TransInpaint: Transformer-based Image Inpainting with Context Adaptation

Pourya Shamsolmoali[1], Masoumeh Zareapoor[2], Eric Granger[3]
[1]East China Normal University, China, [2]Shanghai JiaoTong University, China, [3]ETS Montreal, Canada
pshams55@gmil.com, mzarea222@gmail.com, Eric.Granger@etsmtl.ca

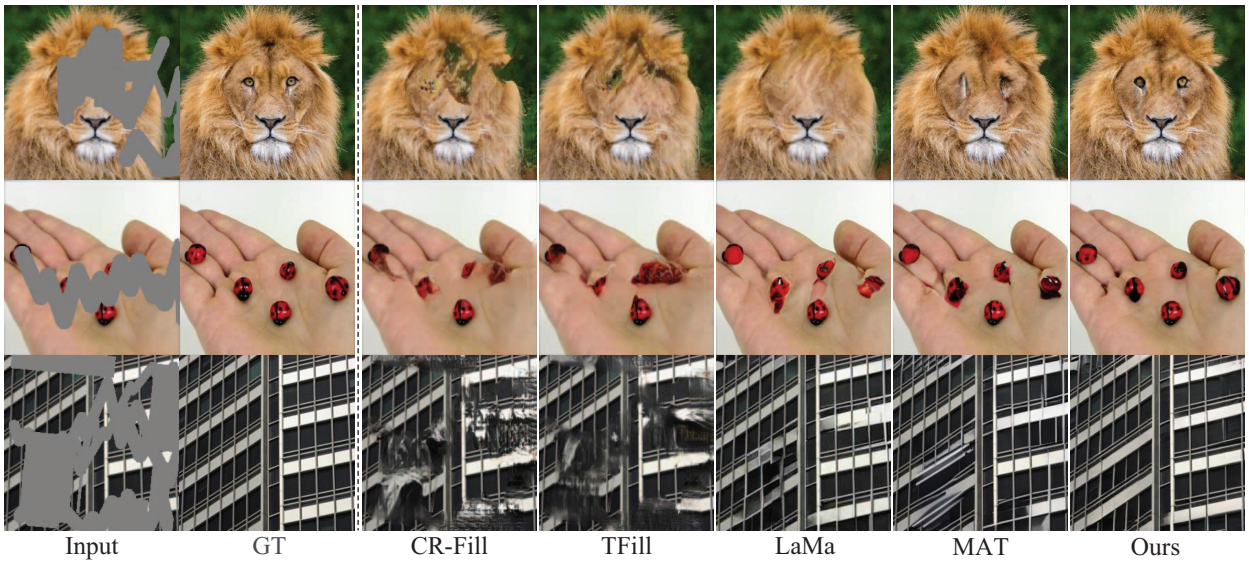

Figure 1: A visual comparison of inpainting methods. Existing models struggle when applied to complex-structured images such as animal faces or images that contain multiple objects.

Abstract

Image inpainting aims to generate realistic content for missing regions of an image. Existing methods often struggle to produce visually coherent content for missing regions of an image, which results in blurry or distorted structures around the damaged areas. These methods rely on surrounding texture information and have difficulty in generating content that harmonizes well with the broader context of the image. To address this limitation, we propose a novel model that generates plausible content for missing regions while ensuring that the generated content is consistent with the overall context of the original image. In particular, we introduce a novel context-adaptive transformer for image inpainting (TransInpaint) that relies on the visible content and the position of the missing regions. Additionally, we design a texture enhancement network that combines skip features from the encoder with the coarse features produced by the generator, yielding a more comprehensive and robust representation of image content. Based on extensive evaluations on challenging datasets, our proposed TransInpaint outperforms the cutting-edge generative models for image inpainting in terms of quality, textures, and structures.

1. Introduction

Image inpainting (a.k.a. completion) is a challenging problem in computer vision due to its broad range of applications and ill-posed nature. Its goal is to generate plausible content for damaged or missing regions of an image. It has a wide range of real-world applications, including image manipulation [21] and object removal [33]. The most challenging aspect of image completion is predicting realistic content for distorted regions while maintaining consistency and coherence across the entire image. However, this challenge is accentuated when dealing with complex patterns or structures, such as images that contain multiple objects or animals' faces [15, 16]. Because these images have intri-

cate details and fine-grained textures, which can be difficult to accurately capture and reconstruct in the completed image. Fig. 1 illustrates the limitations of current inpainting methods in generating high-quality content for regions that contain complex structures. For example, CR-Fill [48] and TFill [50] are unable to generate an accurate replacement for the missing regions, resulting in an unrealistic and incoherent final image. While MAT [15] performs better in generating high-resolution completed results, it still struggles to generate textures that smoothly integrate with the overall image context, especially when used on images with complex patterns like those seen on the building. Furthermore, existing image completion methods often rely on surrounding textures to fill in missing regions, which may not be effective for all types of images. For instance, in Fig. 1 (first row), we can observe that MAT, struggles to accurately reconstruct the lion's eyes. This is mainly due to the lack of contextual information in the surrounding regions, which prevents the model from accurately capturing the structure and shape of the eyes in the lion's face and leads to an incomplete structure estimation.

Convolutional neural network (CNN)-based image completion methods have shown promising results on small and aligned masked images [25, 48]. However, when dealing with images that contain complex structures, these methods are unable to capture the semantic relationships between distant regions. This difficulty is due to the inherent properties of CNNs, such as the slow growth of their effective receptive field and the dominance of nearby pixels. To address these challenges, recent studies have explored the use of more flexible models, such as transformers [5, 12, 22, 32, 36]. The transformer is suitable for non-local modeling and can effectively attend to relevant features across the entire image, even those far from the missing region. While some recent works have utilized transformers for completion [37, 43, 50], they have mainly focused on generating low-resolution predictions, which can result in coarse image structures and compromised quality. Other studies have used auto-regressive transformers to handle complex structures [14, 45, 47], which are well-suited for small and regular masks. However, when it comes to missing regions with arbitrary shapes and sizes, these approaches cannot be as effective [16].

In this paper, TransInpaint is introduced to effectively generate plausible content for missing regions of an image while maintaining the original structure of the image. Our model proposes a Context-adaptive Transformer (CT) to analyze the image context by capturing the intricate co-occurrence features and using this information to estimate the type and shape of the missing region. Additionally, a texture enhancement network (TENet) is proposed that combines the generated coarse features and skip features from the encoder to effectively build repeating textures. As shown in Fig. 1, TransInpaint generates high-quality missing content that seamlessly blends with the overall image structure. Our main contributions can be summarized as: **(1)** A novel image completion architecture is proposed that includes a CT network for predicting the missing contents. We also introduce TENet which integrates local and global layers into a compact and discriminative representation. This helps the model effectively handle complex structures and improve the quality of the generated images. **(2)** Empirical results on challenging CelebA, Places2, and ImageNet datasets indicate that our proposed TransInpaint outperforms previous completion methods by a large margin across various evaluation metrics. Notably, our model provides higher completion fidelity when dealing with images containing multiple objects and complex patterns.

2. Related Works

Image completion, compared to image generation [30], is a more challenging task, particularly when substantial portions of the image are missing. Traditional diffusion-based approaches [1, 2] transfer information from nearby, undamaged regions to the missing areas. Patch-based or exemplar-based methods [4, 13] select patches with similar appearances to fill in the missing regions. However, diffusion-based methods can introduce blurs and tend to fail when the missing regions are large [15], while patch-based methods struggle with completing large missing regions in complex scenes as they rely heavily on the patch-wise matching of low-level features [47]. Deep learning has achieved remarkable progress in image inpainting in recent years. Contextual information is exploited by Pathak et al. [24] using adversarial training in combination with an autoencoder-based architecture to fill incomplete images. Several variations [18, 42, 10] of the CNN architecture have been proposed for image inpainting. Besides, more advanced learning methods have been introduced, such as local and global discrimination [9], gated convolution [46], and contextual attention [19, 41], etc. Multi-stage generation, which leverages intermediate clues like object edges [23], structures [27], and semantic segmentation maps [33], has gained significant attention for its ability to generate realistic textures and reasonable structures. Moreover, recent research has focused on addressing more challenging image completion tasks, such as filling large holes with irregular shapes. LaMa [34] is a one-stage network that combines multi-scale receptive fields to capture both global and local context information and generate patterns for mission regions. However, LaMa can generate faded or blurry structures when the missing region is large and extends beyond the object boundary. CR-Fill [48] proposes a contextual framework by incorporating a learnable loss function, but fails to preserve the texture of the original input image and produces unrealistic outputs. Denoising diffusion

models [20, 28, 38, 40] and autoregressive transformer approaches [17, 37, 47] show promise in generating realistic content by using accurate likelihood computation and iterative sampling. However, their performances are limited by variational learning and raster scan-order-based generation, which make it difficult to generate realistic content when the input image has a large and irregularly missing part. On the other hand, diffusion models have recently gained attention due to their ability to generate high-quality images [16, 44].

TFill [50] is a transformer-based architecture designed to capture long-range dependence between pixels in the encoder, but its performance heavily relies on the size of the missing regions. In cases where the missing regions are large or contain complex structures, TFill cannot produce plausible results, as shown in Fig. 1 and reported by [31]. MAT [15] is a GAN-based image completion model that contains transformer blocks with style manipulation and a mask-updating approach. However, MAT suffers from distortion at the interface between the produced instances and their surroundings [16], especially in complex images with intricate patterns and structures, where the completed region may appear inconsistent with the surrounding context. Our TransInpaint model differs from previous approaches in that it improves visual consistency and generates high-quality outputs that are more semantically plausible, especially at the boundaries between the generated and unmasked regions. This is achieved by leveraging the strengths of our CT and TENet, which allow for better modeling of the context of the input image and the generation of more realistic and visually coherent outputs.

3. Proposed Method

Our proposed model aims to reconstruct distorted images where the visual instances have been damaged. This is a challenging task as the model not only needs to produce plausible instances but also ensure that the constructed instances flawlessly match the rest of the image. Our TransInpaint model addresses this problem in two steps. First, our context-adaptive transformer (CT) is used to reconstruct the masked image by determining a contextually relevant instance. This is achieved by adapting the object detection with the Detection Transformer (DETR) [3] to identify the missing instances. Second, the reconstructed and masked images are fed to another CNN (TENet), which uses appearance priors and unmasked pixels to replenish texture details and convert the masked input into a realistic image. The TransInpaint pipeline is shown in Fig. 2.

(A) Predicting Missed Instances: TransInpaint uses the pre-trained DETR model [3] to predict the classes of instances and determines the relationships between objects and content in a given scene. The input (non-masked) image is denoted by x, while the masked image is represented by x_m. Let $P = [p_1, ..., p_i]$ and $c = [c_1, ..., c_i]^\top$ be the bounding box (BB) coordinates and object classes of the visible instances, respectively, that are extracted from the segmentation map $S_M = DETR(x_m)$, and i is the expected number of instances. The CT network uses extracted object classes c to create learnable input tokens. These tokens are then concatenated with the tokens for the masked region to form the input sequence for the transformer. The transformer generates a probability distribution over the possible classes for the missing context y_{out}, based on the available context y_{in} and the masked region. The class with the highest probability is selected as the predicted class for y_{out}. This process allows the network to determine the most likely class for the missing context based on the available context and the masked region. Once the class of the missing context is determined, appearance priors are used to generate the appearance of the missing instance. Indeed, the appearance priors are learned from the unmasked pixels of the input image and the available instances P, using our network. These appearance features are then combined with the predicted class of the missing context to generate the appearance of the missing instance. Another solution is to use DETR object queries as input tokens directly, which yields lower performance compared to using learnable class embedding (see "TransInpaint w/o CT" in Table 1).

The positional encoding provides location information about the available instances to the learnable class embedding to improve the accuracy of the prediction. Positional encoding vectors are generated using a sigmoid activation function applied to a linear layer that takes as input the normalized center coordinates, height (H), and width (W) of the bounding boxes (BB). The CT network computes the following functions:

$$
\begin{aligned}
z_0 &= \mathrm{E}_{\mathrm{cor}} + \mathrm{E}_{\mathrm{class}} = \mathrm{MLP}(c') + \rho(\mathrm{MLP}(P')), z_0 \in \mathbb{R}^{(i+1)\times v} \\
z'_t &= \mathrm{MSA}(\mathrm{LN}(z_{t-1})) + z_{t-1}, \qquad t = 1, ..., T \\
z_t &= \mathrm{MLP}(\mathrm{LN}(z'_t)) + z'_t, \qquad t = 1, ..., T \\
y &= \mathrm{LN}(z^0_T),
\end{aligned}
\tag{1}
$$

where MLP is a multi-layer perceptron, MSA is multi-head self-attention, LN denotes layer normalization, ρ represents a sigmoid activation function, v denotes the embedding vectors' dimension, P' is $[p_0] \cup P$, $[p_0]$ denotes BB coordinate for the masked region, and c' is $[c_0] \cup c$ in which c_0 represents additional class tokens for contents that are lost. CT has eight heads and twelve layers of transformer encoders $(T = 12)$. The output of the last transformer layer is then projected to an element-wise distribution using 512 components of visual vocabulary. Additionally, the model uses the masked language (ML) objective, which is similar to the one used by DETR. Following the ML objective, the input sequence is discretized, and the indexes of the missing region tokens are represented by $I = h_1, h_2, ..., h_J$, where J is the total number of missing region tokens. The model is

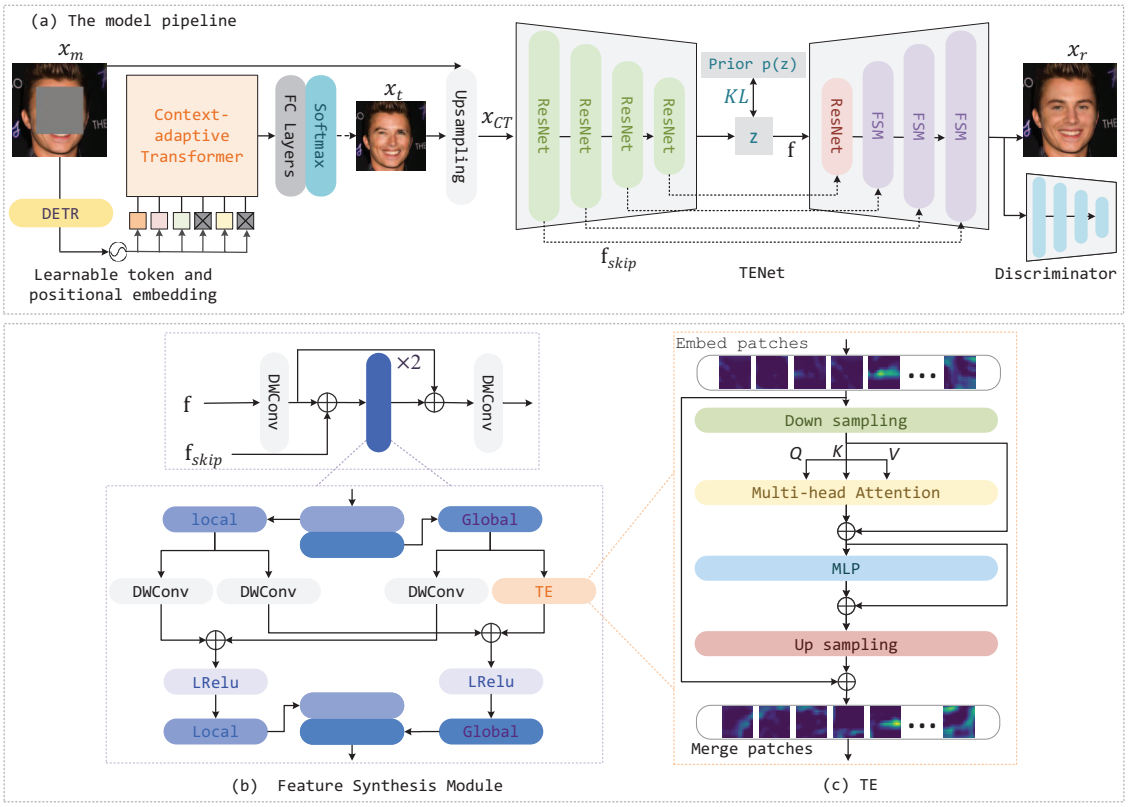

Figure 2: Our proposed TransInpaint pipeline consists of two steps: a CT, and a TENet. The CT network detects lost instances and damaged regions in the input image, generating appearance priors through sampling. The TENet then uses these appearance priors, along with the CT output, to generate high-quality and natural-looking outputs. It achieves this through a combination of Down/Up ResNet blocks, Feature Synthesis, and TE modules.

trained to minimize the negative log-likelihood of the missing region tokens (X_I) based on the visible region tokens (X_{-I}) in the coarse prior X. This means that the model is trained to predict the missing region tokens based on the information provided by the visible region tokens as,

$$\ell_{\mathrm{ML}} = \mathop{\mathbb{E}}_{\mathrm{X}} \left[\frac{1}{J} \sum_{j=1}^{J} -\log\ p(\varkappa_{h_J} | X_{-I}, \theta) \right], \qquad (2)$$

where θ denotes the parameters of the transformer. It is important to highlight that in our model, the combination of the appearance priors and unmasked pixels provides valuable information to generate texture details that are consistent with the overall structure and appearance of the completed image.

Sampling Method. In this section, we introduce how to obtain realistic appearance priors using the proposed CT network. However, directly sampling the entire set of masked positions can lead to unrealistic results because the CT learns the distribution of the missing region tokens based on the visible region tokens. Indeed, sampling them independently can result in unnatural-looking outputs. To address this issue, the missing region tokens are sampled using a Gibbs sampling procedure that accounts for the correlations between the missing region tokens. During each iteration of Gibbs sampling, a grid position from $p(\varkappa_{h_J} | X_{-I}, X_{<h_J}, \theta)$ is sampled using the top-k predicted components, where $X_{<h_J}$ represents the previously generated tokens. The sampled token is then replaced with its corresponding masked token, and this process is continued until all missing region tokens have been filled. The positions are selected sequentially, similar to the PixelCNN [35]. After sampling, a set of completed token sequences is obtained. The appearance priors $X \in \mathbb{R}^{\mathbb{L}\times 3}$ are then reconstructed using a vocabulary query for each discrete sequence derived from the CT.

(B) Texture Enhancement Network (TENet): After obtaining the low-dimensional appearance priors, we transform X into x_t for further processing. The challenge now is to learn a deterministic mapping that can re-scale x_t to its original resolution and maintain consistency between the missed regions and the unmasked regions. To achieve this, we propose TENet guided by x_m to generate high-quality

features from the reconstructed appearance.

$$x_{CT} = \mathcal{B}_\delta(x_t^{\uparrow} \otimes x_m) \in \mathbb{R}^{H,W\times 3}, \tag{3}$$

where $\mathcal{B}$ is the network's backbone that is parametrized by δ, $x_t^{\uparrow}$ is the result of x_t bilinear interpolation, and $\otimes$ is the function of concatenation along channel dimension. Then, we feed the x_{CT} to our TENet. To create realistic textures and semantic structures within the missing regions of the image, we introduce the Feature Synthesis module (FSM) as shown in Fig. 2 (b). FSM uses skip connections between encoder and decoder layers that have the same resolution scale. It takes as input the features ƒ from the previous layer in the generator (which correspond to the created textures from the preceding layers), as well as the encoded skip features $\mathfrak{f}_{\text{skip}}$ (which correspond to the existing textures in the original image). This integration of features allows us to accurately generate repeated textures, leading to more realistic and visually appealing results. FSM divides channels into two parallel branches (local and global). The local branch uses depth-wise convolutions (DWConv) to extract local features and provide positional information for our Texture Enhancement (TE) module. The global branch, on the other hand, uses the TE to obtain global contexts by processing the entire feature map. Finally, the outputs of the two branches are concatenated to obtain the final feature representation. ViT [5] is selected as the basis for the TE module. In TE, the input is a set of feature vectors $\mathbf{x}_p \in \mathbb{R}^{N\times C}$, where N is the number of tokens and C is the channel quantity. Feature vectors are obtained by splitting the input feature map $\mathbf{m} \in \mathbb{R}^{C\times H\times W}$ into patches $\mathbf{m}_p \in \mathbb{R}^{N\times D^2\cdot C}$ and down-sampling them. D is the down-sampling rate, and (H, W) is the feature map's resolution.

The TE consists of a MSA module and a MLP without LN to improve its effectiveness. The MLP has a GELU [7] activation and two linear layers to address the rank collapse issue. The output feature map of TE is computed using up-sampling (U) of the TE's output token embeddings and adding them to the original feature patches $\mathbf{m}_p$, resulting in $\mathbf{m}_{\text{out}} = \text{U}(\mathbf{z}'_p) + \mathbf{m}_p$, where $\mathbf{z}'p$ is the output token embeddings of TE. Finally, the output sequence $\mathbf{m}_{\text{out}}$ is reshaped into an enhanced feature map $\mathbf{m}_c \in \mathbb{R}^{C\times H\times W}$.

Given the token embeddings $\mathbf{x}_p$ as the input sequence of the TE, while ($\mathbf{Q} = \mathbf{x}_p\mathbf{W}_q,\ \mathbf{K} = \mathbf{x}_p\mathbf{W}_k,\ \mathbf{V} = \mathbf{x}_p\mathbf{W}_v$) a single self-attention head can be written as:

$$\mathbf{Q} = \mathbf{x}_p\mathbf{W}_q,\ \mathbf{K} = \mathbf{x}_p\mathbf{W}_k,\ \mathbf{V} = \mathbf{x}_p\mathbf{W}_v, \tag{4}$$

$$\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}})((\lambda_c\mathbf{I}_v + \mathbf{W}_c)\mathbf{V}). \tag{5}$$

in which $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$ denote linear projection matrices. $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ indicate the query, key, value matrices, and D_k denotes the channel quantity of the token embeddings. $\mathbf{I}_v$ is an identity matrix, whose rank is the same as $\mathbf{V}$. $\mathbf{W}_c$ is the diagonal matrix. λ_c is the scaling factor, which is set to 0.5. Therefore, our TE can be written as:

$$\mathbf{z}_p = \text{MSA}(\mathbf{x}_p) + \mathbf{x}_p,\ \mathbf{x}_p = \text{D}(\mathbf{m}_p), \tag{6}$$

$$\mathbf{z}'_p = \text{MLP}(\mathbf{z}_p) + \mathbf{z}_p, \tag{7}$$

$$\mathbf{m}_{\text{out}} = \text{U}(\mathbf{z}'_p) + \mathbf{m}_p. \tag{8}$$

where the D indicates the down-sampling layer (average pooling). The MLP, composed of two linear layers and a GELU activation layer to mitigate the rank collapse issue. $\mathbf{m}_{\text{out}} = \text{U}(\mathbf{z}'_p) + \mathbf{m}_p$, where the U indicates the up-sampling layer, which is a DWConv. Finally we reshape the output sequence $\mathbf{m}_{\text{out}}$ to an enhanced feature map $\mathbf{m}_c \in \mathbb{R}^{C\times H\times W}$.

We use instance normalization instead of batch normalization in our TENet generation network since different missing regions will have an impact on each batch's means and variances. In our architecture, ResNet block down is the same as ResNet block up, in which we apply the average pooling layer after 3×3 and 1×1 Convolution layers.

(C) Network Training: The masked pixels are represented by $x_{\mathfrak{m}}$, and the model uses the unmasked pixels to recover the corresponding pixels in the input. Feature vectors are obtained using our TransInpaint. The encoder extracts feature maps, which are then used to compute a set of latent vectors $\mathcal{V}$. These vectors $\mathcal{V}$ are processed by a decoder to reconstruct the image from the latent vectors, which results in the reconstructed image x_r. Additionally, the model reconstructs an earlier image x_{CT} from CT, which is a reference image used for comparison during training. To perform this reconstruction we use the following loss function:

$$\begin{aligned}\ell_{TransInpaint} = \ell_{\text{mask}}(x_{\mathfrak{m}}, x_r) + \ell_{\text{pix}}(x_{CT}, x_r) + \\ \beta\|\hat{\nabla}[\mathcal{V}] \ominus f\|_2^2 + \lambda\ell_{\text{st}}\|f - f_r\|_2^2 + \ell_{\text{adv}}(x_{\mathfrak{m}}, x_r),\end{aligned} \tag{9}$$

where f_r and f are the feature maps from the reconstructed and GT images. Mask loss measures the reconstruction error between $x_{\mathfrak{m}}$ and the predicted image x_r. Pixel loss ensures the consistency between x_r and the earlier reconstructed image x_{CT}. Commitment loss measures the difference between the latent feature vectors $\mathcal{V}$ and their corresponding feature vectors f. The gradient of $\mathcal{V}$ is estimated by the decoder and compared to the true gradient f. $\hat{\nabla}[\cdot]$ is a pause-gradient operation that stops gradients from flowing into its argument, and $\beta = 0.25$, and Style loss, in which $\lambda = 250$. Indeed, the ℓ_{mask} determines the dissimilarity between the corrupted $x_{\mathfrak{m}}$ and reconstructed images x_r. It is composed of three components, the ℓ_1 between two images' pixel values, the perceptual loss (ℓ_P), and the style loss (ℓ_S). Following is a detailed description of the losses listed above.

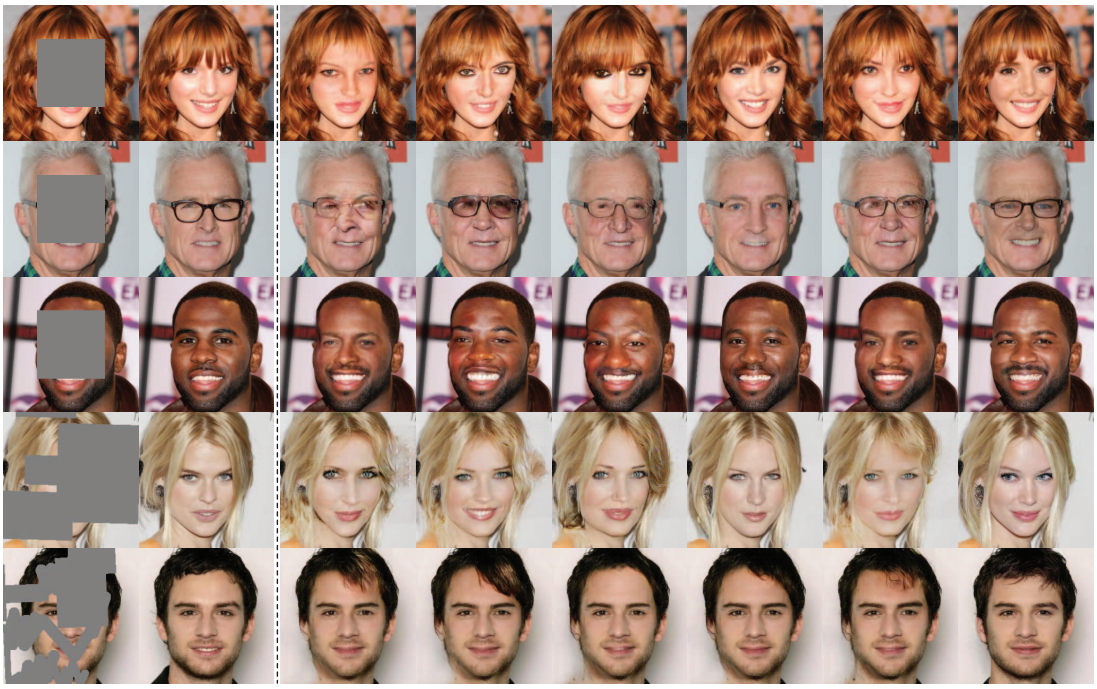

Figure 3: Comparison of qualitative completion results on CelebA images.

On the basis of the activation maps from VGG-16, the conceptual loss (ℓ_P) and style loss (ℓ_S) are computed as:

$$\ell_P = \sum_{l}^{L_P} \mathcal{M}(|\hbar_l(x_{\mathfrak{m}}) \ominus \hbar_l(x_r)|), \tag{10}$$

$$\ell_S = \sum_{l}^{L_S} \mathcal{M}(|\mathcal{G}(\hbar_l(x_{\mathfrak{m}})) \ominus \mathcal{G}(\hbar_l(x_r))|), \tag{11}$$

in which $\hbar_l(.)$ denotes different layers in VGG-16, and $\mathcal{G}(.)$ represents the function that returns the Gram matrix of its argument. For ℓ_P and ℓ_S, L_P is set to {lrelu1-1, lrelu2-1, ..., lrelu5-1}, and L_S is set to {lrelu2-2, lrelu3-4, lrelu4-4, lrelu5-2} respectively. ℓ_{Mask} is, therefore, equal to the sum of the above two losses and the ℓ_1 .
The adversarial loss ℓ_A of our model is calculated using a discriminator network $\mathcal{D}_A(.)$:

$$\ell_A = -\mathcal{M}(\log[1 \ominus \mathcal{D}_A(x_r)]) - \mathcal{M}(\log[\mathcal{D}_A(x_{\mathfrak{m}})]), \tag{12}$$

in which $\mathcal{M}(.)$ denotes a mean-value operation and $\log[.]$ represents the element-wise logarithm operation. The network architecture of the discriminator is identical to TFill.

4. Experimental Validation

(A) Experimental Methodology: Our model is evaluated on three datasets: CelebA-HQ [11], Places2 [51], and ImageNet [29]. We follow the standard training, validation, and testing splits for each dataset. For ImageNet, only 1K images from the test split are randomly selected for evaluation, the same as in [37]. To generate the missing image regions for training, we re-split the train and validation subsets, and randomly cropped the images to include between 10% and 50% of the total image area, including the missing rectangular part. The BB boundary is randomly selected at 50 points, and irregular sections are created by drawing lines between those points. We evaluate our model using three metrics commonly used in image inpainting: learned perceptual image patch similarity (LPIPS) [49] and fréchet inception score (FID) [8] provide quantitative measurements of how well the completed images match the original images in terms of perceptual quality. We also use the structural similarity index (SSIM) [39] that evaluates the similarity between the original image and the completed image based on their luminance, and structural information.
Our model is implemented in PyTorch and we use four RTX 3090 GPUs to train the model with a batch size of 32 for 1M iterations. We apply spectral normalization to all networks and initialize them using Orthogonal Initialization. The networks are trained from scratch using the Adam optimizer with a fixed learning rate of 1e-4, β_1 = 0, and β_2 = 0.9. The transformer is optimized with Adam (learning rate = 3e-4). To ensure a fair comparison with other inpainting methods, the training and testing images are of size 256×256 with random regular or irregular masks. We compare our TransInpaint with several inpainting approaches including

Figure 4: Comparison of qualitative completion results on Places2 and ImageNet images.

Table 1: Quantitative comparisons on CelebA, Places2, and ImageNet datasets on center-masked images. Results are based on LPIPS (↓), FID (↓) and SSIM (↑). "*" indicates TransInpaint trained only with DETR, instead of CT. TransInpaint w/o TENet indicates our model trained with standard encoder/decoder network. TransInpaint w/o FSM indicates our model with TENet, but we use two convolution layers instead of FSM to aggregate $\mathfrak{f}$ and $\mathfrak{f}_{\text{skip}}$.

Method	CelebA			Places2			ImageNet		
	LPIPS	FID	SSIM	LPIPS	FID	SSIM	LPIPS	FID	SSIM
JPGNet [6] ACM-MM'21	0.129	14.62	0.869	0.163	21.26	0.784	0.196	24.35	0.709
CR-Fill [48] ICCV'21	0.107	6.37	0.914	0.131	10.85	0.837	0.173	11.42	0.785
TFill [50] CVPR'22	0.092	5.23	0.919	0.039	9.11	0.843	0.162	10.81	0.797
LaMa [34] WACV'22	0.085	4.93	0.925	0.121	8.75	0.851	0.159	10.52	0.810
GAVQ [26] CVPR'23	0.088	4.96	0.920	0.124	8.84	0.842	0.163	10.70	0.787
MAT [15] CVPR'22	**0.074**	4.54	0.936	0.110	8.39	0.862	0.128	10.24	0.816
TransInpaint w/o CT *	0.101	6.34	0.908	0.130	10.97	0.844	0.166	11.25	0.776
TransInpaint w/o TENet	0.113	6.97	0.915	0.144	13.34	0.838	0.178	15.84	0.789
TransInpaint w/o FSM	0.098	6.11	0.926	0.123	10.14	0.852	0.153	10.67	0.797
TransInpaint (ours)	0.079	**4.46**	**0.941**	**0.104**	**8.05**	**0.884**	**0.115**	**9.93**	**0.834**

JPGNet [6], CR-Fill [48], TFill [50], LaMa [34], GAVQ [26], MAT [15] using the provided pre-trained weights.

(B) Qualitative Results: Figs. 3 and 4 provide a qualitative comparison of state-of-art inpainting methods on the CelebA, Places2, and ImageNet datasets. In general, CR-Fill does not produce plausible structures, and TFill generates textures with unnatural artifacts, indicating that structural information has minor contributions to texture synthesis. TransInpaint outperforms LaMa and MAT in capturing the global context and preserving realistic textures, especially on challenging Places2 and ImageNet images. It also outperforms other models in generating instances that are realistic and visually similar to the original images.

(C) Quantitative Results: Table 1 provides a comparison of our TransInpaint model with several state-of-art methods. For a fair comparison, we tested against models on the same masks. TransInpaint outperforms other methods in terms of the three evaluation metrics. MAT outperforms other methods on CelebA, as it achieves a lower LPIPS score. LaMa performs better than TFill in generating the target instance on Places2. However, as shown in Table 1 and Figs. 3 and 4, our model can fill missing regions with more visually realistic instances that are close to the original instances, resulting in better LPIPS, FID, and SSIM performance. For example, on ImageNet, TransInpaint achieves LPIPS and FID scores of 0.115 and 9.93, respectively, which are 11%

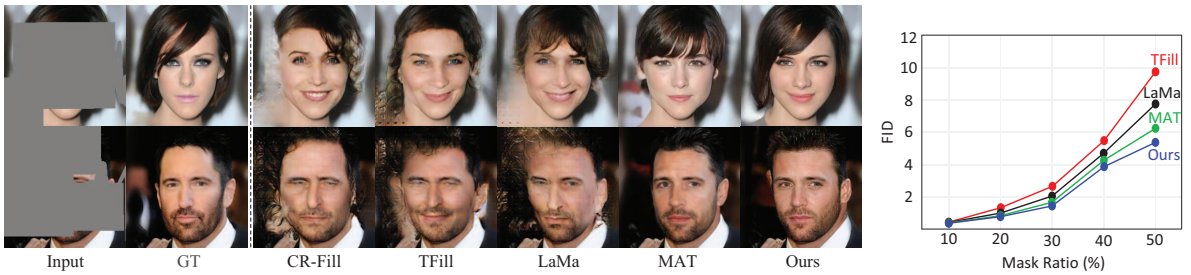

Figure 5: Left: Comparison of qualitative completion results on CelebA images with large missing regions. Right: Plot of FID score versus the mask ratio on the CelebA dataset.

and 4% better than other methods. In Fig. 5 (left), it can be observed that state-of-the-art models cannot reconstruct images with large missing regions, resulting in distorted outputs, while TransInpaint generates plausible outputs without any artifacts. Furthermore, Fig. 5 (right) depicts the FID score versus the mask ratio. Although MAT has a similar FID score to our model for the min mask ratio, our image completion outperforms MAT by a higher margin for large masks. The high level of TransInpaint performance can be the result of two factors: (1) our training strategy, and use of the CT module to generate relevant content for the missing instances, and (2) the TENet that reuses high-frequency features. Combining skip features from the encoder with coarse features from the generator allows for better modeling of long-range dependencies and global context.

(D) Ablation Study: Qualitative and quantitative comparisons were conducted to justify the effectiveness of our proposed TransInpaint. Table 1 and Figs. 3, and 4 show inpainting results with and without our proposed modules. By using CT and TENet networks, our TransInpaint learns to generate texture details in the output image without any distortions, artifacts, or blurriness. In particular, without the CT module, our TransInpaint model cannot have satisfactory performance on images with multiple objects or complex textures. For example, without the CT module on Places2, our model has a FID of 10.97. On the other hand, without the TENet module, TransInpaint struggles with texture reconstruction (FID of 15.84 on ImageNet). Moreover, as the qualitative evaluations in Figs. 3, and 4 show, our FSM module plays a pivotal role in achieving high-quality inpainting results, particularly in the generation of repeating textures. The use of skip connections in the FSM enhances the flow of information between the layers, facilitating efficient feature reuse and preserving structural information. Use of the FSM in our model results in 21% improvement of FID on the Places2. The proposed modules are the key components of our framework and play an important role in reconstructing the texture of the original image.

(E) Visual Inspection: We perform a user study against other models to more accurately assess the subjective quality. We take twentyfive masked images at random from the Places2 test set. Different approaches are used to produce reconstructed outputs for a test image. In this assessment, participants are given two separate generated images at the same time, one created by our model and the other by one of the baselines. Participants are asked to select the most photorealistic and visually natural image. We collect responses from thirty participants and calculate the ratios of each approach using the data given in Fig. 6. Our model has 69% likelihood to be selected.

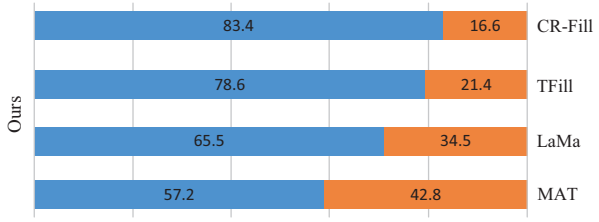

Figure 6: The results of a study on visual inspection. The values represent the preference for the comparison pair.

5. Conclusion

We have proposed a new Context-adaptive Transformer (TransInpaint) for image inpainting. Rather than filling up the damaged regions of an image with surrounding textures, TransInpaint synthesizes context-adaptive visual contents. TransInpaint provides high prediction accuracy while preserving the pixels with the least uncertainty using our proposed context-adaptive transformer and transformer-based feature enhancement network, resulting in high-quality and plausible completion. Through a set of experiments on three challenging benchmarks, we show that TransInpaint provides an advantage over the baseline models. Although our model has demonstrated superior performance compared to existing state-of-art approaches, it still faces limitations when it comes to accurately reconstructing complex shapes, such as the intricate details of animal eyes. We are motivated to address this challenge as part of our future work by leveraging the power of diffusion techniques to capture these intricate structures, thereby enhancing the realism and quality of completed images.

References

[1] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE TIP*, 10(8):1200–1211, 2001.

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proc. SIGGRAPH*, pages 417–424, 2000.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. ECCV*, 2020.

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

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, and et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2021.

[6] Qing Guo, Xiaoguang Li, Felix Juefei-Xu, Hongkai Yu, Yang Liu, and Song Wang. Jpgnet: Joint predictive filtering and generative network for image inpainting. In *Proc. ACM-MM*, 2021.

[7] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Proc. NIPS*, 30, 2017.

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

[10] Jitesh Jain, Yuqian Zhou, Ning Yu, and Humphrey Shi. Keys to better image inpainting: Structure and texture go hand in hand. In *Proc. WACV*, pages 208–217, 2023.

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *Proc. ICLR*, 2017.

[12] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Comput. surveys*, 54(10s), 2022.

[13] Joo Ho Lee, Inchang Choi, and Min H Kim. Laplacian patch-based image synthesis. In *Proc. CVPR*, pages 2727–2735, 2016.

[14] Henry Li and Yuval Kluger. Autoregressive generative modeling with noise conditional maximum likelihood estimation. *Proc. ICLR*, 2023.

[15] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proc. CVPR*, pages 10758–10768, 2022.

[16] Wenbo Li, Xin Yu, Kun Zhou, Yibing Song, Zhe Lin, and Jiaya Jia. Image inpainting via iteratively decoupled probabilistic modeling. *arXiv preprint arXiv:212.023*, 2023.

[17] Jian Liang, Chenfei Wu, Xiaowei Hu, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zicheng Liu, Yuejian Fang, and Nan Duan. Nuwa-infinity: Autoregressive over autoregressive generation for infinite visual synthesis. *Proc. NIPS*, 35, 2022.

[18] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *Proc. ECCV*, pages 725–741, 2020.

[19] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *Proc. ICCV*, pages 4170–4179, 2019.

[20] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proc. CVPR*, pages 11461–11471, 2022.

[21] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *Proc. ICLR*, 2021.

[22] Abdelrahman Mohamed, Rushali Grandhe, KJ Joseph, Salman Khan, and Fahad Khan. D3former: Debiased dual distilled transformer for incremental learning. In *Proc. CVPR*, 2023.

[23] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *Proc. ICCV Workshop*, 2019.

[24] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, 2016.

[25] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proc. CVPR*, pages 10775–10784, 2021.

[26] Shruti S Phutke, Ashutosh Kulkarni, Santosh Kumar Vipparthi, and Subrahmanyam Murala. Blind image inpainting via omni-dimensional gated attention and wavelet queries. In *Proc. CVPR*, 2023.

[27] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proc. ICCV*, pages 181–190, 2019.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, 2022.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[30] Pourya Shamsolmoali, Masoumeh Zareapoor, Swagatam Das, Salvador García, Eric Granger, and Jie Yang. Gen: Generative equivariant networks for diverse image-to-image translation. *IEEE TCyber.*, 2022.

[31] Pourya Shamsolmoali, Masoumeh Zareapoor, and Eric Granger. Image completion via dual-path cooperative filtering. *ICASSP*, 2023.

[32] Pourya Shamsolmoali, Masoumeh Zareapoor, Huiyu Zhou, Dacheng Tao, and Xuelong Li. Vtae: Variational transformer autoencoder with manifolds learning. *IEEE TIP*, 2023.

[33] Jookyung Song, Yeonjin Chang, Seonguk Park, and Nojun Kwak. Semantics-guided object removal for facial images: with broad applicability and robust style preservation. *arXiv preprint arXiv:2209.14479*, 2022.

[34] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proc. WACV*, pages 2149–2159, 2022.

[35] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Proc. NIPS*, 29, 2016.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. NIPS*, 30, 2017.

[37] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *Proc. ICCV*, 2021.

[38] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022.

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4), 2004.

[40] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. *arXiv preprint arXiv:2303.09472*, 2023.

[41] Chaohao Xie, Shaohui Liu, Chao Li, Ming-Ming Cheng, Wangmeng Zuo, Xiao Liu, Shilei Wen, and Errui Ding. Image inpainting with learnable bidirectional attention maps. In *Proc. ICCV*, pages 8858–8867, 2019.

[42] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *Proc. ECCV*, pages 1–17, 2018.

[43] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proc. CVPR*, pages 7508–7517, 2020.

[44] Ahmet Burak Yildirim, Vedat Baday, Erkut Erdem, Aykut Erdem, and Aysegul Dundar. Inst-inpaint: Instructing to remove objects with diffusion models. *arXiv preprint arXiv:2304.03246*, 2023.

[45] Tackgeun You, Saehoon Kim, Chiheon Kim, Doyup Lee, and Bohyung Han. Locally hierarchical auto-regressive modeling for image generation. *Proc. NIPS*, 2022.

[46] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proc. ICCV*, pages 4471–4480, 2019.

[47] Yingchen Yu, Fangneng Zhan, Rongliang Wu, Jianxiong Pan, Kaiwen Cui, Shijian Lu, Feiying Ma, Xuansong Xie, and Chunyan Miao. Diverse image inpainting with bidirectional and autoregressive transformers. In *Proc. ACM-MM*, pages 69–78, 2021.

[48] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M Patel. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In *Proc. ICCV*, 2021.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, pages 586–595, 2018.

[50] Chuanxia Zheng, Tat-Jen Cham, Jianfei Cai, and Dinh Phung. Bridging global context interactions for high-fidelity image completion. In *Proc. CVPR*, 2022.

[51] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2017.