Signal, Image and Video Processing
https://doi.org/10.1007/s11760-020-01749-6

ORIGINAL PAPER

A robust and efficient image de-fencing approach using conditional generative adversarial networks

Divyanshu Gupta[1] · Shorya Jain[1] · Utkarsh Tripathi[1] · Pratik Chattopadhyay[1] · Lipo Wang[2]

Received: 18 February 2020 / Revised: 3 July 2020 / Accepted: 17 July 2020

Abstract
Image de-fencing is one of the most important aspects of recreational photography in which the objective is to remove the fence texture present in an image and generate an aesthetically pleasing version of the same image without the fence texture. In this paper, we present an automated and effective technique for fence removal and image reconstruction using conditional generative adversarial networks (cGANs). These networks have been successfully applied in several other domains of computer vision, focusing on image generation and rendering. Our approach is based on a two-stage architecture involving two cGANs in succession, in which the first cGAN generates the fence mask from an input fenced image, and the next one generates the final de-fenced image from the given input and the corresponding fence mask obtained from the previous cGAN. Training of these networks is carried out independently using suitable loss functions, and during the deployment phase, the above two networks are stacked together in an end-to-end manner to generate the de-fenced image from an unknown test image. Extensive qualitative and quantitative evaluations using challenging data sets emphasize the effectiveness of our approach over state-of-the-art de-fencing techniques. The data sets used in the experiments have also been made available for further comparison.

Keywords Automated de-fencing · Two-stage cGAN network · Fence mask detection · Image inpainting

1 Introduction

Despite technological advances in the domain of digital photography, capturing a clear snapshot of an object of interest often becomes difficult if some obstructions are present in the front or behind the object. For example, in a zoo, the presence of fence/cage bars occludes the field of view of the camera, which hinders capturing an unobstructed view of the bird or animal behind the cage bars. Image de-fencing refers to the task of generating an aesthetically pleasing image of the intended object by eliminating the fence structure. Traditionally, this problem has been viewed as a combination of two separate subproblems: (i) *fence mask generation*, which essentially clusters the image region into fenced and non-fenced regions, and (ii) *image inpainting*, which involves artificially synthesizing colors to the fenced regions to make the rendered image look realistic [1–9]. This is explained diagrammatically with the help of Fig. 1a–c.

Conditional generative adversarial networks (cGANs) [10–13] have already demonstrated strong potential in performing image-to-image translation by satisfying a set of user-defined conditions [14–17]. Hence, in this work, we propose to use cGAN-based model for image de-fencing. To the best of our knowledge, this is the first-ever work that attempts to develop an end-to-end cGAN-based architecture for the image de-fencing task. Specifically, we propose a two-stage algorithm consisting of two sub-networks (cGANs) to carry out the fence mask generation and image inpainting stages in succession. Encouraging results are obtained

✉ Pratik Chattopadhyay
pratik.cse@iitbhu.ac.in

Divyanshu Gupta
divyanshu.gupta.cse15@iitbhu.ac.in

Shorya Jain
shorya.jain.cse16@iitbhu.ac.in

Utkarsh Tripathi
utkarsh.tripathi.cse16@iitbhu.ac.in

Lipo Wang
elpwang@ntu.edu.sg

[1] Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi 221005, India

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

Published online: 30 July 2020

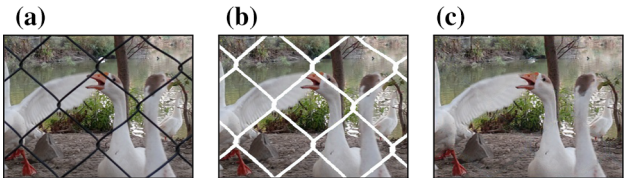

Fig. 1 **a** Input image, **b** fence mask detection stage, **c** inpainting stage

from extensive qualitative and quantitative analysis of our approach. The overall paper has been organized as follows: Sect. 2 summarizes the related literature in automatic fence detection, image inpainting, and image de-fencing, while the proposed approach is described in detail in Sect. 3. Experimental evaluation and analysis of results are presented in Sect. 4, while conclusions, along with future research scopes, are highlighted in Sect. 5.

2 Related work

We provide an overview of the existing approaches to fence mask generation, image inpainting, and image de-fencing in the following three subsections.

2.1 Fence mask detection

Fence mask detection is the process of segmenting an input image with a fence into two clusters, such that all the fence pixels are assigned a particular cluster, and each of the other pixels is assigned a different cluster. To date, several research articles exist in the domain of regular and near-regular pattern detection [2,18–20]. The work in [18] uses higher-order feature matching to discover the lattices of near-regular patterns in real images based on the principal eigenvector of the affinity matrix. In [19], a method for detection of deformed 2D wallpaper patterns in real-world images has been proposed by mapping the 2D lattice detection problem into a multi-target tracking problem, which is solved within a Markov random field framework. In [21], the problem of near-regular fence detection is handled by employing an efficient mean-shift belief propagation method to extract the underlying deformed lattice in the image. In [22], a soft fence detection method is discussed that uses visual parallax as a cue to distinguish between fenced and non-fenced regions.

2.2 Image inpainting

Image inpainting is the process of restoration of the unfilled portions of an image with appropriate plausible content/color. Image inpainting methods used in the literature can be broadly divided into two categories: (a) diffusion-based methods [23,24] and (b) exemplar-based methods [25–28]. The former category of approaches uses smoothness priors to propagate information from known regions to the unknown region, while the latter category fills in the occluded regions by employing similar patches from other locations in the image. Exemplar-based methods have the potential of filling up large occluded regions and recreate missing textures to reconstruct large regions within an image. But these methods are unable to recover the high-frequency details of the image properly. To the best of our knowledge, context encoder [29] is the first deep learning approach used for image inpainting in which an encoder maps an image with missing regions to a low-dimensional feature space that is used by the decoder to generate the inpainted output image. The work in [30] uses a pre-trained VGG network to minimize the feature differences in the image background, thereby improving the work of [29]. In [14], a GAN-based approach is described that maps the image inpainting task into a constrained image generation problem, such that the encoding of the inpainted image in a latent space is close to that of the unfilled input image in the same latent space in terms of weighted context loss and prior loss. The technique in [31] estimates missing regions in an unfilled image, following which it employs an attention-based mechanism for fine-tuning the results. A two-stage adversarial model is described in [32] that consists of an edge generator network to detect boundaries of the unfilled patches within an image and an image completion network to fill these patches with appropriate colors.

2.3 Image de-fencing

The first work on image de-fencing has been developed in [1], in which the fence patterns are segmented based on spatial regularity, and an inpainting algorithm [25] is applied to fill in the fence pixels with appropriate colors. An improvement to this work is suggested in [2], which employs an online learning algorithm for lattice detection and segmentation, and finally, a multiview inpainting technique is adopted to improve the image restoration process. Both approaches assume the fence structure to be near regular. In [3], a multi-frame de-fencing technique is described that uses loopy-belief propagation [33] across frames and uses an image matting technique [34] for fence segmentation with the assumption that the color of the fence pixels is significantly different from the background. Another video de-fencing approach discussed in [35] extracts a fence mask from each frame with the aid of depth maps captured by a Kinect sensor, following which it employs an optical flow algorithm to find correspondences between adjacent frames. By modeling the de-fenced image as a Markov random field, the maximum a posteriori estimate obtained by applying loopy-belief propagation is considered to be the final de-fenced image. A similar technique to identify the occluded regions using multiple image frames has also been described

in [36]. In [7], signal de-mixing has been used to detect the fence structure by capturing the sparsity and regularity of the different image regions. A popular inpainting algorithm [25] is next applied to output the de-fenced images. This approach requires manual specification of a large number of parameters due to which it is not very suitable for practical use. Another semi-automated approach for image de-fencing is described in [6], where several fence pixels in the image are manually marked, following which a Bayesian classifier is used to classify each pixel as *fence* or *non-fence* based on the color distribution of the marked pixels and the non-marked pixels. As understandable, both the approaches [6,7] are prone to human error and are also time intensive. Histogram of oriented gradients has been used for fence mask detection in [8], and the same inpainting algorithm [25] (as in most of the previous de-fencing techniques) has been used for inpainting.

Recently, a few deep learning-based video de-fencing approaches have been developed. For example, the work in [4] employs a convolutional neural network (CNN) to detect the fence pixels in an input image and then uses a sparsity-based optimization framework to fill in the fence pixels, which is time intensive. The work in [5] utilizes a pre-trained CNN coupled with an SVM classifier for fence texel joint detection and then connects these joints to obtain scribbles for image matting. However, this approach fails to provide satisfactory performance if a fenced image with an irregular pattern is provided as the input. Another approach for fence segmentation using fully convolutional neural networks is presented in [37], which is accompanied by an efficient occlusion-aware optical flow-based image recovery algorithm. In [9], fenced images, artificially synthesized from natural images, have been used to train a ResNet-based image recovery network to predict the de-fenced images. However, the initial spatial filtering step involved in this algorithm requires the specification of a set of user-defined parameters, which is likely to vary for different images. Moreover, the results of de-fencing presented in this work also lack in visual quality.

As seen from the extensive literature survey, previous approaches to fence detection suffer from either the unrealistic assumption of (i) color consistency in fence structures and (ii) the presence of near-regular repeated fence elements in any given fenced image. In contrast to these previous techniques, our method uses cGAN-based prediction for both the fence mask detection and the inpainting stages. Qualitative and quantitative results presented in Sect. 4 verify that our approach performs robustly and carries out de-fencing effectively even in challenging situations such as irregular fence structures and occlusion, which the previous methods have failed to achieve. The main contributions of our work are:

- Proposing a novel two-stage end-to-end image de-fencing network involving a fence mask generator and an image recovery sub-networks, each of which is based on cGANs.
- Training the models suitably to handle challenging situations such as images with occlusion or broken/irregular fence structures in a time-efficient manner.
- Performing extensive experimental evaluation and making the pre-trained models and data sets used in the experiments publicly available to the research community.

3 Proposed approach

A schematic diagram of the proposed de-fencing approach is shown in Fig. 2.

With reference to the figure, two cGANs have been used to carry out the fence mask detection and the image inpainting steps. The generator and discriminator pair of the two networks are denoted by (G_1,D_1) and (G_2,D_2). The generator G_1 takes as input a fenced image along with its Canny edge map and outputs a fence mask, while the generator G_2 uses the fence mask generated by G_1 along with the input image to output the final de-fenced image. Adversarial losses at the discriminators D_1 and D_2 are used to train the networks separately. During deployment, the generators G_1 and G_2 are stacked one after another to form an end-to-end network for translating any fenced image at the input of G_1 to its de-fenced version at the output of G_2. The cGAN architectures and the training steps are explained in Sects. 3.1 and 3.2.

3.1 Fence mask generator

The task of the fence mask generator G_1 is to predict a binary fence mask image from a given fenced image. Using the available ground truth, this network learns a function $\mathcal{G}_1$ that takes as input a fenced image I_f as well as a Canny edge map of I_f (denoted by I_c) to generate a fence mask I_p. This function can be represented as:

$$I_p = \mathcal{G}_1(I_f, I_c), \tag{1}$$

which is learned using the first cGAN (Fig. 2). The generator G_1 consists of an encoder with seven down-sampling layers, followed by a decoder with seven up-sampling layers. The detailed architecture of G_1 is as follows:

It can be seen that the G_1 generator consists of a total of 14 layers, in which $conv_k^i$ represents a 2D convolution layer with a stride value of two, and k filters at the i^{th} layer, and $dconv_k^i$ represents up-sampling followed by a 2D convolution layer with k filters at the i^{th} layer. Skip connections are used between i^{th} $dconv$ and $(14-i)^{th}$ $conv$ layers. The final

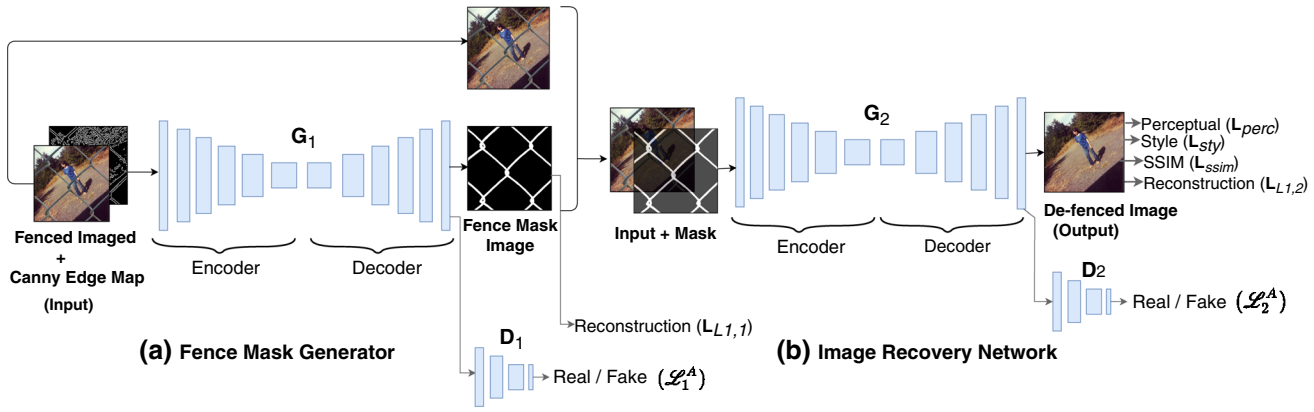

Fig. 2 Two-stage image de-fencing network

Detailed architecture of the fence mask generator
$conv^1_{64}$- $conv^2_{128}$- $conv^3_{256}$- $conv^4_{512}$- $conv^5_{512}$- $conv^6_{512}$- $conv^7_{512}$- $dconv^8_{512}$- $dconv^9_{512}$- $dconv^{10}_{512}$- $dconv^{11}_{256}$- $dconv^{12}_{128}$- $dconv^{13}_{64}$- out^{14}_c

layer, represented by out^i_c, performs up-sampling by a factor of two followed by 2D convolution with c filters along with $tanh$ activation. The c filters correspond to the c channels in the generated image. Since for the fence mask generator, the output image is a binary mask, the value of c is one. The discriminator D_1 is a 16×16 PatchGAN Markovian discriminator that is also a deep convolution classification network with the following configuration:

Discriminator architecture of the mask generator
$conv^1_{64}$ - $conv^2_{128}$ - $conv^3_{256}$ - $conv^4_{512}$ - val^5_1

Here, val^5_1 represents the final 2D convolution layer with a single filter and outputs classification of each 16×16 patch of an image as either real or fake.
We use I_m and I_p, conditioned on I_f as inputs to the discriminator to predict the fence mask image as real or fake. The network is trained with the objective function comprising of the adversarial loss $\mathcal{L}^A_1$ and the $L1$ loss as shown in (2):

$$\min_{\mathcal{G}_1}\max_{\mathcal{D}_1} \mathrm{L}_{G_1} = \min_{\mathcal{G}_1}\Big(\alpha_1 \max_{\mathcal{D}_1}(\mathcal{L}^A_1) + \beta_1(\mathrm{L}_{L1,1})\Big), \tag{2}$$

where $\mathcal{D}_1$ represents the function learned by the discriminator of the cGAN, and α_1 and β_1 are regularization parameters. Here, we choose $\alpha_1 = 1$ and $\beta_1 = 10$. The above two loss functions are mathematically defined as follows:

$$\mathcal{L}^A_1 = \mathbf{E}_{I_f,I_m}\Big[log(\mathcal{D}_1(I_m, I_f))\Big] + \mathbf{E}_{I_f,I_p}\Big[log(1-\mathcal{D}_1(I_p, I_f))\Big], \tag{3}$$

$$\mathrm{L}_{L1,1} = \mathbf{E}\Big[||I_p - I_m||_1\Big]. \tag{4}$$

Here, $\mathbf{E}$ denotes the expectation operator and other symbols carry their usual meanings. The network is trained in multiple epochs, and the training is stopped once the absolute difference between the network loss values in two successive epochs reaches less than a small threshold ϵ (we use $\epsilon = 10^{-3}$). It may be noted that the generator can also be trained using I_f alone (i.e., without using the Canny edge map I_c). However, appending the Canny edge map along with the input image during training helps in improving the fence mask detection results, since the Canny edge map captures crucial structural information and properties present in an image and thus provides high-frequency detail of an image which is useful for fence mask detection, thereby enabling the cGAN to precisely differentiate between the fenced and the non-fenced structures present in the image.

3.2 Image recovery network

As explained before, the *image recovery network* architecture is also based on cGAN. Let I_d be the ground-truth de-fenced image and $\tilde{I}_f$ be its ground-truth fence mask. The cGAN learns a function $\mathcal{G}_2$ to generate the final de-fenced image $\tilde{I}_p$, conditioned on I_f as shown in (5):

$$\tilde{I}_p = \mathcal{G}_2(\tilde{I}_f, I_f). \tag{5}$$

The generator and discriminator architectures used here are similar to those used in the case of the *fence mask generator* network, as explained in Sect. 3.1 with the exception that in the final layer, the convolution operation considers

three filters instead of just *one*. This is since the inpainted image to be generated by this cGAN is an RGB image with *three* channels. This network is trained by a joint objective function consisting of the adversarial loss, perceptual loss, style loss, and *L1* loss, along with SSIM loss. The adversarial loss computed at the discriminator D_2 of this cGAN is defined as:

$$\mathcal{L}_2^A = \mathbf{E}_{I_f, I_d}\Big[\log(\mathcal{D}_2(I_d, I_f))\Big] + \mathbf{E}_{I_f, \tilde{I}_p}\Big[\log(1-\mathcal{D}_2(\tilde{I}_p, I_f))\Big]. \tag{6}$$

The perceptual loss term L_{perc} provides the differences between the high-level feature representations between the ground-truth and the GAN-generated images, and this is computed using a pre-trained CNN [38]. It is computed as:

$$\mathrm{L}_{perc} = \mathbf{E}\Big[\sum_i \frac{1}{N_i}||\mathbf{a}_\mathrm{i}(I_d) - \mathbf{a}_\mathrm{i}(\tilde{I}_p)||_1\Big], \tag{7}$$

where $\mathbf{a}_\mathrm{i}$ denotes the activation map, and N_i represents the number of filters in the i^{th} layer of the VGG-19 network. As seen from (7), this loss imposes a higher penalty if the output image is not perceptually similar to the ground truth. To train the cGAN, we also minimize a style loss term quantified by the amount of correlation present between the features maps of the generated and ground-truth images at a particular layer and it is calculated with the help of *Gram matrix* G [39]. Each element of this matrix represents the inner product between a pair of vectorized feature maps at a CNN layer. For the feature map of size $C_j \times H_j \times W_j$, the style loss is mathematically defined as:

$$\mathrm{L}_{sty} = \mathbf{E}_j\Big[||\mathrm{G}_\mathrm{j}^{\mathbf{a}}(\tilde{I}_p) - \mathrm{G}_\mathrm{j}^{\mathbf{a}}(I_{def})||_1\Big], \tag{8}$$

where $\mathrm{G}_\mathrm{j}^{\mathbf{a}}$ is the $\mathrm{C_j} \times \mathrm{C_j}$ Gram matrix corresponding to feature map $\mathbf{a}_\mathrm{j}$. Both L_{perc} and L_{sty} enable the generator G_2 to learn the input data distribution at a high resolution. The *L1* loss function is computed as:

$$\mathrm{L}_{L1,2} = \mathbf{E}\Big[||\tilde{I}_p - I_d||_1\Big]. \tag{9}$$

For obtaining visually pleasing images from the generator, we also incorporate a structural similarity loss term [40,41] as shown in (10), which indicates the differences between the luminance, contrast, and structure between the generated de-fenced image and the ground-truth de-fenced image. Mathematically,

$$\mathrm{L}_{SSIM} = \frac{1}{N}\sum_p (1 - \mathrm{SSIM(p)}), \tag{10}$$

where p refers to a particular pixel position and N corresponds to the number of pixels in the image. Mathematically,

$$\mathrm{SSIM(p)} = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2{\mu_y}^2 + C_1}.\frac{2\sigma_{xy} + C_2}{{\sigma_x}^2{\sigma_y}^2 + C_2}, \tag{11}$$

and it refers to the structural similarity index between the ground-truth and the generated image at pixel position p. In the above expression, μ_x and μ_y represent mean intensities in the neighborhood of p, while σ_x and σ_y represent standard deviations for two nonnegative image signals x and y, respectively, σ_{xy} represents the covariance of x and y, and C_1 and C_2 are constants [1]. The overall loss function for the image recovering network is computed as:

$$\min_{G_2}\max_{D_2}\mathrm{L}_{G_2} = \min_{G_2}\Big(\alpha_2 \max_{D_2}(\mathcal{L}_2^A) + \beta_2(\mathrm{L}_{L1,2}) + \gamma(\mathrm{L}_{perc}) + \delta(\mathrm{L}_{sty}) + \eta(\mathrm{L}_{SSIM})\Big), \tag{12}$$

where α_2, β_2, γ, δ, and η are regularization parameters. In our experiments, we set α_2=0.1, β_2=10, γ=2, δ=1, and η=1.

4 Experiments and results

Our experiments have been performed on a system with three graphics processing units (GPUs), out of which one is Nvidia Titan Xp with 12-GB RAM, total FB memory as 12196 MB and total BAR1 memory as 256 MB, and the other two are Nvidia GeForce GTX 1080 Ti with 11-GB RAM, total FB memory as 11178 MB and total BAR1 memory as 256 MB. For training the fence mask detector, a public fence segmentation data set [37] has been used, which consists of fences with regular patterns only. Along with this, we also use a synthetic data set containing images with irregular fence patterns that are made by adding artificial fence structures on a set of natural images from the Pascal VOC data set [42]. We construct a large number of irregular fence structures by applying warping and other image transformation techniques on the ground-truth fence masks from the first data set and next superimpose these fence structures on the selected images from the Pascal VOC data. The total number of pairs of fenced images and their corresponding fence masks in the training set is 29013. To train the image recovery network also, we had to construct an artificial data set since no public data set exists with both fenced and the corresponding de-fenced images. For this purpose, we create another synthetic data set by superimposing artificial fence structures on images selected from the Pascal VOC data [42] and the

[1] Appropriate values for C_1 and C_2 to compute the SSIM loss can be found in https://github.com/keras-team/keras-contrib/blob/master/keras_contrib/losses/dssim.py.

COCO data [43]. The total number of images in this gallery set is 29413. For evaluating our approach, we construct the following sets of test images: (a) 150 images with regular fence patterns, (b) a set of images with varying fence patterns (referred to as the *Test-1* set), (c) a set of images with occlusion (referred to as the *Test-2* set), and (d) a set of images with broken/irregular structures (referred to as the *Test-3* set). The total number of images in *Test-1*, *Test-2*, and *Test-3* sets is equal to 150. Since the two sub-networks accept 256×256-dimensional image input only, at the outset, each image is resized to dimensions 256×256. The data sets used during training and testing the two sub-networks, as well as the pre-trained models, have been made available here.

4.1 Evaluation with different Canny thresholds

As explained in Sect. 3, the input to the fence mask generator is a Canny edge map along with the fenced image. Since the computation of the Canny map requires specification of two threshold parameters, in the first experiment, we make an appropriate choice of these threshold parameters by considering different combinations of choices and selecting the one that performs the best on the training set. The ground-truth fence mask is compared with the generated fence mask, and the *F1-Score* is computed based on the number of fence and non-fence pixels that are predicted correctly. The different sets of minimum/maximum threshold parameters for the Canny edge detector and the corresponding *F1-Scores* are shown in Table 1.

It can be seen that the *F1-Score* does not alter much for the different sets of threshold parameters. Still, we select the minimum and maximum thresholds as 100 and 200 since this combination yields the maximum *F1-Score*, and these are also used to report all the future results.

Table 1 F1-Score for different Canny thresholds

Min threshold	200	100	100	100	100
Max threshold	500	500	400	300	200
F1-Score	0.954	0.958	0.957	0.956	0.959

4.2 Results on regular fence images

Figure 3 shows the qualitative performance of the proposed image de-fencing network on a sample test set consisting of regular fence structures.

The first row of the figure represents the input image, the second row represents the output of the fence mask generator, while the last row represents the output of the image recovery network. Visually, it can be observed that the generated images after inpainting are of high quality. In all cases, the de-fenced images are able to successfully remove the fence texture and produce realistic de-fenced results. Use of cGAN-based image de-fencing has twofold benefits: (i) Firstly, due to the powerful generalization capability of generative neural networks, high-quality image de-fencing results can be obtained if trained with a sufficiently large amount of data, and (ii) secondly, it is highly time efficient. We observe that the two generators used in our de-fencing approach have average response times of 24 and 27 milliseconds, respectively. Thus, the de-fencing of an input fenced image can be completed in only about 51 milliseconds.

4.3 Results on challenging real-world images

To evaluate the effectiveness of our approach in handling complex irregular, or occluded fence structures, we consider the test sets *Test-1*, *Test-2*, and *Test-3*, as explained before. Samples images from the *Test-1* set are shown in Fig. 4a, while few samples from test sets *Test-2* and *Test-3* are shown in Figs. 5a and 6a, respectively. The output of the fence mask generator corresponding to each image in Figs. 4a, 5a, and 6a is shown in Figs. 4b, 5b, and 6b, while the final de-fenced outputs provided by the image recovery network are shown in Figs. 4c, 5c, and 6c.

It can be seen that the de-fenced images generated by the recovery network for each of the three challenging scenarios are visually quite appealing. It can also be concluded from the results that the synthetic data sets constructed by our team are appropriate enough to train a model to perform satisfactorily on any real-world fenced image.

Fig. 3 Qualitative results of the proposed approach, **a** input, **b** generated mask, and **c** de-fenced image

Fig. 4 Sample de-fenced results on *Test-1* images

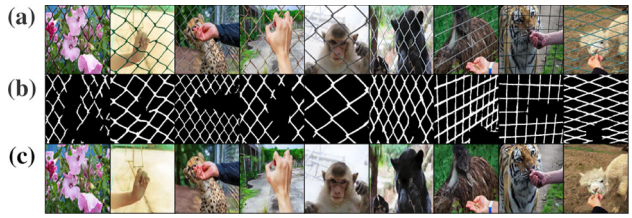

Fig. 5 Sample de-fenced results on *Test-2* images

Fig. 6 Sample de-fenced results on *Test-3* images

4.4 Comparison with existing de-fencing techniques

To verify the efficacy of our image de-fencing approach, we perform both qualitative and quantitative comparative analysis of our work with five other existing image de-fencing approaches, namely [1,6–9]. Apart from these, we also compare the results of our work with two other deep learning-based approaches [13,44]. While [13] introduces the popular *pix2pix* GAN that is trained using adversarial loss and $L1$ loss, the work in [44] proposes a deep network that behaves as an image processing operator to carry out tasks such as image smoothing, photographic style transfer, and non-local de-hazing. Both these networks have demonstrated strong effectiveness in solving various image-to-image translation tasks and have thus been used in the comparative study. For the work of [6], 30 random fence pixels are manually selected from each input image, and next, the MATLAB code of this work shared by the authors in [45] has been used to generate the de-fenced results. Qualitative results obtained from the different approaches are shown in Fig. 7. In this figure, the first row shows a set of input fenced images used during testing, while each of the other rows (except the last one) represents the output de-fenced images corresponding to the input images as given by the different approaches used in the comparative study. Proper citations to each method are specified on the left side of the corresponding row. The final row represents the de-fenced results of our work.

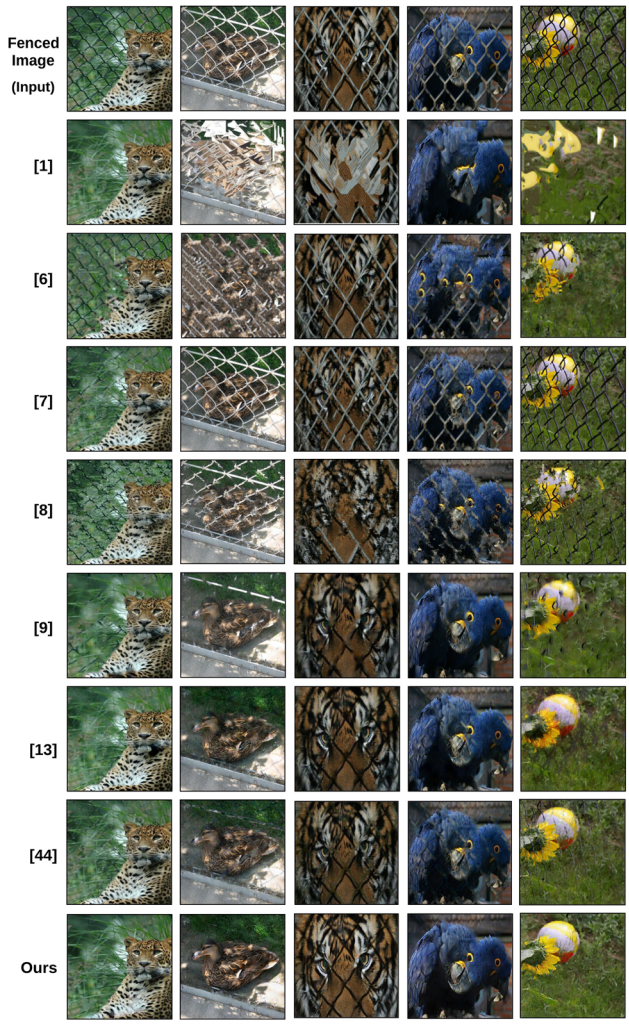

Fig. 7 Qualitative results of image de-fencing approaches

It can be observed that the proposed method outperforms each of the other approaches used in the comparative study in terms of the visual quality of generated images. Although the performance of the proposed method on the *tiger face image* (shown in the third column of the last row) does not appear to be satisfactory, a careful observation reveals that only the shadow of the fence exists on the de-fenced image, whereas the actual fence has been properly removed through de-fencing.

The approaches in [1,6,7] are non-machine learning-based and are dependent on several user-defined parameters and thresholds, due to which these cannot perform robustly against varying input conditions, e.g., non-regular fence structures. As can also be seen from the second, third, and fourth rows in Fig. 7, these methods are unable to remove the fence structures effectively from all the input images. The approach of [1] has the worst overall performance, since it

Table 2 Quantitative results of image de-fencing approaches

Quantitative metric	De-fencing algorithm							
	[1]	[6]	[7]	[8]	[9]	[13]	[44]	Ours
MOS	2.96	2.82	2.76	3.24	3.76	4.12	3.94	4.34
SSIM	0.39	0.27	0.57	0.78	0.81	0.83	0.80	0.87
LPIPS	0.43	0.44	0.41	0.28	0.24	0.21	0.25	0.17

performs fence detection by finding regular lattice structures in an image, and hence, it fails once images with irregular fence patterns are provided. Additionally, here inpainting is done with the assumption that the foreground, i.e., the fence structure, is more regular than the background, and hence, its performance degrades if the image background is textured. The methods in [6,7] require user intervention to manually select a set of fence pixels or texel (i.e., the smallest repeating structure) in the input image, and hence, practical application of these methods is limited. The de-fencing approach in [8] performs fence segmentation based on the histogram of gradient features with the unrealistic assumption of uniform background, and as shown in Fig. 7, it fails to perform well, if the background is textured, or if certain edge-like objects are present in the background. Due to the use of only MSE loss during training, the inpainting network described in [9] fails to retain detailed texture-level information present in the images and hence provides unsatisfactory de-fenced results. The *pix2pix* GAN [13] directly generates the de-fenced output from the input image without any intermediate fence mask detection stage and shows a reasonably good performance. However, due to skipping the mask estimation stage, it sometimes fails to eliminate each and every fence structure present in the input image, as can be seen from the last column of the seventh row in Fig. 7. Similarly, for [44], traces of fence structures are found to be present in the output images (refer to the second-last row of Fig. 7), and hence, this is also not an effective de-fencing method. The LPIPS metric is computed from the internal activation of a VGG-16 network, and its value is closer to human-level perceptual judgments compared to other feature-based metrics. Unlike MOS and SSIM, a lower value of the LPIPS metric indicates a better match between two images.

It can be seen from the table that our approach outperforms the existing de-fencing techniques in terms of each of the three quantitative metrics used in the study. All the above results and discussions emphasize the effectiveness of our work over the previous de-fencing techniques.

5 Conclusions and future work

We explore the applicability of conditional generative adversarial networks (cGANs) for image de-fencing. The proposed approach makes use of two networks: a fence mask generator network and an image recovery network. Results on a set of test images with varying fence patterns show that in addition to being significantly time efficient, our method outperforms each of the existing image de-fencing techniques in terms of the visual quality of generated images as well as in terms of three metrics: MOS, SSIM, and LPIPS.

Our work will also serve as a baseline approach for deep neural network-based image de-fencing, and future researchers can use the extensive data set created by our team to come up with further improved models. Improving the efficiency of the de-fencing process and robustly handling videos with high frame rate may be considered as future scopes of work. This can be achieved through simplification of the cGAN architecture, and/or proposing improved loss functions, and needs to be further studied. Our approach can also be suitably extended to perform other popular image-to-image translation tasks like image de-hazing, de-noising, etc.

Acknowledgements We acknowledge NVIDIA for supporting our research with a TITAN Xp graphics processing unit. Our sincere gratitude goes to Dr. V. Maurya for helping us with the implementation of his team's work that has been used in the comparative study. We are also thankful to everyone who contributed to obtaining the MOS, as reported in Table 2.

References

1. Liu, Y., et al.: Image de-fencing. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
2. Park, M. et al.: Image de-fencing revisited (2010)
3. Khasare, V.S., et al.: Seeing through the fence: image de-fencing using a video sequence (2013)
4. Jonna, S., et al.: My camera can see through fences: a deep learning approach for image de-fencing (2015)
5. Jonna, S., et al.: Deep learning based fence segmentation and removal from an image using a video sequence (2016)
6. Farid, M.S., et al.: Image de-fencing framework with hybrid inpainting algorithm. SIViP **10**(7), 1193–1201 (2016)
7. Kumar, V., et al.: Image defencing via signal demixing (2016)
8. Khalid, M., et al.: Image de-fencing using histograms of oriented gradients. SIViP **12**(6), 1173–1180 (2018)
9. Matsui, T., Ikehara, M.: Single-image fence removal using deep convolutional neural network. IEEE Access **8**, 38846–38854 (2019)
10. Reed, S., et al.: Generative Adversarial Text to Image Synthesis (2016). arXiv:1605.05396

11. Zhang, H., et al.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks (2017)
12. Huang, H., et al.: An introduction to image synthesis with generative adversarial nets (2018). arXiv:1803.04469
13. Isola, P., et al.: Image-To-image translation with conditional adversarial networks (2017)
14. Yeh, R.A., et al.: Semantic image inpainting with deep generative models (2017)
15. Brkic, K., et al.: I know that person: generative full body and face de-identification of people in images (2017)
16. Radford, A., et al.: Unsupervised representation learning with deep convolutional generative adversarial networks (2015). arXiv:1511.06434
17. Chen, X., et al.: InfoGAN: interpretable representation learning by information maximizing generative adversarial nets (2016)
18. Hays, J., et al.: Discovering texture regularity as a higher-order correspondence problem (2006)
19. Park, M., et al.: Deformed lattice detection in real-world images using mean-shift belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1804–1816 (2009)
20. Lin, W.-C., Liu, Y.: A lattice-based MRF model for dynamic near-regular texture tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(5), 777–792 (2007)
21. Park, M., et al.: Deformed lattice discovery via efficient mean-shift belief propagation (2008)
22. Mu, Y., et al.: Video de-fencing. IEEE Trans. Circuits Syst. Video Technol. **24**(7), 1111–1121 (2014)
23. Bertalmio, M., et al.: Simultaneous structure and texture image inpainting. IEEE Trans. Image Process. **12**(8), 882–889 (2003)
24. Levin, A., et al.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2008)
25. Criminisi, A., et al.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9), 1200–1212 (2004)
26. Xu, Z., Sun, J.: Image inpainting by patch propagation using patch sparsity. IEEE Trans. Image Process. **19**(5), 1153–1165 (2010)
27. Darabi, S., et al.: Image melding: combining inconsistent images using patch-based synthesis. ACM Trans. Graph. **31**(4), 1–10 (2012)
28. Huang, J.-B., et al.: Image completion using planar structure guidance. ACM Trans. Graph. **33**(4), 129 (2014)
29. Pathak, D., et al.: Context encoders: feature learning by inpainting (2016)
30. Yang, C., et al.: High-resolution image inpainting using multi-scale neural patch synthesis (2017)
31. Yu, J., et al.: Generative image inpainting with contextual attention. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 5505–5514 (2018)
32. Nazeri, K., et al.: EdgeConnect: generative image inpainting with adversarial edge learning (2019). arXiv:1901.00212
33. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Intl. J. Comput. Vis. **70**(1), 41–54 (2006)
34. Zheng, Y., Kambhamettu, C.: Learning based digital matting (2009)
35. Jonna, S., et al.: A multimodal approach for image de-fencing and depth inpainting (2015)
36. Xue, T., et al.: A computational approach for obstruction-free photography. ACM Trans. Graph. **34**(4), 79 (2015)
37. Du, C., et al.: Accurate and efficient video de-fencing using convolutional neural networks and temporal information (2018)
38. Johnson, J., et al.: Perceptual losses for real-time style transfer and super-resolution (2016)
39. Gatys, L.A., et al.: Image style transfer using convolutional neural networks (2016)
40. Wang, Z., et al.: Image quality assessment: from error visibility to structural similarity. IEEE Trans.Image Process. **13**(4), 600–612 (2004)
41. Zhao, H., et al.: Loss functions for image restoration with neural networks. IEEE Trans. Comput. Imaging **3**(1), 47–57 (2017)
42. Everingham, M., et al.: The pascal visual object classes challenge. Intl. J. Comput. Vis. **88**(2), 303–338 (2010)
43. Lin, T.-Y., et al.: Microsoft COCO: common objects in context (2014)
44. Chen, Q., et al.: Fast image processing with fully-convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2497–2506 (2017)
45. Farid, M.S., et al.: Source code for image defencing (2016).https://www.researchgate.net/publication/296635266_Source_Code_Image_Defencing. Accessed 03 2016