https://doi.org/10.2352/ISSN.2470-1173.2020.10.IPAS-026

Real-world fence removal from a single-image via deep neural network

Takuro Matsui, Takuro Yamaguchi and Masaaki Iheara

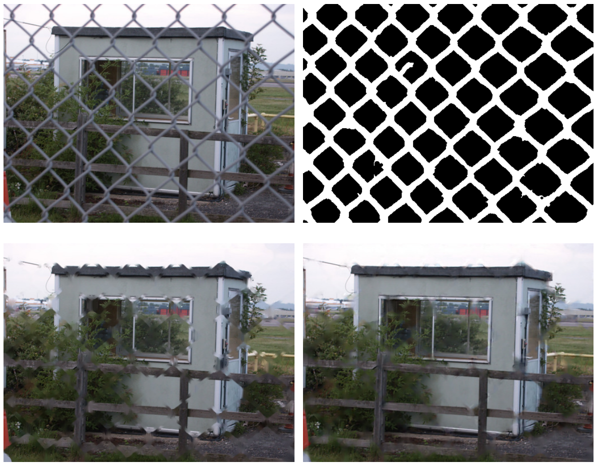
Abstract

At public space such as a zoo and sports facilities, the presence of fence often annoys tourists and professional photographers. There is a demand for a post-processing tool to produce a non-occluded view from an image or video. This "de-fencing" task is divided into two stages: one is to detect fence regions and the other is to fill the missing part. For a decade or more, various methods have been proposed for video-based de-fencing. However, only a few single-image-based methods are proposed. In this paper, we mainly focus on single-image fence removal. Conventional approaches suffer from inaccurate and non-robust fence detection and inpainting due to less content information. To solve these problems, we combine novel methods based on a deep convolutional neural network (CNN) and classical domain knowledge in image processing. In the training process, we are required to obtain both fence images and corresponding non-fence ground truth images. Therefore, we synthesize natural fence image from real images. Moreover, spacial filtering processing (e.g. a Laplacian filter and a Gaussian filter) improves the performance of the CNN for detecting and inpainting. Our proposed method can automatically detect a fence and generate a clean image without any user input. Experimental results demonstrate that our method is effective for a broad range of fence images.

Introduction

Image de-fencing, which means removing a fence from an image, is an important problem. Public spaces which includes a zoo and historical places have to install fences and barricades to enclose the dangerous area. However, amateur and professional photographers such as tourists, journalists and wild-animal lovers are often annoyed by the fences. Fence removal methods are required in various situations. As we can easily access to image processing software such as Photoshop, we are able to remove them with our own hands. Nevertheless, that might be time-consuming and require experience and skill. Image de-fencing is challenging because real-world fences have various type of shapes, textures and colors. In addition, although common fences have regular structures, some fences have completely irregular shapes and are sometimes partly distorted and broken. For this reason, robust and automatic fence removal methods are required for variety of applications.

As far as we know, Liu *et al*. [1] first propose an automatic de-fencing algorithm. They detect fence regions based on the assumption that most fences have a regular or near-regular repeating structure. After segmenting foreground and background, the missing fence region is filled by a basic inpainting method [2]. Thus, the de-fencing task is divided into two phases, which are a fence detection phase and a content recovery phase. According to this separation approach, many methods have been proposed. Existing de-fencing methods are roughly categorized into two types of methods: video-based methods and image-based methods. In this paper, we focus on a single-image fence removal.

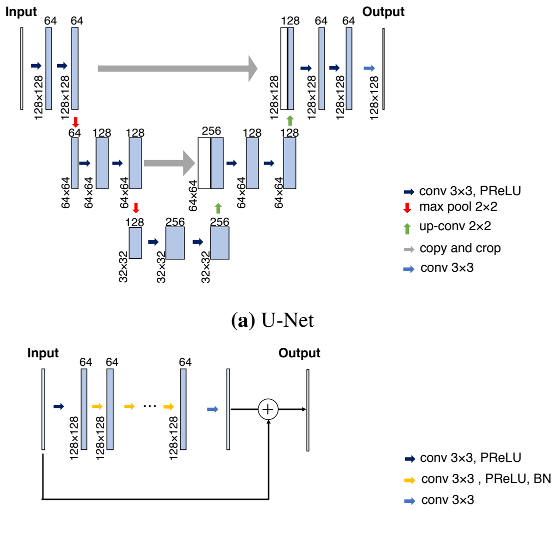


(c) Pre-processed image **(d)** De-fenced image

Figure 1: Sample image de-fencing results.

In video-based methods [3, 4, 5, 6, 7], multiple frames are used to remove fence regions. For example, method [3] estimated the global relative motion of background pixels by matching the corresponding points using affine SIFT descriptor [8]. In static captured videos, hidden part at a certain frame will become visible in another frame. Method [4] and [5] tackles not only static scenes but also dynamic scenes. They introduce CNN (convolutional neural networks) to find fence regions. The fence segmentation task is solved by learning the relationship between fence texel joints and non-fence texel joints. Their proposed algorithms achieve great performance in video-based de-fencing. In addition to RGB video-based methods, method [9] incorporates depth map to enhance the estimated fence mask.

On the other hand, image-based de-fencing methods are more challenging because we have less information to detect fence regions and to fill-in the hidden part. Method [10] uses multi-focus images to remove occluders. Foreground occluders can be removed by synthesizing an object focusing image, an occluder focusing image and an image with flashlight. In method [11], stereo-pair of fenced images are used to remove fences. They compute disparity map corresponding to a pair of images using CNN. As described above, these multi-image methods are difficult for the need to prepare some images which meet the desired conditions. Unlike these methods, there are few single-image based de-fencing methods. Method [1] can automatically find near-regular foreground with [12] and complete the



(a) U-Net

(b) ResNet

Figure 2: Architectures of two networks.

missing region using texture-based inpainting [2]. This method is improved by [13] based on online learning approach. It can be stated that they make great progress in autonomous de-fencing. However, their lattice detection approaches are not able to detect irregular fences. Farid *et al.* [14] tackle this problem using a color based fence estimation algorithm and a hybrid inpainting algorithm. In spite that they can overcome weakness of existing lattice detection methods, we are required to input several fence pixels to predict fence regions.

In this paper, we propose a completely autonomous de-fencing algorithm. Fig. 1 illustrates example results of our proposed de-fencing network. By combining novel CNN methods and classical image processing techniques, our proposed network can detect fence regions and recover the hidden background. Experimental results demonstrate that ours outperforms other state-of-the-arts on various real-world fence images. Furthermore, since we identify this fence detection task as a regression problem, our proposed detection network can deal with irregular fence patterns. Additionally, in the image completion phase, we create synthetic fence images to train the network. Our learning-base approaches enable the network to be robust to a wide range of fence images.

Supporting methods

A convolutional neural network (CNN) is one technique of deep learning, which consists of an input and an output layer, as well as multiple hidden layers. Recent studies have reported that CNN-based methods achieve great success in image recognition and restoration. As numerous CNN architectures have been proposed, we refer to two popular networks: U-Net [15] and ResNet [16].

U-Net

In a CNN-based image classification task, while a convolution layer extracts the local features, a pooling layer ambiguates the detail location information. The image classification requires robustness to the object scale and the position aberration. On the other hand, a region segmentation task [17, 18, 15] needs to combine the local features with global position information. The architecture of U-Net [15] consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. Concatenating these captured context and precise location results in success in image segmentation tasks. Fig. 2a shows the revised version of U-Net. Compared to the original architecture, ours has less layers and zero padding is added before convolution to keep an output the same size as input.

ResNet

Residual leaning [16] of CNN overcomes the trade-off between training accuracy and the number of layers. As can be seen ResNet in Fig. 2b, the network has a skip connection, in which the output of the network is added to the input. That is based on the assumption that learning the residual mapping is more simple than directly learning mapping between the output and the input. This Residual-learning-based approach can solve several CNN problems and succeed in several image restoration tasks [19, 20].

Proposed methodology

Our proposed network is divided into two phases as Fig. 3. In the first phase, fence regions are automatically detected from an image and the binary fence mask is generated. In the second phase, the de-fenced image $\boldsymbol{y}$ is created from the fence image $\boldsymbol{x}$ and the estimated fence mask $\boldsymbol{m}$. We find that end-to-end network which directly produces clean images from fence images does not work well.

Fence detection

Our proposed fence detection network is based on U-Net architecture[15]. However, in order to adapt U-Net to a fence detection task, we improve the input data and the output data as follows.

Fence detection methods are categorized into two approaches. Methods in one group use colors of a fence chain. In an image, the range of the fence color is very limited. The similar pixels are grouped and fence regions can be estimated using the color pattern. It is experimentally found that those features are not enough to classify fence regions in difficult cases. The other methods find repeated elements from a given image. Neighbor relationships are assigned among a set of interest points and then similar interest points are collected. This regulatory-based approach achieves good results but is yet ineffective in distorted fence. For this reason, a fence detection task needs to extract both global features and local features. The architecture of U-Net is suitable to capture those sophisticated features. However, even though we directly input an image to U-Net, it results in overfitting. To solve this, we embed classical image processing approaches. First of all, we add luminance channel of image as input to prevent the fence detection network from strongly depending on colors. The Y channel component is calculated by Eq. 1, where $\boldsymbol{x}$ denotes a column vector of RGB fence image.

$$f_Y(\boldsymbol{x}) = (0.299 \quad 0.587 \quad 0.114)\,\boldsymbol{x} \tag{1}$$

Then, edge detection filters make the U-Net robust to fence scale, shape and color. After applying a Sobel filter and a Laplacian filter, the filtered images and RGBY components are concatenated before inputing to U-Net.

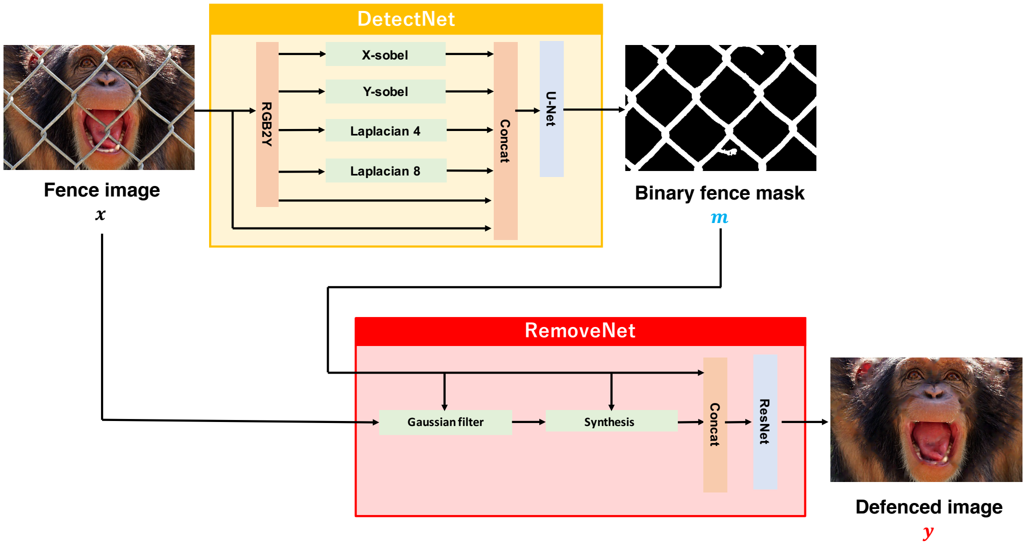


Figure 3: Our proposed network framework. "X-sobel" and "Y-sobel" correspond a vertical Sobel filter and a horizontal Sobel filter, respectively. Also, "Laplacian 4" means a four-nearest-neighbor Laplacian filter, and the same with "Laplacian 8". In RemoveNet, two images are synthesized by Eq. 4

There are other CNN-based fence detection approaches. Supervised learning in Machine learning tasks can be classified into two types: classification algorithms and regression algorithms. Classification algorithms are used in case of a limited number of outputs. For an example, in a classification algorithm that identifies fence images, the output would be the prediction of either "fence" or "not fence". Many methods define fence joints in which wires are crossed like the letter X as positive data. Negative data is defined as non-joints which even includes a part of wire. This approach is not enough for distorted fences and fences that does not include chain nodes. Therefore, we adopt a regression-based fence detection. We randomly crop patches from fence images to create fence image dataset as shown in Fig. 4. Since the output data takes values between 0 and 1, the binary mask $\boldsymbol{m}$ is produced by comparing against a threshold value. The network parameters $\boldsymbol{\Theta}_D$ are learned by minimizing the following objective function:

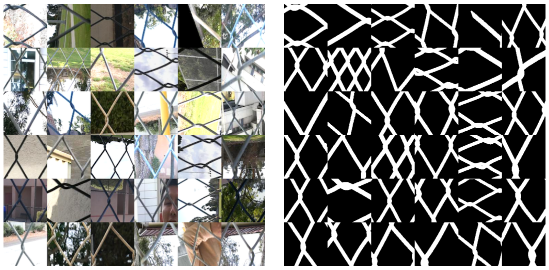
$$E_D(\boldsymbol{\Theta_D}) = \frac{1}{2N}\sum_{n=1}^{N} \|\boldsymbol{m}_n - f_D(\boldsymbol{x}_n;\boldsymbol{\Theta}_D)\|_2^2, \qquad (2)$$

where n indicates an image index of a total of N patches.

Fence removal

The restoration of lost regions is a quite challenging task because the fence is spread out over entire image and occludes a significant portion of the image. In this paper, we apply pre-processing before inputting to the ResNet [16]. In order to fill the hidden part by a fence, we predict an appropriate pixel from known surrounding pixels. We use simplest inpainting method as a pre-processing. Within a 11 × 11 window, by applying a Gaussian filter, the central missing part is replaced by smoothed version. In the window centered at (i_c, j_c), a kernel of a Gaussian filter at pixel (i, j) is defined as:

$$g(i,j) = \frac{1}{2\pi\sigma} e^{-\frac{(i-i_c)^2+(j-j_c)^2}{2\sigma^2}} \left(1 - m(i,j)\right), \qquad (3)$$



(a) Sample fence patches **(b)** Sample mask patches

Figure 4: Sample patches of our training dataset, which includes a wide range of fence patches with different scale, orientation and luminance.

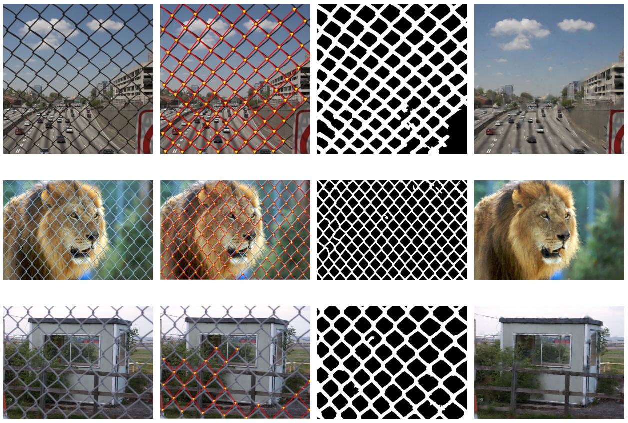
where standard deviation is empirically set to $\sigma = 2$. Note that fence regions do not have values and are not referenced in filtering. By using a Gaussian filter $\mathscr{F}_{\boldsymbol{g}}$, the restored image $\hat{\boldsymbol{x}}$ is obtained by Eq. 4.

$$\hat{\boldsymbol{x}} = \boldsymbol{x} \circ (\boldsymbol{1} - \boldsymbol{m}) + \mathscr{F}_{\boldsymbol{g}} \boldsymbol{x} \circ \boldsymbol{m}. \qquad (4)$$

where ∘ denotes an element-wise multiplication operator. Final de-fenced images are generated by the trained ResNet from pre-processed images. This is based on the hypothesis that ResNet recover high frequency domain of the missing portion. However, it is difficult to get ground truth clean images from real-world fence images. Hence, we create synthetic fence image in Eq. 5 to train the network.

$$\boldsymbol{x} = \boldsymbol{y} \circ (\boldsymbol{1} - \boldsymbol{m}) + (\boldsymbol{c} + \boldsymbol{n}) \circ \boldsymbol{m}. \qquad (5)$$

Note that $\boldsymbol{c}$ denotes color of the fence chain. Our dataset includes 5 colors of fence, which are dark gray, light gray, dark green, light green, blown and white. For robustness, Gaussian noise $\boldsymbol{n}$ is added on the colored fence. The objective function of the removal



(a) Fence image **(b)** Detected fence [25] **(c)** Our estimated mask **(d)** Our de-fenced image

Figure 5: Comparison of fence detection on real-world fence images. "Road" (top), "Lion" (middle) and "Prefab" (bottom).

fence network can be described as:

$$E_R(\mathbf{\Theta}_{\boldsymbol{R}}) = \frac{1}{2N}\sum_{n=1}^{N} \|\boldsymbol{y}_n - f_R(\hat{\boldsymbol{x}}_n, \boldsymbol{m}_n; \mathbf{\Theta}_R)\|_2^2, \qquad (6)$$

where $\mathbf{\Theta}_R$ indicates learned parameters including weights and biases.

Training dataset

We use different datasets in U-Net for detection and in ResNet for restoration. To train U-Net, we collect 545 real-world fence images and binary masks created by Du *et al.* [21]. From these images, we cropped $128 \times 128 \times 3$ patches. In order to increase the amount of data for training improvement, the cropped patches are randomly flipped, rotated, zoomed and brightened. A total of 27088 patches are used for training our detection network. In dataset for ResNet, we assemble $128 \times 128 \times 3 \times 30944$ patches. Fence images are created by combining fence mask in [21] with the clean outdoor images UCID dataset [22] and from the BSD dataset [23] used in [24].

Experimental results

To assess the performance of our proposed network, we newly test on real-world fence images. At first, we compare our fence detection performance to method [25]. Next, we compare fence removal performance to some conventional methods [1, 13, 14]. Last, we introduce interesting results and the limitation of our proposed method.

Parameter settings

Our proposed network have two phases for de-fencing. Each network is trained by Caffe framework developed by Berkeley AI Research (BAIR) and by community contributors. We start the training with a base learning rate of $\alpha_0 = 0.001$. The learning rate is decaying as $\alpha(t) = \alpha_0(1+\gamma t)^p$, where $\gamma = 0.0001$ and $p = 0.75$.

In the first step, we train the revised U-Net as Fig. 2a to detect fence regions from an image. Input data has eight channels that include a RGB-Y image and the filtered images. In order to combine local features and global features, we concatenate convolved features and downsampled one. Now, local features are the difference in gradient and global features are regular patterns of the fence. Downsampling is processed in a max pooling layer with a 2×2 filter. Downsampled features are convolved with 3×3 filters and a weight initializer Xavier [26]. To speed up the training time, we train every mini batch of eight patches. It takes approximately two hours to iterate calculation 10 thousands times.

In the second step, we use ResNet as Fig. 2b to restore the missing regions behind the fence. Input data is the Gaussian filtered RGB image and estimated binary fence mask. The convolutional layers are empirically set to $L = 20$. The weight of 3×3 layer convolutional layers is started with Xavier initializer [26]. A total of 10 thousands iterations with a mini bath of 12 patches run for about two and a half hours.

Fence detection evaluation

We compare the proposed fence detection algorithm to the other lattice detection method [25] on dataset from [1] and our test set. As a subjective comparison, Fig. 5 shows the visual comparison on real-world fence images. Since method [25] finds near regular structure, it suffers from distorted fence and drastic changes of background. On the other hand, our proposed method can detect even twisted part of fence on "Lion" and is unaffected by steep change of scenery. Thus, experimental results indicate that our proposed method is more robust to irregular pattern and complex background.

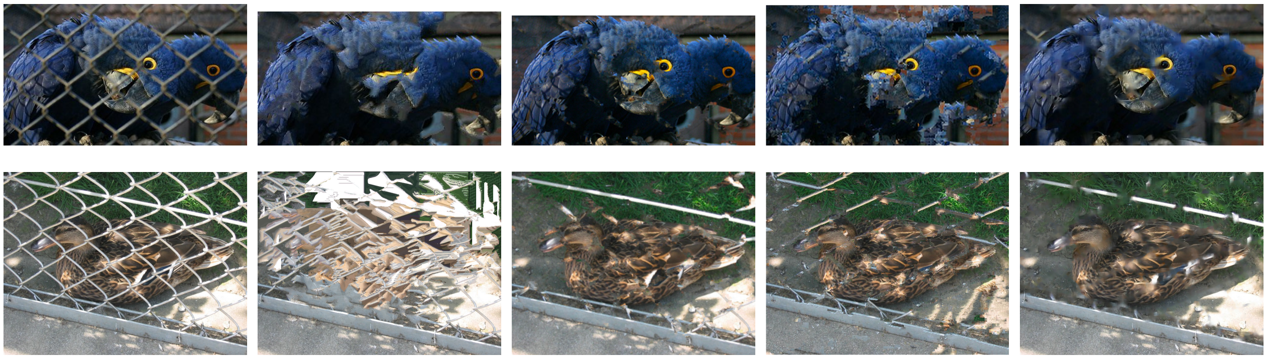
De-fencing evaluation

In this section, we compare the proposed de-fencing method with three state-of-the-arts [1, 13, 14]. Two of them [1, 13] use a lattice detection approach to detect fence regions. In recovery phase, method [1] and [13] adopt exemplar-based inpainting [2] for a single-image. On the other hand, method [14] propose the color-based fence detection and original hybrid inpainting algorithm. The de-fenced results are shown in Fig. 6. It is observed that method [1] and [14] fail to recover images with significant textures preserved. On the contrary, method [13] achieves better results than two conventional methods thanks to improved detection algorithm. However, unnatural artifacts remain upper right of image "Duck" due to their inpainting algorithm. Our method is able to precisely detect fence regions and to clearly fill the missing portions.

Irregular occluders

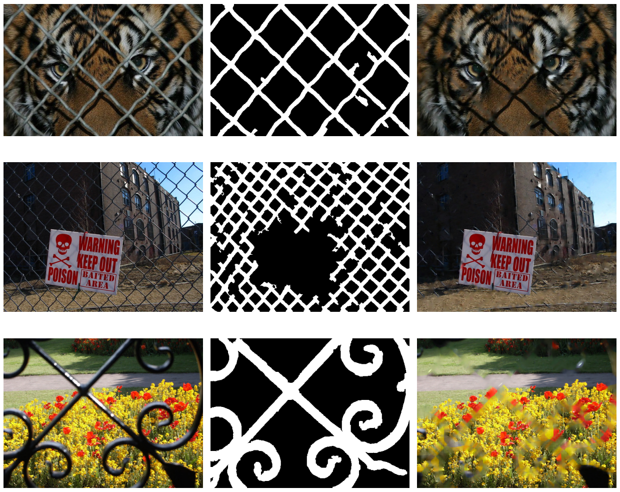
In real world, fences and barricades have diverse shapes and they are often occluded by other objects as bottom two of Fig.5. In the image "Warning", a signboard is hung on the fence. Although ours can discern the difference between the signboard region and fence regions, a part of fence is still remained. This is caused by the similar fence color with the background. Finally, an irregular shape of barricade in image "Garden" can be accurately detected by our methods. It appears that this result can be caused by our dataset which include diverse orientation and scale of fences.

Although our model achieves great performance on various real-world images, there are limits in some cases. Since our training dataset contains not so many fence types, it is hard to recognize certain types of fences. From the results in Fig. 8, it can be stated that our method does not work well when images are taken from a sharp angle and the fence shapes are unique. Nevertheless,



(a) Fence image **(b)** Liu *et al* [1] **(c)** Park *et al* [13] **(d)** Farid *et al* [14] **(e)** Ours

Figure 6: Comparison of de-fencing on real-wold images. “Bird” (top) and “Duck” (bottom).



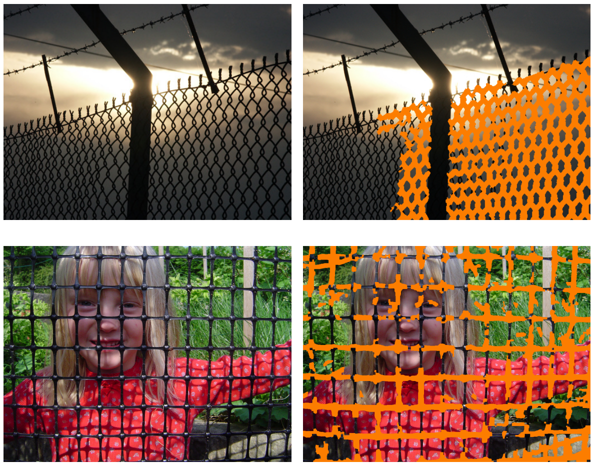
(a) Fence image **(b)** Estimated mask **(c)** De-fenced image

Figure 7: Irregular situations. From top to bottom, “Tiger”, “Warning” and “Garden”.

our trained network can partially detect fence regions. In order to tackle these difficult situations, our dataset and framework have room for improvement.

Conclusion

We have proposed an approach for de-fencing from a single-image using novel deep learning techniques. Ours is not directly removing fence regions but separating a fence detection task and a context recovery task. In our fence detection network, we adopt not only the U-Net architecture, but also combine the classical edge detection filters. Moreover our original patch dataset enables the network robust to irregular real-world fences. In our recovery network, we use ResNet after applying a Gaussian filter with the fence mask. The residual learning can restore missing part and add high frequency components such as textures. In addition, since we do not possess ground truth clean image corresponding to the fence images, we newly synthesize fence image for training dataset. Experimental results demonstrate that our proposed method achieves better performance for de-fencing than several state-of-the-arts. However, it also has limitation in cases such as images taken from a sharp angle.



(a) Fence image **(b)** Estimated mask

Figure 8: Failure cases of fence detection.

References

[1] Y. Liu, T. Belkina, J. H. Hays, and R. Lublinerman, “Image de-fencing,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[2] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[3] V. S. Khasare, R. R. Sahay, and M. S. Kankanhalli, “Seeing through the fence: Image de-fencing using a video sequence,” in *2013 IEEE International Conference on Image Processing*, Sep. 2013, pp. 1351–1355.

[4] S. Jonna, K. K. Nakka, and R. R. Sahay, “My camera can see through fences: A deep learning approach for image de-fencing,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov 2015, pp. 261–265.

[5] S. Jonna, K. K. Nakka, and R. R. Sahay, “Deep learning based fence segmentation and removal from an image using a video sequence,” in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 836–851.

[6] ——, “Towards an automated image de-fencing algorithm using sparsity,” *arXiv preprint arXiv:1612.03273*, 2016.

[7] R. Yi, J. Wang, and P. Tan, "Automatic fence segmentation in videos of dynamic scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 705–713.

[8] J.-M. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM journal on imaging sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[9] S. Jonna, V. S. Voleti, R. R. Sahay, and M. S. Kankanhalli, "A multimodal approach for image de-fencing and depth inpainting," in *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, Jan 2015, pp. 1–6.

[10] A. Yamashita, F. Tsurumi, T. Kaneko, and H. Asama, "Automatic removal of foreground occluder from multi-focus images," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 5410–5416.

[11] S. Jonna, S. Satapathy, and R. R. Sahay, "Stereo image defencing using smartphones," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 1792–1796.

[12] J. Hays, M. Leordeanu, A. A. Efros, and Y. Liu, "Discovering texture regularity as a higher-order correspondence problem," in *European Conference on Computer Vision*. Springer, 2006, pp. 522–535.

[13] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, "Image de-fencing revisited," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 422–434.

[14] M. S. Farid, A. Mahmood, and M. Grangetto, "Image defencing framework with hybrid inpainting algorithm," *Signal, Image and Video Processing*, vol. 10, pp. 1193–1201, 2016.

[15] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[19] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[20] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[21] C. Du, B. Kang, Z. Xu, J. Dai, and T. Q. Nguyen, "Accurate and efficient video de-fencing using convolutional neural networks and temporal information," *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2018.

[22] M. S. Gerald Schaefer, "Ucid: an uncompressed color image database," *Proc. SPIE*, vol. 5307, pp. 5307 – 5307 – 9, 2003. [Online]. Available: http://dx.doi.org/10.1117/12.525375

[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[24] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, June 2017.

[25] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, "Deformed lattice detection in real-world images using meanshift belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1804–1816, Oct 2009.

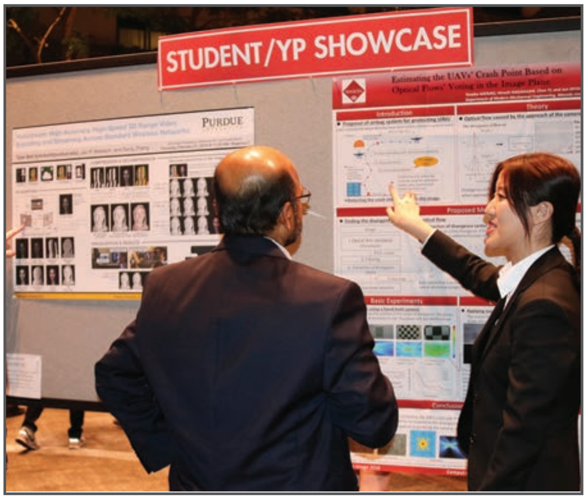
[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

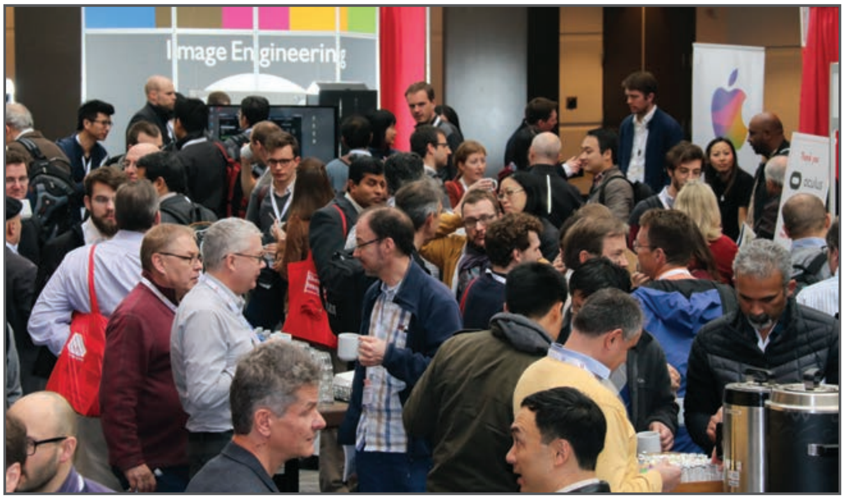
Author Biography

Takuro Matsui received the B.E. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2018. He is currently an M.E. student at Keio University, Yokohama, Japan, under the supervision of Prof. Masaaki Ikehara. His research interests are in the area of image restoration, image de-noising, and neural networks.

Takuro Yamaguchi received the B.E. and M.E. degrees and the Ph.D. in electrical engineering from Keio University, Yokohama, Japan, in 2014, 2016 ,and 2018, respectively. He is currently an associate professor at Keio University, Yokohama, Japan. His research interests are in the field of image reconstruction.

Masaaki Ikehara received the B.E., M.E. and Dr.Eng. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1984, 1986, ,and 1989, respectively. He was Appointed Lecturer at Nagasaki University, Nagasaki, Japan, from 1989 to 1992. In 1992, he joined the Faculty of Engineering, Keio University. From 1996 to 1998, he was a visiting researcher at the University of Wisconsin, Madison, and Boston University, Boston, MA. He is currently a Full Professor with the Department of Electronics and Electrical Engineering, Keio University. His research interests are in the areas of multi-rate signal processing, wavelet image coding, and filter design problems.





AR is a fundamental and transformative interface and platform.

20+ years from my initial working demo to today's AR displays

Optimistic that it won't take another 20 to reach ubiquitous consumer AR!

My first working AR prototype

Today's platforms

Optimis this is << 20

Ubiquitous consumer AR