Wide-Context Semantic Image Extrapolation

Yi Wang[1,2] Xin Tao[2] Xiaoyong Shen[2] Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong [2]YouTu Lab, Tencent
yiwang@cse.cuhk.edu.hk {xintao, dylanshen, jiayajia}@tencent.com

Abstract

This paper studies the fundamental problem of extrapolating visual context using deep generative models, i.e.*, extending image borders with plausible structure and details. This seemingly easy task actually faces many crucial technical challenges and has its unique properties. The two major issues are size expansion and one-side constraints. We propose a semantic regeneration network with several special contributions and use multiple spatial related losses to address these issues. Our results contain consistent structures and high-quality textures. Extensive experiments are conducted on various possible alternatives and related methods. We also explore the potential of our method for various interesting applications that can benefit research in a variety of fields.*

1. Introduction

Humans have the natural ability to perceive unseen surroundings based on limited visual content. For computer vision, accomplishing this task requires generating semantically meaningful and consistent structure and texture. In this paper, we focus on the special task to *infer unseen content outside image boundaries*.

This task finds several related methods and topics in image processing and graphics. It was treated as an intriguing application in view expansion [35, 43, 49], image editing [2], texture synthesis [10, 11, 41], to name a few. These methods exploit information from either external images or internal statistics. For example, algorithms of [35, 43, 49] enlarge the view by matching and stitching similar candidates. Another line [15] uses retargeting. It is also a natural choice to use inpainting methods [1, 5, 7, 20, 23, 25, 37] for extrapolating images. We note that these methods are not specially designed for our task and thus have their respective limitations when applied to content generation. External-image-based algorithms require a large amount of or structurally very similar reference images while internal pixels/patches-based methods mostly produce apparently similar or repeated patterns.

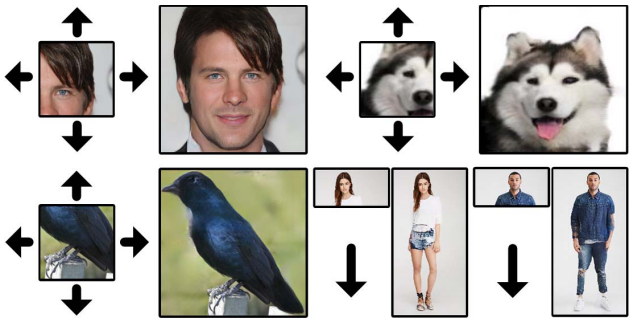


Figure 1. Illustration of our pursuit with examples of face, dog, bird, and human body, which are all highly semantically sensitive and representative.

Different from the results shown in previous work, the illustration in Figure 1 indicates that our method has its unique and strong capability. It can generate the full portrait with hair and background even from a small part of faces (top row of Figure 1), create bird head and tail based on body shape (bottom left of Figure 1), or produce a full human body given only upper body information (bottom right of Figure 1). Note that in all these examples, the algorithm needs to suitably take vastly different context of each incomplete image into account and predict up to 3 times more unknown pixels than known ones.

In regard to technical strategies, deep learning becomes popular and effective in low-level vision [8, 26, 39, 46, 48]. Applying it to this context generation task, however, still needs to consider the following two issues.

Image Size Change Image expansion extends image size beyond boundaries. A similar task is super-resolution [8, 24, 36, 38], which produces high-res (HR) results from low-res (LR) input. Current SR frameworks either upsample input before fed into networks [8], or use spatial expansion modules [24, 36, 38] within the network. So the first issue to conquer in our framework is to properly increase size with structure and detail generation.

One-sided Constraints The boundary condition in context generation has only one side, as illustrated in Figure 1 where black arrows show inference direction. This configuration is different from that of general image-to-image

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00149

translation (*e.g*. image synthesis, deblur), where the latter has a one-to-one spatial correspondence between the prediction and input. The unknown pixels away from image border are less constrained than those near border, potentially accumulating errors or repeated patterns. To deal with it, we design the relative spatial variant loss, context adversarial loss, and context normalization to regularize the generation procedure.

Our Contribution To address these key issues, we propose a Semantic Regeneration Network (SRN) to regenerate the full object from a small portion of visual clues. SRN can generate arbitrary-size semantic structure beyond image boundary without training multiple models. It directly learns semantic features from small-size input, which is both effective and efficient by avoiding bias in common padding and upsampling procedures [33, 40, 26].

In the structure level, SRN contains two components of Feature Expansion Network (FEN) and Context Prediction Network (CPN). FEN takes small-size images as input and extracts features. Such features and extrapolation indicator are fed to CPN for reconstructing final expansion results. With the separation of feature extraction and image reconstruction, learning and inference of our network becomes appropriate and efficient. Further, the designed losses and other processing modules adapt our network to one-sided constraints, generating semantically meaningful structure and natural texture. Our major contribution is twofold.

- We propose an effective deep generative model SRN for image extrapolation. Practical context normalization (CN) module and relative spatial variant (RSV) loss are proposed. They are evaluated along with several other alternatives.
- We apply our solution to various intriguing and important applications.

2. Related Work

2.1. Image Extrapolation

Prior extrapolation solutions [35, 43, 49] usually turn to an external library for solutions in a data-driven manner. This type of methods formulates the problem into matching and stitching, where the new content is retrieved from a pre-constructed dataset. For example, Wang *et al*. [43] exploited this method on the graph representation of images. They retrieve candidate images by subgraph matching, and stitch these wrapped images into the input. Shan *et al*. formulated the image composition into a MRF problem, able to process a large library with high robustness regarding viewpoint, appearance, and layout variation [35]. Zhang *et al*. [49], with the retrieved large image candidate, aligned the small input and candidate. The relative position between similar patches in the known and unknown regions of the candidate is applied to the input in a copy-and-paste manner. As a non-parametric method, data-driven image extrapolation is limited by the used dataset. Moreover, sophisticated or fine textures along expanding boundary hinder the application of this type of methods.

2.2. Conditional Image Generation

Image extrapolation belongs to conditional image generation in deep learning. The most related problem is inpainting. Recent advance in inpainting lies in applying deep generative models to repair large missing pieces [47, 46, 31, 48, 44]. Pathak *et al*. [31] first applied adversarial loss to learn an encoder-decoder network. To create realistic textures based on given context, MRF-based style transfer via patch matching in the deep feature space was employed as post-processing [46]. Further, Yu *et al*. [48] proposed the contextual attention layer, which replaces deep features with its neighborhood weighted average and improves both texture quality and inference efficiency. The other related topic is image retargeting [34, 3]. In [3], a CNN was designed to learn the shift map for each pixel. Salient objects are preserved while background is seamlessly modified. Retargeting has no intention to extend surrounding content.

2.3. Spatial Expansion Operators

Spatial expansion operators are indispensable components in various tasks, when output is with a larger size. Prevalent spatial expansion operators include padding, interpolation, deconvolution [30, 9], sub-pixel convolution [36], and a warping-based SPMC module [38]. We discuss and experiment with these operators except SPMC in Section 4 since SPMC only works with sequential input.

3. Our Method

Given an input image $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$ and filling margin $m = (top, left, bottom, right)$, semantic image expansion (or extrapolation) intends to generate a visually convincing image $\hat{\mathbf{Y}} \in \mathbb{R}^{h' \times w' \times c}$, where $h' = h + top + bottom$, $w' = w + left + right$, and $\mathbf{X}$ is a sub-image of $\hat{\mathbf{Y}}$. Contrary to the inpainting process, which fills interior holes of an image, image extrapolation is meant to expand image borders. For convenience, we denote $h' = r_1 h$ and $w' = r_2 w$ (where $r_1 \geq 1$, $r_2 \geq 1$, and $r_1 r_2 > 1$).

3.1. Framework Design

Our model G consists of two sub-networks of *feature expansion network* (FEN) and *context prediction network* (CPN), as shown in Figure 2. FEN extracts deep features from the given image, and CPN decodes these features into images considering filling margin and size. The input to our network contains an image $\mathbf{X}$ and a margin variable $m = (top, left, bottom, right)$ indicating extension.

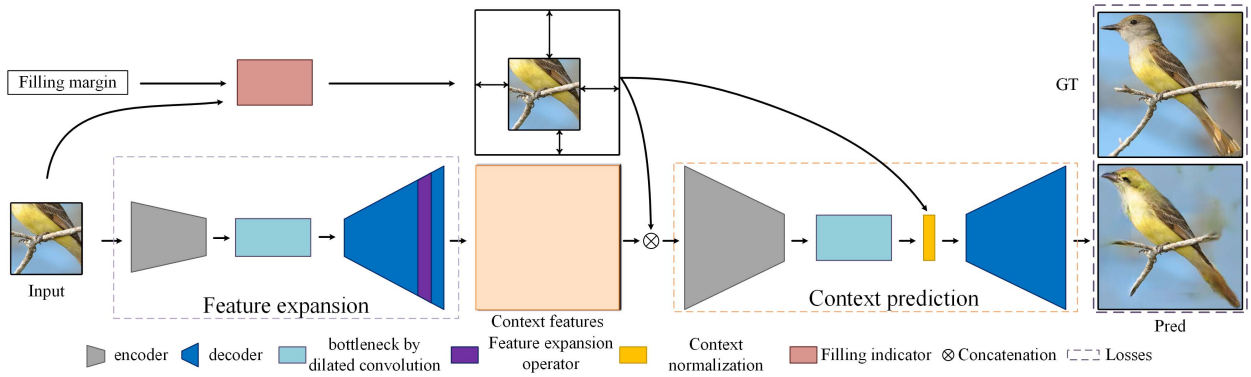


Figure 2. Our framework illustration.

3.1.1 Feature Expansion

This module employs an encoder-decoder-like structure, where input is only $\mathbf{X}$ of size $h \times w \times c$, and output is its feature maps $f(\mathbf{X})$ of size $r_1 h \times r_2 w \times c'$. Increasing the feature size is realized by nearest-neighbor upsampling followed by convolution, except the last layer, which is otherwise achieved by a sub-pixel convolution [36] variant. It is a vanilla convolution followed by reshuffling feature channels. Given a feature map $F \in \mathbb{R}^{h \times w \times r_1 r_2 c'}$, such reshuffling operation $s(\cdot)$ is defined as

$$s(F)_{i,j,k} = F_{\lfloor i/r_1 \rfloor, \lfloor j/r_2 \rfloor, c' r_2 \cdot mod(i,r_1) + c' \cdot mod(j,r_2) + k}, \quad (1)$$

where $s(F) \in \mathbb{R}^{r_1 h \times r_2 w \times c'}$. i, j, and k denote index height, width, and channel, respectively. Compared with the original sub-pixel convolution [36], the presented variant relaxes the constraint that $r_1 = r_2$. It handles scenarios when $r_1 \neq r_2$ while the method of [36] cannot. This ability is useful in human body generation ($r_1 = 4$ and $r_2 = 1$ in Figure 1) and view expansion ($r_1 = 1$ and $r2 = 2$ in Section 4).

We discuss and compare alternative trainable operators, *i.e.*, deconvolution layer and convolution after padding (termed as *unfold* operator in the following) or interpolation. Here deconvolution is not considered since it causes visual artifacts in generation due to the overlap problem [30, 9]. Interpolation or padding methods have their respective properties. Specifically, interpolation assumes that the filling region is similar to that in the corresponding location of the input; zero padding assumes a constant value for missing part; symmetric/mirror padding makes the context feature the mirror version along the image border. Comparing with deconvolution and unfold, sub-pixel convolution expands features with less bias. This is experimentally validated in Session 4.3.

Feature Expansion Network (FEN) is to learn latent context features. Experimental results show that filled pixels in early batches serve as a kind of prior for later generation. Computation directly conditioned on available pixels could yield better performance in terms of both fidelity and visual naturalness [33, 40, 26]. Thus, our model directly infers upon the given visual data without predefined priors.

3.1.2 Context Prediction

We also use encoder-decoder-like network for this component. The input is the concatenation of $f(\mathbf{X})$ and filling indicator, *i.e.* a binary mask, where 0 is for known pixels and 1 for unknown ones, denoted by $\mathbf{M}$. The output is $\hat{\mathbf{Y}}$ of size $r_1 h \times r_2 w \times c$. A context normalization module is developed for coordinating the feature distribution between filling and known regions.

Rather than a simple refinement stage commonly used in the coarse-to-fine framework, the rationale behind Context Prediction Network (CPN) is twofold. First, it incorporates filling margin, which is excluded in FEN, to indicate where to predict. Second, besides the filling margin, input to the network also includes context features learned by FEN instead of coarse prediction. These features are properly handled by compression via an encoder-decoder and our designed context normalization module.

Context Normalization To improve style consistency of the generated image, a *context normalization* (CN) module is proposed. Recent study shows that image style is characterized by its feature statistics. Various image statistical losses [12, 14] and normalization operations [18, 42, 16] were explored to capture such statistics implicitly or explicitly. Inspired by instance normalization [42] and AdaIN [16], our proposed CN function ($t(\cdot)$) is defined as

$$\begin{aligned} t(f(\mathbf{X}), \rho) = & [\rho \cdot n(f(\mathbf{X}_{\mathbf{\Omega}}), f(\mathbf{X}_{\bar{\mathbf{\Omega}}})) \\ & + (1-\rho) f(\mathbf{X}_{\mathbf{\Omega}})] \odot \mathbf{M}\downarrow + \\ & f(\mathbf{X}_{\bar{\mathbf{\Omega}}}) \odot (1 - \mathbf{M}\downarrow), \end{aligned} \quad (2)$$

$$n(x_1, x_2) = \frac{x_1 - \mu(x_1)}{\sigma(x_1)} \cdot \sigma(x_2) + \mu(x_2), \quad (3)$$

where $\mathbf{X}_{\bar{\mathbf{\Omega}}}$ and $\mathbf{X}_{\mathbf{\Omega}}$ indicate known and unknown image regions respectively, $f(\cdot)$ extracts bottleneck features based

on the input-expanded feature maps, and $\rho \in [0, 1]$. $\downarrow$ is the nearest-neighbor downsampling operator. $\mathbf{M} \downarrow$ shares the same height and width with $f(\mathbf{X})$. $\mu(\cdot)$ and $\sigma(\cdot)$ compute the mean and standard deviation. Essentially, it transfers mean and variance from known features to unknown area, which regularizes the generated content beyond one-side constraints and enhances the color/texture consistency between input and predicted regions.

Note that CN and AdaIN [16] are fundamentally different. AdaIN replaces the feature statistics of an image with those from another image. For CN, feature statistics in known/unknown regions of the same image are considered. Moreover, a blending step is incorporated in CN. Because the feature statistics from known and unknown regions could be different for semantically sensitive targets like face and body, blending these feature statistics is crucial for our system. Detailed comparisons are given in the supplementary material.

3.2. Loss Design

The optimization target comprises the reconstruction loss, texture consistency loss, and the adversarial loss, which are detailed as follows.

Relative Spatial Variant Loss Reconstruction loss stabilizes the training procedure by providing pixel-wise supervision. Due to the one-sided property of content extrapolation, spatial variant supervision [48, 44] is needed. We design a relative spatial variant (RSV) reconstruction loss for incorporating such spatial regularization. For the confidence-driven (CD) loss [44], it is formulated as

$$\mathbf{M}_w^i = (g * \overline{\mathbf{M}}^i) \odot \mathbf{M}, \tag{4}$$

where g is a normalized Gaussian filter, $\overline{\mathbf{M}}^i = \mathbf{1} - \mathbf{M} + \mathbf{M}_w^{i-1}$, and $\mathbf{M}_w^0 = \mathbf{0}$. $\odot$ is the Hadamard product operator. Eq. (4) is repeated c times to generate $\mathbf{M}_w^c$.

In RSV, our used weight matrix is

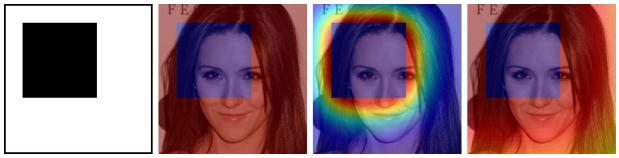
$$\mathbf{M}_w = \mathbf{M}_w^{c-1} / max(\mathbf{M}_w^c, \epsilon). \tag{5}$$

The final reconstruction loss is

$$\mathcal{L}_s = ||(\mathbf{Y} - G(\mathbf{X}, m; \theta)) \odot \mathbf{M}_w||_1, \tag{6}$$

where $G(\mathbf{X}, m; \theta)$ is the output of our generative model G, $\mathbf{Y}$ is the corresponding ground truth, and θ denotes parameters that can be learned.

The repetitive convolution of g over $\overline{\mathbf{M}}^i$ propagates the confidence of known pixels to unknown ones. However, since existing pixels are fewer than unknown ones, and they are almost separated (only a handful of unknown pixels have neighboring known pixels), the confidence propagation is hindered by its scarce neighborhood support. To remedy it, we apply the ratio of two adjacent convolutional results $\mathbf{M}_w^{c-1}$ and $\mathbf{M}_w^c$ to describe the confidence. Intuitively, unknown pixels close to existing regions have high-confidence neighboring pixels. So their relative increase is quicker than that of unknown pixels away from it. As shown in Figure 3, CD does not constrain distant areas while RSV assigns meaningful weight. More comparisons are given in Section 4.3.



(a) (b) (c) (d)

Figure 3. Visualization of weight mask $\mathbf{M}_w$ used in Eq. (5). (a) Input mask (0 and 1 for known and unknown color), (b) use $\mathbf{M}$ as $\mathbf{M}_w$, (c) $\mathbf{M}_w$ in CD [44], (d) $\mathbf{M}_w$ in RSV. (b)-(d) are shown in the jet colormap.

Implicit Diversified MRF Loss Along with pixel-wise reconstruction loss, implicit diversified MRF regularization [29, 44] is introduced as part of the optimization goal for creating crisp texture by bringing close feature distributions of $G(\mathbf{X}, m)$ and $\mathbf{Y}$.

We use $\hat{\mathbf{Y}}_\Omega^L$ and $\mathbf{Y}^L$ to denote features extracted from the L_{th} feature layer of a pretrained network, where $\hat{\mathbf{Y}}_\Omega$ indicates the prediction of the regions to be filled. The ID-MRF loss [29, 44] between $\hat{\mathbf{Y}}_\Omega^L$ and $\mathbf{Y}^L$ is defined as

$$\mathcal{L}_M(L) = -\log(\frac{1}{Z} \sum_{\mathbf{s} \in \mathbf{Y}^L} \max_{\mathbf{v} \in \hat{\mathbf{Y}}_\Omega^L} \overline{\mathrm{RS}}(\mathbf{v}, \mathbf{s})), \tag{7}$$

with respect to

$$\overline{\mathrm{RS}}(\mathbf{v}, \mathbf{s}) = \mathrm{RS}(\mathbf{v}, \mathbf{s}) / \sum_{\mathbf{r} \in \rho_{\mathbf{s}}(\mathbf{Y}^L)} \mathrm{RS}(\mathbf{v}, \mathbf{r}), \tag{8}$$

$$\mathrm{RS}(\mathbf{v}, \mathbf{s}) = \exp((\frac{\beta(\mathbf{v}, \mathbf{s})}{\max_{\mathbf{r} \in \rho_{\mathbf{s}}(\mathbf{Y}^L)} \beta(\mathbf{v}, \mathbf{r}) + \epsilon})/h), \tag{9}$$

where Z is a normalization factor. Eq. (8) is a normalized version of Eq. (9), which defines the similarity between two extracted patches $\mathbf{v}$ and $\mathbf{s}$ from $\hat{\mathbf{Y}}_\Omega^L$ and $\mathbf{Y}^L$ respectively. $\beta(\cdot, \cdot)$ is the cosine similarity. $\mathbf{r} \in \rho_{\mathbf{s}}(\mathbf{Y}^L)$ means $\mathbf{r}$ belonging to $\mathbf{Y}^L$ excluding $\mathbf{s}$. h and ϵ are two positive constants. If $\mathbf{v}$ is like $\mathbf{s}$ more than other neural patches in $\mathbf{Y}^L$, $\mathrm{RS}(\mathbf{v}, \mathbf{s})$ turns large.

In our experiments, we compute the sum of $\mathcal{L}_M$ between $G(\mathbf{X}, m; \theta)$ and $\mathbf{Y}$ on conv3_2 and conv4_2 extracted from pre-trained VGG19 network as $\mathcal{L}_{mrf}$.

Compared with other losses, *e.g.*, style loss and its variants, focusing on restoring texture or style, ID-MRF loss reinforces local image details by referring their most relatively similar patches in ground truth.

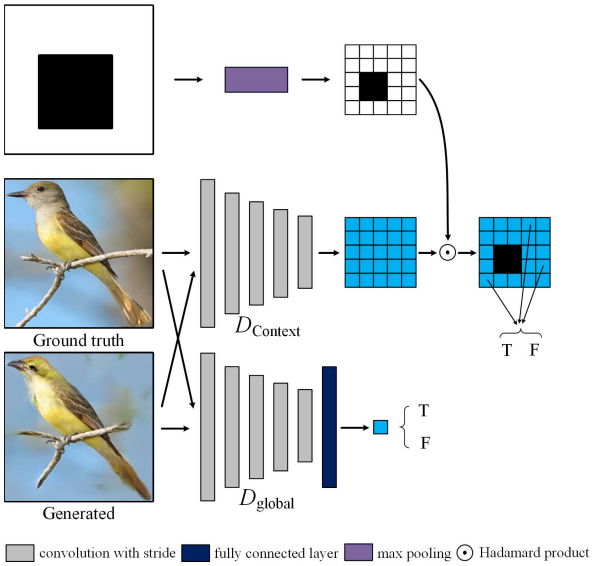


Figure 4. Context discriminator illustration.

Contextual Adversarial Loss Various generation tasks using generative adversarial networks have validated the effectiveness of adversarial training in image creation and synthesis. The adversarial loss, which is an optimization measure that can be learned, is an indispensable ingredient in producing convincing details. In our work, the global and local discriminators [17] with improved Wasserstein distance [13] are employed.

It is noteworthy of the specialty in our design. Unlike restoring a local rectangle region in inpainting tasks where local information can be easily extracted, the contextual region (to be predicted) surrounds the given input region, leading to the difficulty of aggregating local regions into a single probability. To tackle this issue, a masked patch discriminator is adopted as the context discriminator (Figure 4). The output $D_{context}(\hat{\mathbf{Y}})$ of context discriminator for the input prediction $\hat{\mathbf{Y}}$ is defined as

$$\begin{aligned} D_{context}(\hat{\mathbf{Y}}) &= \frac{\sum_{p\in P(\hat{\mathbf{Y}})} p}{\sum_{q\in \mathbf{M}\downarrow} q}, \\ \textit{w.r.t.} \quad P(\hat{\mathbf{Y}}) &= d_{context}(\hat{\mathbf{Y}}) \odot \mathbf{M}\downarrow, \end{aligned} \tag{10}$$

where $d_{context}(\hat{\mathbf{Y}})$ denotes the feature maps of $\hat{\mathbf{Y}}$, and $\downarrow$ is the max pooling operator. For SRN, the global/context adversarial loss is defined as

$$\begin{aligned} \mathcal{L}^n_{adv} = &- E_{\mathbf{X}\sim\mathbb{P}_{\mathbf{X}}}[D_n(G(\mathbf{X};\theta))] + \\ &\lambda_{gp} E_{\hat{\mathbf{X}}\sim\mathbb{P}_{\hat{\mathbf{X}}}}[(||\nabla_{\hat{\mathbf{X}}} D_n(\hat{\mathbf{X}}) \odot \mathbf{M}_w||_2 - 1)^2], \end{aligned} \tag{11}$$

where $\hat{\mathbf{X}} = tG(\mathbf{X}, m; \theta) + (1-t)\mathbf{Y}$, $t \in [0,1]$, $\mathbf{Y}$ is the ground truth corresponding to $\mathbf{X}$, and $n \in \{context, global\}$. Thus, the employed $\mathcal{L}_{adv} = (\mathcal{L}^{context}_{adv} + \mathcal{L}^{global}_{adv})/2$.

Final Learning Objective With relative spatial variant reconstruction loss, ID-MRF loss, and adversarial loss, the model objective of our network is expressed as

$$\mathcal{L} = \lambda_s \mathcal{L}_s + \lambda_{mrf} \mathcal{L}_{mrf} + \lambda_{adv} \mathcal{L}_{adv}, \tag{12}$$

where λ_s, λ_{adv}, and λ_{mrf} are coefficients used to balance the effect among regression, local structure regularization, and adversarial training.

3.3. Learning Scheme

To better stabilize the adversarial training, our model is pre-trained first with only reconstruction loss ($\lambda_s = 5$). Afterwards, we let $\lambda_{mrf} = 0.05$ and $\lambda_{adv} = 0.001$ for fine-tuning SRN until convergence. During training, Adam solver [22] with learning rate $1e-4$ is adopted where $\beta_1 = 0.5$ and $\beta_2 = 0.9$. Training batch size is 16. The input and output are linearly scaled within range $[-1, 1]$.

4. Experiments

Our models are implemented with TensorFlow v1.4 and trained on a PC with Intel Xeon E5 (2.60GHz) CPU and an NVidia TITAN X GPU. We evaluated our method on a variety of datasets, including CelebA-HQ [21], CUB200 [45], DeepFashion [27, 28], ETHZ Synthesizability [6], Paris street view [31], Places2 [50], and Cityscapes [4]. For each dataset, models are trained on the training set and tested on the validation set. Exceptions are CUB200 and ETHZ Synthesizability, which we split as described in the supplementary material.

We train our models on three different resolution settings. 1) $128 \times 128 \rightarrow 256 \times 256$ (used for CelebA-HQ, ETHZ Synthesizability, and CUB200). 2) $64 \times 128 \rightarrow 256 \times 128$ (used for DeepFashion); 3) $256 \times 256 \rightarrow 256 \times 512$ (on Paris street view, Places2, and Cityscapes). We use input image size to indicate setting names in the following.

For visual and quantitative evaluation. We choose 3 models for comparison. Model CA is current state-of-the-art inpainting method using contextual attention layer [48]. We feed a zero-value padded full size image as input, and retrain this model using publicly available codes but with context adversarial loss instead of global and local adversarial loss for fairness. Besides, we compare with baseline model ED and SRN-HR, which have different network architectures, which will be detailed in Section 4.3.

4.1. Quantitative Evaluation

As indicated in previous image generation papers [46, 48], the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) are *not* optimal metrics for evaluating conditional image generation tasks. Thus we only provide these values for reference in Table 1. It is notable that our method yields competitive PSNR and SSIM.

Method	CelebA-HQ-2K		CUB200-1.7K		DeepFashion-3K	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ED	13.88	0.5859	14.90	0.5744	12.50	0.5677
SRN-HR	13.88	**0.6183**	**15.70**	0.6035	**12.72**	0.5686
CA [48]	13.56	0.6010	15.56	0.6467	12.58	**0.5769**
SRN	**14.01**	0.6171	15.59	**0.6473**	12.58	0.5686

Table 1. Quantitative results on the validation data.

	CelebA-HQ	CUB200	DeepFashion
SRN > CA [48]	97.54%	96.42%	93.68%
SRN > ED	96.02%	92.69%	91.13%
SRN > SRN-HR	77.69%	69.63%	62.25%

Table 2. User study statistics. Each entry gives the percentage of cases where results by our approach are judged more realistic than another solution.

Method	64×128	128×128	256×256
CA	17.35	30.56	60.44
ED	18.92	26.66	41.81
SRN-HR	17.73	28.95	52.50
SRN	11.07	18.15	36.75

Table 3. Running time for different structures (ms/image).

More convincing blind user studies of pairwise A/B tests are conducted. Each questionnaire includes 40 pairwise comparisons, regarding results from two different methods on the same input. There are 40 participants invited to user study. They are required to select the more realistic image in each pair. The images are all shown at the same resolution (256×128, 256×256, or 256×512). The comparisons are randomized across different methods, as well as in the left-right order. Participants have unlimited time to decide. In all conditions given in Table 2, our method outperforms the baselines.

Regarding efficiency, Table 3 presents the evaluation time on images of various resolutions. Note that SRN only takes up to $60\% \sim 65\%$ testing time of CA, with similar network depth, width, and capacity (17.14M *vs*. 20.62M).

4.2. Qualitative Evaluation

As shown in Figures 5 and 6, our method produces more convincing objects, portraits, and scene layouts with fine details, inferred from a limited-view input. Compared with the baseline CA, our method performs better with regard to quality of semantic structure, texture and border consistency. Moreover, since the filling margin of our model is arbitrary, SRN can infer visual context from different locations as shown in Figure 7. More results are presented in the supplementary material.

4.3. Ablation Studies

Network architectures We analyze multiple possible network designs. The compared network architectures cover three large-to-large designs and one small-to-large design. Large-to-large means the input is padded into the same size as the output first, while small-to-large directly processes the input like SRN. Large-to-large frameworks compromise vanilla encoder-decoder, SRN-HR, and coarse-to-fine networks, which are formed by two sequential encoder-decoder. Here we directly employ CA [48] as the coarse-to-fine network. The SRN-HR is an variant of SRN, which replaces the feature expansion operator in FEN with common convolution and preserves all the remaining components. Small-to-large design is SRN. The network depth and parameters are set to similar values for fairness.

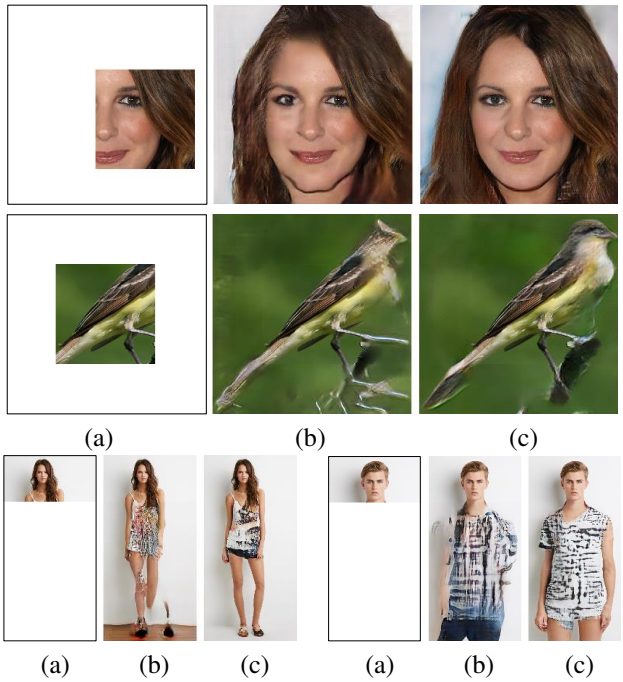


Figure 5. Visual comparisons on CelebA-HQ (top), CUB200 (middle), and DeepFashion (down). (a) Input images. (b) Results of CA [48]. (c) Our results.

Feature expansion operator	deconv	unfold	sub-pixel
PSNR	14.95	**15.06**	15.02
SSIM	0.6409	0.6412	**0.6452**

Table 4. Quantitative results of different feature expansion operators in SRN on CelebA-HQ dataset in the pre-training phase.

Figure 8 shows comparison between the given architectures. Note SRN and SRN-HR give better predictions than CA and ED on creating more natural hair and face shape with fewer visual artifacts, which validates the effectiveness of SRN design. Compared with SRN-HR, SRN produces more realistic hair texture with less inference time (Table 3), which indicates pre-filling padding for the input harms final filling performance as well as efficiency.

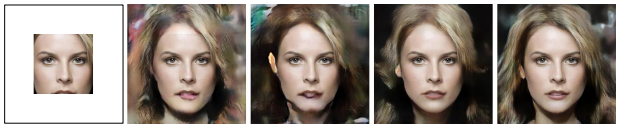
Feature Expansion Operator In our experiments, three feature expansion operators, including deconv, unfold (symmetric padding plus conv.), and sub-pixel conv., are evaluated in SRN structure. Except for these operators, other components in three SRNs are identical. We evaluate the

(a) (b) (c)

Figure 6. Visual comparison on Paris street view (top) and Cityscapes (down). (a) Input image. (b) Results of CA [48]. (c) Our results.

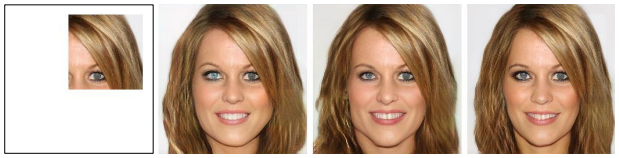
(a) (b) (a) (b) (a) (b) (a) (b)

Figure 7. Extrapolation on CelebA-HQ (top) and CUB200 (down) with arbitrary filling margin. (a) Input images. (b) Our results.



(a) (b) (c) (d) (e)

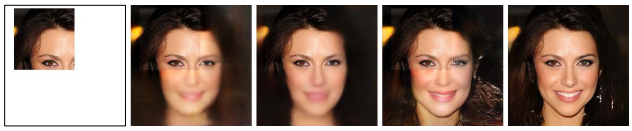
Figure 8. Visual comparison of different network structures on CelebA-HQ. (a) Input image. (b) Coarse-to-fine. (c) Naive encoder-decoder. (d) SRN-HR. (e) SRN.



(a) (b) (c) (d)

Figure 9. Visual comparison of different feature expansion operators on CelebA-HQ. (a) Input image. (b) Deconv. (c) Unfold. (d) Sub-pixel conv.

fidelity of the three SRNs on CelebA-HQ with their pre-trained models. The corresponding quantitative results of pre-trained models are given in Table 4 and the example images of full models are shown in Figure 9. Notably, the PSNR and SSIM of these three SRNs are close to each other. Results using SRN in sub-pixel level are more visual pleasing compared with that with deconv and unfold. Figure 9 shows details of facial structure and texture.



(a) (b) (c) (d) (e)

Figure 10. Visual comparison of using CN (or not) on CelebA-HQ. (a) Input image. (b) SRN w/o CN in pre-training. (c) SRN w/ CN in pre-training. (d) SRN w/o CN. (e) SRN w/ CN.

W/O Context Normalization Two SRNs are evaluated on CelebA-HQ. One of them is with context normalization (CN) module, while the other is not. Their fidelity tests

(a) (b) (c)

Figure 11. Visual comparison of different adversarial losses on CelebA-HQ. (a) Input image. (b) Vanilla global adversarial loss. (c) Context adversarial loss.

(a) (b) (c) (d)

Figure 12. Visual comparisons of different reconstruction losses on CelebA-HQ. (a) Input image. (b) Vanilla l_1 loss. (c) Confidence-driven loss. (d) Relative spatial variant loss.

Using CN	Pre-training		Full-training	
	NO	YES	NO	YES
PSNR	14.48	**15.02**	13.92	**14.01**
SSIM	0.6084	**0.6452**	0.5961	**0.6171**

Table 5. Quantitative results of using context normalization (CN) (or not) in SRN on CelebA-HQ dataset.

	RSV loss	CD loss	vanilla l_1 loss
PSNR	15.02	14.41	**15.06**
SSIM	0.6452	0.6229	**0.6478**

Table 6. Quantitative results of only using different reconstruction losses in SRN on CelebA-HQ dataset (RSV loss: relative spatial variant loss, CD loss: confidence-driven loss).

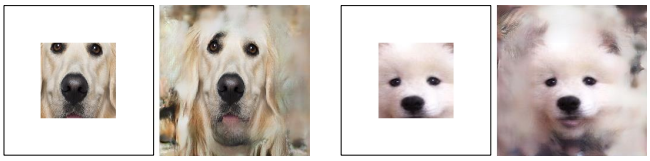
are given in Table 5 and the resulting visual prediction is shown in Figure 10. Clearly, CN improves the SRN quantitatively and qualitatively. In Figure 10, CN harmonizes color and border consistency both in pre-training and full-training phases.

Contextual Adversarial Loss *vs*. Vanilla Impr. WGAN Loss We give qualitative evaluation (Figure 11) on CelebA-HQ of these two types of GAN losses since PSNR, SSIM, and other metrics may not reflect true visual quality. The base model is SRN where relative spatial variant loss and ID-MRF loss are also employed. In Figure 11, SRN with context adversarial loss predicts clearer hair details than that with only global adversarial loss.

Relative Spatial Variant Loss *vs*. Confidence-driven Loss *vs*. Vanilla l_1 Loss Compared with common l_1 loss (where $\mathbf{M}_w = \mathbf{M}$), SRN pre-training with relative spatial variant loss (Eq. (5)) gives comparable fidelity (Table 6). However, it produces more distinctive semantic boundaries (hairline and face shape in Figure 12) than that with confidence-driven loss (where $\mathbf{M}_w = \mathbf{M}_w^c$) [44] and common l_1 loss.

(a) (b) (c)

Figure 13. Visual comparison of texture synthesis on ETHZ Synthesizability. (a) Input image. (b) CA [48]. (c) Our result.



(a) (b) (a) (b)

Figure 14. Morphing of dog images with SRN model trained on CelebA-HQ. (a) Input images. (b) Our results.

4.4. Other Applications and Limitations

Other than content extrapolation for uncropping pictures, SRN also finds applications of texture synthesis (Figure 13) and morphing (Figure 14).

About limitations, each trained model now is with specific expanding ratios (*e.g.*, a model trained for predicting three times more pixels based on the input only produces results in the same setting). Moreover, a gigantic dataset with more than thousands of scene types like Places2 is difficult to fit by a generative model. This problem may be lessened with new research breakthrough for the GAN model.

5. Concluding Remarks

We have explored a deep learning model to conduct image extrapolation for semantically sensitive objects. We summarize that the challenge lies in size expansion and one-sided constraints, and tackle them via proposing new network modules and loss design. Our method achieves promising semantic expansion effect. In future work, semi-parametric approaches will be studied when efficiency is not an issue. As shown in recent work [32, 19], this line of methods use retrieved object segments matched by input to fill the unknown region in advance, and regress raw material. Further, it is interesting to apply image expansion to videos with temporal consistency and redundant spatial information.

References

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

[2] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. *ACM Trans. Graph.*, 28(5):124, 2009.

[3] Donghyeon Cho, Jinsun Park, Tae-Hyun Oh, Yu-Wing Tai, and In So Kweon. Weakly-and self-supervised learning for content-aware deep image retargeting. In *ICCV*, 2017.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[5] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *CVPR*, 2003.

[6] Dengxin Dai, Hayko Riemenschneider, and Luc Van Gool. The synthesizability of texture examples. In *CVPR*, 2014.

[7] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.*, 31(4):82, 2012.

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2016.

[9] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[10] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.

[11] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NeurIPS*, 2015.

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017.

[14] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and improving stability in neural style transfer. In *CVPR*, 2017.

[15] Kaiming He, Huiwen Chang, and Jian Sun. Rectangling panoramic images via warping. *ACM Trans. Graph.*, 32(4):79, 2013.

[16] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[17] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4):107, 2017.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] Karim Iskakov. Semi-parametric image inpainting. *arXiv preprint arXiv:1807.02855*, 2018.

[20] Jiaya Jia and Chi-Keung Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. In *CVPR*, 2003.

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Johannes Kopf, Wolf Kienzle, Steven Drucker, and Sing Bing Kang. Quality prediction for image completion. *ACM Trans. Graph.*, 31(6):131, 2012.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[25] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *ICCV*, 2003.

[26] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018.

[27] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[28] Ziwei Liu, Sijie Yan, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Fashion landmark detection in the wild. In *ECCV*, 2016.

[29] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. *arXiv preprint arXiv:1803.02077*, 2018.

[30] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

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

[32] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *CVPR*, 2018.

[33] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *NeurIPS*, 2015.

[34] Michael Rubinstein, Diego Gutierrez, Olga Sorkine, and Ariel Shamir. A comparative study of image retargeting. *ACM Trans. Graph.*, 29(6):160, 2010.

[35] Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M Seitz. Photo uncrop. In *ECCV*, 2014.

[36] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

[37] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM Trans. Graph.*, 24(3):861–868, 2005.

[38] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, 2017.

[39] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018.

[40] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. *arXiv preprint arXiv:1708.06500*, 2017.

[41] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, 2016.

[42] D Ulyanov, A Vedaldi, and V Lempitsky. Instance normalization: the missing ingredient for fast stylization. cscv. *arXiv preprint arXiv:1607.08022*, 2017.

[43] Miao Wang, Yukun Lai, Yuan Liang, Ralph Robert Martin, and Shi-Min Hu. Biggerpicture: data-driven image extrapolation using graph matching. *ACM Trans. Graph.*, 33(6), 2014.

[44] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *NeurIPS*, 2018.

[45] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[46] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017.

[47] Raymond A Yeh, Chen Chen, Teck-Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, 2017.

[48] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint*, 2018.

[49] Yinda Zhang, Jianxiong Xiao, James Hays, and Ping Tan. Framebreak: Dramatic image extrapolation by guided shift-maps. In *CVPR*, 2013.

[50] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2018.