# CISI-net: Explicit Latent Content Inference and Imitated Style Rendering for Image Inpainting

**Jing Xiao,[1,2,4] Liang Liao,[1,2,4]* Qiegen Liu,[3] Ruimin Hu[1,2]**

[1]National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, China

[2]Collaborative Innovation Center of Geospatial Technology, China

[3]School of Electronic Information Engineering, Nanchang University, China

[4]Research Institute of Wuhan University in Jiangsu, Jiangsu, China

Email: {jing, liaoliangwhu, hrm}@whu.edu.cn liuqiegen@ncu.edu.cn

**Abstract**

Convolutional neural networks (CNNs) have presented their potential in filling large missing areas with plausible contents. To address the blurriness issue commonly existing in the CNN-based inpainting, a typical approach is to conduct texture refinement on the initially completed images by replacing the neural patch in the predicted region using the closest one in the known region. However, such a processing might introduce undesired content change in the predicted region, especially when the desired content does not exist in the known region. To avoid generating such incorrect content, in this paper, we propose a content inference and style imitation network (CISI-net), which explicitly separate the image data into content code and style code. The content inference is realized by performing inference in the latent space to infer the content code of the corrupted images similar to the one from the original images. It can produce more detailed content than a similar inference procedure in the pixel domain, due to the dimensional distribution of content being lower than that of the entire image. On the other hand, the style code is used to represent the rendering of content, which will be consistent over the entire image. The style code is then integrated with the inferred content code to generate the complete image. Experiments on multiple datasets including structural and natural images demonstrate that our proposed approach out-performs the existing ones in terms of content accuracy as well as texture details.

## Introduction

Image inpainting refers to the task of filling in missing or masked regions with synthesized contents. Recently, we have witnessed success of learning-based image inpainting through scene understanding (Pathak et al. 2016; Iizuka et al. 2017; Liao et al. 2018). Benefit from large scale training data, they can produce plausible inpainting result by encoding an incomplete image to a latent code and decoding the code to a complete image. However, the latent code usually has some difficulties to represent the high-dimensional distribution of the complex real scene, such as the changes in spatial structures, illumination and seasons, the inpainting results are still quite blurry and contain notable artifacts.

To reduce the difficulty of using one latent code to represent complex scene, two-stage methods have been proposed to do content generation and texture refinement separately (Yang et al. 2017; Song et al. 2018; Demir et al. 2016; Yu et al. 2018; Zhang et al. 2018). In the first stage, the missing regions are filled by a content generation network, targeting at initializing the correct content. In the second stage, the styles are propagated from known region by matching and adapting neural patches with the most similar mid-layer feature in a texture refinement network, aiming to update initially filled region with fine textures. In this way, it not only preserves contextual structures but also produces high-frequency details. However, the matching process is quite time consuming. Moreover, since the neural patch is a mixture of content and style, copying them from known region into missing region in the second stage will introduce change to the originally generated content, still leading to some notable artifacts.

Referring to the way how people restore corrupted picture: the scene of picture should be understood prior to the restoration; then, the main structures of missing region will be delineated, followed by rendering of the main structures according to the painting style of picture. According to the way of human in picture restoration, separation of content and style will help us to focus on the easier-to-solve subproblems of inpainting: 1) how to predict the missing con-

*Liang Liao is the corresponding author.

tent based on the inference of the semantic content from known region; and 2) how the predicted content can be rendered by imitating how the known region is drawn.

In this paper, we propose a Content Inference and Style Imitation Network (CISI-net), in which content and style are explicitly separated by two encoders and integrated by a decoder. During the encoding process, we focus on inference of the semantic content of the missing region, whilst during the decoding process, we focus on imitating the style of the known region to render the predicted content. Compared to the two-stage inpainting methods, we combine the content inference and style refinement in a unified architecture, and only generate the completed image once.

In the encoding process, the inference of content code is conducted under the assumptions: 1) the inferred content code of corrupted image should be same with the one of the uncorrupted image; and 2) the inferred content codes should be same for the same image with different corrupted regions. To realize the assumptions, we adopted a content code loss in latent space, which uses the content code from original image as guidance, and makes the inferred content codes from all corrupted images to be as similar as it.

In the decoding process, we attempt to make the style of filled region to be same with the known region considering that style is a global feature of whole image. Inspired by the Adaptive Instance Normalization (AdaIN) model for style transfer (Huang et al. 2017), we integrate the encoded style code and inferred content code before decoding. Then, the integrated code is decoded to the complete image. In this way, we make the decoded complete image having consistent style with the known region.

Our contributions are summarized as follows:

1) We design a learning based inpainting system which reduces the high-dimensional distribution of image data into two relatively low-dimensional distributions of content and style, which are easier for the network to model.

2) We introduce an inference of content in latent space guided by a content code loss, which is better agree with the way of human inpainting and more interpretable than only guided by the reconstruction loss in the pixel domain.

3) We show that our trained model can achieve performance comparable with state-of-the-art on structural and natural images.

## Related Work

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

### CNN-based Inpainting

CNN based image inpainting methods introduce the semantic prior of image dataset during training and predict the content of missing region by understanding the context. A pioneer approach is context encoder (CE) (Pathak et al. 2016), which is trained to extract latent feature representation from corrupted image and decode it to predict the content of missing region by combining reconstruction loss and adversarial loss. Based on structure of CE, other losses are proposed to improve the quality of inpainted image, e.g. the global adversarial loss to keep the consistency between the synthesized region with known region (Iizuka et al. 2017), transformation-invariant image feature loss to enhance the perceptual similarity of the synthesized region (Dosovitskiy et al. 2016; Larsen et al. 2016) and recognition performance (Zhang et al. 2017). However, these losses still cannot guarantee fine textures in the inpainted region.

For better recovery of detail textures, multi-scale neural patch synthesis is presented to iteratively optimize the textures through matching and adapting predicted patches using textures features in the known region (Yang et al. 2017) To reduce the computational cost for texture optimization, learning-based texture refinement methods are proposed to simplify the inpainting task into two forward inference stages, e.g. learning residuals (Demir et al.2018), or using features from known regions to guide the refinement of missing region (Yu et al. 2018; Yan et al. 2018). Rather than using two-stage process, examplar-based inpainting model is adopted in (Yan et al. 2018) to generate a Shift feature, which is to replace the features from missing region by similar features in the known region. The Shift feature is then concatenated in the decoding layers to enhance the textures of missing region. In this work, we also try to integrate the texture refinement in the decoder. Rather than using the local similar features to refine the texture, we introduce a style code to represent the overall style of the textures

### Style Transfer

Our texture refinement process can be related to recent works in image style transfer, where both the content and the style (texture) of missing part are estimated and transferred from the known region. Style transfer is first formulated as an optimization problem to transfer style and texture of the style image to the content image (Gatys et al. 2015), by minimizing the difference between the Gram matrix of the generated image and that of the style image. As an alternative, (Wand et al. 2016; Elad et al 2017) use neural-patch based similarity matching between content image and style image to suppress distortions. However, the above methods require iterative optimization in the

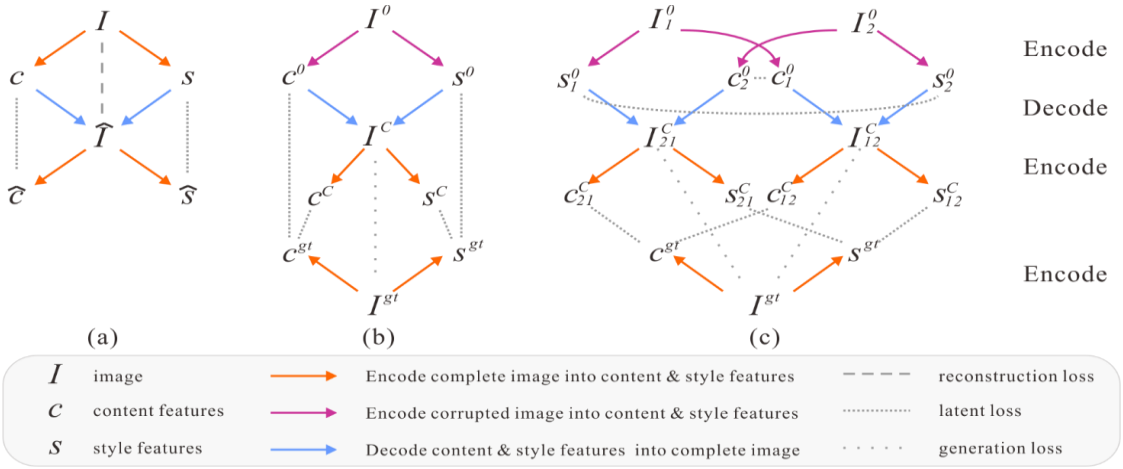



*Figure 1. Consistency constraints for learning encoder and decoder for image inpainting. (a) self-consistency; (b) inferring-consistency; (c) mutual-consistency. The latent code of each encoder is composed of a content code c and a style code s. The orange and red arrows indicate the encoding process for complete images and corrupted images respectively, whilst the blue arrow indicates the decoding process. We train the model with reconstruction objectives (dashed lines) that ensure the correct mapping between latent space and image space, the latent feature objectives (tight dotted lines) that make correct latent content and style feature extraction, as well as the generation objectives (loose dotted lines) that ensure the inpainted images to be similar to ground truth image and indistinguishable from real images*

pixel domain, which is time and computational resource consuming.

Style network models are proposed to realize an end-to-end style transfer. At the beginning, each style is presented by a forward network model (Johnson et al. 2016; Ulyanov et al. 2016), which is hard to generalize. Then, multiple styles are integrated into one model by only recoding the different parameters for new styles (Dumoulin et al. 2017; Zhang et al. 2017). AdaIN network is the first work to represent arbitrary styles in one model (Huang et al. 2017). It uses VGG network to extract style features and content features, and normalizes content into different styles. The AdaIN model is close to our requirement, but content code in our work is predicted from corrupted images and the style code is learned from known region of the image.

### Learning Disentangled Representation

Our work draws inspiration from recent works on disentangled representation learning. The content is firstly disentangled from style for characters with bilinear models (Tenenbaum et al. 1997). More recent work focuses on learning hierarchical feature representations using deep convolutional neural networks to separate content and style (Villegas et al. 2017; Denton et al. 2017; Yang et al. 2019). Although different works use different definitions for content and style for different tasks, we follow the paper (Huang et al. 2018) to define the content as "the underlining spatial structure" and style as "the rendering of the structure". In this work, we attempt to separate the known region into content and style, and use style to render the predicted content for the missing region.

## Proposed Approach

### Problem Description

Suppose we are given a corrupted input image $I^0$. The image inpainting aims to restore the ground truth image $I^{gt}$ by filling the missing region with plausible content $I^R$ to form a completed image $I^C$. While deep generative models can complete this task using reconstruction loss and adversarial loss, the result are still quite blurry and contain notable artifacts. Inspired by recent works on multimodal style transfer (Huang et al. 2018), we assume that each image can be generated by a latent content code $c$ and a latent style code $s$, namely $I = G(c, s)$. Moreover, there exist inverse encoder $E$ of $G$ to separate an image into the latent content code and style code, $(c, s) = \big(E_c(I),\ E_s(I)\big) = E(I)$.

In order to solve the inpainting task, the content code of $I^{gt}$ should be able to be inferred from $I^0$ ($E_c^0(I^0) \approx E_c^{gt}(I^{gt})$), and the style codes from $I^{gt}$ and $I^0$ should be the same since the style is a global feature ($E_s^0(I^0) \approx E_s^{gt}(I^{gt})$). In this way, the completed image can be similar with the ground truth image: $I^C = G(E_c^0(I^0),\ E_s^0(I^0)) \approx G\big(E_c^{gt}(I^{gt}),\ E_s^{gt}(I^{gt})\big) = I^{gt}$. The difference between $E_c^0$ and $E_c^{gt}$ is that $E_c^{gt}$ only needs to extract content feature from an uncorrupted image, but $E_c^0$ needs to be able to infer the overall content from the incomplete image. Since the style represent rendering of the image, it should be consistent over the entire image, thus $E_s^0$ and $E_s^{gt}$ can be the same. Our goal is to learn the underlying generator and encoder functions with neural networks.

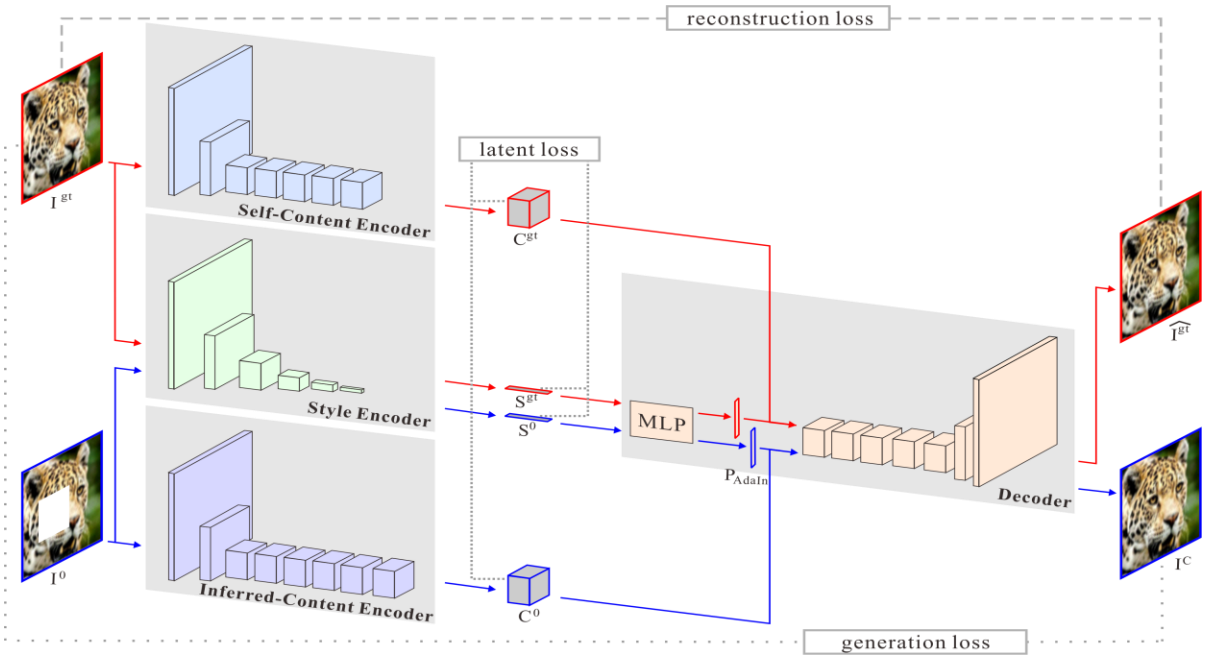



*Figure 2. Architecture of inpainting network*

Figure 2 shows an overview of our model, consisting of three encoders and a decoder. In order to train the model, we define three consistency assumptions:

1) **Self-consistency:** an image can be factorized into a content code $c$ and a style code $s$, and it can also be ideally reconstructed from its factors (Figure 1 (a)).

2) **Inferring-consistency**: the inferred content code $c^0$ and extracted style code $s^0$ from corrupted image should be consistent with $c^{gt}$ and $s^{gt}$ from ground truth image, and the completed image $I^C$ should also be similar with $I^{gt}$ (Figure 1 (b)).

3) **Mutual-consistency**: the content and style of an image is consistent no matter how does it corrupted. Namely, if we have two corrupted images from the same $I^{gt}$, their encoded content and style codes should be the same with each other. Moreover, if we exchange their content codes, the generated images should also be similar to the ground truth image (Figure 1 (c)).

Our loss functions comprise losses in both latent space and image space to represent three consistency constraints.

**Reconstruction loss**. Given a complete image, e.g. a ground truth image, it should be able to be ideally reconstructed. We use $\mathcal{L}_1$ loss to encourage per-pixel reconstruction accuracy and perceptual loss to encourage higher level feature similarity by projecting these images with an ImageNet-pretrained VGG-16 (Dosovitskiy et al. 2016).

$$\mathcal{L}_1(I^C, I^{gt}) = \mathbb{E}_{I^c \sim p(I^c), I^{gt} \sim p(I)} \|I^C - I^{gt}\|_1 \quad (1)$$

$$\mathcal{L}_p(I^C, I^{gt}) = \mathbb{E}_{I^c \sim p(I^c), I^{gt} \sim p(I)} \sum_{n=1}^{N} ||\Psi_n(I^C) - \Psi_n(I^{gt})||_1 \quad (2)$$

$$\mathcal{L}_{recon}(I^C, I^{gt}) = \mathcal{L}_1(I^C, I^{gt}) + \lambda_p \mathcal{L}_p(I^C, I^{gt}) \quad (3)$$

where $\Psi_n$ is the activation map of $n$th selected layer, $N$ is the number of selected layers, $\lambda_p$ is the weight. We use layers *pool*1, *pool*2 and *pool*3 for our loss.

**Latent loss.** Given a latent code (content and style) encoded from a ground truth image, we should be able to recover them after decoding and encoding. Moreover, the latent codes from a corrupted image or its completed image should also be the same with the latent codes from the corresponding ground truth image. It is measured by the $\mathcal{L}_1$ loss for content and style codes separately. We take the loss term for inpainted image for example, and the other loss terms are defined in a similar manner.

$$\mathcal{L}^{c0}_{latent}(c^0, c^{gt}) = \mathbb{E}_{I^0 \sim p(I^0), I^{gt} \sim p(I)} \left\| E^0_c(I^0) - E^{gt}_c(I^{gt}) \right\|_1 \quad (4)$$

$$\mathcal{L}^{s0}_{latent}(s^0, s^{gt}) = \mathbb{E}_{I^0 \sim p(I^0), I^{gt} \sim p(I)} \left\| E^0_s(I^0) - E^{gt}_s(I^{gt}) \right\|_1 \quad (5)$$

where $\mathcal{L}^{c0}_{latent}(c^0, c^{gt})$ and $\mathcal{L}^{s0}_{latent}(s^0, s^{gt})$ are the latent content loss and latent style loss between $I^0$ and $I^{gt}$.

**Generation loss**. In order to generate plausible inpainted image, we expect the inpainted image to be indistinguishable from real image. Besides the above mentioned image reconstruction loss, we also employ GANs (Goodfellow et al. 2014) to match the distribution of inpainted image to the data distribution of ground truth image.

$$\mathcal{L}_{adv}(I^C, I^{gt}) = \mathbb{E}_{I^{gt} \sim p(I)} log D(I^{gt}) + \mathbb{E}_{I^c \sim p(I^c)}[1 - log D(I^c)] \quad (6)$$

where $D$ is a discriminator that tries to distinguish between inpainted images and real images.

Note that the current generation loss treats each pixel of the output image equally, which is not desired. It leads a large portion of the loss will be from the known region and

make the model pay more attention to the generation of this region rather than the hole. On the other hand, due to the known region as input image, the quality of reconstructed content in this region is Inevitably better than that in the holes, which need to be inferred from this available information. Therefore, this is inconsistence between the distributions of reconstructed content in the two regions and distribution of the known region is naturally closer to that of real image, which makes the local patch discriminator cannot distinguish between the output images and real images. To address this issue, we propose a weighted reconstruction loss and multi-scale patch adversarial loss to improve generated quality in the missing region.

Firstly, a weighted $\mathcal{L}_1$ loss and perceptual loss considering the mask region is used and weight of missing region is higher than that of known region. The $\mathcal{L}_1$ loss and perceptual loss are modified as:

$$\mathcal{L}_1(I^C, I^{gt}) = \mathbb{E}_{I^c \sim p(I^c), I^{gt} \sim p(I)} \| M_r \odot (I^C - I^{gt}) \|_1 \quad (7)$$

$$\mathcal{L}_p(I^C, I^{gt}) = \mathbb{E}_{I^c \sim p(I^c), I^{gt} \sim p(I)} \sum_{n=1}^{N} || M_p^n \odot (\Psi_n(I^C) - \Psi_n(I^{gt})) ||_1 \quad (8)$$

where $\odot$ is pixelwise multiplication, $M_r$ and $M_p^n$ are weighted masks in pixel space and feature space respectively.

Then we propose to use a multi-scale PatchGAN to classify global and local patches across the image at multiple scales. The discriminator at each scale is identical and only the input is a differently scaled version of the entire image. Each discriminator is a fully convolutional PatchGAN and outputs a vector of real/fake predictions and each value corresponds to a local image patch. For differentiating the hole patches and the known patches, we propose to compute the PatchGAN loss only on the regions, which overlap with the holes. More formally, our multi-scale patch adversarial loss is defined as:

$$\mathcal{L}_{adv}(I^C, I^{gt}) = \sum_{k=1,2,3} \mathbb{E}_{\left(p_k^{gt}, I_k^{gt}\right)} \left[ logD\left(p_k^{gt}\right) \right] + \mathbb{E}_{\left(p_k^C, I_k^C\right)} \left[ 1 - logD\left(p_k^C\right) \right] \quad (9)$$

where $k$ is the image scale, $p_k^{gt}$ and $p_k^C$ are the patches, which overlap with the holes, on the scaled version images of $I_k^{gt}$ and $I_k^C$.

## Framework

The overall architecture of the inpainting network is shown in Figure 2. It consists of two content encoders, a style encoder and a joint decoder. We also adopt a discriminator for the adversarial loss. Since the content feature encodes the complex spatial structure of the data, we use a high-dimensional spatial map for content code; whilst the style feature has a global and relatively simple effect, we adopt a low-dimensional vector for style code.

**Self-content encoder (SCE).** This encoder is used to extract content feature from complete images. Similar with the completion network (Iizuka et al. 2017), it consists of several strided convolutional layers to downsample the image, followed by several residual blocks (He et al. 2016) to extract content feature. All the convolution layers are normalized by instance normalization (Ulyanov et al. 2017).

**Inferred-content encoder (ICE).** This encoder attempts to extract and infer the intact content features from corrupted images. Different from other image translation tasks such as super-resolution, etc., content inference usually not only rely on local statistics, but also on global context. For increasing the size of receptive field, we adopt dilated convolution in all residual blocks. Dilated convolutions use spaced kernels, making it compute each output value with a wider view of input without increasing the number of parameters and computational burden. At the end of ICE, one extra convolution block without stride is added for content inference.

**Style encoder (SE).** Considering that the style feature is a global effects telling how to render the content, it can be extracted the same from corrupted images ignoring the missing region. So that, we use a uniform style encoder for both complete and corrupted images. The style encoder includes several strided convolutional layers, followed by a global average pooling layer and a fully connected layer. Since we need to preserve the feature mean and variance for style codes, we do not use instance normalization in style encoder.

**Decoder.** The decoder generates an image from its content and style codes. It processes the content code by several residual blocks and using upsampling and convolution layers to reconstruct images. Inspired by the works from style transform that use affine transformation parameters in normalization layers to represent styles (Dumoulin et al. 2017; Wang et al. 2017), we use Adaptive Instance Normalization (AdaIN) (Huang et al. 2017) layers on the residual blocks to modify the style for generate image. The parameters for AdaIN layers are computed by a multilayer perceptron (MLP) from style code.

$$AdaIN(z, \gamma, \beta) = \gamma \left( \frac{z - \mu(z)}{\sigma(z)} \right) + \beta \quad (10)$$

where $z$ is activation produced by previous convolutional layer, $\mu$ and $\sigma$ are channel-wise mean and standard deviation, $\gamma$ and $\beta$ are parameters generated from style code.

**Discriminator**. We use the LSGAN objective proposed by Mao et al. (Mao et al. 2017) and employ multi-scale discriminators (Wang et al. 2018) to guide the generators to produce both realistic details and correct global structure.

## Implementation Details

In the previous subsections, we applied a weighted scheme for reconstruction loss. All the weighted masks are generated based on binary mask (with values of 0 and 1), but contain higher weight for unknown region and lower weight on known region (set to 5 and 1 in experiment). The

mask $M_r$ for $\mathcal{L}_1$ loss is easy to compute since the unknown region is obvious in pixel space. For computing the missing region in the feature space, we define a CNN with convolutional layers and pooling layer similar to VGG-16 but having the elements of convolutional filter set to 1/9. The input of such a CNN is the binary mask in pixel space. Thus, the weighted mask $M_p^n$ is obtained by setting a threshold to the CNN feature map in the corresponding layer. When counting the adversarial loss, the calculation of binary mask region in the feature space is the same way.

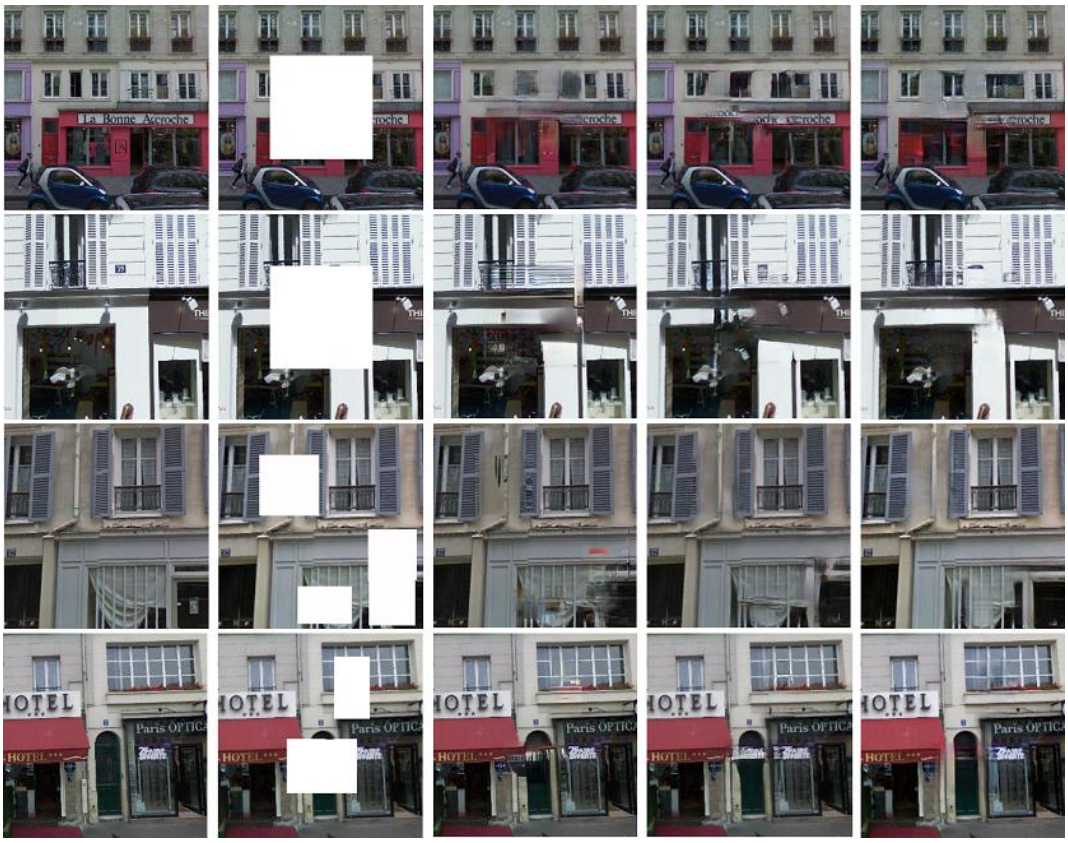
We implement this network using Pytorch toolbox, and optimize this network using the Adam algorithm with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and a learning rate of 0.0001. In all experiments, we use a batch size of 4 and the training is stopped after 500000 iterations. We choose the dimension of the style code to be 8 across all datasets. Random mirroring is applied during training. For balance training of the two encoder, we first train one iteration for the self-content encoder to update the parameters of this encoder with the ground truth image. Then we take the content code of the image to guide the content encoding of corrupted image and update the inferred-content encoder with the generation loss while keep the parameters of the self-content encoder unchanged.

For the model to have "mutual consistency", we train the model with pairs of corrupted images with random missing regions. Then we make the content codes and style codes from two images in a pair to be the same using latent loss.

## Experimental Results

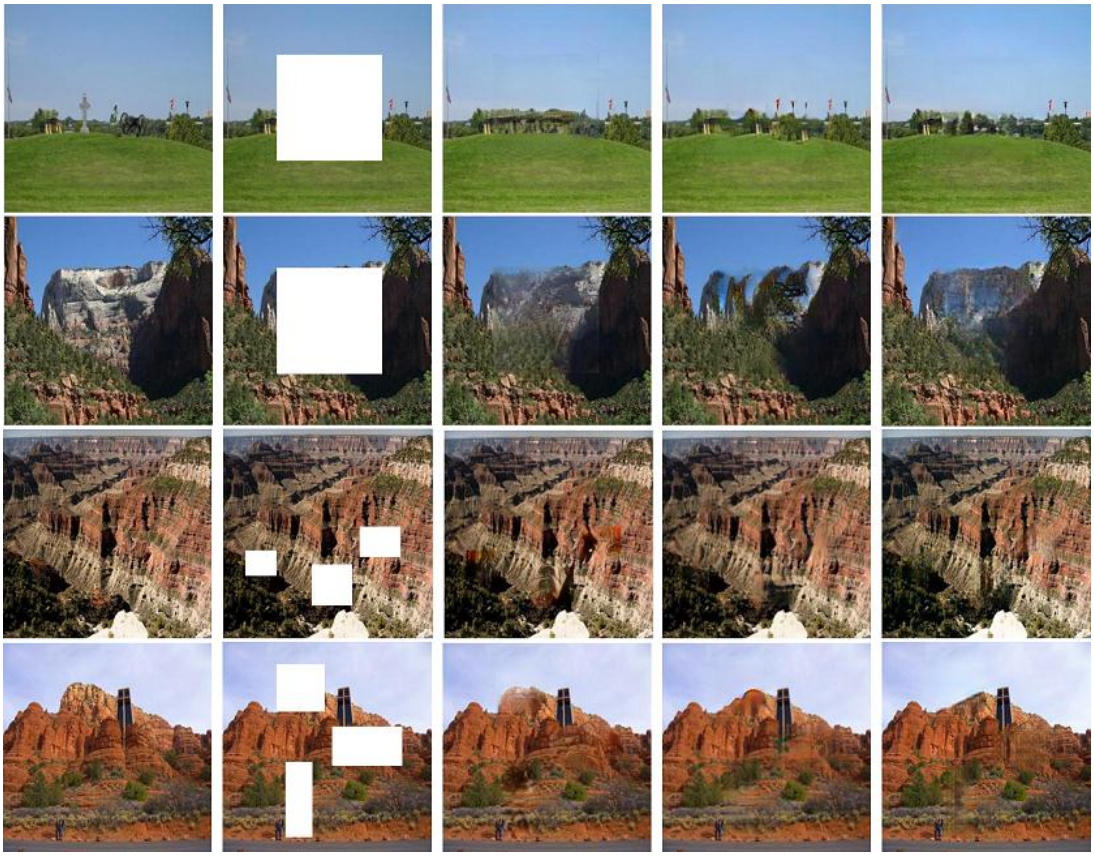
We evaluate our method on two datasets: Paris StreetView (Doersch et al. 2012) with 14,900 training images and 100 test images, and six scenes from Places365-Standard dataset (Zhou et al. 2017). The scene categories selected from Places365-Standard are *butte, canyon, field, synagogue, tundra* and *valley*. Each category has 5000 training images, 900 test images and 100 validation images. For both datasets, we resize each training images to let its minimal height/width be 286, and randomly crop subimage of size 256 × 256 as input to our model.

We compare our results with two learning based methods. GL (Iizuka et al. 2017) adopt a fully convolutional neural network to complete the content and style of image as a whole. GntIpt (Yu et al. 2018) introduces a texture refinement model with contextual attention to leverage the surrounding textures and structures.



*(a) Original image* *(b) masked image* *(c) GL* *(d) GntIpt* *(e) CISI-net*

*Figure 3. Qualitative comparisons of testing results on the Paris StreetView images*



*(a) Original image* *(b) masked image* *(c) GL* *(d) GntIpt* *(e) CISI-net*

*Figure 4. Qualitative comparisons of testing results on the Places2 images*

*Figure 5. Results for object removal using CISI-net.*

## Qualitative Comparisons

Figure 3 and Figure 4 show the visual comparisons of our method, which is denoted as CISI-net, with GL and GntIpt on Paris StreetView and Places2 datasets respectively. The damaged area is simulated by sampling a central hole (128 × 128) or multiple placed missing rectangles randomly. The reported results are direct outputs from the trained models without using any post-processing.

As shown in the figures, GL is effective in understand the context of entire image, but the results tend to be deformed or to mix with the surrounding environment, thus not look realistic or recognizable. GntIpt can generate more realistic results than GL due to the introduction of style from known region. However, some adverse effects, such as incorrect textures in the known areas, are introduced while borrowing the style information from surrounding. In comparison to the competing methods, our CISI-net can generate more semantically plausible and visual-pleasing results with much less artifacts, owing to the separation of content and style. Lower dimensional distribution of content enabled more correct inference, and no other content can be introduced from surroundings.

In Figure 5, we also show some example results for the inpainting of object removal in real world images.

*(a) masked image* *(b) Experiment 1* *(c) Experiment 2* *(d) Experiment 3* *(e) CISI-net*

*Figure 6. Qualitative comparisons of Internal analysis of CISI-net*

### Quantitative Comparisons

In the image inpainting task, many visual-pleasing results can be produced to complete the image, which may be totally different from original image content. For reference, we also compare our model quantitatively with the competing methods on the Paris StreetView in the case of missing center region. Table 1 reports the quantitative results in terms of mean $\mathcal{L}_2$ loss, peak signal-to noise ratio (PSNR) and structural similarity index (SSIM) of the completed region and the execution time of the three models to complete an image of 256 × 256. In general, the proposed CISI-net gets better $\mathcal{L}_2$ loss, PSNR and SSIM with the competing methods, and it can highly reduce the inpainting time due to only one-time forward inference.

Table 1. Quantitative comparison on Paris StreetView dataset

| Method | $\mathcal{L}_2$ loss | PSNR (dB) | SSIM | Time (ms) |
|---|---|---|---|---|
| GL | 7.11% | 19.53 | 0.49 | 168 |
| GntIpt | 6.68% | 19.85 | 0.53 | 285 |
| CISI-net | **6.53%** | **20.05** | **0.55** | **161** |

### Internal Analysis of CISI-net

The main contributions of our CISI-net are the separation of image into content and style, the introduction of latent inference of content, and weighted loss for missing region. To analyze the effectiveness of those operations, experiments are conducted. We use the fully convolutional neural network in CISI-net as the base network.

*Experiment 1*: removing separation (using a fully-convolutional neural network to treat the content and style as a whole), removing latent guidance for content inference and weighted loss for missing region.

*Experiment 2*: keeping image separation, removing latent guidance for content inference and weighted loss for missing region

*Experiment 3*: keeping image separation and guidance for content inference, removing weighted loss for missing region.

Figure 6 shows inpainting results of the three experiments and CISI-net. Comparing (b) and (c), we can notice that reduction from complex high dimensional distribution of image data to relatively low dimensional distribution of content is effective for generating detailed content. (d) shows that the latent loss of content inference leads to more correct structure. The added weighted loss results in finer structure and texture (e) by given more content in the missing region.

## Conclusion

This paper has proposed a novel architecture, i.e. CISI-net, for image completion with only once generation but promising content and details. The explicit separation of content and style has shown its effectiveness on representing the image, where the mapping function for each has smaller dimensionality. We also show that the guided inference in latent space for content can efficiently generating correct structures. Experiments show that our CISI-net can generate fine-detailed and perceptually realistic images. Future studies will be given to further exploring the relationship between content and style and their representations to further improve inpainting performance. We will also further look into the no reference quality assessment (Fang et al. 2018) for inpainting and the generalization to applications, such as context-based coding (Xiao et al. 2016) and disocclusion-based analysis (Xiao et al. 2017).

## Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant 61502348, 61671336, 91738302, by the Natural Science Foundation of Jiangsu Province under Grant BK20180234, by the Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University under Grant 17E03, by the National Key R&D Program of China under Grant 2018YFB1201602.

## References

Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context Encoders: Feature Learning by Inpainting. In *CVPR*, 2536-2544. Las Vegas, Nevada.

Iizuka, S.; Simo-Serra, E.; and Ishikawa, H. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics* 36(4): 1-14.

Liao, L.; Hu, R.; Xiao, J.; and Wang, Z. 2018. Edge-Aware Context Encoder for Image Inpainting. In *ICASSP*, 3156-3160. Calgary, Canada.

Yeh, R. A.; Chen, C.; Schwing, A. G.; Johnson, M. H.; and Do, M. N. 2017. Semantic Image Inpainting with Deep Generative Models. In *CVPR*, 5485-5493. Honolulu, Hawaii.

Yang, C.; Lu, X.; Lin, Z.; Shechtman, E.; Wang, O.; and Li, H. 2017. High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis. In *CVPR*, 4076-4084. Honolulu, Hawaii.

Song, Y.; Yang, C.; Lin, Z.; Liu, X.; Huang, Q.; Li, H.; and Kuo, C. J. 2018. Contextual-based Image Inpainting: Infer, Match, and Translate. In *ECCV*, 3-19. Munich, Germany.

Demir, U., and Unal, G. 2018. Deep Stacked Networks with Residual Polishing for Image Inpainting. *arXiv: 1801.00289*.

Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang T. S. 2018. Generative Image Inpainting With Contextual Attention. In *CVPR*, 5505-5514. Salt Lake City, Utah.

Zhang, H.; Hu, Z.; and Luo, C. 2018. Semantic Image Inpainting with Progressive Generative Networks. In *ACM MM*, 1939-1947. Seoul, Republic of Korea.

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal Unsupervised Image-to-Image Translation. In *ECCV*, 172-189. Munich, Germany.

Huang, X., and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 1501-1510. Venice, Italy.

Dosovitskiy, A., and Brox, T. 2016. Generating images with perceptual similarity metrics based on deep networks. In *NeurIPS*, 1-9. Barcelona, Spain.

Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; and Winther, O. 2016. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 1558-1566. New York City, New York.

Zhang, S.; He, R.; and Tan, T. 2017. DeMeshNet: Blind Face Inpainting for Deep MeshFace Verification. *IEEE Transactions on Information Forensics & Security* 13(3):637-647.

Yan, Z.; Li, X.; Li, M.; Zuo, W.; and Shan, S. 2018. Shift-Net: Image Inpainting via Deep Feature Rearrangement. In *ECCV*, 1-17. Munich, Germany.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style. *arXiv:1508.06576*.

Li, C., and Wand, M. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *CVPR*, 2479–2486. Las Vegas, Nevada.

Elad, M., and Milanfar, P. 2017. Style transfer via texture synthesis. *IEEE Transactions on Image Processing* 26(5) 2338–2351.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 694–711. Amsterdam, the Netherlands.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, 1349–1357. New York City, New York.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2017. A learned representation for artistic style. In *ICLR*.

Zhang, H., and Dana, K. 2017. Multi-style generative network for real-time transfer. *arXiv:*1703.06953.

Tenenbaum, J. B., and Freeman, W. T. 1996. Separating Style and Content. In *NeurIPS*, 662-668. Denver, CO.

Villegas, R.; Yang, J.; Hong, S.; Lin, X.; and Lee, H. 2017. Decomposing motion and content for natural video sequence prediction. In *ICLR*.

Denton, E., and Birodkar, V. 2017. Unsupervised learning of disentangled representations from video. In *NeurIPS*, 4417-4426. Long Beach, CA.

Yang, S.; Liu, J.; and Yang, W. 2019. Context-Aware Text-Based Binary Image Stylization and Synthesis. *IEEE Transactions on Image Processing* 28(2): 952-964.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*, 2672–2680. Montreal, Quebec, Canada.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770-778. Las Vegas, Nevada.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2017. Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis. In *CVPR*, 6924-6932. Honolulu, Hawaii.

Wang, H.; Liang, X.; Zhang, H.; Yeung, D.-Y.; and Xing, E. P. 2017. Zm-net: Real-time zeroshot image manipulation network. *arXiv:1703.07255*.

Mao, X.; Li, Q.; Ren, H.; Lau, R.; Wang, Z.; and Smolley, S. P. 2017. Least Squares Generative Adversarial Networks. In *ICCV*, 2794-2802. Venice, Italy.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In *CVPR*, 8798-8807. Salt Lake City, Utah.

Doersch, C.; Singh, S.; Gupta, A.; Sivic, J.; and Efros, A. A. 2012. What makes paris look like paris? *ACM Transactions on Graphics* 31(4): 101:1-101:9.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(6): 1452-1464.

Xiao, J.; Hu, R.; Liao, L.; Chen, Y.; Wang, Z.; and Xiong, Z. 2016. Knowledge-Based Coding of Objects for Multisource Surveillance Video Data. *IEEE Transactions on Multimedia* 18(9): 1691-1706.

Xiao, J.; Wang, Z.; Chen, Y.; Liao, L.; Xiao, J.; Zhan, G.; and Hu, R. 2017. A sensitive object-oriented approach to big surveillance data compression for social security applications in smart cities. *Software: Practice and Experience* 47(8): 1061-1080.

Fang, Y.; Yan, J.; Li, L.; Wu, J.; and Lin, W. 2018. No Reference Quality Assessment for Screen Content Images With Both Local and Global Feature Representation. *IEEE Transactions on Image Processing* 27(4): 1600-1610.