>
<thead>
<tr>
<th>Method</th>
<th>FFHQ 512</th>
<th>Places2 512</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID(↓)</td>
<td>LPIPS(↓)</td>
</tr>
<tr>
<td>CoModGan (small)</td>
<td>3.9420</td>
<td>0.2497</td>
</tr>
<tr>
<td>CoModGan (official)</td>
<td>3.6996</td>
<td>0.2469</td>
</tr>
<tr>
<td>LaMa</td>
<td>19.5577</td>
<td>0.2871</td>
</tr>
<tr>
<td>DeepFillV2</td>
<td>32.8696</td>
<td>0.3283</td>
</tr>
<tr>
<td>CR-Fill</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Onion-Conv</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MADF</td>
<td>17.1962</td>
<td>0.2688</td>
</tr>
<tr>
<td>AOT-GAN</td>
<td>36.1344</td>
<td>0.3403</td>
</tr>
<tr>
<td>(ours - small)</td>
<td>3.7460</td>
<td>0.2491</td>
</tr>
<tr>
<td>(ours - regular)</td>
<td>3.4134</td>
<td>0.2447</td>
</tr>
</tbody>
</table>

Table 2: This table compares the performance of prior models with our SH-GAN on datasets FFHQ and Places2 under resolution 512.

and the high-frequency information is encoded in the high-resolution feature maps (e.g., $x_{64}$). We then add back $x_i$ to the corresponding $x^{[i]}$. For $i > 64$, we directly pass $x^{[i]}$ to synthesis blocks. Recall that we adopt StyleGAN2 [24] as our synthesis network. We then modulate the synthesis network using the concatenation of $w$ and $w_0$, in which $w$ is the projected vector of the latent code $l$ using the mapping network, and connecting $x^{[i]}$ with each synthesis block via addition.

During the training time, we use a StyleGAN2 discriminator for our adversarial loss. We also use path length regularization on the generator with $w_{pl} = 2$, and $R_1$ regularization on the discriminator with $\gamma = 10$. Other training details can be found in Section 4.2.

3.4. Mask generation

We use the same free-form mask generation algorithm as DeepFillV2 [59] and CoModGAN [63]. These masks are drawn using multiple brush strokes and rectangles. The width of the brush stroke is sampled from $U(12, 48)$ and the number of strokes is chosen randomly from $U(0, 20)$, where $U(\cdot)$ represents the discrete uniform distribution. Meanwhile, we sample $U(0, 5)$ rectangles up to the full size of the input image, and $U(0, 10)$ rectangles up to the half-size.

For more details, please see supplementary.

4. Experiments

This section goes through the details about our dataset, metrics, settings, experiments, and other studies. We provide comprehensive comparisons between our SH-GAN and other prior works through scores and images.

4.1. Datasets and Metrics

We use three datasets: FFHQ [26], Places2 [65], and DTD [11]. FFHQ contains 70,000 high-resolution well-aligned face images, in which we split out 60,000 images for training and use the remaining 10,000 images for validation. Places2 contains 8,026,628 images in its training set and 36,500 images in its validation set. The contents of Places2 are regular scenes and objects. We maintain the original train-validation split for our experiments. DTD contains 5,640 categorized texture images, among which 3,760 are from the train and validation set, and 1,880 are from the test set. We train our models using the train and validation sets and evaluate them using the test set.
distance between the distributions of real and synthetic features. Besides, we also adopt Learned Perceptual Image Patch Similarity (LPIPS) [62], Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) [52] to gauge models from different angles. We will show all the metric scores in Section 4.3.

4.2. Training Details

Many of the training settings of this work closely follow CoModGAN [63] and StyleGAN2 [24]. We use Adam [28] optimizer with $\beta = (0, 0.99)$ for our generator and discriminator. The learning rates are 0.001 for FFHQ/Places2, and 0.002 for DTD. Besides, the training lengths are 25 million images on FFHQ, 50 million on Places2, and 10 million on DTD. We apply both path length regularization and $R_1$ regularization during the training, in which we set $w_{pl} = 2$ and $\gamma = 10$. The batch size is 32 for all models on all datasets. Like StyleGAN2, we compute the exponential moving average of our generator with momentum 0.99993 (i.e. half-decay at 20,000 images with batch size 32). As shown in Tables 1 and 2, we evaluate SH-GAN with small and standard settings under resolutions 256 and 512. SH-GAN (small) is a reduced version of SH-GAN with base channel decreased from 32,768 to 16,384 (See supplementary). Such small and standard settings match the small and official versions of CoModGAN in terms of the model size. We train the small model with 4 GPUs and the standard model with 8 GPUs. Besides, we use 2080Ti for resolution 256 and A100 for resolution 512.

4.3. Results Comparison

Tables 1 and 2 compare the performances of SH-GAN with other prior works [48, 59, 60, 61, 63, 66] on FFHQ and Places2. As mentioned in Section 3.4, we adopt the free-from masks originated from the CoModGAN paper. Among the four metrics, FID/LPIPS gauge perception quality, and PSNR/SSIM gauge pixel accuracy. Note that these metrics reveal image quality in a different way so they may disagree in numbers. For most of the prior works, except for CoModGAN, we have downloaded the official code and models for evaluation. We re-implemented CoModGAN in
Pytorch, and trained and tested the replicated version. The FID scores on the replicated CoModGAN match the FID scores in the original paper. As a result, SH-GAN reaches 4.3459 and 7.5036 on FFHQ and Places2 datasets respectively for the resolution 256, and 3.5713 and 7.8482 for the resolution 512. SH-GAN surpasses all other approaches in terms of FID and becomes the new state-of-the-art.

In addition to the free-form mask experiments, we also tried other types of masks such as the LaMa-style [48] narrow and wide masks. These detailed performance can be found in our supplementary material.

4.4. Extended Studies

For this section, we carry out extended experiments to justify our new designs (i.e. SHU, Heterogeneous Filtering, and Gaussian Split) over the prior challenges such as pattern unawareness, blurry texture, and structure distortion.

Our first experiment is an ablation study using the DTD dataset, in which we trained several models and excluded SHU, Heterogeneous Filter, or Gaussian Split, respectively. We focused on the DTD dataset because texture images are highly structured images that could make obvious cases for comparison. In Table 3, we show that the full version of our model performed FID 48.58, lower than the model without Gaussian Split (i.e. ours - no_GS) by 1.48, lower than the model without Heterogeneous Filtering (i.e. ours - no_HF) by 1.83, and lower than the baseline (i.e. CoModGAN [63]) by 3.35. The LPIPS, PSNR and SSIM scores agree with our FID score. Moreover, we clearly show in Figure 7 that our model generates sharp and robust patterns even when the mask coverage is large. Our HeFilter is very helpful in cases with complex structures and our Gaussian Split helps to remove the aliasing effect to make the pattern sharper.

<table>
<thead>
<tr>
<th>Models</th>
<th>SHU</th>
<th>HF</th>
<th>GS</th>
<th>FID(↓)</th>
<th>LPIPS(↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>51.9289</td>
<td>0.3628</td>
</tr>
<tr>
<td>ours - no_HF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>50.4074</td>
<td>0.3614</td>
</tr>
<tr>
<td>ours - no_GS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>50.0634</td>
<td>0.3573</td>
</tr>
<tr>
<td>ours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td><strong>48.5814</strong></td>
<td><strong>0.3519</strong></td>
</tr>
</tbody>
</table>

Table 3: The ablation study on DTD [11] with SHU, Heterogeneous Filtering (HF), and Gaussian Split (GS) removed respectively.

In our second experiment, we extract the skip feature maps $x[i]$ generated by the encoder, and the intermediate feature maps $y[i]$ generated by the synthesis blocks, $i \in \{16, 32, 64\}$. We then compute the 2-norm of each feature map along the channel axis. To make clear comparisons, our SHU is only connected on resolution 32 without splitting. Figure 8 shows the impact of SHU over these features, in which readers may notice that SHU provides critical hints on patterns to its downstream synthesis blocks for texture generation.

5. Conclusions

We introduce SH-GAN, a novel image completion approach that transforms deep features with spectral hints in the modulated GAN framework. Throughout this paper, we reveal the details of our newly designed module: SHU, and introduce our new spectral transform strategies: Heterogeneous Filtering and Gaussian Split. With inclusive experiments, we show that all our designs are very useful in solving challenging inpainting cases with large-scale free-form missing regions. We believe that our SHU and spectral transform strategies are worth exploring further in other computer vision tasks.
References


computer vision and pattern recognition, pages 5505–5514, 2018.


