11-2022

Delving deep into pixelized face recovery and defense

Zhixuan ZHONG

Yong DU

Yang ZHOU

Jiangzhong CAO

Shengfeng HE
Singapore Management University, shengfenghe@smu.edu.sg

Neurocomputing 513 (2022) 233–246

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Delving deep into pixelized face recovery and defense

Zhixuan Zhong [a], Yong Du [b], Yang Zhou [a], Jiangzhong Cao [c,d,*], Shengfeng He [a,*]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[b] School of Computer Science and Engineering, Ocean University of China, Qingdao, China
[c] School of Information Engineering, Guangdong University of Technology, Guangzhou, China
[d] Guangdong Provincial Key Laboratory of Intellectual Property & Big Data, Guangzhou, China

ARTICLE INFO

Article history:
Received 26 November 2021
Revised 29 August 2022
Accepted 24 September 2022
Available online 27 September 2022
Communicated by Zidong Wang

Keywords:
Face pixelization
Face depixelization
Image recovery

ABSTRACT

Pixelization is arguably one of the most well-adopted deterministic obfuscation techniques for privacy preservation purposes. Although the recovery of pixelized faces is underexplored, the powerful deep neural networks might combat this problem in a data-driven manner. As a consequence, an unbreakable pixelization approach is desired. To achieve this goal, in this paper, we delve into two contradictory problems of unrecoverable pixelization and its counterpart, depixelization, by leveraging the best recovery to strengthen the robustness of the unrecoverable pixelized patterns. In particular, on the offensive end of recovery, we combat the large and continuous nature of pixelized regions by proposing two strategies, 1) an iterative depixelization network that progressively decomposes and predicts the pixelized regions and thus outer results are used to support inner inferences; 2) a dynamic dilated convolution operation is proposed to stride over the redundant identical pixels from the same pixelized region, enabling the network to adaptively extract valid feature representations. We show that our tailored depixelization method significantly outperforms several baselines or inpainting approaches by over 1.0 FID and 2% ID-SIM improvements on CelebA dataset which includes 182,732 human face images, and therefore we study how to defend this advanced recovery and produce unrecoverable pixelized patterns. To balance the visual perception and robustness of pixelization, we propose to generate two types of adversarial examples, pixel-wise and block-wise perturbations, which make different trade-offs between quality and robustness. By deploying our depixelization network in a semi-whitebox setting, our pixelization method can generate imperceptible perturbations while being robust to depixelization.

> "Attack is the secret of defense; defense is the planning of an attack."
>
> — Sun Tzu, The Art of War

1. Introduction

Ubiquitous surveillance cameras and mobile devices capture a massive amount of image data everyday. While this amount of data may be beneficial to applications like smart city, it also captures the sensitive individual information like identities. Image obfuscation, *e.g.*, pixelization or mosaic, is arguably the most widely-used technique to preserve identity privacy, as it can obscure sensitive information while leaving basic image content perceivable (unlike cropping out).However, pixelization procedure still leaves a small amount of information, which keeps a possibility of recovering the pixelized face images to its original appearance. With the development of deep learning, neural networks have been widely used in neuroscience and computer science. LSTM has a powerful ability to analyze and process time series data, and CNN is able to handle a wide variety of image tasks. Large-scale data, the model parameters and the capability of the features paly an important role in deep learning [1,2]. Although no previous attempt has been made to recover a pixelized face, deep neural network is a potential solution due to its ability in "synthesizing" plausible faces [3,4].

To avoid the leak of pixelized identity, in this paper, we aim to develop a pixelization approach that provides rigorous privacy guarantees. Following the spirit of "the best defense is a good offense", we study two contradictory problems, pixelization and

* Corresponding authors at: School of Information Engineering, Guangdong University of Technology, Guangzhou, China; and School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.
E-mail addresses: zxzhong20@gmail.com (Z. Zhong), csyongdu@ouc.edu.cn (Y. Du), matrixGle19@gmail.com (Y. Zhou), cjz510@gdut.edu.cn (J. Cao), hesfe@scut.edu.cn (S. He).

https://doi.org/10.1016/j.neucom.2022.09.141
0925-2312/

depixelization, to evolve an unrecoverable pixelization approach with a better depixelization model (see Fig. 1). For the offensive end of depixelization, the main barrier is the large, continuous, repetitive pixelized regions. The inference of a pixelized face requires building the correlation between blurred blocks and their surroundings, but repetitive pixels may provide redundant and even invalid feature representations. We overcome this barrier from two aspects. First, instead of inferring the pixelized face in one shot, we propose an iterative depixelization network that progressively recovers the original face. More importantly, unlike the progressive models in other applications performed in the image space [5–7], we model this iterative process as a feature recovering problem. In this way, we inject outer predictions as the clues for inner inference at each step, meanwhile avoiding the distortions from multiple inter-space transformations. Cross-stage attention and fusion are proposed to fully propagate the in-process information to every stage. Second, we combat the problem of large repetitive pixelized block by presenting a novel dynamic dilated convolution operation. Traditional convolution operation has a small kernel size that cannot stride out the pixelized block, and the dilated convolution works with a larger receptive field but may skip the important neighboring face regions. Instead, our dynamic dilated convolution operation can adaptively adjust the dilation rate, such that a larger receptive field can be obtained in pixelized regions while a smaller stride is used for face regions. Extensive experiments demonstrate that the proposed depixelization method significantly outperforms several baseline models as well as state-of-the-art inpainting architectures.

Our tailored depixelization model is further deployed as the "attacker" for our pixelization model. We aim to adversarially generate small perturbations on the pixelized regions such that our depixelization model cannot recover the original identity successfully. For this purpose, we devise two types of adversarial examples, pixel-wise and block-wise perturbations, to balance the visual quality and robustness of pixelization. We demonstrate that the proposed pixelization methods are unrecoverable and can be integrated to arbitrary depixelization models.

In summary, our contributions are threefold:

- We present the first attempt to recover a pixelized human face. To this end, we propose an iterative depixelization network with dynamic dilated convolution, both two elements are tailored for recovering large and repetitive pixelized regions. The iterative depixelization network recurrently recovers the pixelized boundaries and then uses the inferred intermediate results as additional information for further recovery.
- We present the first attempt to generate unrecoverable pixelized patterns. Two types of perturbations are proposed and both of them are robust to depixelization.
- We delve deep into both the face depixelization and pixelization. Extensive experiments show that the proposed depixelization and pixelization approaches outperforms several baselines and state-of-the-art architectures by a large margin. We demonstrate the first feasible privacy-preserved pixelization solution.

The remainder of this paper is organized as follows. We review the relevant approaches including image inpainting, image super-resolution and adversarial examples in Section 2. After that, we elaborate the proposed depixelization method named Iterative Depixelization Network in Section 3. And our adversarial pixelization approach including pixel-wise perturbations and block-wise perturbations are described in Section 4. Evaluations from extensive experiments are provided in Section 5, and we conclude the paper in Section 6.

2. Related Work

As we are the first to address the problems of human face depixelization and unrecoverable pixelization,we discuss the most relevant researches in this section, including image inpainting, image super-resolution, and adversarial examples.

2.1. Image Inpainting

Image inpainting aims to recover the missing regions of a given damaged or deteriorating image, which shares a similar objective to depixelization. Recent researches mainly apply deep neural networks to synthesize the missing regions. Context-encoder [8] first apply a conditional GAN [9] to generate large missing regions with exquisite details. Iizuka et al. [10] adopt local and global discriminators to ensure both the local continuity and the global composition of the scene. Liu et al. [11] introduce the partial convolution layer to inpaint irregular missing holes, which classifies pixels as valid/invalid to make better use of the original image information. Yu et al. [12] devise the gated convolution to utilize pixels information with a learnable mask. Liu et al. [13] propose to mutually learn the representations of structure and texture for generating coherent image content. Unlike image inpainting that focuses only on the surrounding pixels, depixelization has a unique emphasis on the blurred information in the pixelized regions, leading to a completely different design principle.

2.2. Image Super-resolution

The purpose of image super-resolution is to reconstruct a corresponding high-resolution image from its low-resolution image. Considering the pixelized faces as low-resolution images, recovering the original identity can be also treated as upsampling with a ultra high scaling factor. A shallow three-layer CNN is firstly proposed by Dong et al. [14] to learn a LR-HR mapping. Kim et al. [15] introduce a 20-layer CNN based on residual learning for utiliz-

(a) Original (b) Pixelized Face (c) Our Depixelization (d) Our Pixelization (e) Depixelized from (d)

Fig. 1. We study two contradictory problems of depixelization and unrecoverable pixelization. We first tailor an iterative depixelization network with dynamic dilated convolution (result shows in (c)), both have a consistent aim to extract valid feature representations over the large and continuous repetitive information of pixelized regions. By taking this advanced recovery model in a semi-whitebox setting, we next propose a robust pixelization approach to defend depixelization (as shown in (d) & (e)).

ing more contextual information. Lim et al. [16] devise a deeper and wider network by stacking modified residual blocks [16]. Furthermore, MemNet [17] and RDN [18], which are based on dense blocks [19], focus on utilizing hierarchical features from convolutional layers to achieve impressive performance. Different with super-resolution which reconstructs the whole image just based on low-resolution information, depixelization utilizes low-resolution information in pixelized regions and high-resolution information around pixelized regions to generate more realistic images. Thus, depixelization could be regarded as a combination of image inpainting and image super-resolution. To the best of our knowledge, we are the first to study the depixelization task.

2.3. Adversarial Examples

Deep neural networks have been demonstrated to be vulnerable to adversarial examples by adding imperceptible perturbations to input images, which can mislead the network to predict wrong results. Previous research on adversarial attacks mainly focus on classification model [20,3,21–23], but paid less attention to generative models such as image-to-image translation task [24]. In addition, a number of adversarial attack strategies are gradient-based [3,22,23,25] or optimization-based [26–28], which need to have white-box access to the architecture and parameters of the model. While for generator-based strategies [29,4], once the network is trained, it could instantly produce adversarial examples without requiring access to the model. We adopt the latter strategy for generating unrecoverable pixelized patterns.

3. Depixelization Approach

3.1. Overview

Fig. 2 show the pipeline of our proposed depixelized method. Inspired by deep learning, we design a nerual network model which is able to restore the pixelized images after training by feeding plenty of ground-truth-pixelized image pairs. It can restore faithful and realistic images from pixelized images without any help of ground-truth during testing. Given a pixelized face input, we propose a depixelization model named Iterative Depixelization Network that could progressively recover a pixelized human face. Since the pixelized regions may always be large and continuous, restoring the entire image in one shot would lead to an ambiguous result due to the lack of valid information. In contrast, our method separates the depixelization process into several stages. At each stage, the network predicts a depixelized boundary of the pixelized area which is then to be shrunk and updated for the next stage. In this way, each outer prediction could provide additional cues for inner inference, facilitating a final realistic and natural reconstruction.

Our model consists of two modules, including a Dynamic Feature Recovering (DFR) Module and a Cross-stage Fusion Module. The former one is used to recurrently repair the boundary of the pixelized area as well as shrinking it at each stage. To reinforce the ability of DFR module in inferring semantic content, we also embed a cross-stage attention mechanism into it. And the latter one is designed to organically merge all the generated intermediate feature maps from DFR module at each recurrence to produce a final result. The pipeline of our depixelization approach is illustrated in Fig. 3. (See Fig. 4).

Except for the framework, we also present a new convolution operation, named dynamic dilated convolution. This operation is proposed to cope with the problem of repetitive pixels in the neighboring pixelized regions. It can adaptively adjust the dilation rate of the convolution, locating the most discriminative features of the pixelized faces.

To train our network, we need to first obtain pixelized/non-pixelized image pairs. Different with other corrupted image inverse problems (*e.g.*, image inpainting), image depixelization is commonly without any indicated mask of the missing or invalid region. In order to provide the hint on the pixelized regions, we synthesize the training samples with the aid of a specific binary mask that labeled the pixelized region and non-pixelized region with 0 and 1. Note that such a mask could be provided by users, or obtained by performing plain grouping/clustering methods on pixelized images. Below we will elaborate the architecture of our depixelization network and the tailored dynamic dilated convolution operation.

3.2. Iterative Depixelization Network

Dynamic Feature Recovering Module. As shown at the top of Fig. 3, the DFR module is our main backbone to recover the pixelized faces. Particularly, we utilize this module for the repeated inferences in different recurrences. Such a module is able to restore semantic content in specific regions of different depixelization subtasks. We first cascade several Dynamic Dilated Convolutional layers (which would be explained detailedly in Section 3.3) to integrate the features of non-pixelized and pixelized regions. With the introduction of such layers, the duplicated information in the pixelized region would be reduced, and the spatial distant valid features in the non-pixelized region can be fully exploited. Meanwhile, these layers could help to mark the recovered area in the

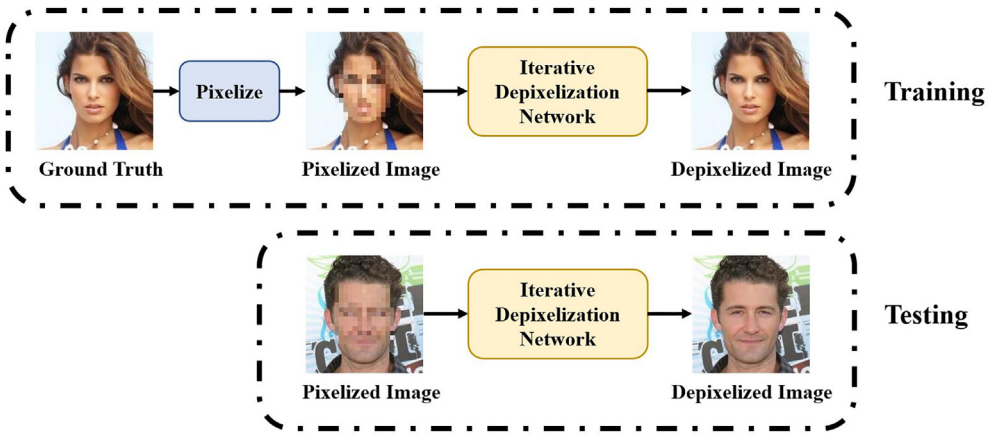


Fig. 2. Overview of our proposed depixelized methods based on deep learning.

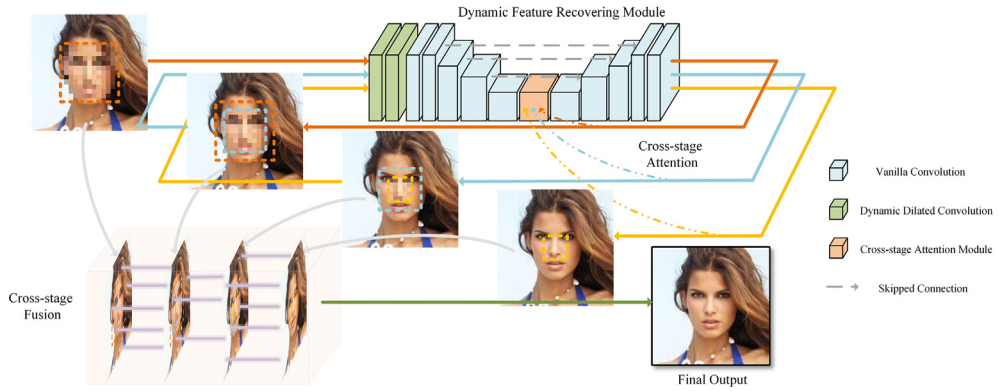


Fig. 3. Overview of our Iterative Depixelization Network. We resolve the large and repetitive nature of the pixelized region by decomposing it intro several sub-regions, such that previous inferences can be used as clues for next stages. A cross-stage attention module is used to inherit stage-specific information, and all side-outputs are integrated by a cross-stage fusion module into the final prediction.

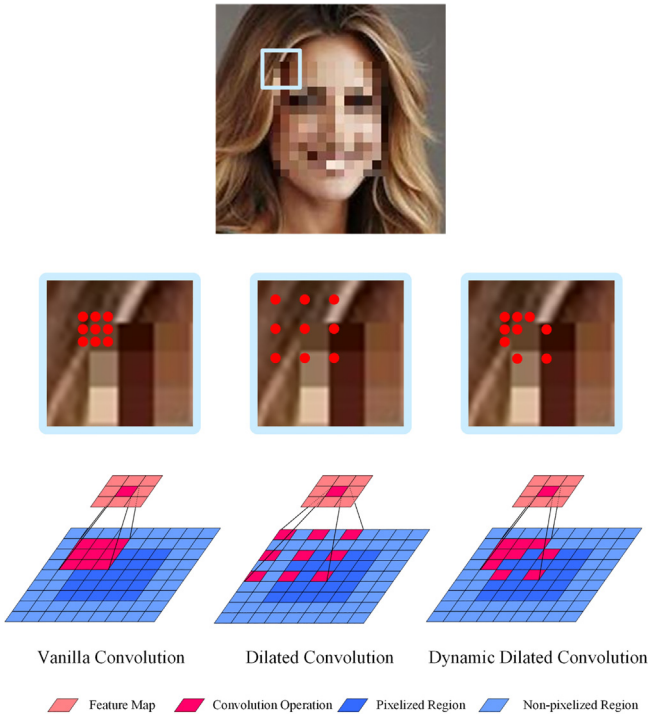


Fig. 4. Illustration of different convolution operations. Due to the large and repetitive pixelized regions, vanilla convolution extracts redundant pixelized information, while dilated convolution skips surrounding face information. Our dynamic dilated convolution adaptively adjusts the dilation rate, optimizing the extracted features from both face and pixelized regions.

current recurrence and update the mask of the pixelized region. After that, we place an encoder-decoder network composed of several convolution layers with skip connection to generate high-quality features in pixelized region for efficiently inference. Finally, the predicted result would be fed into the next recurrence for further recovery.

Cross-stage Attention. Convolutional neural network with filters focus on the local texture details but hardly take into account the global semantic information in image processing task. Especially for large region reconstruction, it is inefficient in capturing features from distant spatial locations. The attention strategy which borrows reasonable features from the known regions to generate patches in unknown regions is a solution to address this issue. However, existing attention mechanisms are unsuitable for our model since they just take effect in individual recurrence.

In order to reconstruct the high-quality feature maps, we present a novel attention module named Cross-stage Attention which adaptively combines the attention scores from every recurrence. The attention applied to patches swapping passes across the whole depixelization process to enhance the consistency among different recovering stages. We follow the attention strategy in [8] to calculate each component of the proposed attention map, and let $score_i$ denotes the attention score map in the ith recurrence, then the attention score map in the next recurrence $score_{i+1}$ is the weighted sum of scores in previous recurrences, which is formulated as follows:

$$score_{i+1} = \lambda_{i+1} \hat{score}_{i+1} + (1 - \lambda_{i+1}) score_i, \tag{1}$$

where λ_{i+1} is a learnable parameter, and $\hat{score}_{i+1}$ indicates the corresponding attention map of the current recurrence before the weighted average operation.

Cross-stage Fusion Module. DFR module would generate several groups of intermediate feature maps. To produce explicit depixelized results, just utilizing the features at the last recurrence only would lead to a gradient vanishing problem. On the other hand, directly summing up all the features for reconstruction would make the result ambiguous. Therefore, we design a Cross-stage Fusion module to adaptively combine all intermediate feature maps. The feature values in pixelized regions of each group of feature maps have no contribution to the depixelized result and should be firstly discarded before feeding into this module. Then the feature maps that only contain valid values are smoothly merged for further recovery. Specifically, let F_i denotes the features generated by DFR module and M_i denotes the corresponding binary mask which identifies the pixelized and non-pixelized regions in the ith recurrence, then the final features $\overline{F}$ for reconstruction are calculated by averaging all the intermediate feature maps as follows:

$$\overline{F} = \frac{\sum_{i=1}^{N} F_i \odot M_i}{\sum_{i=1}^{N} M_i}, \tag{2}$$

where $\odot$ denotes element-wise multiplication, and N denotes the number of recurrences. The division operation here is element-wise division rather than matrix division.

3.3. Dynamic Dilated Convolution

In pixelized images, the embed information in the non-pixelized regions is very rich, while that is extremely sparse in

the pixelized areas. This implies that it is not suitable to use vanilla convolution operation to capture features, since the values in pixelized and non-pixelized regions should not be treated equally. Besides, vanilla convolution is usually limited in the size of its receptive field, such that could not fully consider spatial dependencies.

As for dilated convolution [30], which expands receptive field by skipping pixels in feature maps without increasing the filter size, focuses on excavating global structural information of the whole features. Even such a strategy could be helpful to integrate the sparse information in pixelized regions, it ignores certain values in the non-pixelized areas, resulting in a loss of feature details.

To tackle such problems, we propose a novel convolution operation named Dynamic Dilated Convolution. It consists of a vanilla convolution operation over non-pixelized regions and a dilated convolution operation over pixelized regions, and thus integrates rich non-pixelized representations and sparse pixelized information to depixelize pixelized images with exquisite details. Specifically, the dynamic dilated convolution operation in the $(i+1)$th recurrence can be formulated as follows:

$$X\prime = W^{\mathrm{T}}_{vanilla}(X \odot S_{i+1}) + W^{\mathrm{T}}_{dilated}(X \odot (1 - S_{i+1})), \tag{3}$$

where X and $X\prime$ respectively denote the input and output features in the current convolution sliding window, *e.g.*, X is a 3×3 feature map if the kernel size of the convolution operation is 3×3, $W_{vanilla}$ and $W_{dilated}$ respectively indicate the vanilla part and the dilated part of the convolutional filter. The superscript T denotes the matrix transposition operation. And S_{i+1} is the binary mask of X, which is sampled from the mask M_i according to the location of X in the whole feature map, with 1 for identifying non-pixelized regions and 0 for pixelized ones.

At the same time, we will update the mask M_{i+1} as follows: if there is any non-pixelized pixel in the sliding window, then we mark all pixels in this window to be non-pixelized, that is

$$m\prime = \begin{cases} 1, & \text{if sum}(S_{i+1}) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $m\prime$ denotes the values inside the corresponding window of the mask M_{i+1}. In this way, the mask M will be shrunk after each recurrence and finally we can recover the whole pixelized image.

3.4. Loss Function

The goal of our depixelization network is to recover pixelized images in both local details as well as global plausibility. Thus, we consider to optimize our depixelized model from two aspects, that is, a pixel-wise accuracy and a perceptual consistency. Let H, W, and C respectively denote the height, weight, and channel size of the corresponding image or feature map, I_{GT} denotes the ground truth image while I_{pred} indicates the predicted depixelized image by our depixelization network, M denotes the binary mask of input image identifying non-pixelized and pixelized regions with 1 and 0 respectively. We first define the losses of a pixel-wise accuracy respectively in pixelized and non-pixelized regions, which are formulated as follows:

$$\mathscr{L}_{pixelized} = \frac{1}{HWC}||(I_{GT} - I_{pred}) \odot (1 - M)||_1, \tag{5}$$

$$\mathscr{L}_{non-pixelized} = \frac{1}{HWC}||(I_{GT} - I_{pred}) \odot M||_1. \tag{6}$$

Next, the perceptual loss [31] from an ImageNet-pretrained VGG16 [32] model is used. It compares the features generated by VGG16 of the ground truth images with that of the predicted images, so that the high-level information (texture, content, and global structure) can be maintained. The perceptual loss can be expressed as follows:

$$\mathscr{L}_{perceptual} = \sum_{i=1}^{3} \frac{1}{H_i W_i C_i}||\phi_i^{GT} - \phi_i^{pred}||_1, \tag{7}$$

where ϕ_i^* denotes the output feature maps of the ith selected pooling layer (in our case they are pool 1, pool 2 and pool 3 layers) in the fixed VGG16 when given I_{GT} or I_{pred}.

We also adopt the style loss which can be written as:

$$\mathscr{L}_{style} = \sum_{i=1}^{3} \frac{1}{C_i \times C_i}||\frac{1}{H_i W_i C_i}\left(\phi_i^{GT}(\phi_i^{GT})^{\mathrm{T}} - \phi_i^{pred}\left(\phi_i^{pred}\right)^{\mathrm{T}}\right)||_1. \tag{8}$$

The total variation (TV) loss which aims to smooth images is the final term of our objective, that is:

$$\mathscr{L}_{TV} = \frac{1}{HWC}\sum_{h,w,c}||I_{pred}^{h,w+1,c} - I_{pred}^{h,w,c}||_1 + ||I_{pred}^{h+1,w,c} - I_{pred}^{h,w,c}||_1. \tag{9}$$

where $I_{pred}^{h,w,c}$ denotes the pixel value at the location of (h, w) in the cth channel.

In summary, the total objective of our depixelization model can be formulated as follows:

$$\begin{aligned}\mathscr{L}_{total} = {} & \lambda_{pixelized}\mathscr{L}_{pixelized} + \lambda_{non-pixelized}\mathscr{L}_{non-pixelized} \\ & + \lambda_{perceptual}\mathscr{L}_{perceptual} + \lambda_{style}\mathscr{L}_{style} + \lambda_{TV}\mathscr{L}_{TV}.\end{aligned} \tag{10}$$

where λ_* is the weight with respect to different loss terms.

4. Unrecoverable Pixelization

4.1. Problem Definition

Most of adversarial attack researches focus on image classification task [3,23,26,29], which require the softmax probabilities corresponding to the confidence of classifying images to each label, such that it can mislead the model to predict incorrect results. Different from the classification task, there is no quantitative layer such as the softmax layer to directly affect the outcome quality of image generation task. Considering that our goal is to pixelize a face image so that it cannot be recovered by the recovery model, we aim to add imperceptible noise for this purpose. As a consequence, we define the distance between the model outcome and its ground truth pixelized image as the adversarial attack objective function.

Let $I_{pixelized}$ be the pixelized face image of the ground truth image I_{GT}. Given $\mathscr{G}$ is the pixelized face recovery network, we can derive $I_{pred} = \mathscr{G}(I_{pixelized})$, where I_{pred} is similar with I_{GT}. In the adversarial setting, we aim to add some imperceptible noise δ to $I_{pixelized}$ for misleading $\mathscr{G}$ to produce images similar with the original pixelized face images visually. The misleading target can be varied, we decide to maintain the original pixelized appearance due to the clear and direct objective that can be easily trained. And a pixel-wise constraint $\mathscr{L}_1$ is imposed on the misleading prediction $I_{pred}\prime$, which is formulated as follows:

$$\mathscr{L}_1 = ||I_{pred}\prime - I_{pixelized}||_1, \tag{11}$$

where $I_{pred}\prime = \mathscr{G}(I_{pixelized} + \delta)$.

It is easy to apply gradient-based adversarial attack strategies such as FGSM [3] or PGD [23] to generate adversarial examples, but most of them are white-box attack which need to access the architecture and parameters of the model all the time. Instead, we aim to propose a more practical pixelization method that can be widely applied to images or videos to help protecting sensitive information without any constraint on the recovery model. Therefore, a generator-based adversarial attack strategy is proposed in a

semi-whitebox setting. It is able to produce adversarial examples in the black-box setting in testing phase once the model is trained in the white-box setting. In this setting, it is able to attack the recovery network with a high success rate and adapt to other recovery models. In addition, it can accelerate the producing process and generate more natural and undetectable adversarial examples. Let $\mathscr{F}$ denotes the adversarial generator and the process of producing adversarial examples $I_{pixelized}\prime$ can be written as

$$\begin{aligned} \delta &= \mathscr{F}\left(I_{pixelized}\right), \\ I_{pixelized}\prime &= I_{pixelized} + \delta. \end{aligned} \tag{12}$$

To guarantee an imperceptible difference between adversarial examples and the corresponding pixelized images, we use another constraint $\mathscr{L}_2$ for the adversarial examples, that is

$$\mathscr{L}_2 = ||I_{pixelized}\prime - I_{pixelized}||_1. \tag{13}$$

And the whole objective for the generation of unrecoverable pixelized images is thus defined as follows:

$$\mathscr{L}_{adv} = \lambda_1 \mathscr{L}_1 + \lambda_2 \mathscr{L}_2. \tag{14}$$

To balance the pixelization effect and robustness, we explore two types of perturbations in the following.

4.2. Pixel-wise Perturbation

In order to maintain the functionality to preserve identity privacy of pixelization, the adversarial examples should be as close to the original pixelized face image as possible. It is natural to generate imperceptible noises to the pixelized regions. It will seize the weaknesses of the model and concentrate on sensitive regions so as to destroy the recovery effects.

We design a convolutional encoder-decoder network to translate pixelized face images into perturbation images, and then merge perturbation images and pixelized face images to the get adversarial examples (see Fig. 5a). During training, the pixelized recovery network is fixed and placed after the adversarial generator, taking the adversarial examples as input and inferring desired results (pixelized face images) to guide the process of generating adversarial perturbations with the back propagation gradient. Although pixel-wise perturbation can be easily trained to fool the network, it is vulnerable to smoothing algorithms such as Gaussian filter which is able to erase these tiny noises. We nextly explore another alternative.

4.3. Block-wise Perturbation

Different from pixel-wise pixelization, block-wise pixelization aims to generate block-wise noise of the same size as pixelized grids. By adding the perturbation in block level to pixelized images, the colors of some essential pixelized grids are changed slightly. The block perturbation is more harmonious in visual and is hard to be eliminated by smoothing algorithms.

Similar with the pixel-wise pixelization network, we propose a convolutional encoder-decoder to generate block-wise adversarial examples (see Fig. 5b). In order to craft perturbations in block-level, we insert two resize layers, one is down-sampling layer and the other is up-sampling layer, into the adversarial generator to control the size of feature maps. In this way, the learned perturbations are large enough to integrate into the input pixelized regions.

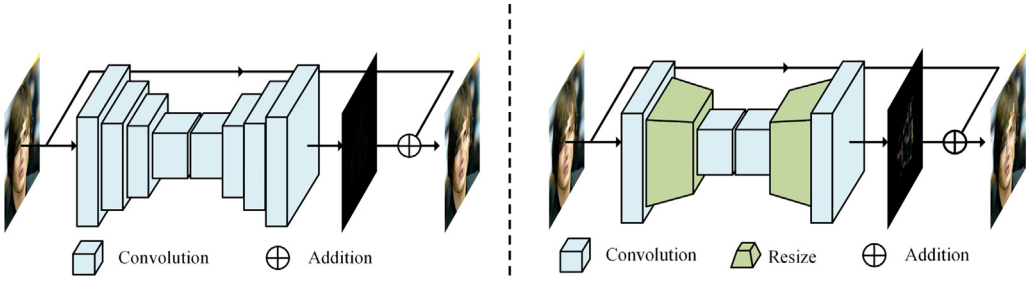
5. Experimental Evaluations

We implement both the depixelization and pixelization algorithms in Pytorch [33] on a PC with an Nvidia GeForce RTX 2080Ti GPU. Both two algorithms can be performed in real-time, and depixelization takes 25 ms to recover a 256×256 image, and 2.8 ms to add perturbations.

Evaluation Settings. To evaluate our two models, we use the CelebA dataset which contains 182,732 human face images. We follow the original splitting, in which 162,770 images are used for training and 19,962 images are used for testing. Regarding the mask, we use a face detector and pixelize the face with a 128*128 binary mask for data generation, and provide a ground-truth of it. We get the pixelized data by downsampling the face regions of images to specific size and then upsampling it to the original size with nearest neighbor algorithm. The specific size is controlled by the *pixelization ratio*, calculated as *specific size = original size* $\times$ *pixelization ratio*. We empirically define all the hyper-parameters, we set $\lambda_{pixelized} = 6, \lambda_{unpixelized} = 1, \lambda_{perceptual} = 0.05, \lambda_{style} = 120$, and $\lambda_{tv} = 0.1$ in the Iterative Depixelization Network. Regarding the case of pixel-wise perturbation, we set $\lambda_1 = 1$ and $\lambda_2 = 3$, while for the case of block-wise perturbation, we set $\lambda_1 = 1$ and $\lambda_2 = 8$.

Both models are optimized by the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The Iterative Depixelization Network is firstly trained for 8 epochs with a learning rate of 0.0002, and then finetuned for 2 epochs with a learning rate of 0.00005. The entire training procedure takes 3 days on a PC with an Nvidia GeForce RTX 2080Ti GPU. As for the perturbation models, with the learning rate 0.0001, pixel-wise perturbation is trained for 2 epochs while block-wise perturbation is trained for 4 epochs.

Evaluation Metrics. To evaluate whether a pixelized face is successfully recovered or defended, we use two perceptual-based metrics. Fréchet Inception Distance (FID) [34] computes the Wasserstein-2 distance between the distribution of GT and output images. Identity similarity (ID-SIM) is computed to examine whether the recovered faces can be recognized as the same identity with the ground truth. We adopt a state-of-the-art face recognizer [35] for computing this score. We also use two traditional



(a) Pixel-wise Network (b) Block-wise Network

Fig. 5. Architectures used to generate two types of perturbations.

metrics, PSNR and SSIM [36]. Although these two metrics cannot measure semantic similarity, and PSNR even biases to blurry synthesis, they are reported here to indicate the numerical gap between our method and the perfect recovery.

5.1. Evaluations on Depixelization

5.1.1. Ablation Studies

We first evaluate the effect of pixelization settings in pixelized images, as well as the contributions of different components of the proposed depixelization model.

Effect of Pixelization Ratio. Pixelization results depend on the pixelization ratio (*i.e.*, block size) defined in advanced. It is of great importance to have a general depixelization approach that can handle arbitrary pixelization ratios in practise. Here we examine different strategies for training our depixelization network with respect to different pixelization ratios. Three strategies are used to generate the training images. (1) *Single*, we use a fixed ratio of 0.1 to produce training images. (2) *Multiple*, we randomly select one of three ratios (0.08, 0.1, 0.12) to generate the training data. (3) *Infinite*, we select a random ratio ranging from 0.08 to 0.12. All these strategies are tested on five ratios, and they are trained with the same amount of data.

Table 1 shows the quantitative results of these three strategies. For all strategies, images processed by smaller pixelization ratios are more difficult to depixelize since there is not enough information (larger block sizes) to recover to their original appearances. Training the network with a fixed ratio of 0.1 (*i.e.*, "*Single*" in the table) cannot generalize to the other ratios. Compared with "*Multiple*", the model trained with the infinite strategy is more robust to pixelization ratio variations and thus more practical. We select infinite strategy in all the other experiments.

Effect of Iterative Recovery Strategy. It is worthy to be noted that our depixelization approach is based on an iterative recovery strategy, of which the recurrence number would influence the depixelization performance. As a result, we examine our iterative recovery strategy by comparisons of 1-stage, 3-stage and 6-stage predictions, *i.e.*, the recurrence number is set to 1, 3 and 6, respectively. Quantitative results are given in Table 2. We can see our final model achieves the best performance, meanwhile we can see by decomposing the predictions into multiple stages, the depixelization quality is better than one-shot prediction. Due to the size of pixelized regions, the performance is converged to 6-stage (our final model). The first row of Fig. 6 shows the qualitative comparison, and we can see that the results generated by 1-stage and 3-stage are semantically ambiguous especially in center regions. This is because the correlations between known and center regions are too weak for few-stage recovery.

Effect of Dynamic Dilated Convolution. To evaluate the effect of our dynamic dilated convolution in depixelization, we replace it with vanilla convolution and dilated convolution. The results are shown in Table 2. We can see that dynamic dilated convolution has the best FID and ID-SIM. Although PSNR and SSIM are similar for three variants, the bottom row of Fig. 6 shows that vanilla convolution tends to blur the face due to the extracted repetitive information, and dilated convolution cannot recover the face regions correctly as it strides over the neighboring pixels.

Effect of Cross-stage Attention. As shown in the third row of Fig. 6, when we remove Cross-stage Attention mechanism or just replace it with ordinary attention module which is widely used in image processing without any modification, the results are unnatural and severe artifacts can be easily found. Due to the tailored design of Cross-stage Attention mechanism which captures relations among different recurrences, it can generate more realistic and semantically consistent results.

Effect of Cross-stage Fusion Module. We compare our Cross-stage Fusion module with other feature fusion variants in Fig. .6, *i.e.*, generating the final feature map from DFR module without any feature fusion strategy, or averaging all the intermediate features. The former one neglects features produced by easiler recurrences, and the latter one combines redundant information from all recurrences. Both results are blurry and lack of details, while our Cross-stage Fusion module is capable of producing faithful and clear results with plausible details.

Effect of Noises and Contrast. To verify the robustness of our proposed method to various input pixelized images with different modifications such as noises or contrast, we conduct experiments by adding random noises to images and modifying images with different gamma curves. As shown in Fig. 7 and Table 3, the performance of our method is insensitive to noises and contrast of input images.

5.1.2. Comparisons with SOTA Architectures

As discussed in Section 2, we are the first to study depixelization problem and there is no other pixelized face recovery approach. Hence, we tend to compare our depixelization approach with inpainting methods. In addition, we try our best to modify these inpainting methods according to the input pixelized images so that they can be regarded as depixelization approaches to some extent. Eight state-of-the-art inpainting architectures, FFI [12], CRA [37], PConv [11], PRVS [38], RN [39], MADF [40], LGNet [41], and DSNet [42] methods and their corresponding modified depixelization-adapted versions are compared in here.

Comparing with Original Inpainting Models. We first directly adopt the original architectures of these competitors. To remove irrelevant affecting factor of data differences, we finetune all these

Table 1
Evaluation on three training strategies.

	Ratio	FID ↓	ID-SIM ↑	PSNR ↑	SSIM ↑
Single	0.08	1.3282	0.37	27.81	0.8967
	0.09	0.8143	0.46	29.32	0.9148
	0.10	0.6803	0.52	30.82	0.9298
	0.11	0.6511	0.51	30.48	0.9268
	0.12	0.7978	0.53	30.93	0.9303
Multiple	0.08	0.7845	0.45	30.07	0.9233
	0.09	0.7753	0.48	30.18	0.9236
	0.10	0.7149	0.52	30.96	0.9311
	0.11	0.6819	0.55	31.00	0.9320
	0.12	0.6769	0.59	31.70	0.9370
Infinite	0.08	0.7807	0.45	30.09	0.9238
	0.09	0.7429	0.50	30.66	0.9283
	0.10	0.6840	0.53	31.00	0.9317
	0.11	0.6504	0.57	31.54	0.9359
	0.12	0.6375	0.59	31.67	0.9369

Table 2
Evaluation on our proposed components.

	FID ↓	ID-SIM ↑	PSNR ↑	SSIM ↑
1-stage Depixelization	0.6997	0.47	30.24	0.9090
3-stage Depixelization	0.6894	0.50	30.61	0.9216
Vanilla Conv	0.7106	0.49	30.64	0.9278
Dilated Conv	0.6962	0.48	30.55	0.9292
w/o Attention	0.7511	0.46	29.65	0.9057
Ordinary Attention	0.6913	0.48	30.27	0.9195
w/o Fusion	0.7364	0.48	30.08	0.9136
Average Fusion	0.6907	0.49	30.48	0.9262
Ours (6-stage + Dynamic Dilated Conv + Cross-stage Attention + Cross-stage Fusion)	**0.6803**	**0.52**	**30.82**	**0.9298**

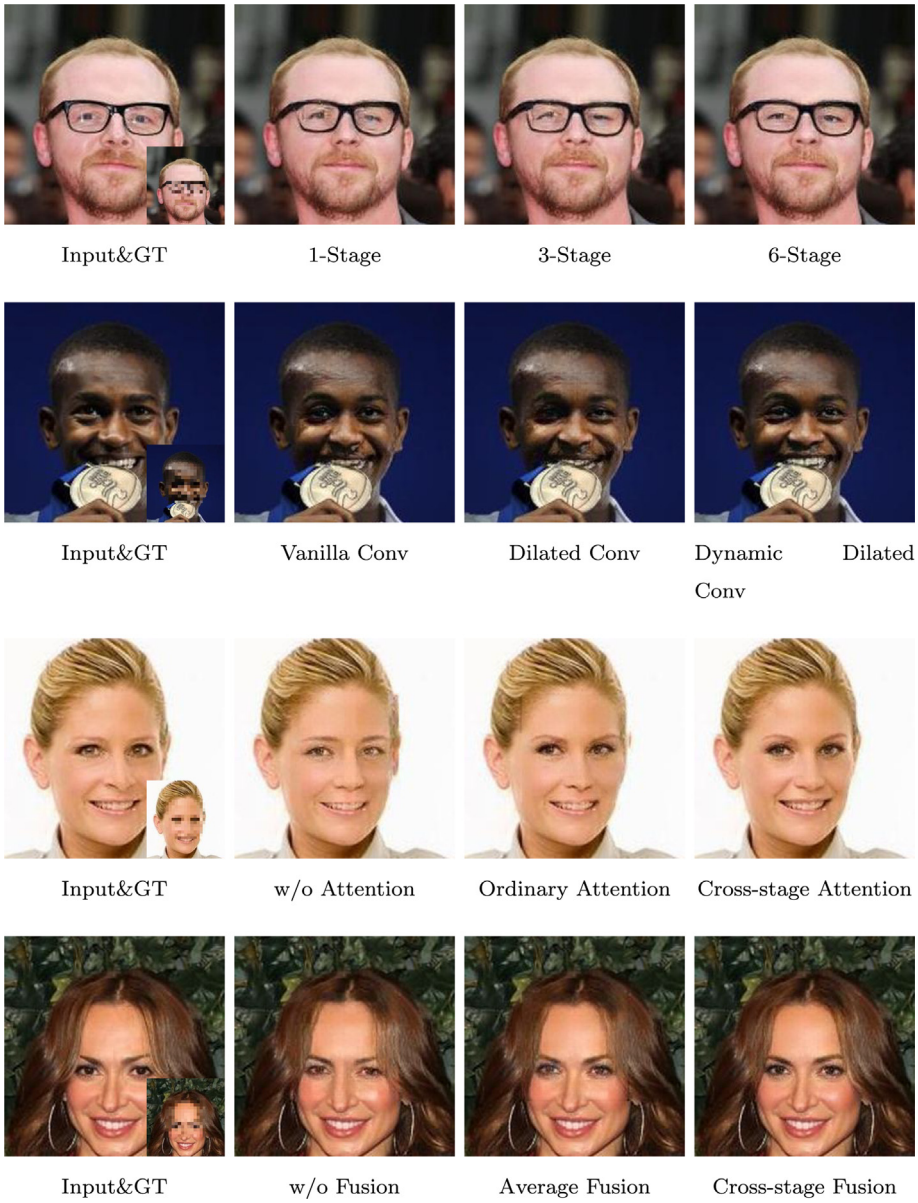


Fig. 6. Qualitative comparison with respect to different variants of our model.

Fig. 7. Qualitative comparison with respect to different modifications of input images.

Table 3
Evaluation on images with different modifications.

	SNR ↑	PSNR ↑	SSIM ↑
w/ Noises	4.4209	29.47	0.9125
w/ Gamma 1.4	4.4376	30.02	0.9216
w/ Gamma 1/1.4	4.4287	29.86	0.9189
w/o Modification	**4.4754**	**30.82**	**0.9298**

models using our face depixelization data. Quantitative and qualitative evaluations (top-6 performers are compared) are shown in Table 4 and Fig. 8, respectively. We can see that all these models perform badly in depixelization. The reason is that all these inpainting models are designed to handle empty user defined holes in the images, which has a completely different emphasis to depixelization. As a consequence, all these models adopt partial convolution operation to prevent extracting meaningless features from the holes. Neglecting the critical information in the pixelized regions ruin the final predictions.

Comparing with Depixelization-adapted Versions. To make these competitors adapt depixelization task, we simply replace all the partial convolution layers with vanilla convolution layers, yielding eight new versions of these methods, FFI++, CRA++, PConv++, PRVS++, RN++, MADF++, LGNet++ and DSNet++. As can be seen in Table 4, by considering pixelized regions, these eight new versions achieve much better performance than the original ones. However, similar to the observation in our ablation study, using vanilla convolution results in blurry faces, and therefore metrics FID and ID-SIM are much lower than ours. This situation can be further observed in Fig. 9 (top-6 performers are compared) that our network generates more semantic and consistent results with exquisite details.

5.2. Evaluations on Pixelization

Given a properly trained depixelization model, we adopt it in the semi-whitebox setting for generating unrecoverable patterns. We examine four different depixelization methods to demonstrate the effectiveness and adaptiveness of our pixelization approach. Each of the method is trained to generate both the pixel-wise and block-wise perturbations, yielding eight pixelization models in total.

Table 5 shows the results of eight pixelization models. We measure the generated adversarial examples as well as the recovered results using PSNR. For both perturbations, we can see that the

Table 4
Quantitative comparison with state-of-the-art inpainting models and their depixelization-adapted versions.

Methods	FID ↓	ID-SIM ↑	PSNR ↑	SSIM ↑
FFI [12]	2.9124	0.25	24.36	0.8657
CRA [37]	2.8535	0.23	24.95	0.8726
PConv [11]	2.6842	0.22	24.89	0.8766
PRVS [38]	2.6453	0.24	25.80	0.8891
RN [39]	3.6743	0.22	25.32	0.8825
MADF [40]	5.6695	0.17	25.81	0.8945
LGNet [41]	2.6378	0.23	25.54	0.8821
DSNet [42]	2.6126	0.24	25.57	0.8982
FFI++ [12]	2.1227	0.35	27.85	0.9057
CRA++ [37]	1.9803	0.37	27.97	0.9005
PConv++ [11]	2.0019	0.37	28.01	0.9012
PRVS++ [38]	1.2809	0.50	30.43	0.9277
RN++ [39]	1.7011	0.49	30.35	0.9266
MADF++ [40]	4.5025	0.19	26.73	0.8949
LGNet++ [41]	2.1057	0.36	28.95	0.9183
DSNet++ [42]	1.7967	0.38	29.68	0.9206
Ours	**0.6803**	**0.52**	**30.83**	**0.9298**

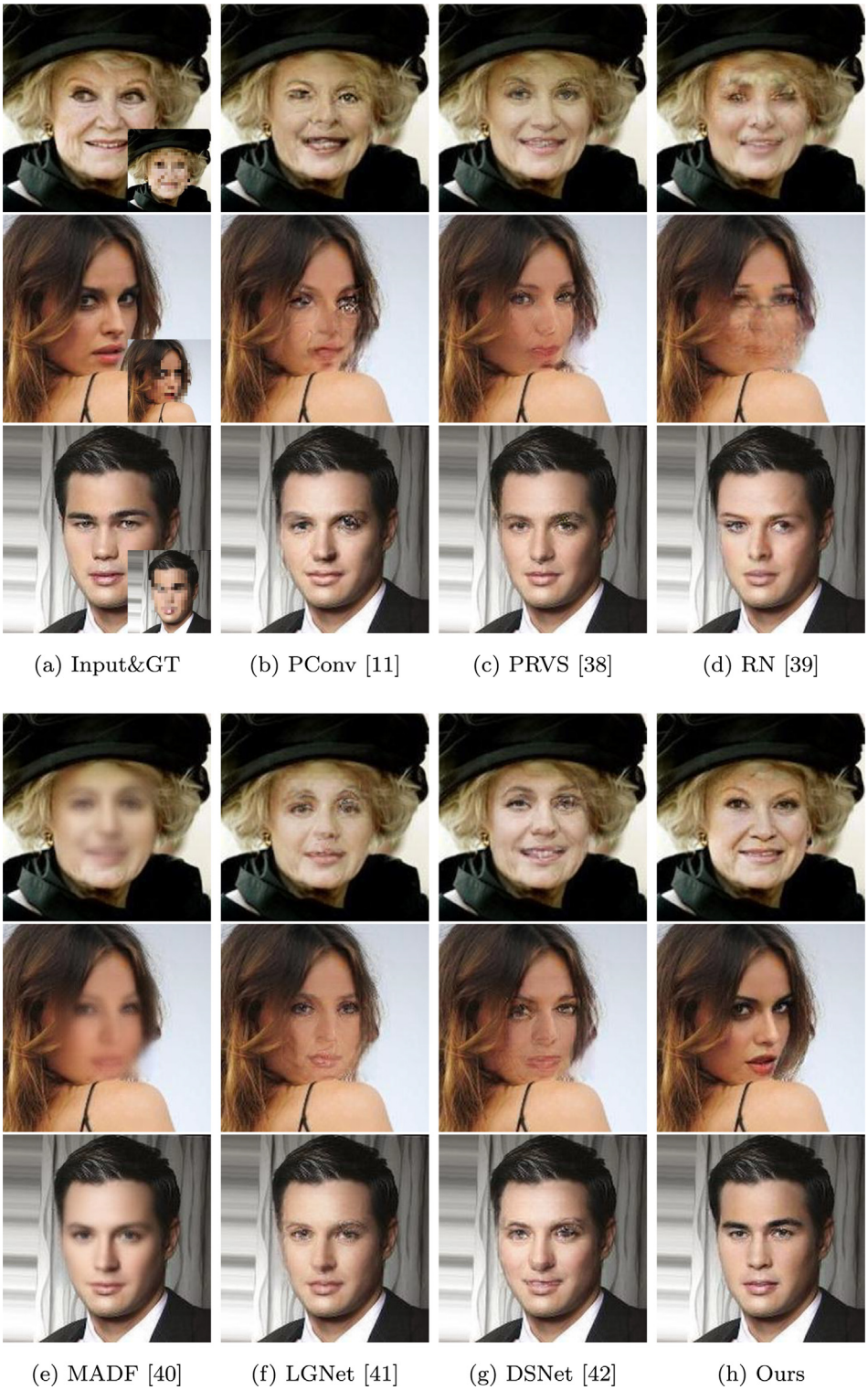


Fig. 8. Qualitative comparison with six state-of-the-art inpainting models.

added perturbations affect the original perception to some extent for all the four models. After recovery, an interesting finding is that all PSNR values increase. This is because our designed adversarial constraint misleads the depixelization network to generate the original pixelized patterns. In other words, the added perturbations force the depixelization network to perform "denoising" and "refinement". Overall, our proposed pixelization strategy works for all the tested models.

Fig. 10 presents the adversarial examples and corresponding recovery results. In terms of pixelization quality, pixel-wise perturbations are small noises, but viewers may discover they are not traditional pixelization. On the other hand, block-wise perturbations

Fig. 9. Qualitative comparison with six state-of-the-art corresponding depixelization-adapted versions of inpainting models.

affect the entire color of the block, providing a more plausible pixelization result. In terms of recovery quality, all the results remain blocky faces without leaking the identity, demonstrating the effectiveness of our pixelization strategy. We also find that the patterns generated from different models vary and therefore they are model-specific. This is a common limitation of white-box and semi-white box adversarial attack, in which the learned adversarial examples cannot generalize to other models.

Table 5
Fidelity comparisons on the adversarial examples and recoveries in terms of PSNR. All PSNR values are compared to the original pixelized images, in which 'Adv' indicates the imperceptible level of added perturbations, and 'Recovery' reveals the robustness of pixelization.

Methods	Pixel-wise		Block-wise	
	Adv	Recovery	Adv	Recovery
PConv++ [11]	31.5868	32.4333	28.6230	30.5142
PRVS++ [38]	32.3347	34.7163	**32.8720**	33.7512
RN++ [39]	27.8219	30.5830	31.3127	26.9752
Ours	**35.5628**	**38.4611**	30.7664	**35.3640**

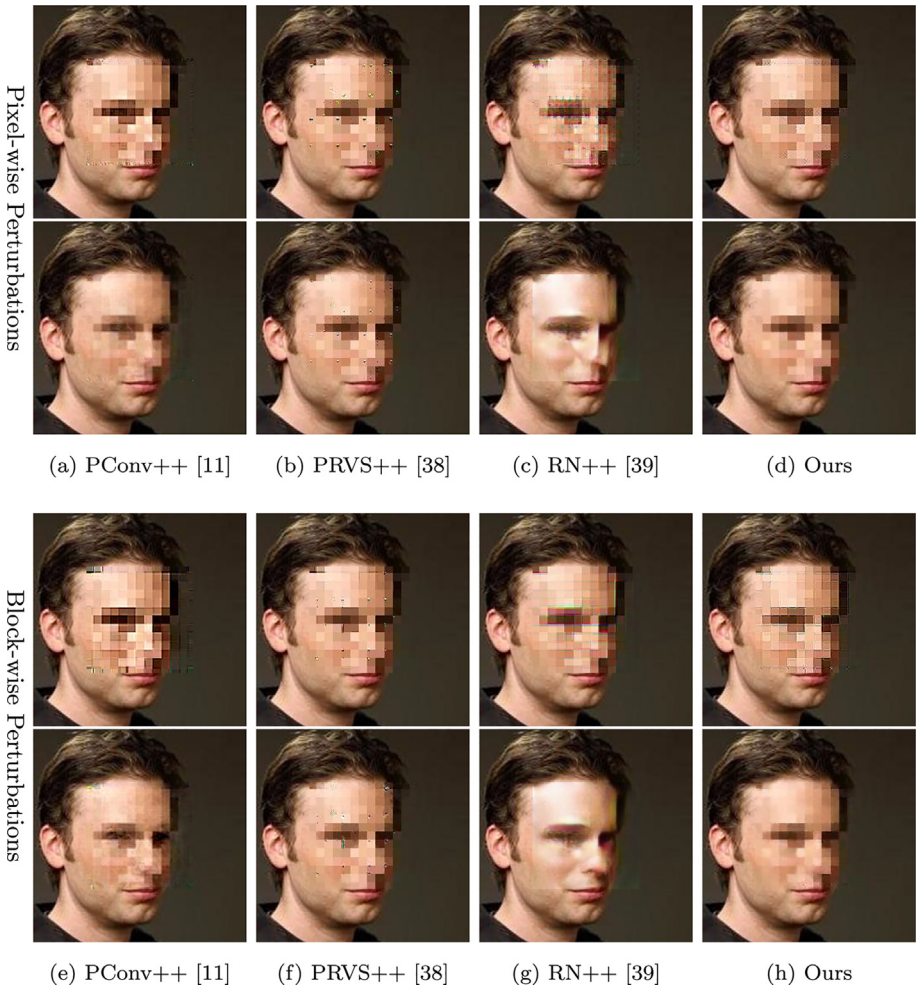


Fig. 10. Recovery comparisons with respect to different pixelization models. Upper row depicts the pixelized results, and the bottom row shows the recovered ones.

6. Conclusion

In this paper, we address the important privacy preservation problem of pixelized faces. To this end, we study two contradictory problems of depixelization and unrecoverable pixelization. We present the first solutions for these two problems. To effectively depixelize the large and repetitive pixelized regions, we accordingly propose an iterative depixelization network with dynamic dilated convolution, both two techniques combat the ambiguity of the large pixel blocks. Second, under the semi-whitebox adversarial attack setting, our adversarial model is used to learn the pixel-wise and block-wise perturbations for defending the recovering process. Extensive experiments demonstrate the effectiveness of both models, yielding the first feasible unrecoverable pixelization approach.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

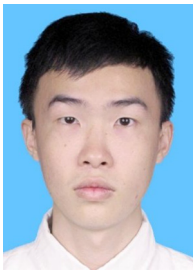
Acknowledgement

This project is supported by the National Natural Science Foundation of China (No. 61972162); Guangdong International Science and Technology Cooperation Project (No. 2021A0505030009); Guangdong Natural Science Foundation (No. 2021A1515012625); Guangzhou Basic and Applied Research Project (No. 202102021074); CCF-Tencent Open Research fund

(RAGR20210114); and the Guangdong Provincial Key Laboratory of Intellectual Property & Big Data (No. 2018B030322016).

References

[1] S. Aydın, Deep learning classification of neuro-emotional phase domain complexity levels induced by affective video film clips, IEEE Journal of Biomedical and Health Informatics 24 (6) (2019) 1695–1702.
[2] S. Aydın, B. Akın, Machine learning classification of maladaptive rumination and cognitive distraction in terms of frequency specific complexity, Biomedical Signal Processing and Control 77 (2022) 103740.
[3] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572.
[4] D. Deb, J. Zhang, A.K. Jain, Advfaces: Adversarial face synthesis, in: IEEE International Joint Conference on Biometrics (IJCB), 2020.
[5] H. Zhang, Z. Hu, C. Luo, W. Zuo, M. Wang, Semantic image inpainting with progressive generative networks, in: ACM MM, 2018, p. 1939–1947.
[6] S. Chen, Y. Fu, Progressively guided alternate refinement network for rgb-d salient object detection, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), ECCV, 2020, pp. 520–538.
[7] Y. Liu, Q. Wen, H. Chen, W. Liu, J. Qin, G. Han, S. He, Crowd counting via cross-stage refinement networks, IEEE Transactions on Image Processing 29 (2020) 6800–6812.
[8] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: CVPR, 2016, pp. 2536–2544.
[9] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.
[10] S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion, ACM Transactions on Graphics 36 (4) (2017) 1–14.
[11] G. Liu, F.A. Reda, K.J. Shih, T.-C. Wang, A. Tao, B. Catanzaro, Image inpainting for irregular holes using partial convolutions, in: ECCV, 2018, pp. 85–100.
[12] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Free-form image inpainting with gated convolution, 2019, pp. 4471–4480.
[13] H. Liu, B. Jiang, Y. Song, W. Huang, C. Yang, Rethinking image inpainting via a mutual encoder-decoder with feature equalizations, in: ECCV, 2020.
[14] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE transactions on pattern analysis and machine intelligence 38 (2) (2015) 295–307.
[15] J. Kim, J.K. Lee, K.M. Lee, Deeply-recursive convolutional network for image super-resolution, in: CVPR, 2016, pp. 1637–1645.
[16] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: CVPR workshops, 2017, pp. 136–144.
[17] Y. Tai, J. Yang, X. Liu, C. Xu, Memnet: A persistent memory network for image restoration, in: ICCV, 2017, pp. 4539–4547.
[18] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: CVPR, 2018, pp. 2472–2481.
[19] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: CVPR, 2017, pp. 4700–4708.
[20] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199.
[21] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: A survey, IEEE Access 6 (2018) 14410–14430.
[22] A. Kurakin, I. Goodfellow, S. Bengio, et al., Adversarial examples in the physical world (2016).
[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083.
[24] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, CVPR.
[25] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: CVPR, 2018, pp. 9185–9193.
[26] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 ieee symposium on security and privacy (sp), IEEE, 2017, pp. 39–57.
[27] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Transactions on Evolutionary Computation 23 (5) (2019) 828–841.
[28] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: CVPR, 2016, pp. 2574–2582.
[29] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, D. Song, Generating adversarial examples with adversarial networks, arXiv preprint arXiv:1801.02610.
[30] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions (2016). arXiv:1511.07122.
[31] L.A. Gatys, A.S. Ecker, M. Bethge, A neural algorithm of artistic style, arXiv preprint arXiv:1508.06576.
[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, 2019, pp. 8024–8035.
[34] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2017, pp. 6626–6637.
[35] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, F. Huang, Curricularface: Adaptive curriculum learning loss for deep face recognition, in: CVPR, 2020.
[36] Zhou Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.
[37] Z. Yi, Q. Tang, S. Azizi, D. Jang, Z. Xu, Contextual residual aggregation for ultra high-resolution image inpainting, 2020, pp. 7508–7517.
[38] J. Li, F. He, L. Zhang, B. Du, D. Tao, Progressive reconstruction of visual structure for image inpainting, in: ICCV, 2019.
[39] T. Yu, Z. Guo, X. Jin, S. Wu, Z. Chen, W. Li, Z. Zhang, S. Liu, Region normalization for image inpainting., in: AAAI, 2020, pp. 12733–12740.
[40] M. Zhu, D. He, X. Li, C. Li, F. Li, X. Liu, E. Ding, Z. Zhang, Image inpainting by end-to-end cascaded refinement with mask awareness, IEEE Transactions on Image Processing 30 (2021) 4855–4866.
[41] W. Quan, R. Zhang, Y. Zhang, Z. Li, J. Wang, D.-M. Yan, Image inpainting with local and global refinement, IEEE Transactions on Image Processing 31 (2022) 2405–2420.
[42] N. Wang, Y. Zhang, L. Zhang, Dynamic selection network for image inpainting, IEEE Transactions on Image Processing 30 (2021) 1784–1798.



Zhixuan Zhong obtained a B.Sc. degree in 2020 and is now a master's student both in the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing and deep learning.

Yong Du is an assistant professor in the Department of Computer Science and Technology, Ocean University of China. He obtained B.Sc. and M.Sc. degrees from Jiangnan University and a Ph.D. degree from South China University of Technology. His research interests include computer vision and image processing.

Yang Zhou obtained a B.Sc. degree in 2020 and is now a master's student both in the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing and deep learning.

Jiangzhong Cao received his Ph.D. degree in communication and information system from School of Information Science and Technology, Sun Yat-sen University, China, in 2013. He is currently an Associate Professor with the School of Information Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include computer vision, pattern recognition and deep learning.

Shengfeng He is an associate professor in the School of Computer Science and Engineering, South China University of Technology. He obtained B.Sc. and M.Sc. degrees from Macau University of Science and Technology in 2009 and 2011 respectively, and a Ph.D. degree from City University of Hong Kong in 2015. His research interests include computer vision, image processing, and computer graphics. He serves on the editorial board of Neurocomputing.