Context-Aware Semantic Inpainting

Haofeng Li, Guanbin Li, Liang Lin, *Senior Member, IEEE*, Hongchuan Yu, and Yizhou Yu

***Abstract*—In recent times, image inpainting has witnessed rapid progress due to the generative adversarial networks (GANs) that are able to synthesize realistic contents. However, most existing GAN-based methods for semantic inpainting apply an auto-encoder architecture with a fully connected layer, which cannot accurately maintain spatial information. In addition, the discriminator in existing GANs struggles to comprehend high-level semantics within the image context and yields semantically consistent content. Existing evaluation criteria are biased toward blurry results and cannot well characterize edge preservation and visual authenticity in the inpainting results. In this paper, we propose an improved GAN to overcome the aforementioned limitations. Our proposed GAN-based framework consists of a fully convolutional design for the generator which helps to better preserve spatial structures and a joint loss function with a revised perceptual loss to capture high-level semantics in the context. Furthermore, we also introduce two novel measures to better assess the quality of image inpainting results. The experimental results demonstrate that our method outperforms the state-of-the-art under a wide range of criteria.**

***Index Terms*—Convolutional neural network, generative adversarial network (GAN), image inpainting.**

I. Introduction

IMAGE inpainting aims at synthesizing the missing or damaged parts of an image. It is a fundamental problem in low-level vision and has attracted widespread interest in the computer vision and graphics communities as it can be used for filling occluded image regions or repairing damaged photographs. Due to the inherent ambiguity of this problem and the complexity of natural images, synthesizing content with reasonable details for arbitrary natural images still remains a challenging task.

Manuscript received April 27, 2018; accepted August 2, 2018. This work was supported in part by the Hong Kong Research Grants Council through General Research Funds under Grant HKU17209714, in part by the National Natural Science Foundation of China under Grant 61702565, in part by the State Key Development Program under Grant 2018YFC0830103, in part by the EU H2020 project-AniAge under Grant 691215, and in part by the CCF-Tencent Open Research Fund. This paper was recommended by Associate Editor L. Shao. *(Haofeng Li and Guanbin Li contributed equally to this work.) (Corresponding author: Guanbin Li.)*

H. Li is with the Department of Computer Science, University of Hong Kong, Hong Kong (e-mail: lhaof@foxmail.com).

G. Li and L. Lin are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: liguanbin@mail.sysu.edu.cn; linliang@ieee.org).

H. Yu is with the National Centre for Computer Animation, Bournemouth University, Poole BH12 5BB, U.K. (e-mail: hyu@bournemouth.ac.uk).

Y. Yu is with the Department of Computer Science, University of Hong Kong, Hong Kong, and also with AI Lab, Deepwise, Beijing 100080, China (e-mail: yizhouy@acm.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2865036

A high-quality inpainted result should be not only realistic but also semantically consistent with the image context surrounding the missing or damaged region at different scales. First, colorization should be reasonable and spatially coherent. Second, structural features such as salient contours and edges should be connected inside the missing region or across its boundary. Third, texture generated within the missing region should be consistent with the image context and contain high-frequency details. In addition, missing object parts need to be recovered correctly, which is challenging and requires capturing high-level semantics.

Deep convolutional neural networks are capable of learning powerful image representations and have been applied to inpainting [3], [4] with varying degrees of success. Recently, semantic image inpainting has been formulated as an image generation problem and solved within the framework of generative adversarial networks (GANs) [5]. GAN trains a generator against a discriminator and successfully generates plausible visual content with sharp details. The state-of-the-art results [2], [6], [7] have been achieved.

However, all existing GAN-based solutions to inpainting share common limitations. First of all, they utilize an encoder–decoder architecture with fully connected (fc) layers as the *bottleneck* structure in the middle of the network. The bottleneck structure contains two fc layers. The first fc layer converts the convolutional features with spatial dimensions to a single feature vector and another fc layer maps the feature vector backward to features with spatial dimensions. The first fc layer collapses the spatial structure of the input image so that the location-related information cannot be accurately recovered during the decoding process. Second, the discriminator only takes a synthesized region without its image context as an input. Thus, neither structural continuity nor texture consistency can be guaranteed between the synthesized region and its image context. Moveover, existing GANs struggle to understand high-level semantics within the image context and yield semantically consistent content.

To overcome the aforementioned limitations, we conceive a novel fully convolutional generative network for semantic inpainting. First, we adopt a fully convolutional design without the bottleneck structure to preserve more spatial information. Second, we composite the synthesized region and its image context together as a whole, and measure the similarity between this composite image and the ground truth. To increase such a similarity, perceptual loss is computed for the composite image. This perceptual loss defined in terms of high-level deep features is promising in capturing the semantics of the image context.

Furthermore, noticing that the $L2$ loss and PSNR are unable to rate blurry results accurately and quantitative measures

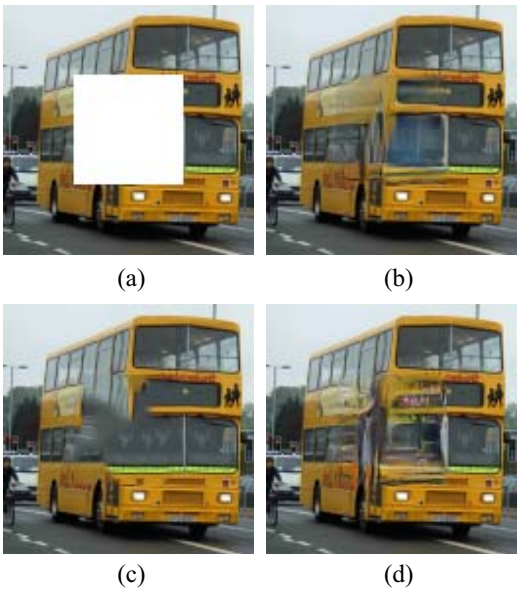


Fig. 1. Our proposed CASI with perceptual loss synthesizes content with a more reasonable colorization and structure than Content-Aware Fill [1] and Context Encoder [2]. (a) Input. (b) CASI. (c) Content-Aware Fill. (d) Context Encoder.

do not exist for assessing how well the intended semantics have been restored, we define a local entropy error and a semantic error (SME) to resolve these two issues, respectively. The SME is defined as the hinge loss for the confidence that a composite image with a synthesized region should be assigned the groundtruth label of its real counterpart, where the confidence value is estimated by a pretrained image classifier. In our experiments, images synthesized by our inpainting model can successfully reduce the SME estimated by a powerful image classifier. This indicates that our model is capable of inferring semantically valid content from the image context.

In summary, this paper has the following contributions.

1) We present a fully convolution GAN to restore images. The proposed network discards fully connected layers for better maintaining the original spatial information in the input image, as shown in Fig. 1. This network can process images with variable size.
2) We introduce a novel context-aware loss function, including a perceptual loss term, which measures the similarity between a composite image and its corresponding groundtruth real image on the semantic feature domain.
3) We propose two novel measures: 1) a local entropy error based on middle-level statistics and 2) an SME based on high-level features, for evaluating inpainting results.

II. Related Work

Recently, deep neural networks including GANs have exhibited great performance in image generation, image transformation, and image completion. This section discusses the previous work relevant to image inpainting and our proposed method.

A. Image Inpainting

Many algorithms on recovering holes in images or videos have been proposed [8]–[14]. Some existing methods for image completion are related to texture synthesis [15], [16] or patch-based synthesis [17]–[19]. Efros and Leung [15] proposed a method for predicting pixels from the context boundary while [16] searches for matching patches and quilts them properly. Drori *et al.* [20] computed a confidence map to guide filling while Komodakis and Tziritas [21] proposed a priority belief propagation method. However, these exemplar-based approaches struggle to generate globally consistent structures despite producing seamless high-frequency textures. Hays and Efros [22] filled large missing regions using millions of photographs and presented seamless results. However, in this method, missing regions need to be prepared carefully by completely removing partially occluded objects. Synthesizing content for arbitrary missing regions remains a challenging task (e.g., recovering body parts for a partially occluded object).

B. Generative Adversarial Networks

GANs, which estimate generative models by simultaneously training two adversarial models, were first introduced by Goodfellow *et al.* [5] for image generation. Radford *et al.* [23] further developed a more stable set of architectures for training GANs, called deep convolutional GANs (DCGAN). Recently, GAN has been widely applied to image generation [24], image transformation [25], image completion [2], and texture synthesis [26]. Context Encoder [2] uses a novel channel-wise fc layer for feature learning but keeps the traditional fc layer for semantic inpainting. Yeh *et al.* [6] employed GAN with both a perceptual loss and a contextual loss to solve inpainting. Notice that the perceptual loss in [6] is essentially an adversarial loss and the contextual loss which only considers the context (excluding the synthesized region). Yang *et al.* [7] conducted online optimization upon a pretrained inpainting model primarily inherited from Context Encoder. The optimization is too expensive for real-time or interactive applications. Common disadvantages exist in these GAN-based approaches. First, the fc layer in the encoder–decoder framework cannot preserve accurate spatial information. Second, the discriminator in current GANs only evaluates the synthesized region but not the semantic and appearance consistency between the predicted region and the image context.

C. Fully Convolutional Networks

Fully convolutional networks (FCNs), which were first proposed in [27] for semantic image segmentation, provide an end-to-end learnable neural network solution for pixel-level image comprehension. Without fc layers, FCNs occupy less computational memory and can perform training and inference more efficiently. Besides, FCNs preserve spatial information and extract location-sensitive features. Recently, FCNs have achieved excellent results on semantic segmentation [27], [28]; saliency detection [29]–[32]; automatic image colorization [33]; as well other pixel-wise inferring-based image restoration tasks [34]–[36]. In this paper, we exploit

the idea of FCN in GAN-based inpainting to better capture object contours, preserve spatial information in features, and infer coherent visual content from context.

D. Context-Aware Perceptual Loss

Perceptual loss is a feature reconstruction loss defined by the deep neural networks [37]. It guides neural models to generate images visually similar to their corresponding targets (e.g., ground truth) and has been widely utilized in style transfer [38]. Dosovitskiy and Brox [24] presented a similar concept, called DeePSiM, which successfully generates images with sharp details. So far, perceptual loss has been applied to style transfer [37], [38]; super resolution [37]; and texture synthesis [39]. However, these topics primarily use the "texture network," a part of the VGG network [40] to extract the middle-level features while high-level features from the fc layers have not been investigated for image completion. In this paper, we exploit high-level deep features in the definition of perceptual loss to synthesize regions semantically consistent with their contexts.

III. METHOD

As shown in Fig. 2(b), our proposed context-aware semantic inpainting method (CASI) is composed of an inpainting generation pipeline (on the left) and a joint loss function (on the right). The fully convolutional generative network takes an image context as the input, where the missing region is filled with the mean pixel value. The missing region is generated by point-wise multiplication (denoted as "mask operation") with a mask. The inverse operation turns one into zero, and zero into one. The output of the generative network is a synthesized image with the same size as the input. Then this output image is cropped using the boundary of the missing region and placed within the image context to form a composite image (denoted as "prediction-context"), via a point-wise addition (denoted as "compose operation"). The discriminator network receives the synthesized content within the missing region and the ground truth within the same region, respectively, and attempts to classify the received content as either "real" or "fake." The classification error is formulated as the adversarial loss, one of the components in the proposed loss. Our joint loss function is a linear combination of a pixel-wise *L*2 loss, the adversarial loss, and a perceptual loss.

A. Fully Convolutional Generative Network

The fully convolutional generative network consists of three blocks: 1) down-sampling; 2) flatting; and 3) up-sampling. First, the down-sampling block plays the role of an encoder, which reduces the spatial dimension to 1/8 of the input size. The flatting block discovers and maintains essential edges without further changing the spatial size. Finally, the up-sampling block plays the role of a decoder, which transforms the feature map to an RGB image with the same resolution as the input.

The down-sampling block consists of three convolutional layers of 4×4 kernels and two convolutional layers using 3×3 kernels. The first layer of this block performs 4×4 convolution. Then these two types of convolutional layers alternate and the block ends with a 4×4 convolutional layer. The 4×4 convolutions use a stride of 2 and 1 pixel padding to reduce the spatial size by half while doubling the number of channels in the feature map. The reduced spatial dimensions allow convolution kernels to have larger receptive fields in the input image. The 3×3 convolutions use a stride of 1 and 1 pixel padding to keep the same spatial size and channel number. Such layers enhance the recognition capacity of the network. The flatting block is composed of three convolutional layers with kernel size 3×3 and two residual blocks. These residual blocks enhance the prediction accuracy for semantic inpainting. The middle layer doubles the number of channels while the last layer reduces it by half. Thus, the flatting block keeps the number of channels the same in the input and output feature maps. The up-sampling block has three de-convolutional layers using 4×4 kernels and three convolutional layers using 3×3 kernels. Similar to the down-sampling block, the two types of layers alternate, and the first layer performs 4×4 deconvolution. In the up-sampling block, 4×4 deconvolution acts as parameterized interpolation which doubles the spatial size while each 3×3 convolutional layer reduces the number of channels by half. The last layer of the up-sampling block generates an RGB image with the same size as the input.

Our proposed generative network does not have a bottleneck fc layer, and enjoys the benefits of fully convolutional architecture. It is capable of locating essential boundaries, maintaining fine details and yield consistent structures in missing regions.

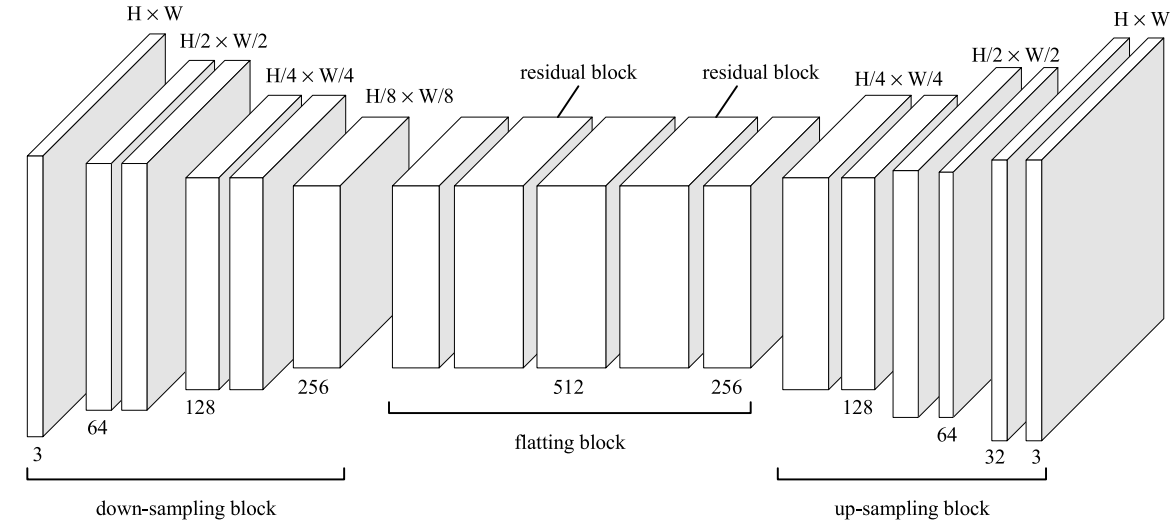
B. Discriminative Network

Our discriminator shares a similar but shallower structure with the down-sampling block in the generator network. Compared with the down-sampling block, the discriminator removes all 3×3 convolutional layers to avoid overfitting. Otherwise, the capacity of the discriminator would be so large that the generator does not have a chance to confuse the discriminator and improve itself. A fc layer is employed to perform binary classification at the end of the discriminator.

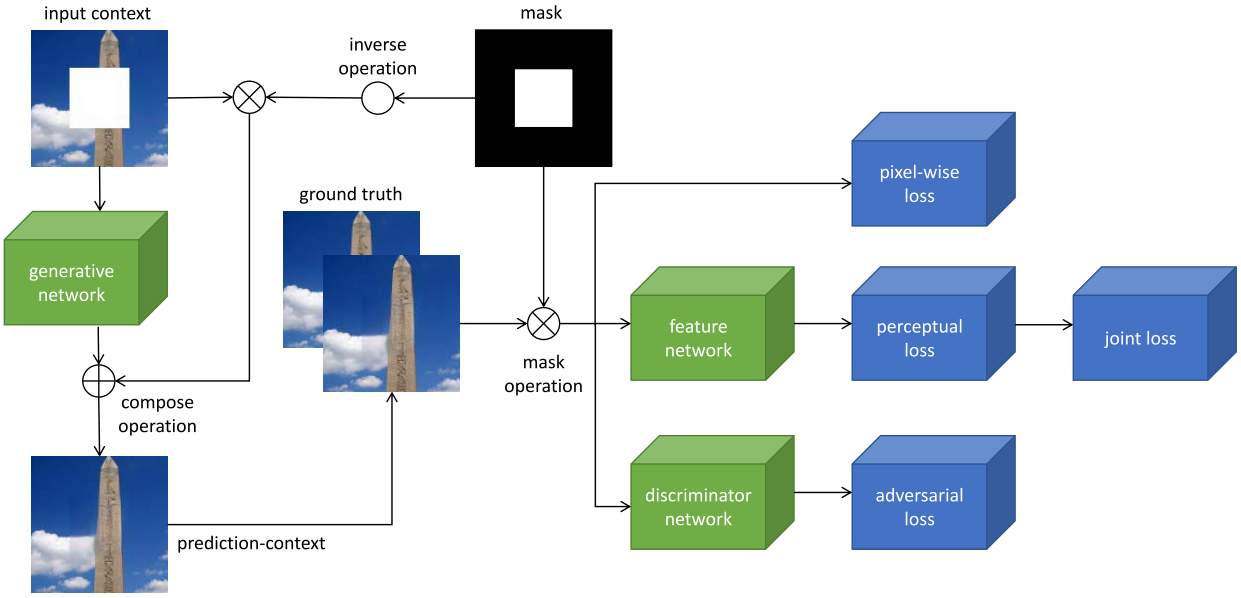
The normalization and nonlinear activations are used in CASI. Except for the last layer, every convolutional layer in the generator and the discriminator is followed with a batch normalization (batchnorm) operation. Rectified linear units (ReLUs) are added following each batchnorm layer in the generator while leaky-ReLUs are used in the discriminator according to the architecture guidelines in DCGAN. A sigmoid layer is adopted in the last layer of the generator and the discriminator to map original pixel to its corresponding confidence value.

C. Loss Function

Given the analysis in Section I, existing GAN-based semantic inpainting methods fail to capture high-level semantics and synthesize semantically consistent content for the missing region. In this paper, we propose to composite the synthesized region and its image context together as a whole, and measures the visual similarity between this composite image and



(a)



(b)

Fig. 2. Network architecture. (a) Fully convolutional generative network. (b) Context-aware semantic inpainting pipeline.

the ground truth using a perceptual loss. Our overall loss function consists of a pixel-wise $L2$ loss, an adversarial loss, and a perceptual loss. It can be formulated as follows:

$$L_{\text{inp}} = \lambda_{\text{pix}} l_{\text{pix}} + \lambda_{\text{adv}} l_{\text{adv}} + \lambda_{\text{per}} l_{\text{per}} \tag{1}$$

where L_{inp} denotes the overall inpainting loss. l_{per}, l_{adv}, and l_{pix} represent our perceptual loss, adversarial loss, and pixel-wise $L2$ loss, respectively, while λ_{per}, λ_{adv}, and λ_{pix} are the weights of the respective loss terms.

Pixel-wise $L2$ loss, l_{pix}, is a straightforward and widely used loss in image generation. It measures the pixel-wise differences between the synthesized region and its corresponding ground truth. l_{pix} is defined as

$$l_{\text{pix}}(x, z) = \|M \odot (x - z)\|_2^2 \tag{2}$$

where M is a binary mask, where a value of 1 indicates the missing region and a value of 0 indicates the known context region, $\odot$ is the element-wise product, x is the ground-truth image, and z is the corresponding inpainting result computed as

$$z = ((1 - M) \odot x) \oplus (M \odot G((1 - M) \odot x)) \tag{3}$$

where $\oplus$ is the element-wise addition, G is the CASI generator, $(1 - M) \odot x$ is the context region of x, and $M \odot G(\cdot)$ is the missing region in the generator's output. $\oplus$ in (3) merges the known context region and the synthesized missing region to obtain the final inpainting result.

However, calculating loss within the image space cannot guarantee to generate an image perceptually similar to the

ground truth as neural networks tend to predict pixel values close to the mean of the training data. In practice, the pixel-wise $L2$ loss only produces blurred images without clear edges or detailed textures. Thus, we exploit an adversarial loss and a novel perceptual loss to overcome this problem.

The adversarial loss l_{adv} is defined on the objective function of the discriminator. As the discriminator aims at distinguishing synthesized content from its corresponding ground truth, its objective is to minimize a binary categorical entropy e in

$$e(D(M \odot x), D(M \odot z)) = \log(D(M \odot x)) + \log(1 - D(M \odot z)) \quad (4)$$

where e denotes the binary categorical entropy and D is the CASI discriminator. The discriminator D predicts the probability that the input image is a real image rather than a synthesized one. If the binary categorical entropy is smaller, the accuracy of the discriminator is better. Note that D is not a pretrained or constant model during the training stage. Instead, G and D are trained alternatively. As minimizing the binary categorical entropy e is equivalent to maximizing the negative of the binary categorical entropy, the final objective value of the discriminator is described in the right hand side of (5). As the generator acts as an adversarial model of the discriminator, it tends to minimize the negative of the binary categorical entropy. Thus, the adversarial loss of the generator l_{adv} can be formally described as

$$l_{\text{adv}} = \max_{D} \big[\log(D(M \odot x)) + \log(1 - D(M \odot z))\big]. \quad (5)$$

l_{adv} makes the synthesized region deviate from the overly smooth result obtained using the pixel-wise $L2$ loss as real images are not very smooth, which typically have fine details. Although, the adversarial loss promotes fine details in the synthesized result, it is still far from perfect. First, existing discriminators are unaware of the image context and do not explicitly consider the composite image consisting of both the synthesized region and the image context. Second, binary classification is not challenging enough for the discriminator to learn the appearance of different objects and parts. Note that semantic inpainting needs to not only synthesize textures consistent with the context but also recover missing object parts, which requires high-level features extracted from the image context. Thus, we propose a perceptual loss based on high-level semantic features.

Our perceptual loss, l_{per}, is defined as

$$l_{\text{per}}(x, z) = e(F(x), F(z)) = \frac{1}{C_j H_j W_j} \big\| F_j(x) - F_j(z) \big\|_2^2 \quad (6)$$

where F is a pretrained feature network that extracts a generic global feature from the input, F_j denotes the activations of the jth layer of F, and $F_j(x)$ and $F_j(z)$ are a $C_j \times H_j \times W_j$ tensor, respectively. In our experiments, we use ResNet-18 pretrained over the ImageNet dataset [41] as the feature network F, and the 512-D feature from the penultimate layer of ResNet-18 as F_j. Similar high-level features extracted by F give rise to similar generated images, as suggested in [24]. In addition, a perceptual loss based on high-level features makes up for the missing global information typically represented in a fc layer in the generator. Different from DeepSiM, our feature is extracted from the composite image consisting of the synthesized region and the image context rather than from the synthesized region alone.

D. Post-Refinement for High-Resolution and Irregular-Shape Case

This section introduces a post-refinement adapted from [7] to extend the proposed neural network model to handle with high-resolution, in-the-wild, or irregular-shape cases. Before the refinement, the proposed model produces a coarse inpainting result as a reference. Given an image with a corrupted region that could be located at any position of the image and may have irregular shape, a *context box* is cropped and reshaped to match the input size of the proposed model. The context box is the double size as the bounding box of the corrupted region and centered on the bounding box. As the context box may exceed the range of the image, the vacant region (including the corrupted region) is filled with the mean pixel value which is computed among all the training samples.

Afterward, the reshaped context box is fed into the proposed CASI model to yield a coarse inpainting result. Then, we formulate refining the initial coarse result as an optimization problem. The optimization objective is the right hand side of (7). I_0 is the initial coarse reference while $\tilde{I}$ is the refined image. The first term is a pixel-wise difference between the refined result and the coarse reference. The second term is a TV loss term which helps to enhance the smoothness and is defined by (8) where i and j denote the position of image pixels. The third term in (7) encourages the similarity between the refined result within the corrupted region and the image patch surrounding the corrupted region. In detail, F denotes a pretrained neural network that takes an image as input and outputs a feature map which contains low-level and middle-level visual features. We take the first three convolutional blocks of VGG-19 [40] as the neural network F. $F(I, i)$ is the extracted feature of patch i in an image I. R denotes a region that contains nonoverlapping patches and approximately covers the corrupted region while $N(i)$ is a set of patches adjacent to the patch i

$$\tilde{I} = \arg\min_{I} \|I - I_0\|^2 + \text{TV}(I) + \frac{1}{|R|} \sum_{i \in R} \min_{j \in N(i) \wedge j \notin R} \Big\{ \|F(I, i) - F(I, j)\|^2 \Big\} \quad (7)$$

$$\text{TV}(I) = \sum_{i,j} \Big(\big(I_{i,j+1} - I_{i,j}\big)^2 + \big(I_{i+1,j} - I_{i,j}\big)^2 \Big). \quad (8)$$

The experimental results of the refinement on high-resolution, in-the-wild, and irregular-shape cases are displayed in Sections V-E and V-F. Notice that these cases have different data source and different kinds of corruption, but they are handled with the same implementation of the refinement.

IV. Implementation

Let us discuss the details of our inpainting pipeline. Training images for CASI require no labels. As shown in Algorithm 1,

Algorithm 1 Training CASI

$F \leftarrow$ LOADMODEL()
$G \leftarrow$ INITWEIGHT(), $D \leftarrow$ INITWEIGHT()
for $i \leftarrow 1$, *maxIterations* **do**
  x, z, M
  for $j \leftarrow 1$, *Diters* **do**
    $x \leftarrow$ SAMPLEBATCH()
    Compute z using Eq. (3)
    Compute l_{adv} using Eq. (4)
    Update D
  end for
  $l_{pix} \leftarrow \text{MSE}(x, z)$
  $f_x \leftarrow F(x), f_z \leftarrow F(z)$
  Compute l_{per} using Eq. (6)
  Compute L_{inp} using Eq. (1)
  Update G
end for

the training stage consists of a limited number of iterations. During each training iteration, the discriminator is updated *Diters* times and the generator is trained once. In each iteration that updates the discriminator, each training image is separated into an image center and an image context. The image center has the same size of the central region, and the image context is the image filled with the mean pixel value in the central region. The image center and the image context of a training image form a training pair. The generator takes the image context as the input and synthesizes the image center. The discriminator attempts to distinguish the synthesized content from the ground-truth image center. The adversarial loss is calculated and then the parameters of the discriminator are updated. In the rest of each training iteration, the pixel-wise *L*2 loss is computed, the feature network extracts a feature from the composite image, and three loss functions are combined to obtain the joint inpainting loss. The generator is finally updated according to the joint loss. This process is repeated until the joint loss converges. In the testing stage, each testing image is first filled with the mean pixel value in the center and then passed to the CASI generator. The central region of the generator's output is cropped and pasted back onto the testing image to yield the final inpainting result.

Our CASI is implemented on top of DCGAN [23] and Context Encoder [2] in Torch and Caffe [42]. ADAM [43] is adopted to perform stochastic gradient descent. As in [2], CASI predicts a larger region which overlaps with the context region (by 4px). 10× weight is used for the pixel-wise *L*2 loss in the overlapping area. Using a TITAN XP GPU, training on a dataset of 20 000 images costs 3 to 4 days. Inpainting a single image takes less than 0.2 s. Recovering a batch of 20 images costs less than 1 s.

V. EVALUATION

This section evaluates our proposed deep neural network architecture and joint loss function on a subset of ImageNet [41] and the Paris StreetView dataset [2], [44]. This subset contains 20 randomly sampled categories, denoted as "ImageNet-20." ImageNet-20 consists of 25 000 training images and 1000 testing images. Paris StreetView contains 14 900 training samples and 100 testing samples.

TABLE I
QUANTITATIVE RESULTS ON IMAGENET-20. CASIS WITHOUT THE ADVERSARIAL LOSS ACHIEVE LOWER MEAN *L*2 ERROR AND HIGHER PSNR BUT BLURRY RESULTS, WHICH INDICATES THAT MEAN *L*2 ERROR AND PSNR INACCURATELY ASSESS OVER-SMOOTH CASES

Method	mean $L1$ error	mean $L2$ error	PSNR
Context Encoder	12.15%	3.31%	15.59dB
CASI,$L2$	11.07%	**2.57%**	**17.08dB**
CASI,$L2$ + per	11.21%	2.64%	16.95dB
CASI,$L2$ + adv	11.15%	2.93%	16.68dB
CASI,$L2$+adv+per	**10.89%**	2.83%	16.81dB

A. Effectiveness of Perceptual Loss

We first verify whether adding a perceptual loss improves the results. CASI is trained using four different loss functions, respectively, to compare their performance. For these loss functions, the hyper-parameters of CASI are set in the same way, and the perceptual loss is defined using the same feature extracted using the same feature network. The four loss functions are: 1) pixel-wise *L*2 loss; 2) *L*2 loss + perceptual loss; 3) *L*2 loss + adversarial loss; and 4) *L*2 loss + adversarial loss + perceptual loss. In the following we use 1)–4) to refer to these loss functions.

Fig. 3 shows the qualitative results of the above loss functions. The resolution of each images is 128 × 128. This result includes four samples representing different cases. All the missing regions are at the center of the image. From left to right, each column corresponds to a loss function from Fig. 3(a)–(d), respectively. As shown in Fig. 3, (a) and (b) generate over-smooth results while (c) and (d) present sharper details. This conforms that the adversarial loss indeed alleviate the blurriness caused by the *L*2 loss. Between (a) and (b), (a) is more blurry while subtle textures or wrinkles can be observed in (b). Between (c) and (d), although they both preserve sharp edges, (d) is more semantically consistent with the context region. These results reveal that the adversarial loss works in the middle level to yield patches with consistent sharp details while the perceptual loss synthesizes consistent high-level contents.

Table I shows the quantitative results from this experiment. It presents numerical errors between the synthesized contents and their ground truth using three commonly employed measures: 1) mean *L*1 error; 2) mean *L*2 error; and 3) PSNR. Notations 1)–4) are used to denote four trained CASI models. As shown in Table I, 1) achieves the smallest mean *L*2 error and largest PSNR while 4) achieves the smallest mean *L*1 error. Mean *L*2 error is smaller for solutions close to the mean value but such solutions are overly smooth and undesirable [see Fig. 3(a) and (b)]. Models trained without the adversarial loss have advantage in mean *L*2 error due to

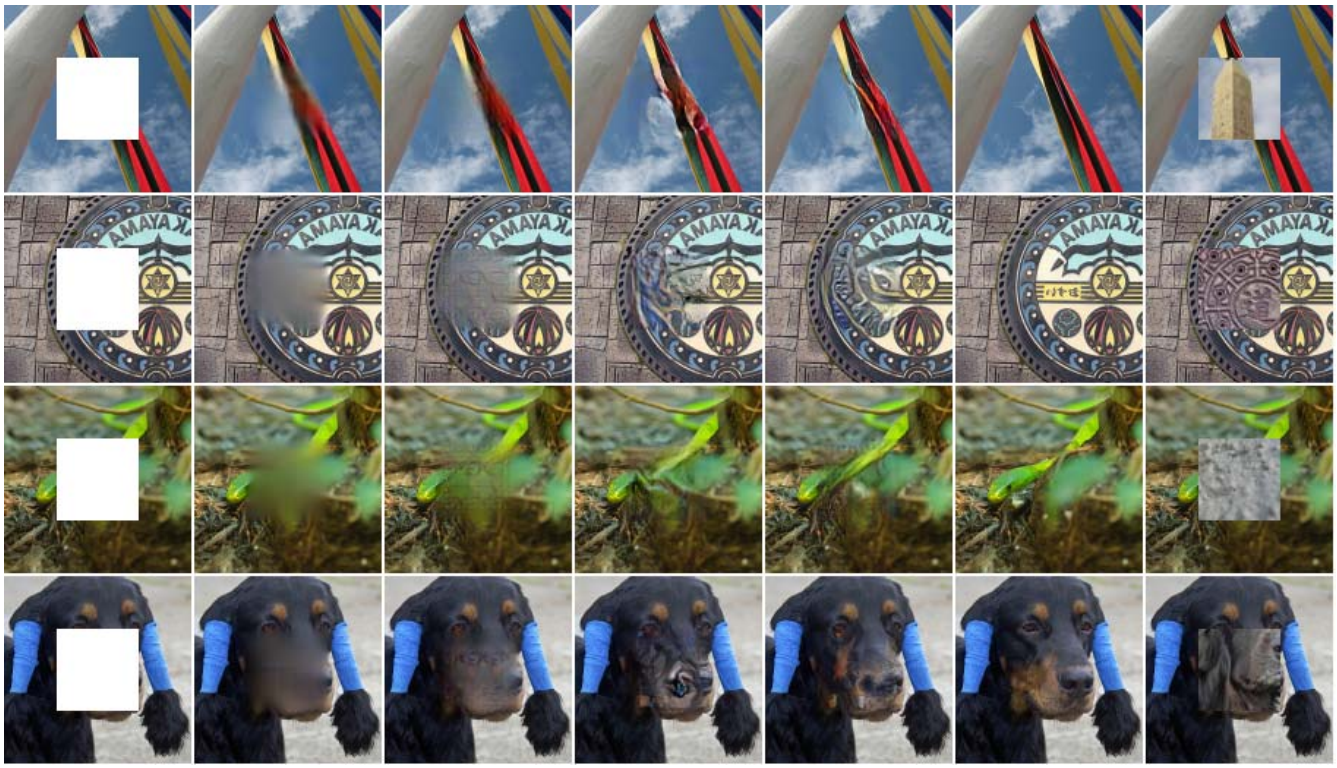


Fig. 3. Comparison among different combinations of loss functions and nearest-neighbor (NN)-inpainting. The adversarial loss promotes low-level sharp details while the perceptual loss improves high-level semantic consistency. (a) $L2$ (b) $L2 +$ per. (c) $L2 +$ adv. (d) $L2 +$ adv $+$ per.

TABLE II
INVESTIGATION OF PERCEPTUAL LOSS

Method	mean $L1$ error	mean $L2$ error	PSNR
$\lambda_{per} = 0$	11.15%	2.93%	16.68dB
$\lambda_{per} = 0.2$	**10.89%**	**2.83%**	**16.81dB**
$\lambda_{per} = 0.4$	11.12%	2.93%	16.60dB
$\lambda_{per} = 0.7$	11.43%	3.06%	16.44dB

TABLE III
EFFECTIVENESS OF FULLY CONVOLUTIONAL ARCHITECTURE

Method	mean $L1$ error	mean $L2$ error	PSNR
CASI+fc	9.70%	1.71%	18.83dB
CASI	**7.49%**	**1.37%**	**20.37dB**

their blurry results. Similar results have been reported in [24]. Between 3) and 4), 4) has smaller mean $L2$ error than 3). And 2) and 4) have smaller mean $L1$ error than 1) and 3), respectively. Thus, the perceptual loss is effective in improving our CASI model.

B. Investigation of Perceptual Loss

This section investigates how the parameter of perceptual loss effect the performance of our method. We set the hyper-parameters in our algorithm as follows. The summation of the weights of all loss terms is 1.0. The weight of the adversarial loss is 0.001, as suggested by [2]. We determine the weight of the perceptual loss λ_{per} by cross validation on the ImageNet-20 dataset. As shown in Table II, setting the weight of the perceptual loss to 0.2 results in the lowest mean $L1$ error, mean $L2$ error, and the highest PSNR value among four different parameter settings.

C. Effectiveness of Fully Convolutional Architecture

This section investigates whether applying fully convolutional architecture benefits semantic inpainting. We design a CASI+fc model by inserting two fc layers after the third layer of the CASI flatting block [described in Fig. 2(a)]. The first fc layer takes a convolutional feature map as input and outputs a 2048-D feature vector which is followed by a Tanh layer. The second fc layer takes the output of the activation layer as input and outputs a feature map with spatial dimensions. Then the fourth layer of the CASI flatting block takes the feature map as input. We compare CASI+fc model and CASI model on Paris StreetView dataset. As Table III shows, CASI outperforms CASI+fc by 2.21% in mean $L1$ error, 0.34% in mean $L2$ error, and 1.54 dB in PSNR although CASI+fc contains more parameters than CASI. The result suggests applying fully convolutional architecture is more conducive for generative network as the fc layers could collapse the spatial structure of the image features.

Fig. 4. High-resolution cases on Paris StreetView.

TABLE IV
EFFECTIVENESS OF RESIDUAL BLOCK

Method	mean $L1$ error	mean $L2$ error	PSNR
CASI-	11.09%	2.93%	16.31dB
CASI	**10.89%**	**2.83%**	**16.81dB**
CASI-	7.79%	1.43%	20.14dB
CASI	**7.49%**	**1.37%**	**20.37dB**

TABLE V
HIGH-RESOLUTION CASE ON PARIS STREETVIEW

Method	mean $L1$ error	mean $L2$ error	PSNR
ContextEncoder	9.04%	1.82%	18.90dB
CASI	**8.04%**	**1.53%**	**19.79dB**
EBIR	11.08%	2.76%	17.59dB
NeuralPatch	9.59%	2.07%	18.42dB
CASI+	**8.62%**	**1.73%**	**19.18dB**

D. Effectiveness of Residual Block

This section verifies whether adding the residual blocks enhance the performance. We design a CASI-model by removing the two residual blocks in CASI model and demonstrate comparison results between them. As shown in the upper part of Table IV, CASI outperforms CASI-by 0.2% in mean $L1$ error, 0.1% in mean $L2$ error, and 0.5 dB in PSNR, on the ImageNet-20 dataset. As shown in the lower part of Table IV, CASI presents better performance than CASI- in mean $L1$ error, mean $L2$ error, and PSNR value, on the Paris StreetView dataset. The above results suggest that adding residual blocks improves the prediction accuracy for the CASI model.

E. High-Resolution Case

This section investigates how our method performs on high-resolution cases. The motivation of investigation on high-resolution cases is that most existing neural network-based inpainting methods can only deal with input images not larger than 128×128. This section demonstrates how the proposed method perform with input images of 512×512. Two groups of experiments are presented. The first group compare our method to [2] by scaling image to match with the input size of [2]. As shown in the upper part of Table V, our CASI model presents lower mean $L1$ error, lower mean $L2$ error, and higher PSNR value than Context Encoder [2] in high-resolution Paris StreetView dataset. The second group investigates whether adding a post-optimization based on our model deals with high-resolution cases. One concurrent work, Neural Patch Synthesis (NeuralPatch) [7], trains its network to synthesize content at the image center and presents high-resolution object removal results during testing. We have integrated our method with post-optimization in [7] (denoted as CASI+) and demonstrate better performance than NeuralPatch [7]. As shown in the lower part of Table V, the CASI+ method achieves lower mean $L1$ error, lower mean $L2$ error, and higher PSNR value in comparison to NeuralPatch [7], which suggests that the proposed CASI can provide more accurate reference content for post-optimization-based image completion methods. Besides, CASI+ also outperforms a nonlearnable method, edge-based image restoration [10], which indicates that neural network learns important prior to reconstruct edges from training data. Fig. 4 is a qualitative comparison among [7] and [10] and CASI+. As Fig. 4 shows, CASI+ extends more reasonable edges and preserves more details than [7] and [10].

F. General and In-the-Wild Case

This section investigates how the proposed method perform on general and in-the-wild cases. The first experiment in this section is to test the proposed method on high-resolution real images that are collected out of ImageNet and Paris StreetView

Fig. 5. High-resolution in-the-wild case.

dataset. The qualitative results of the first experiment are shown in Fig. 5. The resolution of the input images in Fig. 5 are 430×645, 708×1062, and 426×570. The results verify that our proposed method could perform well on in-the-wild cases.

The second experiment in this section is to test the proposed method on real images with irregular corrupted region. The qualitative results of the second experiment are displayed in Fig. 6. These input images are also collected in-the-wild out of ImageNet and Paris StreetView dataset and their resolutions are 357×500, 332×450, and 332×450, respectively. The results suggest that the proposed algorithm is capable of repairing images with irregular corrupted region.

G. Investigation of Generalization Ability

This section investigates the generalization ability of the CASI model. If the CASI model has weak generalization ability and overfits the training data, it may predict what it memorize from the training data. Thus, we conduct a nearest neighbor inpainting (NN-inpainting) experiment. For each testing input image, we search for the most matching patch from the training dataset to complete the image, using the algorithm proposed in [22]. The qualitative results of NN-inpainting are displayed in Fig. 3. The CASI results [in Fig. 3(d)] are quite different from the NN-inpainting results and demonstrate the superiority in preserving both appearance and structure coherence, which indicates that the CASI model does not simply copy or memorize patch from the training dataset while repairing the input images.

H. Comparison With the State-of-the-Art

We compare our proposed CASI model trained using the joint loss with other four state-of-the-art image inpainting methods, including Content-Aware Fill [45], StructCompletion [8], ImageMelding [19], and Context Encoder [2]. As shown in Fig. 7, methods [8], [45], and [19] without using neural network fail to recover the dog face in the first sample, extend the bus window in the second sample, and connect the snake body in the third sample. These methods fail to recover high-level semantics. Context Encoder struggles to display clear structure while the proposed CASI shows visually acceptable results in Fig. 7.

The second experiment in this section compares our method with other state-of-the-art inpainting methods [1], [2], [7], [8], [19], [45] on the Paris StreetView dataset. Table VI shows the quantitative results. Results from PatchMatch [45], NeuralPatch, and Context Encoder are collected from [7] and [2], respectively. As shown in Table VI, our results exceed others by a considerable margin under all three measures. Our method outperforms the second best by 1.58% in mean $L1$ error, 0.53% in mean $L2$ error, and 1.56 dB in PSNR.

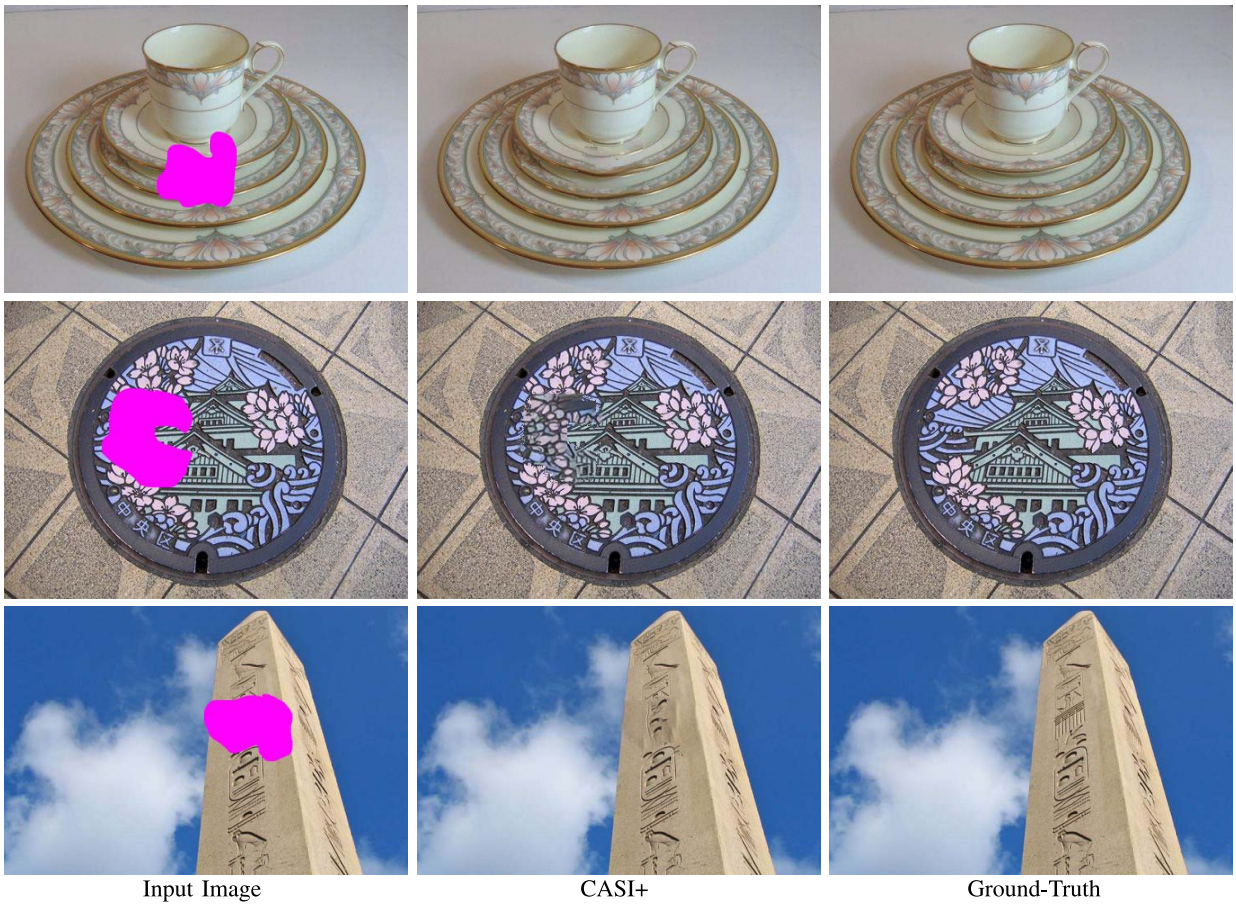


Fig. 6. Irregular-shape case.

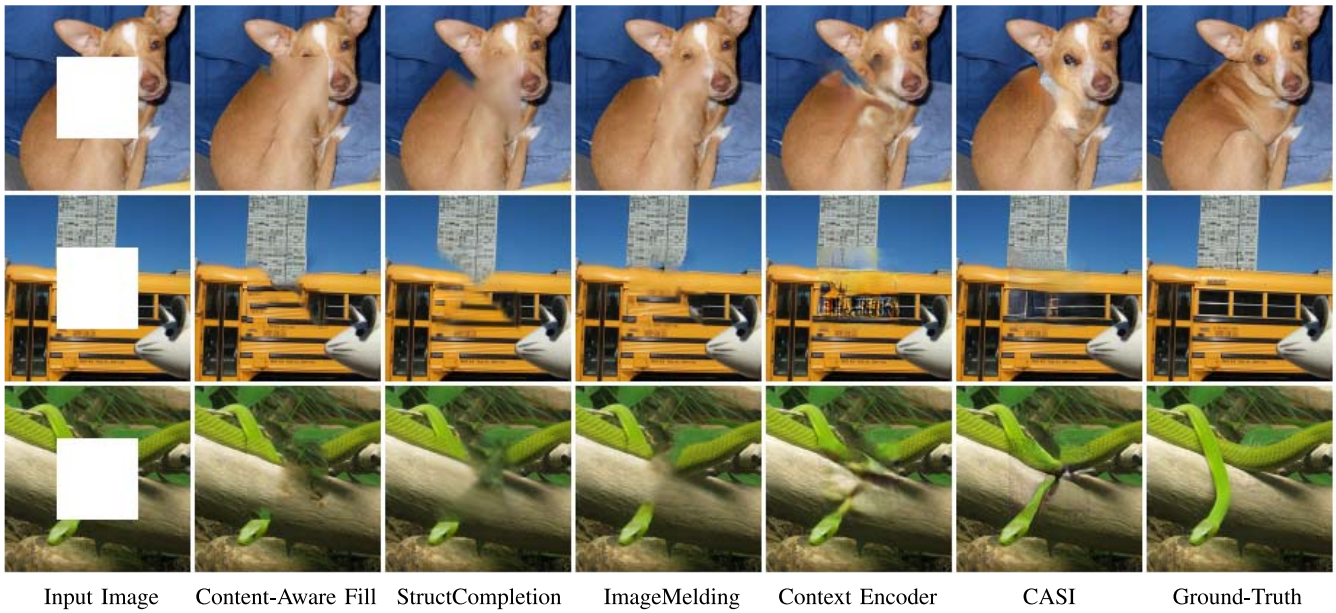


Fig. 7. Comparison on ImageNet-20 dataset.

I. Investigation of Criteria for Inpainting

In this section, we use more criteria to evaluate CASI and Context Encoder, and propose two new criteria for semantic inpainting. There are three major experiments. In the first experiment, we evaluate inpainting methods using structural similarity index (SSIM) [46] and feature similarity index (FSIM) [47]. These indices are originally applied to image quality assessment (IQA) that attempts to quantify

TABLE VI
QUANTITATIVE RESULTS ON PARIS STREETVIEW

Method	mean $L1$ error	mean $L2$ error	PSNR
PatchMatch	12.59%	3.14%	16.82dB
NeuralPatch	10.01%	2.21%	18.00dB
StructCompletion	9.67%	2.07%	18.03dB
ImageMelding	9.55%	2.19%	18.05dB
Context Encoder	9.37%	1.96%	18.58dB
CASI	**7.49%**	**1.37%**	**20.37dB**

TABLE VII
SIMILARITY INDICES ON IMAGENET-20

Method	SSIM	FSIM	FSIMc
Context Encoder	0.2579	0.6977	0.6899
CASI,$L2$	0.5196	0.6255	0.6202
CASI,$L2$ + per	0.4927	0.6843	0.6779
CASI,$L2$ + adv	0.5141	0.7202	0.7148
CASI,$L2$+adv+per	**0.5198**	**0.7239**	**0.7187**
$\lambda_{per} = 0$	0.5141	0.7202	0.7148
$\lambda_{per} = 0.2$	0.5198	0.7239	0.7187
$\lambda_{per} = 0.4$	0.5093	0.7203	0.7149
$\lambda_{per} = 0.7$	0.4951	0.7163	0.7108

the visibility of differences between the two images. Here, we investigate the visual differences between the inpainting results and their corresponding ground truth. Thus, we test inpainting methods using the two IQA indices. SSIM is a classical index defined by structural similarity while FSIM is the state-of-the-art based on two low-level features: 1) phase congruency (PC) and 2) gradient magnitude. FSIM is defined in

$$\text{FSIM} = \frac{\sum S_{\text{PC}}(x) \cdot S_G(x) \cdot \text{PC}_m(x)}{\sum \text{PC}_m(x)} \tag{9}$$

where $S_{\text{PC}}(x)$ and $S_G(x)$ are PC similarity and gradient similarity, respectively, at position x, and $\text{PC}_m(x)$ is the PC value of x as a weight. As shown in Table VII, all CASI models achieve higher similarity with the ground truth than Context Encoder under SSIM, FSIM, and FSIM for color image. It indicates that our method not only recovers more consistent structures but also synthesizes content with higher visual quality. However, SSIM and FSIM are still biased toward blurry results of CASI, $L2$ ($+l_{\text{per}}$).

In the second experiment, we introduce a novel local entropy error to rate blurry predictions more accurately. Entropy in texture analysis is a statistic characterizing the texture within an image region, as defined in [48]. The local entropy at a pixel is defined as the entropy within a 9×9 neighborhood of the pixel. We define local entropy error as the mean squared error (LEMSE) or the mean absolute error (LEMAE) of local entropy within the synthesized region. As shown in Table VIII, our proposed CASI delivers the lowest LEMSE and LEMAE among all methods. In addition, CASI with $L2$ loss and CASI with $L2$ + per loss achieve

TABLE VIII
LOCAL ENTROPY ERRORS ON IMAGENET-20

Method	LEMSE	LEMAE
Context Encoder	0.5872	0.5391
CASI,$L2$	1.8926	1.0795
CASI,$L2$ + per	0.8454	0.7219
CASI,$L2$ + adv	0.4869	0.4945
CASI,$L2$+adv+per	**0.4611**	**0.4847**
$\lambda_{per} = 0$	0.4869	0.4945
$\lambda_{per} = 0.2$	0.4611	0.4847
$\lambda_{per} = 0.4$	0.4470	0.4759
$\lambda_{per} = 0.7$	0.4492	0.4771

the largest and second largest errors under both LEMSE and LEMAE, which is consistent with most of the visual results (a subset is given in Fig. 3) and confirms that our proposed local entropy error is capable of rating over-smooth results accurately.

In the third experiment, we propose a high-level criterion, SME, which aims at measuring how successful an inpainting method recovers the semantics. SME is defined with respect to a pretrained image classifier that outputs a probability of the image being part of each possible category. SME is based on two probabilities that the groundtruth image and the synthesized image belong to the groundtruth category, respectively. It is formulated as follows:

$$\text{SME} = \frac{1}{n} \sum_{i=1}^{n} \max\left(0, P_{x_i}^{y_i} - P_{z_i}^{y_i}\right) \tag{10}$$

where n is the number of testing samples, x_i, z_i, and y_i are the groundtruth image, synthesized image (with real context), and the groundtruth category of the ith sample. $P_{x_i}^{y_i}$ is the probability that image x_i belongs to category y_i, estimated by a pretrained classifier (e.g., residual network [49] or VGG network [40]). Here, we associate the probability of assigning the correct label with our SME because we focus on to what extent a corruption "makes a dog unlike a dog" and to what extent the restored content "makes a dog look like a dog again." A baseline model simply fills the missing region with the mean pixel value. The SME of this baseline measures how much a corrupted region harms the semantic information of an image. In Table IX, SME-rL represents the SME achieved by applying an L-layer residual network as the classifier while SME-vL represents the SME achieved by adopting an L-layer VGG network as the classifier. Notice that our feature network is simpler than the ResNets used for estimating SME, which implies that harvesting knowledge using a low-capacity model can reduce the SME estimated by a high-capacity classifier. As shown in Table IX, our proposed network outperforms other inpainting methods by achieving the smallest SME.

Perceptual loss weight is also investigated on the above new criteria for semantic inpainting, as shown in the lower part of Tables VII–IX. $\lambda_{\text{per}} = 0.7$ performs better on similarity indices and SMEs while $\lambda_{\text{per}} = 0.4$ demonstrates better results

TABLE IX
SMEs ON IMAGENET-20

Method	SME-r50	SME-r101	SME-r152	SME-r200	SME-v16	SME-v19
baseline	0.2063	0.1735	0.1852	0.2063	0.1794	0.2086
Context Encoder	0.1467	0.1462	0.1442	0.1467	0.1001	0.1123
CASI,$L2$	0.1862	0.1908	0.1886	0.1877	0.1444	0.1652
CASI,$L2$ + per	0.1542	0.1631	0.1671	0.1626	0.1213	0.1384
CASI,$L2$ + adv	0.1276	0.1359	0.1349	0.1362	0.0846	0.0952
CASI,$L2$+adv+per	**0.1070**	**0.1180**	**0.1201**	**0.1200**	**0.0721**	**0.0775**
$\lambda_{per} = 0$	0.1276	0.1360	0.1350	0.1363	0.0846	0.0952
$\lambda_{per} = 0.2$	0.1070	0.1180	0.1201	0.1200	0.0721	0.0775
$\lambda_{per} = 0.4$	0.1074	0.1125	0.1218	0.1215	0.0704	0.0767
$\lambda_{per} = 0.7$	0.0994	0.1126	0.1117	0.1131	0.0632	0.0702

Fig. 8. Limitation of our method.

on local entropy error. To compromise different criteria, λ is chosen from 0.2 to 0.4.

VI. CONCLUSION

In this paper, we have presented a fully convolutional GAN with a context-aware loss function for semantic inpainting. This network employs a fully convolutional architecture in the generator, which does not have a fc layer as the bottleneck layer. The joint loss includes a perceptual loss to capture semantic information around the synthesized region. In addition, we have developed two new measures for evaluating sharpness and semantic validity, respectively. In summary, our method delivers state-of-the-art results in qualitative comparisons and under a wide range of quantitative criteria. As shown in Fig. 8, the proposed method has limitation that it struggles to restore a corrupted region with dense strongly curved lines. We aim to address the problem in the future work.

REFERENCES

[1] *Content-Aware Fill*. Accessed: Aug. 26, 2018. [Online]. Available: https://research.adobe.com/project/content-aware-fill
[2] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
[3] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.
[4] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Image inpainting through neural networks hallucinations," in *Proc. IEEE 12th Image Video Multidimensional Signal Process. Workshop (IVMSP)*, Jul. 2016, pp. 1–5.
[5] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[6] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. CVPR*, vol. 2, Jul. 2017, p. 4.
[7] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, p. 3.
[8] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Trans. Graph.*, vol. 33, no. 4, p. 129, 2014.
[9] J. Jia and C.-K. Tang, "Image repairing: Robust image synthesis by adaptive ND tensor voting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2003, pp. 643–650.
[10] A. Rares, M. J. Reinders, and J. Biemond, "Edge-based image restoration," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1454–1468, Oct. 2005.
[11] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, p. 107, 2017.
[12] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-map image editing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 151–158.
[13] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2004, pp. 120–127.
[14] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, 2005.
[15] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, 1999, pp. 1033–1038.
[16] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 341–346.
[17] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2003, pp. 721–728.
[18] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1153–1165, May 2010.
[19] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, "Image melding: Combining inconsistent images using patch-based synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, p. 82, 2012.
[20] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 303–312, 2003.
[21] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2649–2661, Nov. 2007.
[22] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 4.
[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
[24] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 658–666.

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[26] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 702–716.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[28] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[29] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep CNN features," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5012–5024, Nov. 2016.

[30] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 247–256.

[31] G. Li and Y. Yu, "Contrast-oriented deep neural networks for salient object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8337098/

[32] G. Li, Y. Xie, T. Wei, K. Wang, and L. Lin, "Flow guided recurrent neural encoder for video salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3243–3252.

[33] W. Zhang, C.-W. Fang, and G.-B. Li, "Automatic colorization with improved spatial coherence and boundary localization," *J. Comput. Sci. Technol.*, vol. 32, no. 3, pp. 494–506, 2017.

[34] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, "Attention-aware face hallucination via deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1656–1664.

[35] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[36] G. Li *et al.*, "Non-locally enhanced encoder-decoder network for single image de-raining," *arXiv preprint arXiv:1808.01491*, 2018.

[37] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[38] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[39] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1349–1357.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[41] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[42] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 675–678.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[44] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes Paris look like Paris?" *ACM Trans. Graph.*, vol. 31, no. 4, p. 101, 2012.

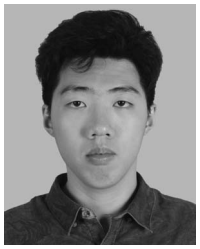
[45] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein, "The patchmatch randomized matching algorithm for image manipulation," *Commun. ACM*, vol. 54, no. 11, pp. 103–110, 2011.

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[47] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[48] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB®*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003, ch. 11.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.



Haofeng Li received the B.S. degree in computer science from the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China, in 2015. He is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science, University of Hong Kong, Hong Kong.

His current research interests include computer vision, image processing, and deep learning.

Mr. Li was a recipient of the Hong Kong Postgraduate Fellowship.

Guanbin Li received the Ph.D. degree in computer science from the University of Hong Kong, Hong Kong, in 2016.

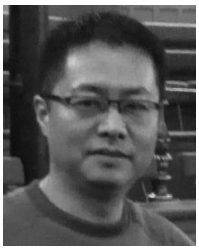
He is currently a Research Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He has authorized and co-authorized over 20 papers in top-tier academic journals and conferences. His current research interests include computer vision, image processing, and deep learning.

Dr. Li was a recipient of the Hong Kong Postgraduate Fellowship. He serves as an Area Chair for the conference of VISAPP. He has been serving as a reviewer for numerous academic journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON COMPUTERS, CVPR2018, and IJCAI2018.

Liang Lin (M'09–SM'15) received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2008.

He is currently the Executive Research and Development Director of SenseTime Group Ltd., Beijing, and a Full Professor with Sun Yat-sen University, Guangzhou, China. He is the Excellent Young Scientist of the National Natural Science Foundation of China. From 2008 to 2010, he was a Post-Doctoral Fellow with the University of California at Los Angeles, Los Angeles, CA, USA. From 2014 to 2015, he was with Hong Kong Polytechnic University, Hong Kong, and the Chinese University of Hong Kong, Hong Kong, as a Senior Visiting Scholar. He currently leads the SenseTime Research and Development Team to develop cutting-edges and deliverable solutions on computer vision, data analysis and mining, and intelligent robotic systems. He has authorized and co-authorized over 100 papers in top-tier academic journals and conferences.

Mr. Lin was a recipient of the Best Paper Runners-Up Award in ACM NPAR 2010, the Google Faculty Award in 2012, the Best Paper Diamond Award in IEEE International Conference on Multimedia and Expo (ICME) 2017, and Hong Kong Scholars Award in 2014. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, *Visual Computer*, and *Neurocomputing*. He served as an Area/Session Chairs for numerous conferences, such as ICME, Asian Conference on Computer Vision, and ACM International Conference on Multimedia Retrieval. He is a fellow of IET.



Hongchuan Yu received the Ph.D. degree in computer vision from the Chinese Academy of Sciences, Beijing, China, in 2000.

He is currently a Principal Academic of computer graphics with the National Centre for Computer Animation, Bournemouth University, Poole, U.K. At PIs, he has secured over 2 million in research grants from EU H2020, Royal Society. He has published over 70 academic articles in reputable journals and conferences. His current research interests include image processing, pattern recognition, and computer graphics.

Dr. Yu is a fellow of High Education of Academy United Kingdom. He regularly served as PC Members/Referees for IEEE journals and conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, and the IEEE TRANSACTIONS ON CYBERNETICS.

Yizhou Yu received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2000.

He is currently a Professor with the University of Hong Kong, Hong Kong, and was a Faculty Member with the University of Illinois at Urbana–Champaign, Champaign, IL, USA, from 2000 and 2012. His current research interests include deep learning methods for computer vision, computational visual media, geometric computing, video analytics, and biomedical data analysis.

Dr. Yu was a recipient of the 2002 U.S. National Science Foundation CAREER Award and the 2007 NNSF China Overseas Distinguished Young Investigator Award. He has served on the editorial board of *IET Computer Vision*, the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, *Visual Computer*, and the *International Journal of Software and Informatics*. He has also served on the program committee of many leading international conferences, including SIGGRAPH, SIGGRAPH Asia, and the International Conference on Computer Vision.