Neural Processing Letters
https://doi.org/10.1007/s11063-019-10080-2

Attentive Semantic and Perceptual Faces Completion Using Self-attention Generative Adversarial Networks

Xiaowei Liu[1] · Kenli Li[1] · Keqin Li[2]

Abstract

We propose an approach based on self-attention generative adversarial networks to accomplish the task of image completion where completed images become globally and locally consistent. Using self-attention GANs with contextual and other constraints, the generator can draw realistic images, where fine details are generated in the damaged region and coordinated with the whole image semantically. To train the consistent generator, i.e. image completion network, we employ global and local discriminators where the global discriminator is responsible for evaluating the consistency of the entire image, while the local discriminator assesses the local consistency by analyzing local areas containing completed regions only. Last but not least, attentive recurrent neural block is introduced to obtain the attention map about the missing part in the image, which will help the subsequent completion network to fill contents better. By comparing the experimental results of different approaches on CelebA dataset, our method shows relatively good results.

Keywords Attention mechanism · Images completion · Non-local neural net · Semantics completion

1 Introduction

1.1 Background

In many application scenarios, image completion is a very useful research area, which allows filling in target regions with alternative contents [31] and it is a fundamental problem of human low-level and high-level visual perception. As it can be used to fill occluded image

✉ Kenli Li
lkl@hnu.edu.cn

Xiaowei Liu
liuxiaowei@hnu.edu.cn

Keqin Li
lik@newpaltz.edu

[1] College of Information Science and Engineering, Hunan University, Changsha 410081, China

[2] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

Published online: 27 July 2019

regions or repair damaged photos, it has aroused widespread interest in computer vision [39]. This technology has also been extended to other related applications, such as video completion [36,37].

Nevertheless, it is still a challenging problem, because it often requires high level semantic understanding of the scene. It is not only necessary to complete the texture in the picture, but also understand semantic anatomy of the scene and object being completed.

Essentially, for image completion tasks, the generator needs to know the data distribution of individual objects in the image and the overall structure of the scene, just like a real life painter can draw something well. However structural characteristics and distributions of different objects are quite different. It is difficult to directly learn distributions of a large number of different objects at once. Generally speaking, learning the distribution of the same kind of object is relatively simple, such as a face image, whose structure is relatively fixed. Once the probability distribution of the object to be completed is known, the completion task will become a natural state of being as a painter knows how to draw the unfinished portrait. So at this stage, many completion methods are based on this idea. Here, we review some of classical methods both in the past and in the present.

There are many methods of image completion, such as patch-based image synthesis [2, 5,8,13,16,34], especially in the task of background inpainting, which has been widely used in practice, eg. the smart filling function in photoshop [13]. However, since they assume that missing patches can be found somewhere in the background area, they can’t draw novelty or contents never appeared in the image. In those challenging situations, the damaged region always involves complex, non-repetitive structures as a whole, but it has a relatively fixed pattern. For example, if a mouth is missing from a face image, a second mouth cannot be found elsewhere in the face, then all faces have a fixed pattern-there is a mouth under the nose. Furthermore, these methods cannot capture high level semantics.

Deep convolution neural network (CNN) has a strong ability to learn image representations, and has been successfully applied to inpainting in varying degrees [10,22,42,48]. Inspired by [9,15,32,38] etc, it is generally better to fuse multiple information to accomplish this task. And in recent years, semantic image inpainting has been regarded as an image generation problem and solved within the framework of Generative Adversarial Networks (GANs) [14], where there is a generator against a discriminator and it can successfully generate seemingly reasonable visual content with clear details. State-of-the-art results have been achieved [22,27,42].

1.2 Motivation

Nevertheless, some deep GANs-based methods [6,22,27,42,48,50] have shown prospective results for challenging tasks of filling missing areas in facial images. However, there are some common limitations in all existing solutions based on GANs and deep convolution network.

First, in most cases of image ranking, image retrieval, classification, gesture estimation as described in [11,19–21,45–47], they take advantage of an auto-encoder architecture, in which the entire input information will be encoded into a multi-dimensional vector, and the decoder in turn. In this way, there will be information loss in the encoding process, which is similar to a lossy compression, while the decoder struggles to restore the defective information to original information. Moreover, local characteristics of CNN limit the ability of discriminators and generators. Even so, existing GANs-based methods strive to understand the high-level semantics in image context and generate semantically consistent content. These methods can generate visually plausible images, but usually produce distorted structures or blurred textures that are inconsistent with surrounding areas, mainly because the convolutional neural network is not available when referring to information from long-distant spatial locations [35]. Combining the above two main reasons, it finally leads to the inaccurate recovery of space-related information during decoding.

Second, how to design the loss or difference between the generated image and the original image, which is very difficult for this similarity measurement. It is obvious that, we need losses or differences not only at the underlying pixel-level, but also high semantics-level. To be able to meet all these requirements, we need to combine various measurement methods.

Third, for image completion in any position and size, existing models [22,27,31,42,48] need to input the mask matrix corresponding to the image to be completed, where this mask matrix requires input or marking manually. This will become a disaster for a large number of completion tasks.

1.3 Our Contributions

To overcome above limitations and achieve better results, we design a novel deep generation network for semantic inpainting. To this end, we mainly make the following three contributions:

First, for solving the first problems in the previous Sect. 1.2, we no longer strictly abide by the traditional auto-encoder structure, even in the flatting block of the model, data flows still maintain multidimensional tensors type. To solve the local limitation of convolution network, we employ non-local neural blocks [35], which are complements to convolutional operations and helps with modeling long range, multi-level dependencies across image regions. At the same time, the use of dilated convolution also enlarges receptive field further.

Second, in terms of similarity measurement, we use a perceptual loss or semantic loss, which is promising in capturing high-level semantic difference. For example, we can use the same neural network to extract features from both, and then calculate the differences between features. Among them, a face parsing net whose task is the face semantic segmentation can be used to maintain semantic consistency of synthesized facial images compared with the ground truth.

Third, for concentrating more on the missing region automatically, we employ an attentive LSTM module to gain attention map of the corrupted region in portrait, with the help of which it is convenient for the generator close behind to focus on the missing/damaged region, i.e. a key part to be synthesized.

In our experiments, our model can successfully reduce semantic errors and get better visual results. This shows that our model can infer visually and semantically valid content from context information of an entire image, especially the missing/damaged area and its surroundings. The experimental results demonstrate that our proposed approach generates higher quality completion results than most of existing ones.

2 Related Work

A large number of literatures exist for image inpainting, and due to space limitations we are unable to discuss all of those in detail. Groundbreaking work in that direction includes the aforementioned works and references therein. Since our method is based on generative models and neural networks, we will review relevant academic researches and technical works below.

2.1 Generative Adversarial Network

GANs was first proposed by Goodfellow et al. [14], which trains two adversarial networks simultaneously to capture data distribution of input images. Therefore, a typical GANs network consists of a generator and a discriminator, in which the generator tries to learn the optimal transport mapping from inputs to outputs, and the discriminator judges the quality of its generation. The generator continuously improves the ability to generate images that may fool the discriminator to determine the artificial as a real one. However, the discriminator keeps improving itself to tell whether the image is generated or real. By mutually learning with alternating iterations between the discriminator and generator, a nash equilibrium will be achieved in theory, when the generator can generate images that are visually plausible enough to make the discriminator unable to determine whether the image is synthetic. The zero-sum game between both can be expressed as the following optimization of min-max problem:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data(x)}} \log D(x) + E_{z \sim p_Z(z)} \log(1 - D(G(z))) \tag{1}$$

where x is obtained by random sampling from the training sample set, which yields to distribution $P_{data}(x)$, and z stands for a random vector in some latent space. In our model, z is not a random vector, but a latent tensor generated by a encoder, one of the components of the completion module as shown in Figs. 1 and 2. Here, $G(\cdot)$ represents a generator who is inspired by the latent tensor and generates a sample, and $D(\cdot)$ is responsible for evaluating the likelihood that a sample is in the training data set. In the final stage of training, $G(\cdot)$ become perfect at last, when $D(\cdot)$ unable to determine where the input generated image comes from, that is, the output is 1/2.

2.2 Non-local Neural Network and Self-attention GAN

Recently, Wang et al. [35] first presented a non-local neural structure as a generic building block for capturing long-range dependencies. Inspired by the classical non-local means method in [3] for image denoising, the proposed non-local operation computes the response at a position as a weighted sum of the features at all positions. This building block can be easily hot-plugged into many neural network architectures involving global awareness. From this, Zhang et al. [51] proposed self-attention GANs (SA-GANs) where the non-local block was integrated into GANs. In SA-GANs, feature cues from all locations can be used to generate details. Moreover, the discriminator can inspect whether far-end details of the image are consistent with others. In our model, we use non-local neural blocks in both the generator and discriminator.

2.3 Image Inpainting

Face inpainting is only one aspect of image restoration technology. Therefore, image restoration technology is very useful for face inpainting.

Nowadays, image completion problems can be considered to be an application of image generation. Especially in recent years, there are a large number of academic papers on image generation, most of which are based on the deep GANs framework. And almost all GANs frameworks are presented with image generation as examples. Such as Radford et al. [33] further developed deep convolutional GANs (DCGANs) that have certain architectural constraints combining convolution neural network, and demonstrate that they are a strong candidate for image generation. Kataoka et al. [25] mixed GANs with a special kind of attention mechanism, and generated more real images. Ouyang et al. [30] proposed a novel learning architecture of LSTM conditional generative adversarial networks to generate plausible images from word descriptions.

It is quite intuitive to apply image generation technology to image completion. Moreover, in the real image completion task, the data distribution of the image to be completed should be loaded, and the perfect image generator has naturally acquired this distribution. Many scholars have done the work of image inpainting and completion, e.g. Pathak et al. [31] used AlexNet architecture as the encoder with a novel channel-wise fully connected layer for feature learning for semantic inpainting. Yu et al. [48] proposed a new method based on the deep generative model, which not only can synthesize novel image structures, but also make better prediction by using the surrounding image features as reference during network training. In particular, Li et al. [27] developed an effective facial completion model using GANs with many losses.

2.4 Visual Attention

Attention-related neural processes have been extensively investigated in neuroscience. Visual attention is a particularly interesting aspect: many animals focus on specific parts of their visual inputs. This principle is of great significance to the visual system as we need to choose the most pertinent part of information, rather than using all available information, a large part of which is independent of the response of the nervous system. This is very similar to the focusing function of a camera, which concentrates on one or two targets only and burs others as a background at a particular moment. A similar idea focusing on specific parts of inputs has been applied to all possible areas of deep learning, such as natural language processing, reasoning, and computer vision [4,7,23].

From the perspective of attention's role, visual attention includes temporal and spatial attention. Our application scenario has nothing to do with time, so we only use spatial attention, which is the ability to focus on specific objects in a visual environment. In deep learning, there have been many studies [1,24,25,29,41,48] on learning spatial attention. Here, we choose to review some representative models related to the proposed contextual attention model. For example, Mnih et al. [29] presented a novel recurrent neural network model, which can extract information from images or videos by adaptively selecting a sequence of regions or locations. Ba et al. [1] presented a multi-objects recognition model based on attention mechanism. The model is a kind of recurrent neural network trained with reinforcement learning, which is used to identify the most relevant regions in input images. Jaderberg et al. [24] first proposed a parameterized spatial attention model, called spatial transformation network for object classification. However, this model is not suitable for modeling patch-wise attention for some reason. Recently, Ouyang et al. [30] described an attention-based model that automatically learns to annotate the content of images and gazes at salient objects and generates equivalent words in output sequences. Qian et al. [32] injected visual attention into both the generative and discriminative networks for Raindrop Removal. Based on the above ideas, we put forward the following model—attention LSTM module as shown on the left in Fig. 1, specially used to obtain the attention map of the corrupted area in the input image. Through end-to-end training the contour weight of a face can be obtained as shown at the bottom of the attention LSTM module in Fig. 1.

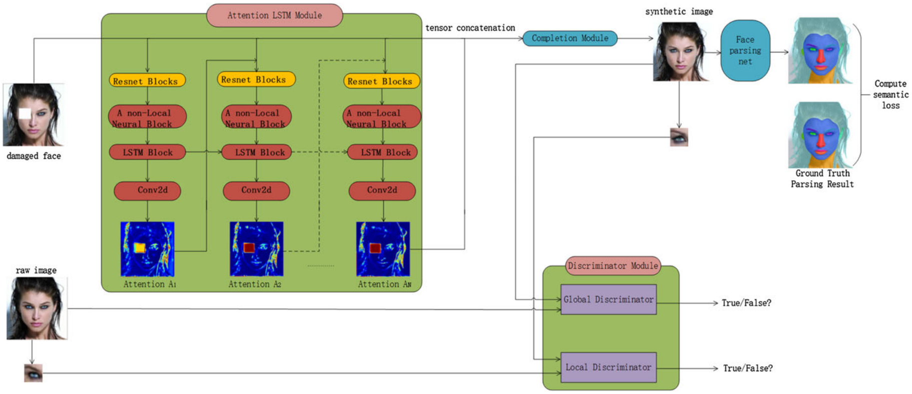

Fig. 1 Our global architecture of proposed model, which is consists of two parts—a generator and a discriminator as the usual GAN. The generator is made up of an attention LSTM module and a completion module, and the global and local discriminators compose the whole discriminator module. Two discriminators are learned to distinguish the synthesize contents in the corrupted region and whole generated image as real and fake. Face parsing network is a semantic segmentation network which is a pre-trained and fixed model and its loss is a regularization term of GAN objective function, which further ensures that the newly generated face and the original face are more semantically consistent. In detail, it is employed to compute the semantic loss between the synthetic and real one. Note that only the generator (attention LSTM module and completion module) is needed during the testing phase

3 Approach

Our method is based on self-attention GANs and consists of two main parts: a generator module and a discriminator module. The generator consists of an attention LSTM module and a completion module. The discriminator module is made up of a global discriminator and local one. Additionally, there is a face semantic segmentation net, called face parsing network, which is a pre-trained model and remains fixed, for further ensuring the new generated face and the corresponding ground truth more consistent semantically. In practice, it is employed to compute the semantic loss between the synthetic and real one. Next we will introduce each module of the entire model in detail.

3.1 Attention LSTM Module

It is a recurrent neural network (RNN) based module, where we use long short-term memory (LSTM) [18] instead of the ordinary RNN unit. The attention LSTM module as shown in Fig. 1 finds the area of interest in the input image for repair. These areas are mainly the missing areas and their surrounding structures necessary to complete the network, in order to help obtain better local image restoration results. As shown in Fig. 1, not only three generated attention maps focus on missing parts, but also concern about the surrounding face contour, such as the other eye, the only nose, mouth and hair in long-distance from the corrupted region. Each unit of the module consists of four parts—a Residual neural network (Resnet) [17] block, a non-local neural block, an LSTM unit and a conv2d operation.

3.1.1 Resnet Block

This block may have multiple resnet blocks and try to extract features from input corrupted images. As shown in Fig. 1, The resnet block is in the front end of each RNN cell. The

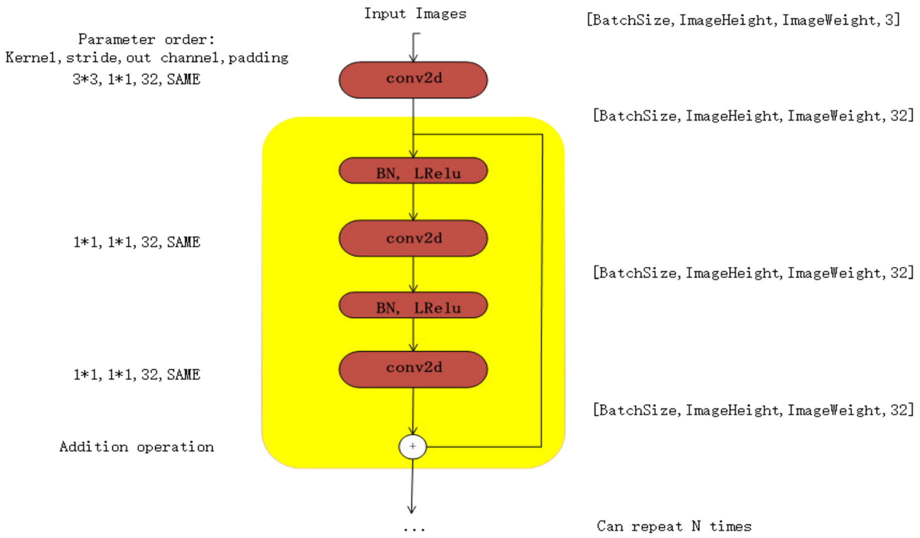
Input Images
Parameter order:
Kernel, stride, out channel, padding
3*3, 1*1, 32, SAME
conv2d
[BatchSize, ImageHeight, ImageWeight, 3]
[BatchSize, ImageHeight, ImageWeight, 32]
BN, LRelu
1*1, 1*1, 32, SAME
conv2d
[BatchSize, ImageHeight, ImageWeight, 32]
BN, LRelu
1*1, 1*1, 32, SAME
conv2d
[BatchSize, ImageHeight, ImageWeight, 32]
Addition operation
+
...
Can repeat N times

Fig. 2 Resnet blocks. BN means batch normalization operation, *Lrelu* is short for Leaky-ReLU and $con2d$ is a convolutional operation whose parameters and their order are shown on the left side of the graph. On the right side, the shape of the flowing tensor have be listed in vertical direction at a specific time. Input images are colorful face images, which have three channels and come from CelebA. E.g. the last item in the upper right corner of the above figure indicates the number of channels in input images

detailed structure is shown in Fig. 2. The residual block marked with orange background can be repeated N times. In our implementation, N is set to 3, which is up to your hardware configuration of computers.

3.1.2 Non-local Neural Block

Next, extracted feature maps flow into non-local neural block which can capture long distance dependencies. Using 1×1 convolutions, matrix multiplication and soft-max operation, the following Eq. (2) is essentially achieved [3,35]:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{2}$$

Here i is the index of an output location in space, time or space–time, and its response will be calculated. j is the index of enumerating all possible locations. x is the input signal (image, video and often their highly enriched features) and y is the output signal of the same size as x. A pairwise function f computes relationship representation (such as affinity) between positions of index i and all j. The unary function g computes a representation of the input signal at the position j, whose response is normalized by a factor $C(x)$. For detailed discussion, please refer to literature [35]. The implementation here is shown in Fig. 3.

3.1.3 LSTM Block

Through a non-local block, we can pick up more global associated feature information. After that, we are arriving at the LSTM block in our RNN module, which strengthen the learning of

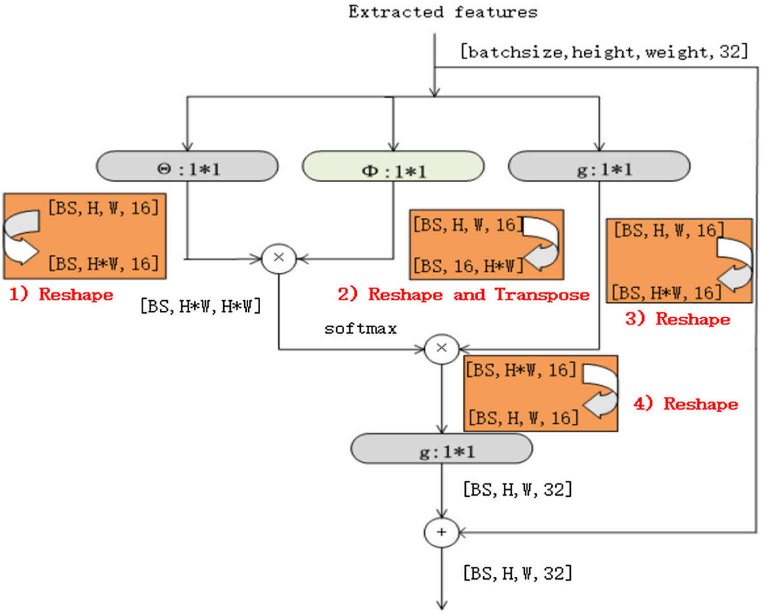

Fig. 3 A space non-local block. The feature maps are shown as a shape of their tensors, e.g., $BS^*H^*W^*32$ for 32 channels, where BS refers to batch size, H and W are the height and width of the input image respectively. "⊗" denotes matrix multiplication, and "⊕" denotes element-wise sum. The gray boxes denote 1×1 convolutions. Here we show the embedded gaussian version, with a bottle-neck of 16 channels. The last thing to note is that the purpose of the second "reshape and transpose" operation marked by red color is to match the dimension of the tensor obtained by the first "reshape" operation when the two are multiplied

the missing part of the input image, and finally the learned attention map will be input into the completion net to guide the image inpainting, actually tells the network to pay more attention to the corrupted part. This guiding process may be implicit, but in our model, we explicitly use our attention information to emphasize new losses that need to be reduced. Finally, a conv2d operation is employed for getting a attention map with the same dimension as the missing labeling matrix and one channel. The detailed LSTM cell diagram is no longer drawn here. For detailed discussion, please refer to the corresponding literatures [12,18,30,40] etc.

3.2 A Completion Module

By and large, the completion network belongs to an encoder–decoder architecture. The difference is that we add dilated convolutions [44] and non-local residual neural blocks at both ends of the flatting block to learn spatial long dependences. The completion network structure is shown in Fig. 4 in details. A down-sampling block, a flatting block and an up-sampling block consist of the whole completion network. The down-sampling block consists of a series of convolutions, batch normalization (BN) and leaky rectified linear unit (LRelu) alternately arranged. This part extracts main features through familiar convolution operations. Later, we are going to the next block, a flatting block that further refine the features and find long range dependences using non-local blocks and dilated convolutions. Finally, we hire a decoder architecture to restore the corrupted image from extracted features at the stage of down-sampling. A more detailed description is given in the notes below the Fig. 4.

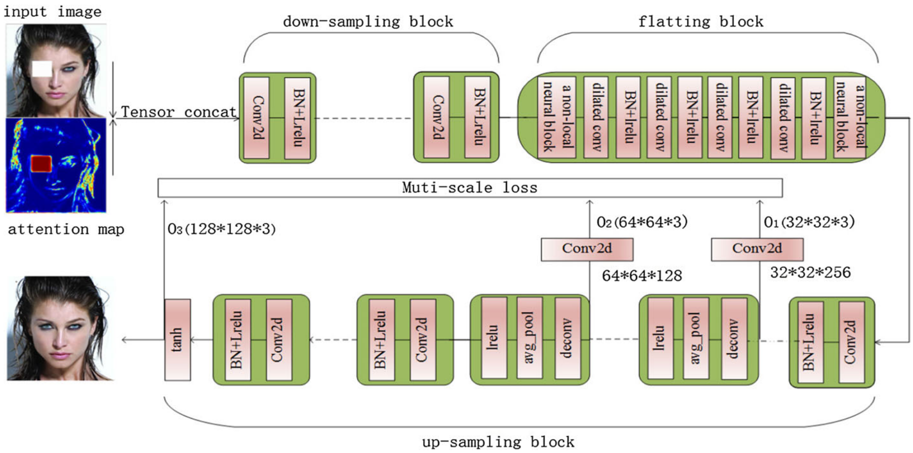

Fig. 4 Completive network diagram. It is a general encoder–decoder architecture that mainly consists of three blocks, namely a down-sampling block, a flatting block and an up-sampling block. The down-sampling block makes use of a series of conv2d operations. From left to right, the size is getting smaller and the thickness of the layer grows. In the flatting block, the height and weight of tensor data flow are relatively minimum. Dilated convolution alternate with batch normalization having non-local neural blocks both ends. The last part is an up-sampling block, which makes up of many convolutions and deconvolution. Finally the size and channel number of the input image are recovered. The last point is to explain, there may be some residual connections between the down-sampling and the up-sampling

3.3 Discriminator Module

Our complete discriminator here is composed of a local and a global network, which are learned to distinguish the synthesize contents in marked regions and assess whole images formed by splicing the generated missing part with the input image. At the end of the discriminator, Two sub-discriminators will judge global and local consistency and facticity respectively, and both of them will output a scalar, that is, the probability of judging to be true. Its rough architecture is shown in the lower right corner of the Fig. 1. The detailed structure is similar to [22], which is no longer drawn here.

The discriminator is only useful in the training stage, but no longer in the testing stage. The reason for the existence of discriminators here is to train better generators. Since only one hole was dug in the original image during training, only a local discriminator was used. If you dig two holes, you should employ two local discriminators and so on. But when there are too many holes, too many local discriminations are unreasonable. Furthermore, if dividing the image into four blocks fixedly and setting up four local discriminators, we will encounter the problem of the number of positive and negative examples mismatching in training. For simplicity's sake here, we only dig one hole when training, whose size and position is random. But the input of local discriminator is a fixed window bigger than the hole.

3.4 Loss Functions

Due to our architecture based on GANs, loss functions, i.e. objective functions are composed of two parts, a generative part and a discriminative one, similar to Eq. (1). Firstly, the representation of image completion is introduced, and then conventional loss functions and additional loss functions as regularization terms are elaborated in detail.

3.4.1 Representation of Problems

The problem of image completion can be represented by the following Eq. (3):

$$I_{rec} = (C(I, A_N) \otimes M) \oplus (I \otimes (1 - M)) \tag{3}$$

Here I_{rec} is the final reconstruction result, which is a combination of the part that has been repaired (corresponding to the missing/corrupted part in the original image) and the one that is not missing/corrupted in the original image. $C(\cdot)$ is the completion network that use missing/corrupted images I and corresponding attention map A_N generated by automatic recognition of input damaged images with attention *LSTM* module. M denotes a label matrix with same sizes as the input image. In the M, the value of an element can only be 0 and 1, label 1 represents the missing/corrupted pix and 0 represents the undamaged. The operation $\otimes$ here is an element-wise multiplication and $\oplus$ denotes an element-wise addition. $\oplus$ in Eq. (3) merges the known context region and the synthesized missing region to obtain the final inpainting result.

In the pre-training phase, the damaged input image I can be generated by Eq. (4), but then you have to turn a black hole white.

$$I = I_{gt} \otimes (1 - M) \tag{4}$$

where I_{gt} is uncorrupted images in training data set such as CelebA and as explained above M is a label matrix which markups holes in images. In the training phase, M can be generated randomly. However, the missing area is filled with 0, i.e. black color now. After that, it should be replaced by 255, i.e. white color in our implementation. It is also common to fill it as the average value of the training image pixels. Anyhow, other areas remain unchanged.

3.4.2 Attentive Reconstruction Loss

To stabilize the training, a mean square error (MSE) loss considering the inpainting region is used. The MSE loss is defined by Eq. (5):

$$L_{mse} = \left\| M \otimes (C(I, A_N) - I_{gt}) \right\| \tag{5}$$

Here M is the label matrix, which can be easily derived from the subtraction of training facial image pairs (the raw face and the corresponding corrupted one). $\|\cdot\|$ denotes a MSE computation.

3.4.3 Attentive LSTM Loss

The attentive LSTM module is hired to automatically detect missing/corrupted parts in the missing/corrupted image so that later completion network can pay more attention to the restoration of missing parts. The loss function in each recurrent block is defined as a mean square error between the output attention map at time step t and the label matrix M corresponding to the input image. Recurrent blocks in the LSTM module is repeatedly for N times. Intuitively, N will be better if it is bigger. In practice, due to the limitation of computer hardware configuration (Our config: 2 Tesla P100), N is set to 3. And at this point, batchsize can only be set to 2 at most.

The N recurrent blocks are used to continuously enhance the position information of the missing part, and the final output attention map will be the most reliable. If the last attention map is limited only, it will be very difficult for training. According to the hypothesis that for attention loss, the earlier the attention map should have less weight, and the larger N is, the greater the confidence of the corresponding attention map. The loss function is expressed as Eq. (6):

$$L_{att} = \sum_{t=1}^{N} \theta^{t} \left\| A_t - M) \right\| / \sqrt[3]{HW} \tag{6}$$

where A_t is the attention map output by the attentive LSTM network at time step t, θ is a hyper-parameter that can be thought of as a fixed weight and its value should be greater than 1 here, θ^t shows that with the increase of time step t, the more importance attach to A_t. H and W refer to the height and width of the input image respectively, which also neutralizes weights greater than one in a sense. Of course, θ here should match the size of the input image. We set N to 3 and θ to 3 here. Obviously, a larger N may produce a better attention map, but it also requires more storage capacity and more powerful computing power.

3.4.4 Adversarial Loss

During the training phase, the adversarial loss causes the reconstructed image to be close to samples in the training set, which is achieved by updating the parameters of the attention LSTM and completion network. At the end of the training, D will predict that the image from $C(\cdot)$ comes from the training set with a high probability. We use the same loss term, just like in the original GANs:

$$L_{adv} = \log(1 - D(I_{rec})) \tag{7}$$

when D is fixed, the goal of training is that G can deceive the existing D. D stands for discriminator module, I_{rec} represents global (I_{global_rec}) or local (I_{local_rec}) reconstruction results. For example, there is a positive example—I_{global_rec} is I_{gt} and I_{local_rec} is the real part corresponding to the missing part in corrupted input images I. At this stage of training, those inputs of negative examples (generated by our model) should make the output of D(·) close to 1. In practice, this original log function is generally not used, but the cross-entropy function or other newer method are used instead.

3.4.5 Weighted Multi-scale Loss

For the weighted multi-scale loss, a loss measure similar to that in [32,49], we extract features from different upsample layers to form maps in different sizes. Using this method, we intend to constrain and refine the training of model parameters through loss of different scales. We define the loss function as:

$$L_{mut} = \sum_{t=1}^{T} \lambda^{T-t} \left\| O_t - label_t) \right\| \tag{8}$$

where O_t indicates the tth output extracted from the upsample layers, and $label_t$ indicates the ground truth that has the same scale as that of O_t. λ^t are the weights for different scales. The closer to the original size, the greater the weight, Namely putting more weight at the larger scale. As shown in the middle of Fig. 4, there are three different scale losses computation that correspond to 1/4, 1/2 and 1 of the original size respectively. So the T here is 3 and we set λ to 0.75.

3.4.6 Semantic Loss

It is easy for traditional GANs to learn texture features, but not easy to learn specific topological structure and geometric features, such as two eyes, a nose and a mouth, and a relatively reasonable position to form a normal person's face. If we want to inpaint images, we should understand what are missing and their contextual information, so is the neural network. Before repair, it is very difficult for neural networks to understand the semantics of missing parts, which is exactly the task of training. Relatively speaking, limiting the semantic loss of its repair results is simple. After the reconstruction, semantic segmentation of all parts of the face can be done using the same network. By reducing the difference of semantic segmentation feature between the two faces of the complement and ground truth, more similar semantic effects can be achieved. The semantic loss can be easily obtained by Eq. (9):

$$L_{sem} = Loss_{softmax}(FaceFarsing(I_{gt}), FaceFarsing(I_{rec})) \tag{9}$$

Here we employ modified Bilateral Segmentation Network (BiSeNet) [43] for face parsing, the state of the art for semantic segmentation recently. A simple element-wise soft-max loss is hired to compute this loss L_{sem}. A more detailed description of the face parsing network will be given in the experimental section.

3.4.7 Summary of Training Objective Functions

With the above definitions of so many losses, we can get final loss functions. The weight and other parameters in our proposed model are updated using back-propagation with the total generative loss:

$$L_g = \alpha L_{adv} + L_{sem} + L_{mut} + L_{att} + L_{mse} \tag{10}$$

where α is weight to balance the effects of L_{adv} and other losses. In practice, α has to be relatively small to constrain the recovered image with input pixels. Here, we set α to 10^{-2}. The following 4 weights are set to 1 by default.

Refer to the normal GANs Eq. (1), our discriminative loss can be expressed as:

$$L_d = -\log(D(I_{gt})) - \log(1 - D(I_{rec})) \tag{11}$$

Here, I_{gt} contains global and local ground truth. I_{rec} includes the synthetic local region and the corresponding global image containing locally generated region. Multiple discriminators need to do cumulative operations. Obviously, minimum values are both required for L_g and L_d.

3.5 Training Algorithmic Description

We train our network effectively by gradually increasing the difficulty level and scale of the network, whose process is scheduled into two stages.

First, we only use the attention LSTM loss L_{att} and MSE loss L_{mse} to train the network, so that we can get rough weights and biases of the attention LSTM module and generator module. Since the proposed model is end-to-end, the LSTM and generator parts are not trained separately.

Then, we train our model with all losses including the generator loss L_g, and global and local adversarial losses L_d. We look forward to optimizing overall parameters of LSTM and generator network through those loss functions. The overall algorithm is shown below:

Algorithm 1 Training procedure of our proposed framework.

```
while $iterations\ t < T_{max}\ train$ do
    Sample a minibatch of images $I_{gt}$ from training data.
    Generate label matrix $M$ with random holes for each image $I_{gt}$ in the minibatch.
    if $t < T_r$ then
        Train attentive LSTM blocks and the completion network with attentive LSTM loss [Eq. (6)] and MSE loss [Eq. (5)].
    else
        Update the completion network and attentive LSTM net with all losses [Eqs. (10) and (11)].
    end if
end while
```

Finally, we can get the reconstructed face image in terms of Eq. (3). In order to eliminate the stitching around missing regions, Poisson blending or other methods can be used for post-processing.

4 Experiments

4.1 Data Set

We use CelebA [28] as an experimental dataset, in which all images cover large pose variations and background diversity. For testing, we remove 2K images from the dataset before training for evaluating our method with two types of corruptions: fixed size and changed position, or fixed position and gradually increasing in size. Completion tasks are very challenging. For the former task, the corrupted hole with fixed size in any position has to be recovered from the surrounding given information. For the latter one, the recovered region in any position must contain semantically correct content, i.e. eyes, nose, mouse and eyebrows etc. on human faces. And more importantly, all the content should be coherent perfectly with surrounding face features.

4.2 Experimental Environment

Software: CentOS 7, cuda 9.0, tensorflow-GPU 1.6 etc.
hardware: 2 computation cards-Tesla P100 16 GB, 1 cpu-xeon e5 etc.
Our model has trained for 60 epochs on celebA dataset, which takes about one mouth. That is to say, it takes more than 10 h for 1 epochs.

4.3 Face Parsing Network

Face parsing is essentially a semantics segmentation of different facial regions, mainly divided into nose, left eye, right eye, upper lip, lower lip, teeth, skin, left brow, right brow, eye glass, left ear, right ear, ear earrings, neck, neck lace, cloth, hair, hat and so on. In our face parsing network, we uses BiSeNet [43] with spatial path and context path in detail, which combines the features of these two paths with feature fusion module. We train this model with CelebAMask-HQ [26], a face semantic segmentation data set corresponding to CelebA-HQ.

We change the size of face images to 128×128 first and then feed them into the parsing network to predict the label for each pixel. There are several parsing results on the

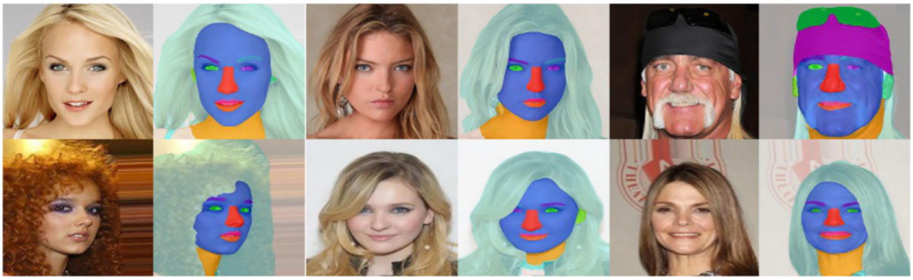

Fig. 5 Examples of our parsing results on CelebAMask-HQ test dataset (top) and CelebA test dataset (bottom). In each pair of original images and corresponding semantic segmentation images, all pixels in face images (left) are classified as specified labels which are shown in different colors (right). Six pairs are shown here

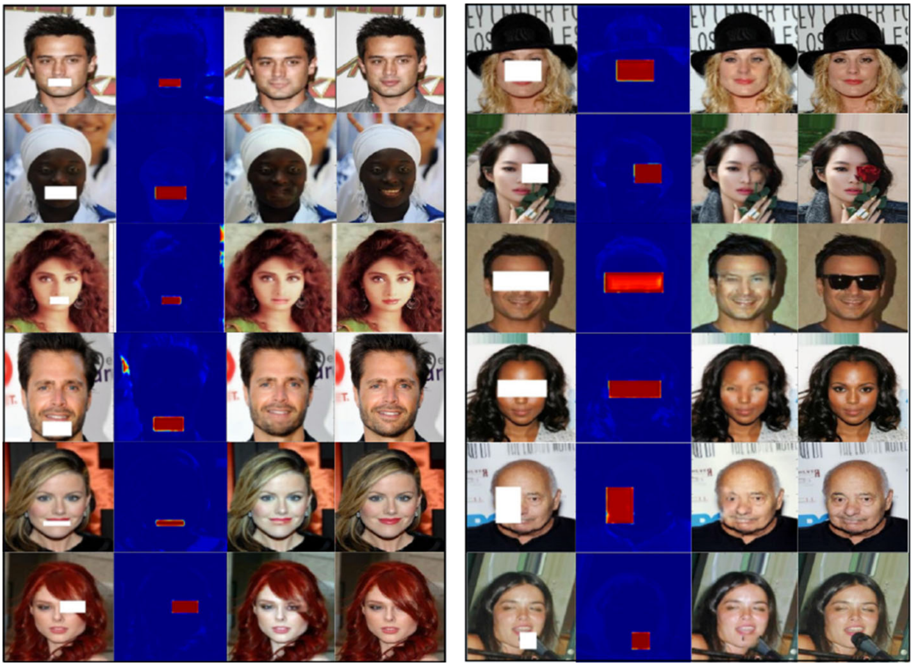

Fig. 6 Sample results fo our model on CelebA. Here are twelve groups of visualization results. Each group consists of four columns, where each column from left to right is represented as the input image to be repaired, the corresponding attention map, the repaired result using our proposed model, the ground truth separately

CelebAMask-HQ and CelebA test images presented in Fig. 5. The parsing network should be pretrained before training, and it remains fixed during training, when we first use the pretrained network on the CelebA set to obtain the parsing results of originally undamaged faces as the ground truth, and compare them with the parsing results on repaired faces. Finally, this parsing loss is added to generator loss as a regularization term, which can be regarded as high-level semantic difference between two facial images.

4.4 Visual Result

Some visual results are shown in Fig. 6, which demonstrates that our method can successfully predict the missing content with high quality. Here is just a small visualization of the results, and the next section will compare our methods qualitatively and quantitatively with other methods.

4.4.1 Comparisons

We compare our method with five methods including:

1. PM: PatchMatch [2], the state-of-the-art non-learning based approach
2. SIIDGM: Semantic Image Inpainting with Deep Generative Models [42]
3. CE: Context Encoders: feature learning by inpainting [31]
4. GLCIC: Globally and Locally Consistent Image Completion [22]
5. GIICA: Generative Image Inpainting with Contextual Attention [48].

A fair comparison with above methods would requires retraining their models on the same data set with the same train-test splits. For fairness, we perform the same scaling on each image in CelebA and retrain them. We evaluate on 2000 images randomly. Unfortunately, subject to hardware resources, all images used in our experiments are resized into 128 × 128. No post-processing has been done for the results obtained by each method.

4.4.2 Qualitative Comparisons

Figure 7 shows the comparisons of six methods on CelebA. It can be seen that:

1. When the missing area is large, PM method always copy semantically incorrect patches to fill holes.
2. The results of SIIDGM method depend heavily on the quality of DCGAN network and the completion results should be pretty good intuitively as long as DCGAN is strong enough. Obviously, DCGAN is not capable enough here. Simple DCGAN structure may be difficult to meet the requirements.
3. There are obvious artifacts and blurry in the generated regions with the CE, GLCIC and GIICA models. It's easy to see visually that the results are not plausible while our model achieves the best performance.

In general, our proposed model can automatically detect holes' location and fuse remote spatial information with global information, which ensures the completion results more plausibel and consistent visually.

4.4.3 Quantitative Comparisons

As mentioned in many studies [10,22,27,42,48], due to the existence of many possible solutions, there is no good numerical metric to evaluate image inpainting results. Nevertheless, we still use PSNR and SSIM to evaluate those inpainting results. Table 1 shows the comparison results. It can be seen that our method outperforms all the other methods on these average measurements on random masks in different size and position.

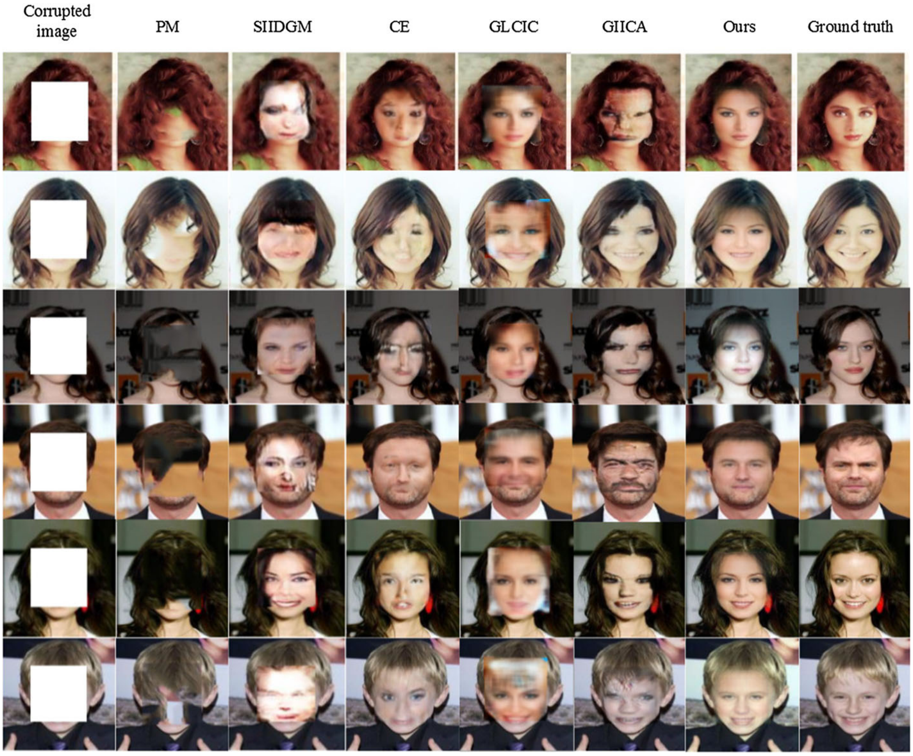

Fig. 7 Comparisons of testing visual results, which of six methods for inpaiting faces with holes (fixed size-64 × 64) in the center. Here are 6 rows and 8 columns. Each row represents different test images and each column represents different results using different methods. Methods represented by each column have been shown above the figure

Table 1 Numerical comparisons of 2000 test images on CelebA

Method	PSNR	SSIM
PM	16.22	0.56
CE	20.02	0.68
SIIDGM	18.22	0.66
GLCIC	21.84	0.75
GIICA	21.46	0.73
Our approach	**22.26**	**0.81**

Our approach is superior to other approaches in terms of PSNR and SSIM

4.5 Discussion

There may be many solutions for image completion, which is good depends mainly on human visual perception and it's impossible to quantify the results. For different painters, results of inpainting the same damaged facial image may be all different. For our method, the artist's mind, namely completion network, depends on the training data set. Our proposed method incorporates a pre-attention map extraction network, namely the attention LSTM module, to identify of missing or damaged regions automatically and guide subsequent completion tasks. Moreover, in our method, we integrate multi-scale losses as regularization terms to achieve better repair results.

Our model can robustly handle holes of any size and location except for the boundary. Compared to other methods, our model is able to grasp the overall situation and the surrounding environment. Such as the sample on the left of Fig. 6 in row 4, the beard was also repaired to a certain extent according to the surroundings. Last but not least, our model testing does not require any manual intervention because of the attention LSTM module that can automatically identify the missing regions for us.

5 Conclusion

In this work we propose a deep generative architecture for face completion. The network is based on a GANs with an encoder–decoder-like model as a generator. Except for the damaged image itself as input to the generator, attention map of the missing part of the image is added as a part of inputs too, which comes from an attentive LSTM module. The proposed model can successfully synthesize semantically valid and visually plausible contents for the missing facial key parts. Both qualitative and quantitative experiments show that our model generates the completion results of high perceptual quality and is quite flexible to handle a variety of holes. Nevertheless, the training model is very time-consuming and hardware resources-consuming (it need to take about a month to train our model), it is necessary to optimize the energy consumption of the model in the future, hoping to use less hardware and get better results in a shorter time.

Acknowledgements This work was supported by the National Key R&D Program of China under Grant 2018YFB1003401, and in part by the National Outstanding Youth Science Program of National Natural Science Foundation of China under Grant 61625202.

References

1. Ba J, Mnih V, Kavukcuoglu K (2014) Multiple object recognition with visual attention. Preprint. arXiv:1412.7755
2. Barnes C, Shechtman E, Finkelstein A, Goldman DB (2009) PatchMatch: a randomized correspondence algorithm for structural image editing. In: ACM transactions on graphics (ToG), vol 28. ACM, New York, p 24
3. Buades A, Coll B, Morel JM (2005) A non-local algorithm for image denoising. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 2. IEEE, New York, pp 60–65
4. Cho K, Courville A, Bengio Y (2015) Describing multimedia content using attention-based encoder–decoder networks. IEEE Trans Multimed 17(11):1875–1886
5. Darabi S, Shechtman E, Barnes C, Goldman DB, Sen P (2012) Image melding: combining inconsistent images using patch-based synthesis. ACM Trans Graph 31(4):82 https://doi.org/10.1145/2185520.2185578
6. Denton EL, Chintala S, Fergus R et al (2015) Deep generative image models using a Laplacian pyramid of adversarial networks. In: Advances in neural information processing systems, pp 1486–1494
7. Desimone R, Duncan J (1995) Neural mechanisms of selective visual attention. Annu Rev Neurosci 18(1):193–222
8. Drori I, Cohen-Or D, Yeshurun H (2003) Fragment-based image completion. In: ACM transactions on graphics (TOG), vol 22. ACM, New York, pp 303–312
9. Duan M, Li K, Li K (2017) An ensemble CNN2ELM for age estimation. IEEE Trans Inf Forensics Secur 13(3):758–772

10. Fawzi A, Samulowitz H, Turaga D, Frossard P (2016) Image inpainting through neural networks hallucinations. In: 2016 IEEE 12th image, video, and multidimensional signal processing workshop (IVMSP). IEEE, New York, pp 1–5
11. Gehring J, Miao Y, Metze F, Waibel A (2013) Extracting deep bottleneck features using stacked autoencoders. IEEE, pp 3377–3381. https://doi.org/10.1109/ICASSP.2013.6638284
12. Gers F (2001) Long short-term memory in recurrent neural networks. PhD thesis, Verlag nicht ermittelbar
13. Goldman B, Shechtman E, Belaunde I (2010) Content-aware fill. https://research.adobe.com/project/content-aware-fill
14. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
15. Guo S, Tan G, Pan H, Chen L, Gao C (2017) Face alignment under occlusion based on local and global feature regression. Multimed Tools Appl 76(6):8677–8694
16. Hays J, Efros AA (2007) Scene completion using millions of photographs. ACM Trans Graph 26(3):4. https://doi.org/10.1145/1276377.1276382
17. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
18. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
19. Hong C, Yu J, Tao D, Wang M (2014) Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. IEEE Trans Ind Electron 62(6):3742–3751
20. Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. IEEE Trans Image Process 24(12):5659–5670
21. Hong C, Yu J, Zhang J, Jin X, Lee KH (2018) Multi-modal face pose estimation with multi-task manifold deep learning. IEEE Trans Ind Inf 15:3952–3961
22. Iizuka S, Simo-Serra E, Ishikawa H (2017) Globally and locally consistent image completion. ACM Trans Graph (ToG) 36(4):107. https://doi.org/10.1145/3072959.3073659
23. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell 11:1254–1259
24. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K (2015) Spatial transformer networks. http://arxiv.org/abs/1506.02025
25. Kataoka Y, Matsubara T, Uehara K (2016) Image generation using generative adversarial networks and attention mechanism. In: 2016 IEEE/ACIS 15th international conference on computer and information science (ICIS). IEEE, New York, pp 1–6
26. Lee CH, Liu Z, Wu L, Luo P (2019) MaskGAN: towards diverse and interactive facial image manipulation. Technical report
27. Li Y, Liu S, Yang J, Yang MH (2017) Generative face completion. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3911–3919
28. Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision, pp 3730–3738
29. Mnih V, Heess N, Graves A et al (2014) Recurrent models of visual attention. In: Advances in neural information processing systems, pp 2204–2212
30. Ouyang X, Zhang X, Ma D, Agam G (2018) Generating image sequence from description with LSTM conditional GAN. In: 2018 24th international conference on pattern recognition (ICPR). IEEE, New York, pp 2456–2461
31. Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA (2016) Context encoders: feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2536–2544
32. Qian R, Tan RT, Yang W, Su J, Liu J (2018) Attentive generative adversarial network for raindrop removal from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2482–2491
33. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint. arXiv:1511.06434
34. Rares A, Reinders MJ, Biemond J (2005) Edge-based image restoration. IEEE Trans Image Process 14(10):1454–1468
35. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7794–7803
36. Wexler Y, Shechtman E, Irani M (2004) Space–time video completion. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004, vol 1. IEEE, New York, p 1
37. Wexler Y, Shechtman E, Irani M (2007) Space–time completion of video. IEEE Trans Pattern Anal Mach Intell 3:463–476

38. Xia C, Zhang H, Gao X (2017) Combining multi-layer integration algorithm with background prior and label propagation for saliency detection. J Vis Commun Image Represent 48:110–121
39. Xie J, Xu L, Chen E (2012) Image denoising and inpainting with deep neural networks. In: Proceedings of the 25th international conference on neural information processing systems, NIPS'12, vol 1. Curran Associates Inc., Lake Tahoe, Nevada, pp 341–349. http://dl.acm.org/citation.cfm?id=2999134.2999173
40. Xingjian S, Chen Z, Wang H, Yeung DY, Wong WK, Woo W (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
41. Xu K, Ba JL, Kiros R, Cho K, Courville A, Salakhutdinov R, Zemel RS, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: 32nd International Conference on Machine Learning, ICML 2015, pp 2048–2057
42. Yeh RA, Chen C, Yian Lim T, Schwing AG, Hasegawa-Johnson M, Do MN (2017) Semantic image inpainting with deep generative models. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5485–5493
43. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Bisenet: bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 325–341
44. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. Preprint. arXiv:1511.07122
45. Yu J, Rui Y, Tao D (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032
46. Yu J, Tao D, Wang M, Rui Y (2014) Learning to rank using user clicks and visual features for image retrieval. IEEE Trans Cybern 45(4):767–779
47. Yu J, Kuang Z, Zhang B, Zhang W, Lin D, Fan J (2018) Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing. IEEE Trans Inf Forensics Secur 13(5):1317–1332
48. Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2018) Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5505–5514
49. Yu J, Zhu C, Zhang J, Huang Q, Tao D (2019) Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2019.2908982
50. Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, Metaxas DN (2017) Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 5907–5915
51. Zhang H, Goodfellow I, Metaxas D, Odena A (2018) Self-attention generative adversarial networks. Preprint. arXiv:1805.08318

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.