Very Long Natural Scenery Image Prediction by Outpainting

Zongxin Yang[1,2] Jian Dong[3] Ping Liu[2] Yi Yang[2] Shuicheng Yan[4]
[1]SUSTech-UTS Joint Centre of CIS, Southern University of Science and Technology
[2]ReLER, University of Technology Sydney [3] Qihoo 360 [4] Yitu Technology
zongxin.yang@student.uts.edu.au, dongjian-iri@360.cn, {ping.liu,yi.yang}@uts.edu.au, shuicheng.yan@yitu-inc.com

Abstract

Comparing to image inpainting, image outpainting receives less attention due to two challenges in it. The first challenge is how to keep the spatial and content consistency between generated images and original input. The second challenge is how to maintain high quality in generated results, especially for multi-step generations in which generated regions are spatially far away from the initial input. To solve the two problems, we devise some innovative modules, named Skip Horizontal Connection and Recurrent Content Transfer, and integrate them into our designed encoder-decoder structure. By this design, our network can generate highly realistic outpainting prediction effectively and efficiently. Other than that, our method can generate new images with very long sizes while keeping the same style and semantic content as the given input. To test the effectiveness of the proposed architecture, we collect a new scenery dataset with diverse, complicated natural scenes. The experimental results on this dataset have demonstrated the efficacy of our proposed network.

1. Introduction

Image outpainting, as illustrated in Fig. 1, is to generate new contents beyond the original boundaries for a given image. The generated image should be consistent with the given input, both on spatial configuration and semantic content. Although image outpainting can be used in various applications, the solutions with promising results are still in shortage due to the difficulties of this problem.

The difficulties for image outpainting exist in two aspects. First, it is not easy to keep the generated image consistent with the given input in terms of the spatial configuration and semantic content. Previous works, e.g., [28] needs local warping to make sure there is no sudden change between the input image and the generated region, especially around the boundaries of the two images. Second, it is hard to make the generated image look realistic since it has less contextual information comparing with image inpainting [2, 16].

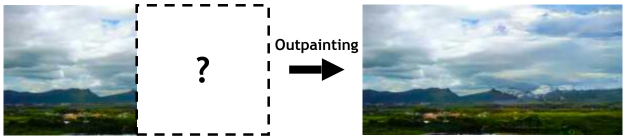


Figure 1. Illustration of image outpainting in one step. Given an image as input, image outpainting generates a new image with the same size but outside the original boundary. The spatial configuration and semantic meaning between generated images and the original input must keep consistent.

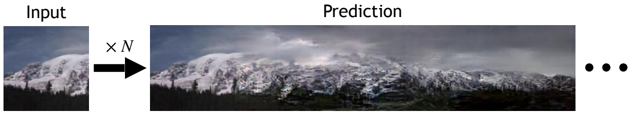


Figure 2. Illustration of image outpainting for natural scenery images horizontally in multi-steps.

For solving image outpainting problems, a few preliminary works were published [14, 23, 34, 28]. However, none of those works [14, 23, 34, 28] utilize ConvNets. Those works attempt to "search" image patch(es) from given candidates, concatenate the best match(es) with the original input spatially. Those works have their limitations: (1) they need handcrafted features to summarize the image; (2) they need image processing techniques, for example, local warping [28], to make sure there is no sudden visual change between input and generated images; (3) the final performance is heavily dependent on the size of the candidate pool.

Inspired by the success of deep networks on inpainting problems [17], we draw on a similar encoder-decoder structure with a global and a local adversarial loss, to solve image outpainting. In our architecture, the encoder is to compress the given input into a compact feature representation, and the decoder generates a new image based on the compact feature representation. More than that, to solve the two challenging problems in image outpainting, we make several innovative improvements in our architecture.

To make the generated images spatial and semantic consistent with original input, it is necessary to take full advantages of the information from the encoder and fuse it into the decoder. For this purpose, we design a **Skip Hori-**

zontal Connection (SHC) to connect encoder and decoder at each same level. By this way, the decoder can generate a prediction with strong regards to the input. Our experimental results prove that the proposed SHC can improve the smoothness and reality of the generated image.

Moreover, we propose **Recurrent Content Transfer (RCT)**, to transfer the sequence from the encoder to the decoder to generate new contents. Compared to channel-wise full connection strategy in the previous work [17], RCT can facilitate our network to handle the spatial relationship in the horizontal direction more effectively. Besides, by adjusting the length of the prediction feature, RCT assists our architecture in controlling the prediction size conveniently, which is hard if utilizing full connection.

By integrating the proposed *SHC* and *RCT* into our designed encoder-decoder architecture, our method can successfully generate images with *extra length* outside the boundary of the given image. As shown in Figure. 2, it is a recursive process since the generation from the last step is utilized as the input for the current step, which, theoretically, can generate smooth, and realistic images with a very long size. Those generated images, although spatially far away from the given input and thus receiving little contextual information from it, still keep high qualities.

To demonstrate the effectiveness of our method, we collect a new scenery dataset with $6,000$ images, which consists of diverse, complicated natural scenes, including mountain with or without snow, valley, seaside, riverbank, starry sky, etc. We conduct a series of experiments on this dataset and not surprisingly beat all competitors [12, 10, 32].

Contributions. Our contributions are summarized in the following aspects:

(1) we design a novel encoder-decoder framework to handle image outpainting, which is rarely discussed before;

(2) we propose Skip Horizontal Connection and Recurrent Content Transfer, and integrate them into our designed architecture, which not only significantly improves the consistency on spatial configuration and semantic content, but also enables our architecture with an excellent ability for long-term prediction;

(3) we collect a new outpainting dataset, which has $6,000$ images containing complex natural scenes. We validate the effectiveness of our proposed network on this dataset.

2. Related Work

In this section, we briefly review the previous works relating to this paper in five sub-fields: Convolutional Neural Networks, Generative Adversarial Networks, Image Inpainting, Image Outpainting, and Image-to-Image Translation.

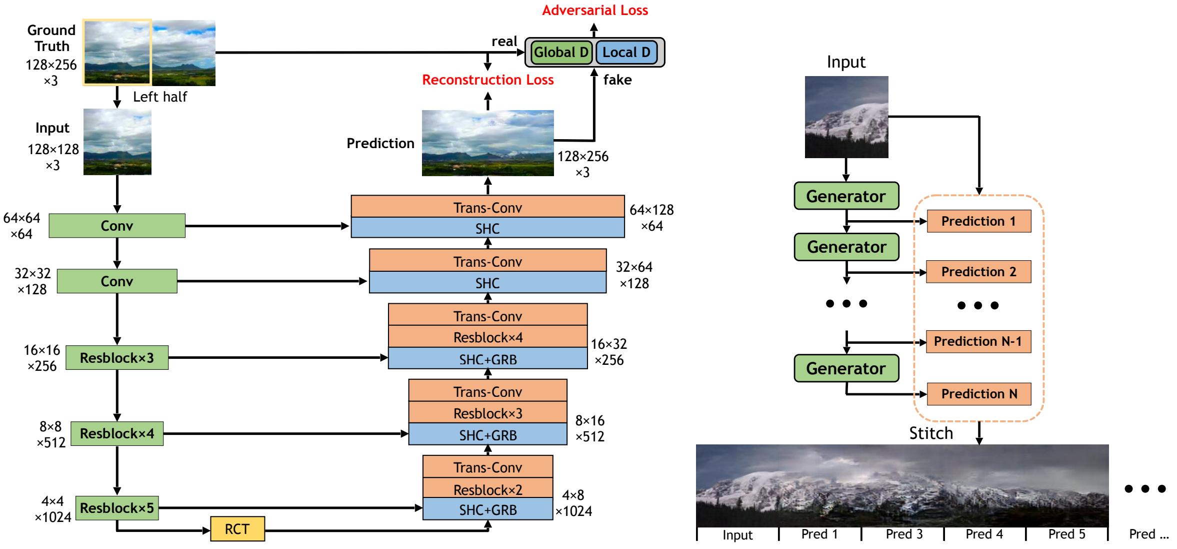
Convolutional Neural Networks (ConvNets) VGGNets[22] and Inception models [25] demonstrate the benefits of deep network. To train deeper networks, Highway networks [24] employ a gating mechanism to regulate shortcut connections. ResNet [7] simplifies the shortcut connection and shows the effectiveness of learning deeper networks through the use of identity-based skip connections. Due to the complexity of our task, we employ a group of ”bottleneck” ResBlocks [7] to build our network and utilize residual connections in Skip Horizontal Connection to improve the smoothness of the generated results.

Generative Adversarial Networks (GANs) GAN [5] has achieved success in various problems, including image generation [3, 18], image inpainting [17], future prediction [15], and style translation [35]. The key to the success of GANs is the introduction of the adversarial loss, which forces the generator to captures the true data distribution. To improve the training of GAN, variants of GANs have been derived. For example, WGAN-GP [6] introduces a gradient penalty and achieves more stable training. And thus we utilize WGAN-GP in this work due to its advantages.

Image Inpainting The classical image inpainting [2, 16] approaches utilize local non-semantic methods to predict the missing region. However, when the missing region size becomes huge, or the context grows complex, the quality of the final results deteriorates [17, 30, 10, 32]. Compared to image inpainting, image outpainting is more challenging. To the best of our knowledge, there is NO other peer-reviewed published work utilizing ConvNets for image outpainting before our work.

Image Outpainting There are a few preliminary published works [14, 23, 34, 27] for image outpainting problems, but none of them utilized ConvNets. Those works employed image matching strategies to “search” image patch(es) from the input image or an image library, and treat the patch(es) as prediction regions. If the search fails, the final “prediction” result will be inconsistent with the given context. Unlike those previous work [14, 23, 34, 27], our approach does not need any image matching strategy but depends on our carefully designed deep network.

Image-to-Image Translation With the development of ConvNets, recent approaches [12, 5, 21, 35] for image-to-image translation design deep networks for learning a parametric translation function. After “Pix2Pix” [12] framework, which use a conditional adversarial network [5] to learn a mapping from input to output images, similar ideas have been applied to related tasks, such as translating sketches to photographs [21], style translation [35, 4], etc. Although image outpainting is similar to the image-to-image translation task, there is a significant difference between them: for image-to-image translation, the input and output keep the same semantic content but change details



(a) Overview

(b) Multi-Step Generation

Figure 3. (a) The overall architecture consists of a generator and a discriminator. The generator exploits an encoder-decoder pipeline. We propose Recurrent Content Transfer (RCT) to link the encoder and decoder. Meanwhile, We deploy Skip Horizontal Connection (SHC) to connect the encoder and decoder at each symmetrical level. Moreover, after the first three SHC layers, we deploy Global Residual Blocks (GRB), which has a large receptive field, to further strengthen the connection between the predicted and original region. (b) We can generate an image with very long sizes by iterating the generator.

or styles; for our work, the style is shared between the input and output, the semantic contents are different but keep consistent.

3. Methodology

We first provide an overview of the overall architecture, which is shown in Fig. 3, then provide details on each component.

3.1. Encoder-Decoder Architecture

We design an encoder-decoder architecture for image outpainting. Our encoder takes an input image and extracts its latent feature representation; the decoder takes this latent representation to generate a new image with the same size, which has consistent content and the same style.

Encoder Our encoder is derived from the *ResNet-50* [7]. The difference is that we replace *max pooling* layers with convolutional layers, and remove layers after *conv4_5*. Given an input image I of size 128×128, the encoder will compute a latent representation with the dimension of 4×4×1024.

As pointed out in [17], it is difficult only to utilize convolutional layers to propagate information from input image feature maps to predicted feature maps. The reason is that there is no one-to-one correspondence between them under this circumstance. In Context Encoders [17], this information propagation is handled by *channel-wise fully-connected* (FC) layers. One of the limitations in FC layers is they can only handle features of fixed sizes. In our practice, this limitation will make predicted results deteriorate when the input size is large (Fig. 7(b)). More than that, as illustrated by [17], FC layers occupy a huge amount of parameters, which makes the training inefficient or imprac-

layer	output size	parameters
Conv	64×64×64	4×4, stride=2
Conv	32×32×128	4×4, stride=2
ResBlock×3	16×16×256	stride of first block=2
ResBlock×4	8×8×512	stride of first block=2
ResBlock×5	4×4×1024	stride of first block=2
RCT	4×4×1024	None
SHC+GRB	4×8×1024	dilated rate=1
ResBlock×2	4×8×1024	None
Trans-Conv	8×16×512	4×4, stride=2
SHC+GRB	8×16×512	dilated rate=2
ResBlock×3	8×16×512	None
Trans-Conv	16×32×256	4×4, stride=2
SHC+GRB	16×32×256	dilated rate=4
ResBlock×4	16×32×256	None
Trans-Conv	32×64×128	4×4, stride=2
SHC	32×64×128	None
Trans-Conv	64×128×64	4×4, stride=2
SHC	64×128×64	None
Trans-Conv	128×256×3	4×4, stride=2

Table 1. The specific parameters of generator. Trans-Conv is transposed convolution.

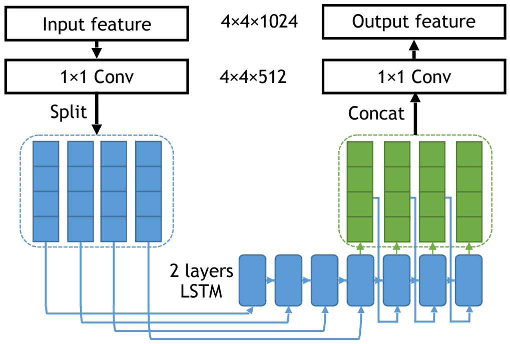


Figure 4. The illustration of Recurrent Content Transfer (RCT). 1×1 convolutional layers are utilized to adjust the channel dimension of input and output of RCT. RCT splits the feature representation of input to a sequence in the horizontal direction, and uses two *LSTM* layers to transfer this sequence to a predicted sequence. The size of the prediction region can be adjusted by setting the length of the prediction sequence in 1-step prediction, which is set to 4 to achieve a satisfactory result in our practice.

tical. To deal with those problems, we propose a Recurrent Content Transfer (RCT) layer for information propagation in our network.

Recurrent Content Transfer RCT, which is shown in Figure. 4, is designed for efficient information propagation between feature sequences from input regions and prediction regions respectively. Specifically, RCT splits the feature maps from the input region to a sequence in the horizontal dimension, and then uses two *LSTM* [9] layers to transfer this sequence to a new sequence corresponding to the prediction region. After that, the new sequence is concatenated and reshaped into predicted feature maps. 1×1 convolutional layers are utilized to adjust the channel dimensions of input and output in RCT. Given input feature maps with a size of $4\times4\times1024$, RCT outputs feature maps with the same dimension.

Benefit from the recurrent structure in RCT, we can control the size of the prediction region by setting the length of the prediction sequence in 1-step prediction. And by iterating the model, we can generate images with high-quality and very long range (Fig. 11, 12).

Decoder Decoder takes $4\times4\times1024$ dimensional features, which are encoded from a 128×128 image (*I*), to generate an image of size 128×256. The left half of the generated image is the same as the input image *I*; the right half is predicted by our architecture. Similar to the most recent methods, we use five *transposed-convolutional* layers [33] in the decoder to expand the spatial size and reduce the channel number. However, unlike the previous work [17], before each *transposed-convolutional* layer, we propose to use our designed **Skip Horizontal Connection (SHC)** to fuse the feature from the encoder into the decoder.

Skip Horizontal Connection Inspired by U-Net [19], we propose SHC, which is shown in Fig. 5(a), to share information from the encoder to the decoder at the same level. The difference between SHC and U-Net [19] is that the spacial size of the encoder feature is different from the decoder in SHC. SHC focuses on the left half of the decoder feature which corresponds to the original input region.

As illustrated in Fig. 5(a), given a feature $D_{h,w,c}$ from decoder and a feature $E_{h,\frac{w}{2},c}$ from encoder, SHC computes a new feature $D^{'}_{h,w,c}$. The procedures are as follows: First, we concatenate the left half of $D_{h,w,c}$, denoted as $D^{left}_{h,\frac{w}{2},c}$, with $E_{h,\frac{w}{2},c}$ on the channel dimension; then, we pass this concatenated feature through three convolutional layers, which have kernels of 1×1, 3×3 and 1×1 size respectively, to get to a feature representation denoted as $E^{'}_{h,\frac{w}{2},c}$. To make the training more stable, we introduce a residual connection to make a element-wise addition between $E^{'}_{h,\frac{w}{2},c}$ and $E_{h,\frac{w}{2},c}$. We denote the addition result as $D^{left'}_{h,\frac{w}{2},c}$. We use $D^{left'}_{h,\frac{w}{2},c}$ to replace the left half of the input feature for SHC, $D_{h,w,c}$, to get the final output for SHC, denoted as $D^{'}_{h,w,c}$.[1]

Besides, to keep a balance between the insufficient context due to small kernel sizes and the high computation cost introduced by large kernel dimensions, we propose to combine the advantage of Residual Block [7] and *Inception* into a novel block: *Global Residual Block (GRB)*, which is shown in Figure. 5(b).

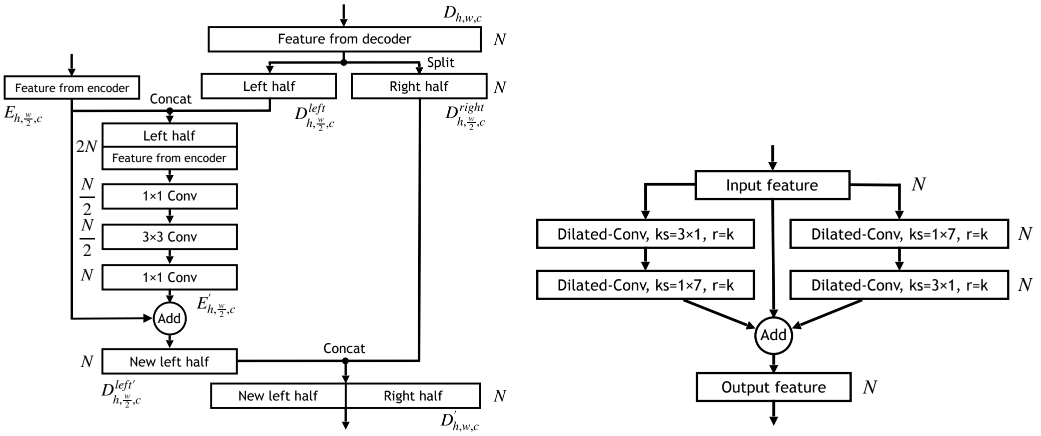
In GRB, a combination of $1\times$n and n$\times1$ convolutional layers replace n$\times$n convolutional layers, the residual connection is introduced to connect the input to output, and *dilated-convolutional* layers [31] is utilized to "support exponential expansion of the receptive field without loss of resolution or coverage". To strengthen the connection between the original and predicted region aligned on the horizontal direction, we set a bigger receptive field on the horizontal dimension in GRB. [2]

3.2. Loss Function

Our loss function consists of two parts: a masked reconstruction loss and an adversarial loss. The reconstruction loss is responsible for capturing the overall structure of the predicted region and logical coherence with regards to the input image, which focuses on low-order information. The adversarial loss [5, 1, 6] makes prediction look more real, which is due to high-order information capturing.

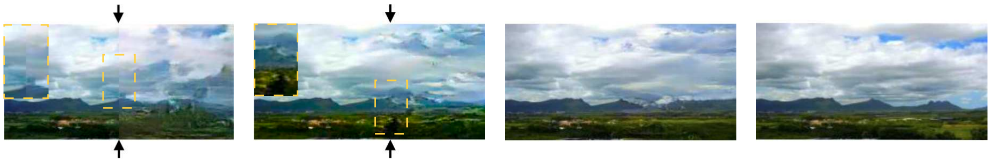
[1] Specially, the SHC before first *transposed-convolutional* layer is different from above. In this layer, we just concatenate the input of RCT to the left of predicted feature map on width dimension, because the predicted feature doesn't include any information from the input region to compute.

[2] We only deploy GRB after first three SHC layers, because we found it fails to achieve good performance when setting GRB too close to the output layer. After GRB, we deploy some ResBlocks to compensate for the performance loss caused by *Inception* architecture and *Dilated convolutions*.



(a) SHC (b) GRB

Figure 5. The details of Skip Horizontal Connection (a) and Global Residual Block (b). N is the channel number, ks is the kernel size, and r is the dilation rate of *dilated-convolutional* layers. In (b), we set a bigger size of receptive field on horizontal dimension (1×7) to strengthen the connection between the input and predicted region.



(a) No SHC (b) One SHC layer (c) Full SHC (d) Groundtruth

Figure 6. (a): When we don't use any SHC layers, there is an obvious boundary between the input region and predicted region, which means an inconsistency during the generation process. (b) When we utilize one SHC layer with GRB in the middle of decoder, the boundary line starts to fade away. (c) After we deploy more SHC layers, there is no obvious boundary.

Masked Reconstruction Loss We use a L2 distance between ground truth image x and predicted image $\tilde{x}$ as our reconstruction loss, denoted as $\mathcal{L}_{rec}(x)$,

$$\mathcal{L}_{rec}(x) = M \odot \parallel x - \tilde{x} \parallel_2^2, \tag{1}$$

where M is a mask used to reduce the weights of L2 along the prediction direction. Masked reconstruction loss is prevalent in generative image inpainting task [17, 10, 32], because less relation is between ground truth and prediction when far away from the border. Different from other mask methods, we use a cos function to decay the weight to zero. In the predicted region, let d be the distance to the border between origin and predicted region and W_p be the width of prediction in 1-step, we have:

$$M(d) = \frac{1 + cos(\frac{d\pi}{W_p})}{2}. \tag{2}$$

The L2 loss can minimize the mean pixel-wise error, which makes the generator to produce a rough outline of the predicted region but results in a blurry averaged image [17]. To alleviate this blurry problem, we add an adversarial loss to capture high-frequency details.

Global and Local Adversarial Loss Following the same strategy utilized in [32], we deploy one global adversarial loss and one local adversarial loss, to make the generated images indistinguishable from the real input image. We choose a modified Wasserstein GANs [6] for our global and local adversarial loss due to its advantages, the only difference between the global and the local adversarial loss is their input.

Specifically, by enforcing a soft version of the constraint with a penalty on the gradient norm for random samples $\tilde{x} \sim \mathbb{P}_{\tilde{x}}$, the final objective in [6] becomes:

$$\begin{aligned}\max_G \min_D \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] \\ + \lambda_{gp} \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(\parallel \nabla_{\tilde{x}} D(\tilde{x}) \parallel_2 - 1)^2].\end{aligned} \tag{3}$$

Hence the adversarial loss for the discriminator, $\mathcal{L}_{dis}$, is

$$\begin{aligned}\mathcal{L}_{dis} = \min_D \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] \\ + \lambda_{gp} \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(\parallel \nabla_{\tilde{x}} D(\tilde{x}) \parallel_2 - 1)^2].\end{aligned} \tag{4}$$

And the adversarial loss for the generator, $\mathcal{L}_{gen}$, is

$$\mathcal{L}_{gen} = \min_G - \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \tag{5}$$

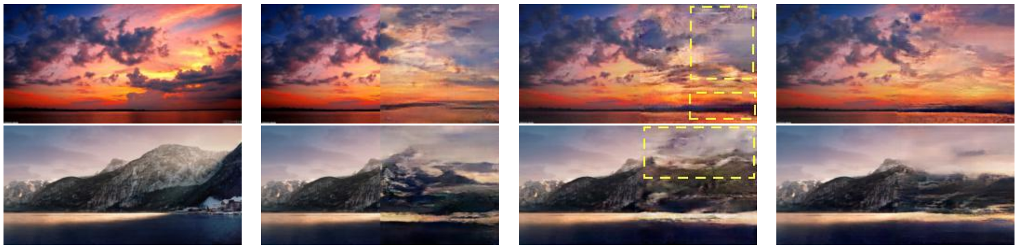

(a) Ground Truth (b) FC (c) FC+SHC (d) RCT+SHC(Ours)

Figure 7. The qualitative results on our collected scenery dataset. The method of (b) uses a fully-connected (FC) layer to connect the encoder and decoder, with which an obvious un-smoothness on the boundary between original and predicted regions. And the method of (c) deploys SHC layers to mitigate the un-smoothness, but there is still a problem that the generated image is easily getting blurred when the prediction region is far away from the input region. We use yellow boxes to highlight the blurred areas in predicted areas. Finally, the method of (d), which replaces the FC layer with RCT, overcomes the problem in (c) and makes the details in the prediction more delicate.

Figure 8. Examples of our scenery dataset. This scenery dataset consists of diverse, complicated natural scenes, including mountain with or without snow, valley, seaside, riverbank, starry sky, etc.

In the global adversarial loss, $\mathcal{L}_{dis}^{global}$ and $\mathcal{L}_{gen}^{global}$, the x and $\tilde{x}$ are the ground truth images and the entire output (including original input on the left, and the predicted region on the right). In the local adversarial loss, $\mathcal{L}_{dis}^{local}$ and $\mathcal{L}_{gen}^{local}$, the x and $\tilde{x}$ are the right half of ground truth images and the right half of entire output (the predicted region).

In a summary, the entire loss for global and local discriminators, $\mathcal{L}_D$, is

$$\mathcal{L}_D = \beta \mathcal{L}_{gen}^{global} + (1-\beta)\mathcal{L}_{gen}^{local}. \tag{6}$$

And the entire loss for the generator, $\mathcal{L}_G$, is

$$\mathcal{L}_G = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_D. \tag{7}$$

In our experiments, we set $\lambda_{gp} = 10$, $\beta = 0.9$, $\lambda_{adv} = 0.002$, and $\lambda_{rec} = 0.998$.

3.3. Implementation Details

In our architecture, we use ReLU as the activation function in the decoder module, Leaky-ReLU as the activation function in other modules. We choose Instance normalization [26] instead of Batch normalization [11] before these activation functions empirically.

4. Experiments

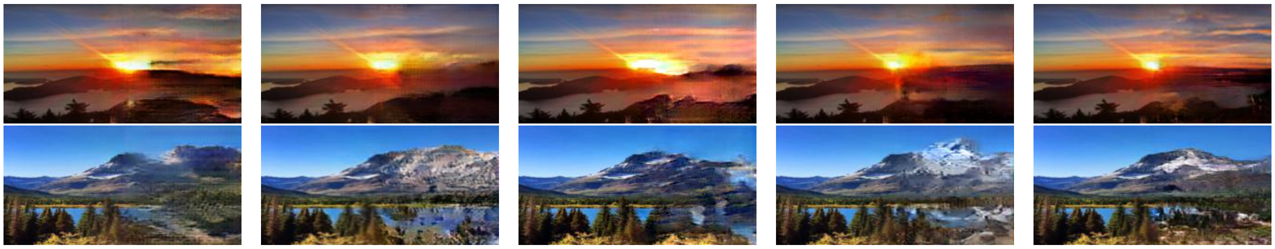
We prepare a new scenery dataset consisting of diverse, complicated natural scenes, including mountain with or without snow, valley, seaside, riverbank, starry sky, etc. There are about $5,000$ images in the training set and $1,000$ images in the testing set. Part of the dataset (about $3,000$) comes from SUN dataset [29], and we collect others on the internet. Fig. 8 shows some examples. We conduct a series of comparative experiments to test our model on 1-step prediction[3]. And we will show the strong representation ability of our architecture on multi-step prediction.

4.1. One-step Prediction

To train our model, we use Adam optimizer [13] to minimize the loss functions defined in Equation 7 and Equation 6. We set base learning rate$= 0.0001$, $\beta_1 = 0.5$ and $\beta_2 = 0.9$. Before the formal training, we set $\lambda_{adv} = 0$, $\lambda_{rec} = 1$ and train generator for 1000 iterations. In the formal training, we set $\lambda_{adv} = 0.002$ and $\lambda_{rec} = 0.998$. Same as the training method in [1], the discriminator updates parameters n_{cir} times but the generator once. When iterations is less than 30 or a multiple of 500, we set $n_{cir} = 30$. In other cases, we set $n_{cir} = 5$. The batch size is 32, and the learning rate is divided by 10 after $1,000$ epochs. The epoch number in our training process is $1,500$.

In training, each image is resized to 144×432, and then

[3] In our experiment, we do natural scenery image outpainting only on horizontal directions because of the limitation of our collected data. But theoretically, our network can work on any directions after modifications.



(a) Pix2Pix [12] (b) GLC [10] (c) CA [32] (d) FC+SHC (e) RCT+SHC (Ours)

Figure 9. Comparisons on 1-step with latest generative methods. Ours RCT+SHC method achieve the best quality.

a 128×256 image is randomly cropped from it or its horizontal flip. In testing, we resize the image to 128×256.

Number of GRB	IS	FID
0	2.756	15.171
1	2.765	14.828
3 (ours)	**2.852**	**13.713**

Table 2. Evaluation of Inception Score (IS) [20] (the higher the better) and Fréchet Inception Distance (FID) [8] (the lower the better) of different number of GRB. 0 means no GRB used in the network. 1 means we keep the GRB where the feature size is 16 × 32 × 256. 3 is the setting utilized by us.

Comparison with Previous Works We make comparisons with latest generative methods[4], including Pix2Pix [12], GLC [10], and Contextual Attention [32], which are originally designed for image inpainting. The comparison result is shown in Fig. 9. We can find that our method achieves the best generation quality due to our designed architecture.

We employ Inception Score [20] and Fréchet Inception Distance [8] to measure the generative quality objectively and report them in Table. 3. Our method achieves the best performance of FID, but its IS is a bit lower than CA [32]. This is because CA employs a contextual attention method, which uses the feature in the original region to reconstruct prediction. But as shown in Fig. 9, 10, the contextual attention makes predictions worse when far away from original inputs. This leads to poor FID score (19.040, while ours 13.713). The contextual attention is an effective method in small region prediction (such as inpainting), but is not suitable in long-range outpainting.

Ablation Study First, we conduct ablation studies to demonstrate the necessity of introduction of SHC and RCT. The qualitative result comparison is shown in Fig. 7, in which we compare our architecture with the models without SHC or RCT. According to the experimental results, SHC successfully mitigates the un-smoothness between the predicted and original region. And RCT effectively improves the representation ability of the model and make the details in the prediction more delicate. Second, we make an ablation study on GRB. As shown in Table.2, the performance improves when using more GRB modules, which demonstrates the effectiveness.

Method	IS	FID
Pix2Pix [12]	2.825	19.734
GLC [10]	2.812	14.825
CA [32]	**2.931**	19.040
FC+SHC	2.845	15.186
RCT+SHC (Ours)	2.852	**13.713**

Table 3. Evaluation of IS [20] and FID [8] scores in 1-step prediction. Images from the validation set have an IS of 3.387. We evaluate FID score between predictions and validation set which has 1, 000 images.

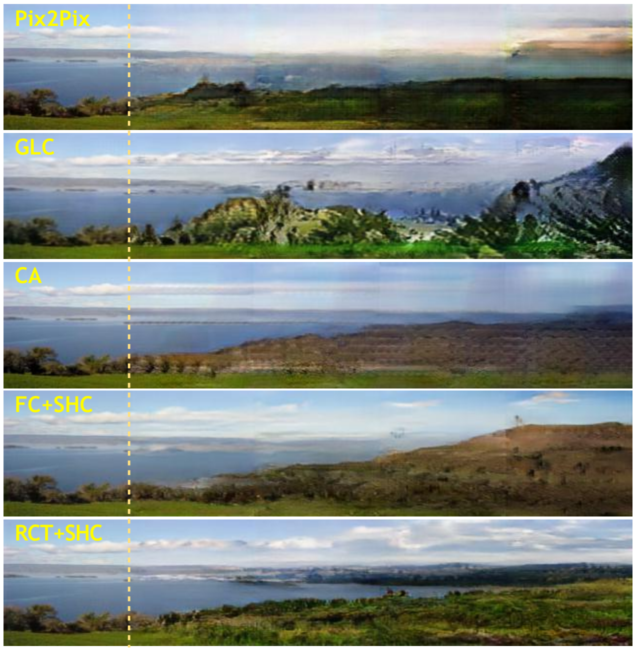


Figure 10. Comparisons on multi-step predictions.

4.2. Multi-Step Prediction

In this section, we use the well-trained model in Section 4.1 for multi-step prediction experiments. To make

[4] We make some modifications on their implementation for image outpainting.

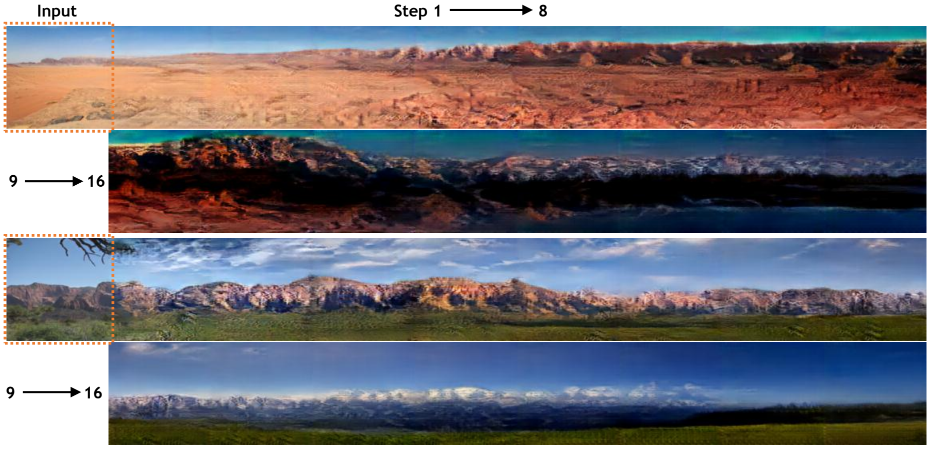


Figure 11. The prediction of very long range. Given an input image (128×128 size), we predict 16 steps to the right direction (128×2176 size). Each example is shown in two lines.

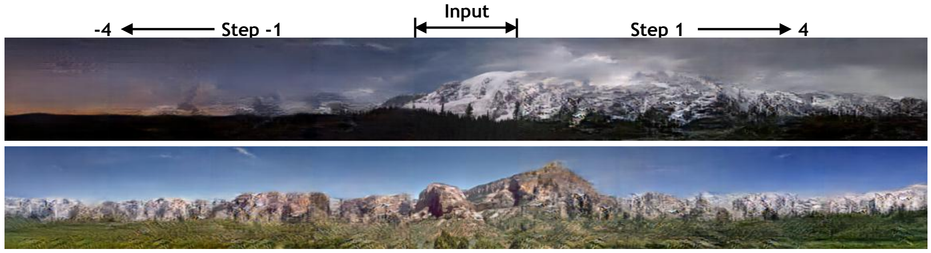


Figure 12. The prediction of an input image on both sides. Given an input image (128×128 size), we predict 4 steps to both the left (step: -1:-4) and right (step: 1:4) directions (128×1152 size). The middle of the example is the input region.

multi-step predictions, we use the predicted output from the previous step as the input for the next step. By concatenating the results from each step, we can get a very long picture.

We experiment with the prediction on one side in a very long range (Fig. 11) and the prediction on both sides (Fig. 12). These two experiments both show the powerful representational capabilities of our architecture. By the benefit of RCT, our model allows for long-term predictions with only a small amount of noise increase.

Besides, we make a comparison between our method and previous works: Pix2Pix [12], GLC [10], and CA [32] on multi-step predictions. The comparison result is shown in 10. Again, the result consistency in Pix2Pix [12], GLC [10], and CA [32] drops dramatically under this circumstance. FC+SHC achieves a better consistency, but still suffers from a large blurry effect. Especially, when far away from original inputs, sharp edges occur in the prediction results. By replacing the FC module with RCT, our method achieves the best performance on both consistency and sharpness.

A Hard Case Example. We test our method on some difficult cases, which are hard for previous works based on image matching. We show one example in Fig. 13. As shown in Fig. 13, when a given input is nearly nonobservable due to its darkness, our method is still able to generate a highly realistic snow mountain.

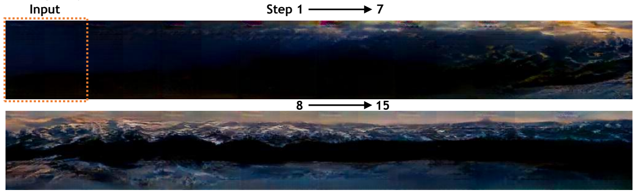


Figure 13. Generation from an input with few observable details, which is a hard case for previous Non-DL methods.

5. Conclusion and Future Work

We design a novel end-to-end network to solve image outpainting problems, which is, to the best of our knowledge, the first approach to utilize a deep neural network for solving this problem. With the introduction of the graceful designed Recurrent Content Transfer, Skip Horizontal Connection, and Global Residual Block, our network can generate images with high quality and extra length. We collect a new natural scenery dataset and conduct a series of experiments on it. Not surprisingly, our proposed method achieves the best performances. More than that, the proposed method can successfully generate extremely long pictures by iterating the model, which is unprecedented.

In future work, we would like to explore how to extrapolate images on horizontal and vertical directions with one same model simultaneously. Besides, we plan to design a specialized training process for the multi-step prediction.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[3] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[4] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Style aggregated network for facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 379–388, June 2018.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 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. In *NIPS*, 2017.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

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

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

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

[14] Johannes Kopf, Wolf Kienzle, Steven Drucker, and Sing Bing Kang. Quality prediction for image completion. *ACM Transactions on Graphics (TOG)*, 31(6):131, 2012.

[15] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[16] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[17] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[18] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[21] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[23] Josef Sivic, Biliana Kaneva, Antonio Torralba, Shai Avidan, and William T Freeman. Creating and exploring a large photorealistic virtual space. 2008.

[24] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[26] D Ulyanov, A Vedaldi, and VS Lempitsky. Instance normalization: The missing ingredient for fast stylization. arxiv 2016. *arXiv preprint arXiv:1607.08022*.

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

[28] Miao Wang, Yu-Kun Lai, Yuan Liang, Ralph R. Martin, and Shi-Min Hu. Biggerpicture: Data-driven image extrapolation using graph matching. *ACM Trans. Graph.*, 33(6):173:1–173:13, Nov. 2014.

[29] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.

[30] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017.

[31] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[32] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018.

[33] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025. IEEE, 2011.

[34] Yinda Zhang, Jianxiong Xiao, James Hays, and Ping Tan. Framebreak: Dramatic image extrapolation by guided shift-maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1171–1178, 2013.

[35] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.