

Shift-Net: Image Inpainting via Deep Feature Rearrangement

Zhaoyi Yan¹[0000-0001-5155-5741], Xiaoming Li¹[0000-0003-3844-9308], Mu Li²[0000-0002-7327-3304], Wangmeng Zuo¹(✉)[0000-0002-3330-783X], and Shiguang Shan³[0000-0002-8348-392X]

- ¹ School of Computer Science and Technology, Harbin Institute of Technology, China
yanzhaoyi@outlook.com, csxmli@hit.edu.cn, wmzuo@hit.edu.cn
- ² Department of Computing, The Hong Kong Polytechnic University, Hong Kong
csmuli@comp.polyu.edu.hk
- ³ Institute of Computing Technology, CAS, Beijing, 100049, China
sgshan@ict.ac.cn

Abstract. Deep convolutional networks (CNNs) have exhibited their potential in image inpainting for producing plausible results. However, in most existing methods, e.g., context encoder, the missing parts are predicted by propagating the surrounding convolutional features through a fully connected layer, which intends to produce semantically plausible but blurry result. In this paper, we introduce a special shift-connection layer to the U-Net architecture, namely Shift-Net, for filling in missing regions of any shape with sharp structures and fine-detailed textures. To this end, the encoder feature of the known region is shifted to serve as an estimation of the missing parts. A guidance loss is introduced on decoder feature to minimize the distance between the decoder feature after fully connected layer and the ground-truth encoder feature of the missing parts. With such constraint, the decoder feature in missing region can be used to guide the shift of encoder feature in known region. An end-to-end learning algorithm is further developed to train the Shift-Net. Experiments on the Paris StreetView and Places datasets demonstrate the efficiency and effectiveness of our Shift-Net in producing sharper, fine-detailed, and visually plausible results. The codes and pre-trained models are available at <https://github.com/Zhaoyi-Yan/Shift-Net>.

Keywords: Inpainting · feature rearrangement · deep learning

1 Introduction

Image inpainting is the process of filling in missing regions with plausible hypothesis, and can be used in many real world applications such as removing distracting objects, repairing corrupted or damaged parts, and completing occluded regions. For example, when taking a photo, rare is the case that you are satisfied with what you get directly. Distracting scene elements, such as irrelevant people or disturbing objects, generally are inevitable but unwanted by the

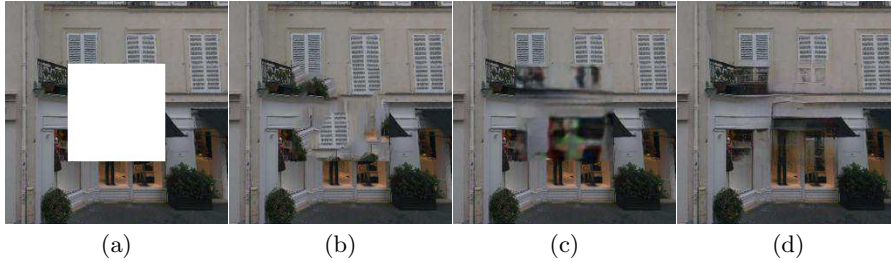


Fig. 1. Qualitative comparison of inpainting methods. Given (a) an image with a missing region, we present the inpainting results by (b) Content-Aware Fill [11], (c) context encoder [28], and (d) our Shift-Net.

users. In these cases, image inpainting can serve as a remedy to remove these elements and fill in with plausible content.

Despite decades of studies, image inpainting remains a very challenging problem in computer vision and graphics. In general, there are two requirements for the image inpainting result: (i) global semantic structure and (ii) fine detailed textures. Classical exemplar-based inpainting methods, e.g., PatchMatch [1], gradually synthesize the content of missing parts by searching similar patches from known region. Even such methods are promising in filling high-frequency texture details, they fail in capturing the global structure of the image (See Fig. 1(b)). In contrast, deep convolutional networks (CNNs) have also been suggested to predict the missing parts conditioned on their surroundings [28, 41]. Benefited from large scale training data, they can produce semantically plausible inpainting result. However, the existing CNN-based methods usually complete the missing parts by propagating the surrounding convolutional features through a fully connected layer (i.e., bottleneck), making the inpainting results sometimes lack of fine texture details and blurry. The introduction of adversarial loss is helpful in improving the sharpness of the result, but cannot address this issue essentially (see Fig. 1(c)).

In this paper, we present a novel CNN, namely Shift-Net, to take into account the advantages of both exemplar-based and CNN-based methods for image inpainting. Our Shift-Net adopts the U-Net architecture by adding a special shift-connection layer. In exemplar-based inpainting [4], the patch-based replication and filling process are iteratively performed to grow the texture and structure from the known region to the missing parts. And the patch processing order plays a key role in yielding plausible inpainting result [22, 40]. We note that CNN is effective in predicting the image structure and semantics of the missing parts. Guided by the salient structure produced by CNN, the filling process in our Shift-Net can be finished concurrently by introducing a shift-connection layer to connect the encoder feature of known region and the decoder feature of missing parts. Thus, our Shift-Net inherits the advantages of exemplar-based and CNN-based methods, and can produce inpainting result with both plausible semantics and fine detailed textures (See Fig. 1(d)).

Guidance loss, reconstruction loss, and adversarial learning are incorporated to guide the shift operation and to learn the model parameters of Shift-Net. To ensure that the decoder feature can serve as a good guidance, a guidance loss is introduced to enforce the decoder feature be close to the ground-truth encoder feature. Moreover, ℓ_1 and adversarial losses are also considered to reconstruct the missing parts and restore more detailed textures. By minimizing the model objective, our Shift-Net can be end-to-end learned with a training set. Experiments are conducted on the Paris StreetView dataset [5], the Places dataset [43], and real world images. The results show that our Shift-Net can handle missing regions with any shape, and is effective in producing sharper, fine-detailed, and visually plausible results (See Fig. 1(d)).

Besides, Yang *et al.* [41] also suggest a multi-scale neural patch synthesis (MNPS) approach to incorporating CNN-based with exemplar-based methods. Their method includes two stages, where an encoder-decoder network is used to generate an initial estimation in the first stage. By considering both global content and texture losses, a joint optimization model on VGG-19 [34] is minimized to generate the fine-detailed result in the second stage. Even Yang *et al.* [41] yields encouraging result, it is very time-consuming and takes about 40,000 millisecond (ms) to process an image with size of 256×256 . In contrast, our Shift-Net can achieve comparable or better results (See Fig. 4 and Fig. 5 for several examples) and only takes about 80 ms. Taking both effectiveness and efficiency into account, our Shift-Net can provide a favorable solution to combine exemplar-based and CNN-based inpainting for improving performance.

To sum up, the main contribution of this work is three-fold:

1. By introducing the shift-connection layer to U-Net, a novel Shift-Net architecture is developed to efficiently combine CNN-based and exemplar-based inpainting.
2. The guidance, reconstruction, and adversarial losses are introduced to train our Shift-Net. Even with the deployment of shift operation, all the network parameters can be learned in an end-to-end manner.
3. Our Shift-Net achieves state-of-the-art results in comparison with [1, 28, 41] and performs favorably in generating fine-detailed textures and visually plausible results.

2 Related Work

In this section, we briefly review the work on each of the three sub-fields, i.e., exemplar-based inpainting, CNN-based inpainting, and style transfer, and specially focus on those relevant to this work.

2.1 Exemplar-based inpainting

In exemplar-based inpainting [1, 2, 4, 6, 8, 15, 16, 20–22, 29, 33, 35, 37, 38, 40], the completion is conducted from the exterior to the interior of the missing part by

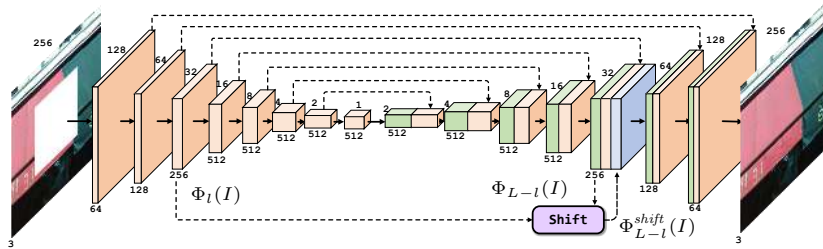


Fig. 2. The architecture of our model. We add the shift-connection layer at the resolution of 32×32 .

searching and copying best matching patches from the known region. For fast patch search, Barnes *et al.* suggest a PatchMatch algorithm [1] to exploit the image coherency, and generalize it for finding k-nearest neighbors [2]. Generally, exemplar-based inpainting is superior in synthesizing textures, but is not well suited for preserving edges and structures. For better recovery of image structure, several patch priority measures have been proposed to fill in structural patches first [4, 22, 40]. Global image coherence has also been introduced to the Markov random field (MRF) framework for improving visual quality [20, 29, 37]. However, these methods only work well on images with simple structures, and may fail in handling images with complex objects and scenes. Besides, in most exemplar-based inpainting methods [20, 21, 29], the missing part is recovered as the shift representation of the known region in pixel/region level, which also motivates our shift operation on convolution feature representation.

2.2 CNN-based inpainting

Recently, deep CNNs have achieved great success in image inpainting. Originally, CNN-based inpainting is confined to small and thin masks [19, 31, 39]. Phatak *et al.* [28] present an encoder-decoder (i.e., context encoder) network to predict the missing parts, where an adversarial loss is adopted in training to improve the visual quality of the inpainted image. Even context encoder is effective in capturing image semantics and global structure, it completes the input image with only one forward-pass and performs poorly in generating fine-detailed textures. Semantic image inpainting is introduced to fill in the missing part conditioned on the known region for images from a specific semantic class [42]. In order to obtain globally consistent result with locally realistic details, global and local discriminators have been proposed in image inpainting [13] and face completion [25]. For better recovery of fine details, MNPS is presented to combine exemplar-based and CNN-based inpainting [41].

2.3 Style transfer

Image inpainting can be treated as an extension of style transfer, where both the content and style (texture) of missing part are estimated and transferred from

the known region. In the recent few years, style transfer [3,7,9,10,12,17,24,26,36] has been an active research topic. Gatys *et al.* [9] show that one can transfer style and texture of the style image to the content image by solving an optimization objective defined on an existing CNN. Instead of the Gram matrix, Li *et al.* [24] apply the MRF regularizer to style transfer to suppress distortions and smears. In [3], local matching is performed on the convolution layer of the pre-trained network to combine content and style, and an inverse network is then deployed to generate the image from feature representation.

3 Method

Given an input image I , image inpainting aims to restore the ground-truth image I^{gt} by filling in the missing part. To this end, we adopt U-Net [32] as the baseline network. By incorporating with guidance loss and shift operation, we develop a novel Shift-Net for better recovery of semantic structure and fine-detailed textures. In the following, we first introduce the guidance loss and Shift-Net, and then describe the model objective and learning algorithm.

3.1 Guidance loss on decoder feature

The U-Net consists of an encoder and a symmetric decoder, where skip connection is introduced to concatenate the features from each layer of encoder and those of the corresponding layer of decoder. Such skip connection makes it convenient to utilize the information before and after bottleneck, which is valuable for image inpainting and other low level vision tasks in capturing localized visual details [14,44]. The architecture of the U-Net adopted in this work is shown in Fig. 2. Please refer to the supplementary material for more details on network parameters.

Let Ω be the missing region and $\bar{\Omega}$ be the known region. Given a U-Net of L layers, $\Phi_l(I)$ is used to denote the encoder feature of the l -th layer, and $\Phi_{L-l}(I)$ the decoder feature of the $(L-l)$ -th layer. For the end of recovering I^{gt} , we expect that $\Phi_l(I)$ and $\Phi_{L-l}(I)$ convey almost all the information in $\Phi_l(I^{gt})$. For any location $\mathbf{y} \in \Omega$, we have $(\Phi_l(I))_{\mathbf{y}} \approx 0$. Thus, $(\Phi_{L-l}(I))_{\mathbf{y}}$ should convey equivalent information of $(\Phi_l(I^{gt}))_{\mathbf{y}}$.

In this work, we suggest to explicitly model the relationship between $(\Phi_{L-l}(I))_{\mathbf{y}}$ and $(\Phi_l(I^{gt}))_{\mathbf{y}}$ by introducing the following guidance loss,

$$\mathcal{L}_g = \sum_{\mathbf{y} \in \Omega} \left\| (\Phi_{L-l}(I))_{\mathbf{y}} - (\Phi_l(I^{gt}))_{\mathbf{y}} \right\|_2^2. \quad (1)$$

We note that $(\Phi_l(I))_{\mathbf{x}} \approx (\Phi_l(I^{gt}))_{\mathbf{x}}$ for any $\mathbf{x} \in \bar{\Omega}$. Thus the guidance loss is only defined on $\mathbf{y} \in \Omega$ to make $(\Phi_{L-l}(I))_{\mathbf{y}} \approx (\Phi_l(I^{gt}))_{\mathbf{y}}$. By concatenating $\Phi_l(I)$ and $\Phi_{L-l}(I)$, all information in $\Phi_l(I^{gt})$ can be approximately obtained.

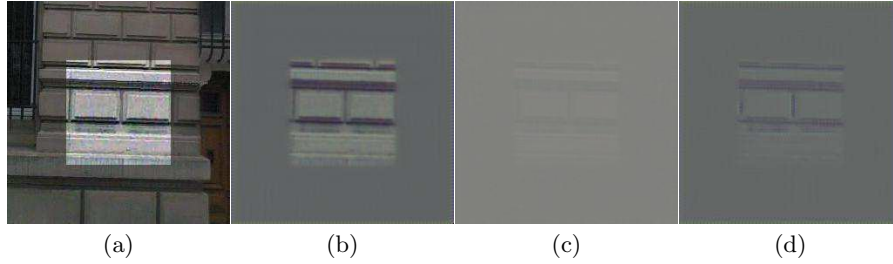


Fig. 3. Visualization of features learned by our model. Given (a) an input image, (b) is the visualization of $(\Phi_l(I^{gt}))_{\mathbf{y}}$ (i.e., H^{gt}), (c) shows the result of $(\Phi_{L-l}(I))_{\mathbf{y}}$ (i.e., H^{de}) and (d) demonstrates the effect of $(\Phi_{L-l}^{shift}(I))_{\mathbf{y}}$.

Experiment on deep feature visualization is further conducted to illustrate the relation between $(\Phi_{L-l}(I))_{\mathbf{y}}$ and $(\Phi_l(I^{gt}))_{\mathbf{y}}$. For visualizing $\{(\Phi_l(I^{gt}))_{\mathbf{y}} | \mathbf{y} \in \Omega\}$, we adopt the method [27] by solving an optimization problem

$$H^{gt} = \arg \min_H \sum_{\mathbf{y} \in \Omega} \left\| (\Phi_l(H))_{\mathbf{y}} - (\Phi_l(I^{gt}))_{\mathbf{y}} \right\|_2^2. \quad (2)$$

Analogously, $\{(\Phi_{L-l}(I))_{\mathbf{y}} | \mathbf{y} \in \Omega\}$ is visualized by

$$H^{de} = \arg \min_H \sum_{\mathbf{y} \in \Omega} \left\| (\Phi_l(H))_{\mathbf{y}} - (\Phi_{L-l}(I))_{\mathbf{y}} \right\|_2^2. \quad (3)$$

Figs. 3(b)(c) show the visualization results of H^{gt} and H^{de} . With the introduction of guidance loss, obviously H^{de} can serve as a reasonable estimation of H^{gt} , and U-Net works well in recovering image semantics and structures. However, in compared with H^{gt} and I^{gt} , the result H^{de} is blurry, which is consistent with the poor performance of CNN-based inpainting in recovering fine textures [41]. Finally, we note that the guidance loss is helpful in constructing an explicit relation between $(\Phi_{L-l}(I))_{\mathbf{y}}$ and $(\Phi_l(I^{gt}))_{\mathbf{y}}$. In the next section, we will explain how to utilize such property for better estimation to $(\Phi_l(I^{gt}))_{\mathbf{y}}$ and enhancing inpainting result.

3.2 Shift operation and Shift-Net

In exemplar-based inpainting, it is generally assumed that the missing part is the spatial rearrangement of the pixels/patches in the known region. For each pixel/patch localized at \mathbf{y} in missing part, exemplar-based inpainting explicitly or implicitly find a shift vector $\mathbf{u}_{\mathbf{y}}$, and recover $(I)_{\mathbf{y}}$ with $(I)_{\mathbf{y}+\mathbf{u}_{\mathbf{y}}}$, where $\mathbf{y} + \mathbf{u}_{\mathbf{y}} \in \bar{\Omega}$ is in the known region. The pixel value $(I)_{\mathbf{y}}$ is unknown before inpainting. Thus, the shift vectors usually are obtained progressively from the

exterior to the interior of the missing part, or by solving a MRF model by considering global image coherence. However, these methods may fail in recovering complex image semantics and structures.

We introduce a special shift-connection layer in U-Net, which takes $\Phi_l(I)$ and $\Phi_{L-l}(I)$ to obtain an updated estimation on $\Phi_l(I^{gt})$. For each $(\Phi_{L-l}(I))_{\mathbf{y}}$ with $\mathbf{y} \in \Omega$, its nearest neighbor (NN) searching based on cross-correlation in $(\Phi_l(I))_{\mathbf{x}}$ ($\mathbf{x} \in \bar{\Omega}$) can be independently obtained by,

$$\mathbf{x}^*(\mathbf{y}) = \arg \max_{\mathbf{x} \in \bar{\Omega}} \frac{\langle (\Phi_{L-l}(I))_{\mathbf{y}}, (\Phi_l(I))_{\mathbf{x}} \rangle}{\|(\Phi_{L-l}(I))_{\mathbf{y}}\|_2 \|(\Phi_l(I))_{\mathbf{x}}\|_2}, \quad (4)$$

and the shift vector is defined as $\mathbf{u}_{\mathbf{y}} = \mathbf{x}^*(\mathbf{y}) - \mathbf{y}$. We also empirically find that cross-correlation is more effective than ℓ_1 and ℓ_2 norms in our Shift-Net. Similar to [24], the NN searching can be computed as a convolutional layer. Then, we update the estimation of $(\Phi_l(I^{gt}))_{\mathbf{y}}$ as the spatial rearrangement of the encoder feature $(\Phi_l(I))_{\mathbf{x}}$,

$$\left(\Phi_{L-l}^{shift}(I) \right)_{\mathbf{y}} = (\Phi_l(I))_{\mathbf{y} + \mathbf{u}_{\mathbf{y}}}. \quad (5)$$

See Fig. 3(d) for visualization. Finally, as shown in Fig. 2, the convolution features $\Phi_{L-l}(I)$, $\Phi_l(I)$ and $\Phi_{L-l}^{shift}(I)$ are concatenated and taken as inputs to the $(L-l+1)$ -th layer, resulting in our Shift-Net.

The shift operation is different with exemplar-based inpainting from several aspects. (i) While exemplar-based inpainting is operated on pixels/patches, shift operation is performed on deep encoder feature domain which is end-to-end learned from training data. (ii) In exemplar-based inpainting, the shift vectors are obtained either by solving an optimization problem or in particular order. As for shift operation, with the guidance of $\Phi_{L-l}(I)$, all the shift vectors can be computed in parallel. (iii) For exemplar-based inpainting, both patch processing orders and global image coherence are not sufficient for preserving complex structures and semantics. In contrast, in shift operation $\Phi_{L-l}(I)$ is learned from large scale data and is more powerful in capturing global semantics. (iv) In exemplar-based inpainting, after obtaining the shift vectors, the completion result can be directly obtained as the shift representation of the known region. As for shift operation, we take the shift representation $\Phi_{L-l}^{shift}(I)$ together with $\Phi_{L-l}(I)$ and $\Phi_l(I)$ as inputs to $(L-l+1)$ -th layer of U-Net, and adopt a data-driven manner to learn an appropriate model for image inpainting. Moreover, even with the introduction of shift-connection layer, all the model parameters in our Shift-Net can be end-to-end learned from training data. Thus, our Shift-Net naturally inherits the advantages of exemplar-based and CNN-based inpainting.

3.3 Model objective and learning

Objective. Denote by $\Phi(I; \mathbf{W})$ the output of Shift-Net, where \mathbf{W} is the model parameters to be learned. Besides the guidance loss, the ℓ_1 loss and the adversarial loss are also included to train our Shift-Net. The ℓ_1 loss is defined as,

$$\mathcal{L}_{\ell_1} = \|\Phi(I; \mathbf{W}) - I^{gt}\|_1, \quad (6)$$

which is suggested to constrain that the inpainting result should approximate the ground-truth image.

Moreover, adversarial learning has been adopted in low level vision [23] and image generation [14, 30], and exhibits its superiority in restoring fine details and photo-realistic textures. Thus, we use $p_{data}(I^{gt})$ to denote the distribution of ground-truth images, and $p_{miss}(I)$ to denote the distribution of input image. Then the adversarial loss is defined as,

$$\mathcal{L}_{adv} = \min_{\mathbf{W}} \max_D \mathbb{E}_{I^{gt} \sim p_{data}(I^{gt})} [\log D(I^{gt})] \quad (7)$$

$$+ \mathbb{E}_{I \sim p_{miss}(I)} [\log(1 - D(\Phi(I; \mathbf{W})))] \quad (8)$$

where $D(\cdot)$ denotes the discriminator to predict the probability that an image is from the distribution $p_{data}(I^{gt})$.

Taking guidance, ℓ_1 , and adversarial losses into account, the overall objective of our Shift-Net is defined as,

$$\mathcal{L} = \mathcal{L}_{\ell_1} + \lambda_g \mathcal{L}_g + \lambda_{adv} \mathcal{L}_{adv}, \quad (9)$$

where λ_g and λ_{adv} are two tradeoff parameters.

Learning. Given a training set $\{(I, I^{gt})\}$, the Shift-Net is trained by minimizing the objective in Eqn. (9) via back-propagation. We note that the Shift-Net and the discriminator are trained in an adversarial manner. The Shift-Net $\Phi(I; \mathbf{W})$ is updated by minimizing the adversarial loss \mathcal{L}_{adv} , while the discriminator D is updated by maximizing \mathcal{L}_{adv} .

Due to the introduction of shift-connection, we should modify the gradient w.r.t. the l -th layer of feature $F_l = \Phi_l(I)$. To avoid confusion, we use F_l^{skip} to denote the feature F_l after skip connection, and of course we have $F_l^{skip} = F_l$. According to Eqn. (5), the relation between $\Phi_{L-l}^{shift}(I)$ and $\Phi_l(I)$ can be written as,

$$\Phi_{L-l}^{shift}(I) = \mathbf{P}\Phi_l(I), \quad (10)$$

where \mathbf{P} denotes the shift matrix of $\{0, 1\}$, and there is only one element of 1 in each row of \mathbf{P} . Thus, the gradient with respect to $\Phi_l(I)$ consists of three terms: (i) that from $(l+1)$ -th layer, (ii) that from skip connection, and (iii) that from shift-connection, and can be written as,

$$\frac{\partial \mathcal{L}}{\partial F_l} = \frac{\partial \mathcal{L}}{\partial F_l^{skip}} + \frac{\partial \mathcal{L}}{\partial F_{l+1}} \frac{\partial F_{l+1}}{\partial F_l} + \mathbf{P}^T \frac{\partial \mathcal{L}}{\partial \Phi_{L-l}^{shift}(I)}, \quad (11)$$

where the computation of the first two terms are the same with U-Net, and the gradient with respect to $\Phi_{L-l}^{shift}(I)$ can also be directly computed. Thus, our Shift-Net can also be end-to-end trained to learn the model parameters \mathbf{W} .

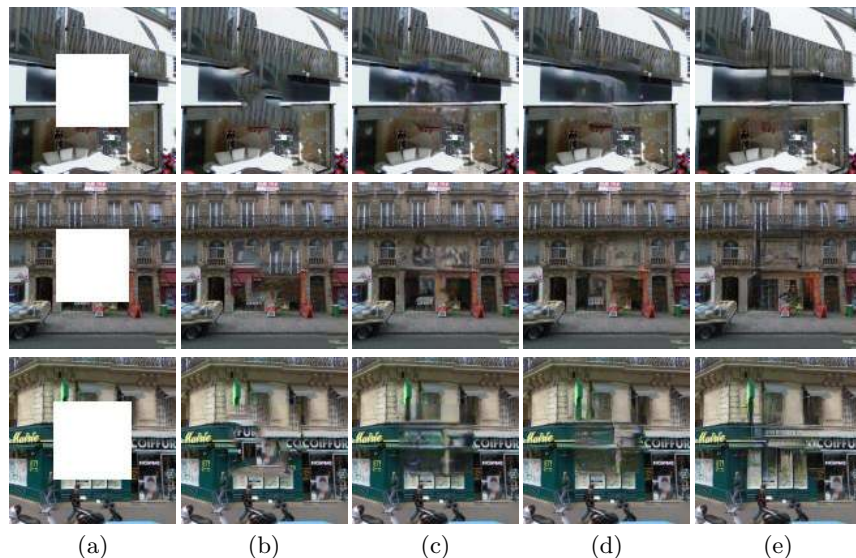


Fig. 4. Qualitative comparisons on the Paris StreetView dataset. From the left to the right are: (a) input, (b) Content-Aware Fill [11], (c) context encoder [28], (d) MNPS [41] and (e) Ours. All images are scaled to 256×256 .

4 Experiments

We evaluate our method on two datasets: Paris StreetView [5] and six scenes from Places365-Standard dataset [43]. The Paris StreetView contains 14,900 training images and 100 test images. We randomly choose 20 out of the 100 test images in Paris StreetView to form the validation set, and use the remaining as the test set. There are 1.6 million training images from 365 scene categories in the Places365-Standard. The scene categories selected from Places365-Standard are *butte*, *canyon*, *field*, *synagogue*, *tundra* and *valley*. Each category has 5,000 training images, 900 test images and 100 validation images. The details of model selection are given in the supplementary materials. For both Paris StreetView and Places, we resize each training image to let its minimal length/width be 350, and randomly crop a subimage of size 256×256 as input to our model. Moreover, our method is also tested on real world images for removing objects and distractors. Our Shift-Net is optimized using the Adam algorithm [18] with a learning rate of 2×10^{-4} and $\beta_1 = 0.5$. The batch size is 1 and the training is stopped after 30 epochs. Data augmentation such as flipping is also adopted during training. The tradeoff parameters are set as $\lambda_g = 0.01$ and $\lambda_{adv} = 0.002$. It takes about one day to train our Shift-Net on an Nvidia Titan X Pascal GPU.

4.1 Comparisons with state-of-the-arts

We compare our results with Photoshop Content-Aware Fill [11] based on [1], context encoder [28], and MNPS [41]. As context encoder only accepts 128×128

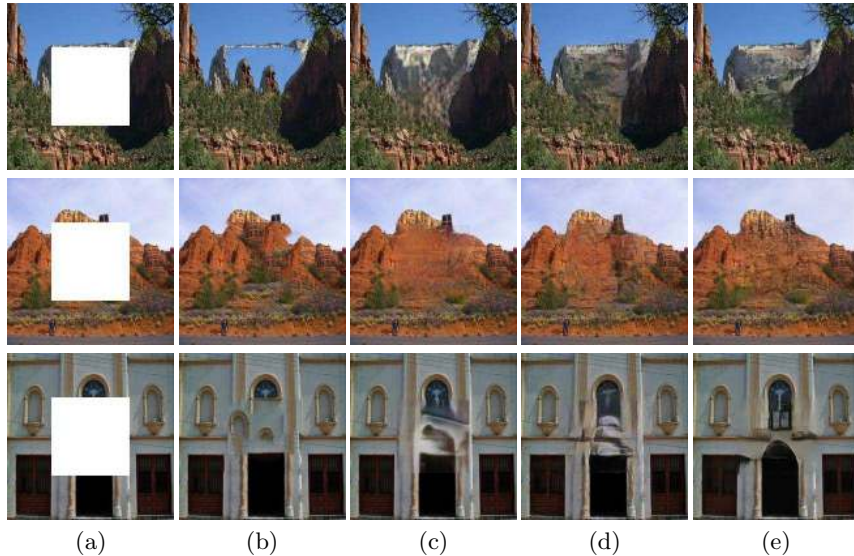


Fig. 5. Qualitative comparisons on the Places. From the left to the right are: (a) input, (b) Content-Aware Fill [11], (c) context encoder [28], (d) MNPS [41] and (e) Ours. All images are scaled to 256×256 .

images, we upsample the results to 256×256 . For MNPS [41], we set the pyramid level be 2 to get the resolution of 256×256 .

Evaluation on Paris StreetView and Places. Fig. 4 shows the comparisons of our method with the three state-of-the-art approaches on Paris StreetView. Content-Aware Fill [11] is effective in recovering low level textures, but performs slightly worse in handling occlusions with complex structures. Context encoder [28] is effective in semantic inpainting, but the results seem blurry and detail-missing due to the effect of bottleneck. MNPS [41] adopts a multi-stage scheme to combine CNN and exemplar-based inpainting, and generally works better than Content-Aware Fill [11] and context encoder [28]. However, the multi-scales in MNPS [41] are not jointly trained, where some adverse effects produced in the first stage may not be eliminated by the subsequent stages. In comparison to the competing methods, our Shift-Net combines CNN and exemplar-based inpainting in an end-to-end manner, and generally is able to generate visual-pleasing results. Moreover, we also note that our Shift-Net is much more efficient than MNPS [41]. Our method consumes only about 80 ms for a 256×256 image, which is about $500\times$ faster than MNPS [41] (about 40 seconds). In addition, we also evaluate our method on the Places dataset (see Fig. 5). Again our Shift-Net performs favorably in generating fine-detailed, semantically plausible, and realistic images.

Quantitative evaluation. We also compare our model quantitatively with the competing methods on the Paris StreetView dataset. Table 1 lists the PSNR, SSIM and mean ℓ_2 loss of different methods. Our Shift-Net achieves the best

Table 1. Comparison of PSNR, SSIM and mean ℓ_2 loss on Paris StreetView dataset.

Method	PSNR	SSIM	Mean ℓ_2 Loss
Content-Aware Fill [11]	23.71	0.74	0.0617
context encoder [28] (ℓ_2 + adversarial loss)	24.16	0.87	0.0313
MNPS [41]	25.98	0.89	0.0258
Ours	26.51	0.90	0.0208

**Fig. 6.** Random region completion. From top to bottom are: input, Content-Aware Fill [11], and Ours.

numerical performance. We attribute it to the combination of CNN-based with exemplar-based inpainting as well as the end-to-end training. In comparison, MNPS [41] adopts a two-stage scheme and cannot be jointly trained.

Random mask completion. Our model can also be trained for arbitrary region completion. Fig. 6 shows the results by Content-Aware Fill [11] and our Shift-Net. For textured and smooth regions, both Content-Aware Fill [11] and our Shift-Net perform favorably. While for structural region, our Shift-Net is more effective in filling the cropped regions with context coherent with global content and structures.

4.2 Inpainting of real world images

We also evaluate our Shift-Net trained on Paris StreetView for the inpainting of real world images by considering two types of missing regions: (i) central region, (ii) object removal. From the first row of Fig. 7, one can see that our Shift-Net trained with central mask can be generalized to handle real world images. From



Fig. 7. Results on real images. From the top to bottom are: central region inpainting, and object removal.

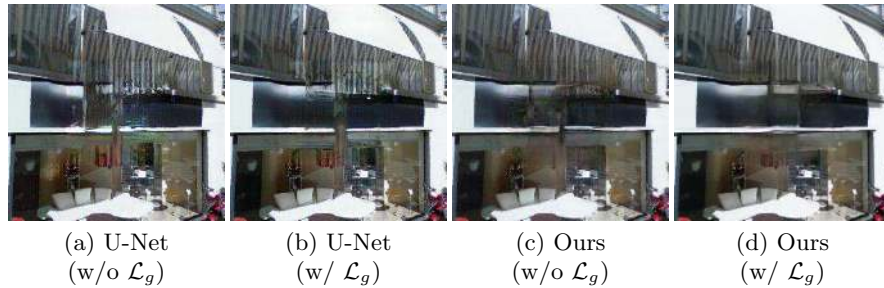


Fig. 8. The effect of guidance loss \mathcal{L}_g in U-Net and our Shift-Net.

the second row of Fig. 7, we show the feasibility of using our Shift-Net trained with random mask to remove unwanted objects from the images.

5 Ablation Studies

The main differences between our Shift-Net and the other methods are the introduction of guidance loss and shift-connection layer. Thus, experiments are first conducted to analyze the effect of guidance loss and shift operation. Then we respectively zero out the corresponding weight of $(L - l + 1)$ -th layer to verify the effectiveness of the shift feature Φ_{L-l}^{shift} in generating fine-detailed results. Moreover, the benefit of shift-connection does not owe to the increase of feature map size. So we also compare Shift-Net with a baseline model by substituting the NN searching with random shift-connection in the supplementary materials.

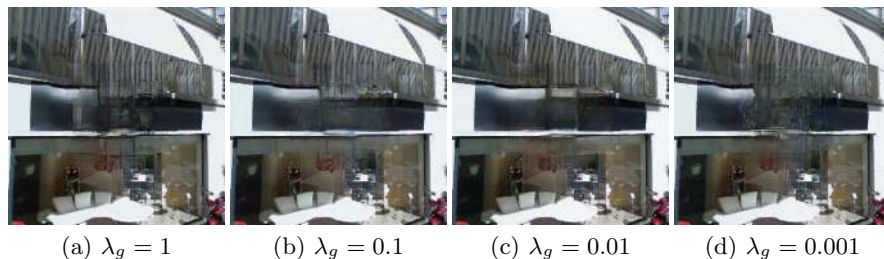


Fig. 9. The effect of the tradeoff parameter λ_g of guidance loss.

5.1 Effect of guidance loss

Two groups of experiments are conducted to evaluate the effect of guidance loss. In the first group, we add and remove the guidance loss \mathcal{L}_g for U-Net and our Shift-Net to train the models. Fig. 8 shows the inpainting results by these four methods. It can be observed that, for both U-Net and Shift-Net the guidance loss is helpful in suppressing artifacts and preserving salient structure.

In the second group, we evaluate the effect of tradeoff parameter λ_g . Note that the guidance loss is introduced for both recovering the semantic structure of missing region and guiding the shift of encoder feature. Thus, proper tradeoff parameter λ_g should be chosen. Fig. 9 shows the results by setting different λ_g values. When λ_g is small (e.g., = 0.001), the decoder feature may not serve as a suitable guidance to guarantee the correct shift of the encoder feature. From Fig. 9(d), some artifacts can still be observed. When λ_g becomes too large (e.g., ≥ 0.1), the constraint will be too excessive, and artifacts may also be introduced (see Fig. 9(a)(b)). Thus, we empirically set $\lambda_g = 0.01$ in our experiments.

5.2 Effect of shift operation at different layers

The shift operation can be deployed to different layer, e.g., $(L-l)$ -th, of the decoder. When l is smaller, the feature map size goes larger, and more computation time is required to perform the shift operation. When l is larger, the feature map size becomes smaller, but more detailed information may be lost in the corresponding encoder layer. Thus, proper l should be chosen for better tradeoff between computation time and inpainting performance. Fig. 10 shows the results of Shift-Net by adding the shift-connection layer to each of the $(L-4)$ -th, $(L-3)$ -th, and $(L-2)$ -th layers, respectively. When the shift-connection layer is added to the $(L-2)$ -th layer, Shift-Net generally works well in producing visually pleasing results, but it takes more time, i.e., ~ 400 ms per image (See Fig. 10(d)). When the shift-connection layer is added to the $(L-4)$ -th layer, Shift-Net becomes very efficient (i.e., ~ 40 ms per image) but tends to generate the result with less textures and coarse details (See Fig. 10(b)). By performing the shift operation in $(L-3)$ -th layer, better tradeoff between efficiency (i.e., ~ 80 ms per image) and performance can be obtained by Shift-Net (See Fig. 10(c)).

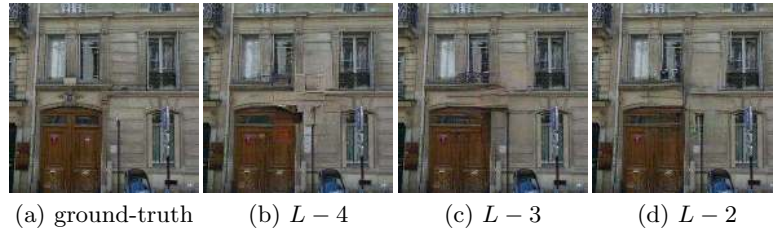


Fig. 10. The effect of performing shift operation on different layers $L - l$.



Fig. 11. Given (a) the input, (b), (c) and (d) are respectively the results when the 1st, 2nd, 3rd parts of weights in $(L - l + 1)$ -th layer are zeroed. (e) is the result of Ours.

5.3 Effect of the shifted feature

The $(L - l + 1)$ -th layer of Shift-Net takes $\Phi_{L-l}(I)$, $\Phi_l(I)$ and Φ_{L-l}^{shift} as inputs. To analyze their effect, Fig. 11 shows the results of Shift-Net by zeroing out the weight of each slice in $(L - l + 1)$ -th layer. When we abandon $\Phi_{L-l}(I)$, the central part fails to restore any structures (See Fig. 11(b)). When we ignore $\Phi_l(I)$, the general structure can be restored (See Fig. 11(c)) but its quality is inferior to the final result in Fig. 11(e). Finally, when we discard the shift feature Φ_{L-l}^{shift} , the result becomes totally a mixture of structures (See Fig. 11(d)). Thus, we conclude that Φ_{L-l}^{shift} acts as a refinement and enhancement role in recovering clear and fine details in our Shift-Net.

6 Conclusion

This paper proposes a novel Shift-Net for image completion that exhibits fast speed with promising fine details via deep feature rearrangement. The guidance loss is introduced to enhance the explicit relation between the encoded feature in known region and decoded feature in missing region. By exploiting such relation, the shift operation can be efficiently performed and is effective in improving inpainting performance. Experiments show that our Shift-Net performs favorably in comparison to the state-of-the-art methods, and is effective in generating sharp, fine-detailed and photo-realistic images. In future, more studies will be given to extend the shift-connection to other low level vision tasks.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under grant No.s 61671182 and 61471146.

References

1. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. In: *ACM Transactions on Graphics (TOG)*. vol. 28, p. 24. ACM (2009)
2. Barnes, C., Shechtman, E., Goldman, D.B., Finkelstein, A.: The generalized patch-match correspondence algorithm. In: *European Conference on Computer Vision*. pp. 29–43. Springer (2010)
3. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337* (2016)
4. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. vol. 2, pp. II–II. IEEE (2003)
5. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.: What makes paris look like paris? *ACM Transactions on Graphics* **31**(4) (2012)
6. Drori, I., Cohen-Or, D., Yeshurun, H.: Fragment-based image completion. In: *ACM Transactions on graphics (TOG)*. vol. 22, pp. 303–312. ACM (2003)
7. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. *arXiv preprint arXiv:1610.07629* (2016)
8. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. vol. 2, pp. 1033–1038. IEEE (1999)
9. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015)
10. Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., Shechtman, E.: Controlling perceptual factors in neural style transfer. *arXiv preprint arXiv:1611.07865* (2016)
11. Goldman, D., Shechtman, E., Barnes, C., Belaunde, I., Chien, J.: Content-aware fill. <https://research.adobe.com/project/content-aware-fill>
12. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. *arXiv preprint arXiv:1703.06868* (2017)
13. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)* **36**(4), 107:1–107:14 (2017)
14. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004* (2016)
15. Jia, J., Tang, C.K.: Image repairing: Robust image synthesis by adaptive nd tensor voting. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. vol. 1, pp. I–I. IEEE (2003)
16. Jia, J., Tang, C.K.: Inference of segmented color and texture description by tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(6), 771–786 (2004)
17. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: *European Conference on Computer Vision*. pp. 694–711. Springer (2016)
18. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. *international conference on learning representations* (2015)
19. Köhler, R., Schuler, C., Schölkopf, B., Harmeling, S.: Mask-specific inpainting with deep neural networks. In: *German Conference on Pattern Recognition*. pp. 523–534. Springer (2014)

20. Komodakis, N.: Image completion using global optimization. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 1, pp. 442–452. IEEE (2006)
21. Komodakis, N., Tziritas, G.: Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing* **16**(11), 2649–2661 (2007)
22. Le Meur, O., Gautier, J., Guillemot, C.: Exemplar-based inpainting based on local geometry. In: Image Processing (ICIP), 2011 18th IEEE International Conference on. pp. 3401–3404. IEEE (2011)
23. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802* (2016)
24. Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2479–2486 (2016)
25. Li, Y., Liu, S., Yang, J., Yang, M.H.: Generative face completion. *arXiv preprint arXiv:1704.05838* (2017)
26. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. *arXiv preprint arXiv:1703.07511* (2017)
27. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5188–5196 (2015)
28. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2536–2544 (2016)
29. Pritch, Y., Kav-Venaki, E., Peleg, S.: Shift-map image editing. In: Computer Vision, 2009 IEEE 12th International Conference on. pp. 151–158. IEEE (2009)
30. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015)
31. Ren, J.S., Xu, L., Yan, Q., Sun, W.: Shepard convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 901–909 (2015)
32. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015)
33. Simakov, D., Caspi, Y., Shechtman, E., Irani, M.: Summarizing visual data using bidirectional similarity. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
35. Sun, J., Yuan, L., Jia, J., Shum, H.Y.: Image completion with structure propagation. *ACM Transactions on Graphics (ToG)* **24**(3), 861–868 (2005)
36. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: Feed-forward synthesis of textures and stylized images. In: ICML. pp. 1349–1357 (2016)
37. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 1, pp. I–I. IEEE (2004)
38. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence* **29**(3) (2007)

39. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: *Advances in Neural Information Processing Systems*. pp. 341–349 (2012)
40. Xu, Z., Sun, J.: Image inpainting by patch propagation using patch sparsity. *IEEE transactions on image processing* **19**(5), 1153–1165 (2010)
41. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017)
42. Yeh, R.A., Chen, C., Lim, T.Y., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with deep generative models. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5485–5493 (2017)
43. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)
44. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593* (2017)