Error feedback denoising network

Ruizhi Hou1 | Fang Li2

1 School of Mathematical Sciences, East China Normal University, Shanghai, China
2 School of Mathematical Sciences and Shanghai Key Laboratory of PMMP, East China Normal University, Shanghai, China

Correspondence
Fang Li, School of Mathematical Sciences and Shanghai Key Laboratory of PMMP, East China Normal University, Shanghai, China.
Email: fli@math.ecnu.edu.cn

Abstract
Recently, deep convolutional neural networks have been successfully used for image denoising due to their favourable performance. This paper examines the error feedback mechanism to image denoising and propose an error feedback denoising network. Specifically, we use the down-and-up projection sequence to estimate the noise feature. By the residual connection, the clean structures are removed from the noise features. The essential difference between the proposed network and other existing feedback networks is the projection sequence. Our error feedback projection sequence is down-and-up, which is more suitable for image denoising than the existing up-and-down order. Moreover, we design a compression block to improve the expression ability of the general 1 × 1 convolutional compression layer. The advantage of our well-designed down-and-up block is that the network parameters are fewer than other feedback networks and the receptive field is enlarged. We have implemented our error feedback denoising network on denoising and JPEG image deblocking. Extensive experiments verify the effectiveness of our down-and-up block and demonstrate that our error feedback denoising network is comparable with the state-of-the-art. The code will be open source. The source codes for reproducing the results can be found at: https://github.com/Houruizhi/EFDN.

KEYWORDS
deep convolutional neural networks, error feedback strategy, image denoising

INTRODUCTION

Image denoising is a fundamental task of computer vision and has many real applications. The aim of image denoising is to restore the clean image \( x \) from a noisy observation \( y \) following the degradation model

\[
y = x + \eta,
\]

where \( \eta \) is commonly assumed to be additive white Gaussian noise (AWGN).

The existing state-of-the-art denoising methods can be classified into non-local similar patch-based methods and deep convolutional neural network (CNN) based methods. The representatives of patch-based methods are block-matching and 3-D filtering (BM3D) [1] and the weighted nuclear norm minimization (WNNM) [2]. The denoising convolutional neural network (DnCNN) [3] becomes the first state-of-the-art network that adopts the convolutional layer and residual learning. The fast and flexible denoising network (FFDNet) [4] works on four sub-images of the noisy image, which expands the size of the receptive field about 2× times and speeds up the computing



This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. IET Image Processing published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology
using the same number of layers. The memory residual network (MemNet) [5] incorporates the recursive architecture and uses the gate unit to extend the memory further. A model-driven network, fractional optimal control network (FOCNet) [6], is inspired by the differential equation and adopts multi-scale operators on features of different sizes, which achieves good performance in the AWGN reduction. [7] uses a more complex noise model to estimate real-world noise.

Though the denoising deep CNNs make significant progress, the existing networks have some limitations. First, many existing denoising networks usually use plain CNN and work on the features of the same size [3, 4], where the receptive field is limited. Second, they only estimate the noise once and remove the noise by a residual connection, which is not enough to approach the true noise.

In order to overcome the above drawbacks of the existing CNNs, we borrow the idea of the error feedback strategy, which has been successfully used in image super-resolution [8]. In this letter, we propose an error feedback denoising network (EFDN) to introduce the feedback strategy into the image denoising task. The structure of EFDN is shown in Figure 1. Specifically, our main contributions are listed in the following.

**Down-and-up projection:** Error feedback [8, 9] is a strategy that corrects the estimating error by iteration. Motivated by this idea, We design a down-and-up (DU) block as our basic unit to realize a denoising network utilizing the error feedback strategy. In this way, the receptive field size improves more than two times compared with the up-and-down (UD) sequence. The detailed structure is given in Section 3.

**A novel compression block:** Many networks adopt $1 \times 1$ convolution to integrate all the preceding features [8, 10]. However, the $1 \times 1$ convolution fails to explore the connection in neighbouring regions when integrating features. To solve this problem, we design a special compression block to connect the neighbouring regions, which enlarges the receptive field further.

**More residual connections:** A single residual connection is not enough to estimate the noise effectively [3, 4]. Therefore we add the residual connection in every down-and-up block. These residual connections can help the down-and-up block remove the clean structure from the estimated noise features.

Extensive experiments are implemented on image denoising and JPEG image deblocking, showing that the proposed EFDN is comparable with the state-of-the-art in both visual quality and quantitative measures.

The paper is structured as follows: In Section 2, we summarize the network design of some current denoising methods and some related works of the feedback mechanism. In Section 3, we describe the structure of our proposed EFDN. In Section 4, we verify the effect of our down-and-up feedback mechanism and evaluate the performance of EFDN on denoising and JPEG image deblocking tasks. Finally, we summarize the advantage and some drawbacks of EFDN in Section 5.

## 2 RELATED WORKS

Recently, the state-of-the-art technologies for many computer vision problems are based on deep CNN, which also happens in the image denoising domain. The early applications of deep learning in the denoising task [11, 12] can hardly compare with the state-of-the-art. Since CNN is successfully used in the computer vision domain [13–15], many CNN denoisers [3, 4] are developed. A Gaussian denoiser based on CNN, i.e. DnCNN [3], outperforms many traditional methods by a large margin. The basic unit of DnCNN consists of the convolutional layer, the batch normalization (BN) layer [16], and the rectified linear unit (ReLU) [13]. This block becomes a typical architecture of the later denoising networks. The residual connection, namely the skip connection, is also adopted by DnCNN, which forces the network to estimate the noise.

The batch normalization layer [16] is first proposed to decrease the distribution fluctuation of the middle features. Many networks for the high-level computer vision task generally apply the BN layer, which enhances the network performance and accelerates the convergence [17–20]. Some denoising networks successfully use the BN layer [3, 5]. In [3], the BN layer and the residual layer benefit from each other and speed up training. However, this is only beneficial when the data is normal and does not work well for small samples. The BN layer can also give a better initialization value for the first vertical layer of the denoiser.
up the training. While in many low-level tasks, such as image super-resolution [10, 21], and the real-world denoising [7], the BN layer sometimes helps little or even degrades the reconstruction quality.

The feedback mechanism allows the algorithm to correct the error by iteration. This procedure is applied to various architectures in many computer vision scenes [9, 22–24]. [8] uses the back-projection stage with dense connection, becoming the first feedback network in the image super-resolution area. The up-and-down sampling operator is used to achieve feedback strategy, which consists of three up- or down- sampling layers and two skip connections. [25] plugs the feedback block as an operator in the recurrent neural network (RNN) architecture as in [23]. While so far, there is no relevant work in the image denoising task. The implementation of the error feedback strategy in denoising is to be explored.

3 PROPOSED METHOD

In this section, firstly, we will explain how we realize the feedback mechanism. Secondly, we will present the structure of the compression block. Finally, we will explain the network structure.

3.1 Feedback strategy

Dense deep back-projection networks (D-DBPN) [8] uses up- and down- projection for super-resolution. We remark that there are some differences between denoising and super-resolution. Super-resolution is to reconstruct details and structures, which are similar to the noise that has a high local variance. If we use up-projection first as in DBPN for image denoising, the noise will be enlarged. Another weakness of the up-and-down sequence is the large memory consumption due to the large feature size.

Since the down-sample layer can reduce the image size and extract the clearer feature, it is better to use the down-sample layer to extract features first. Hence, we design a new down-and-up (DU) block as the basic unit of EFDN, as shown in the top right of Figure 1. The down-projection extracts the noise and remove the structure information. After the up-projection layer and the residual connection, a more accurate noise estimation feature is generated.

Compared with D-DBPN [8], our sampling layers are simpler, and the sequence of down- and up-projection is different. Moreover, we utilize the batch normalization (BN) layer to improve denoising performance.

3.2 Compression block

The feature fusion layer is always a significant proportion of all network layers. Therefore it is necessary to improve the efficiency of feature fusion by a compression layer. We design a compression block (CB), as shown in the bottom right in Figure 1.

The widely used 1×1 convolution feature fusion layer cannot explore the information in neighbour regions. To enlarge the receptive field, we set the middle layer’s kernel size in the compression block as 3×3. Moreover, when the input feature number is relatively small, it is hard to sufficiently extract the information in the input features. Therefore we first increase the channel number of the input channel and the output channel, and k is the kernel size. The feature extraction block consists of two steps: feature mapping and shrinking [27], realized by Conv(ε, 4m, 3) and CB where ε = 1 or 3 is the channel number of the input image. Denote F as the extracted features from the noisy image y.

The body of EFDN consists of T DU blocks to predict the noise features. Let the t-th DU block be DUL(t) and N0 = F, then the t-th middle feature Nt is represented as

\[ N_t = \begin{cases} f_{DL}(N^0), t = 1, \\ f_{DL}(N_{t-1}), t = 2, 3, ..., T, \end{cases} \]

(1)

where [.] refers to the concatenation of the input features. When t > 1, the outputs of the preceding blocks are the input of step t.

The t-th DU block DUL(t) consists of down- and up-projection layers, denoted as fUL(t) and fDL(t). Specifically, fUL(t) and fDL(t) are Conv(m, m, 6) and DeConv(m, m, 6) with stride = 2. All preceding features with the same size are used as the inputs of fUL(t) or fDL(t). Let Ut and Dt be the outputs of fUL(t), fDL(t). When t > 1, they are represented by

\[ D_t = f_{DL}(f_{CB}(U^t, U^t, ..., U^{t-1})), \]
\[ U_t = f_{UL}(f_{CB}(D^t, D^t, ..., D^{t-1})), \]

(2)

where \( U^0 = N^0 \) and \( f_{CB} \) is the CB. There is no CB when \( t = 1 \).
Let \( N_i \) be the final estimated noise, then
\[
N_i = f_R([N^1, \ldots, N^T]),
\]
where \( f_R \) is composed of CB and \( Conv(m, \epsilon, 3) \).

The restoration \( \hat{x} \) is obtained by residual connection [3]:
\[
\hat{x} = y - N_i.
\]

Given the training set \( \{(y_i, x_i)\}_{i=1}^M \), where \( y_i \) is the contaminated image and \( x_i \) is the corresponding clean image, the loss function of the whole network \( F(\cdot, \Theta) \) with weights \( \Theta \) is
\[
\mathcal{L}(\Theta) = \frac{1}{2M} \sum_{i=1}^M \| F(y_i, \Theta) - (y_i - x_i) \|_2^2,
\]
where \( \| \cdot \|_F \) is the Frobenius norm and \( M \) is the number of the training data pairs.

4 | EXPERIMENTS

In this section, we first explain the experimental setting and how to train the network. Secondly, we verify the effectiveness of EFDN via the ablation study. We then evaluate EFDN on denoising and JPEG image deblocking.

4.1 | Experimental setting

We use 800 images from the DIV2K high-resolution images [28] as our train dataset. During training, we generate the clean images \( \{x_i\}_{i=1}^M \) every epoch by randomly cropping \( M = 16 \times 1600 \) patches of size \( 80 \times 80 \) then doing augmentation by a random combination of three modes: Rotating \( 90^\circ \), vertical flipping and horizontal flipping. Then we add AWGN with the specific noise level to the augmented clean images to generate the noisy images \( \{y_i\}_{i=1}^M \). For blind image denoising, we add noise with different noise levels uniformly chosen from \([0,40]\) to different mini-batch. In JPEG deblocking task, the patches number \( M = 16 \times 3200 \), and we generate the low-quality images \( \{y_i\}_{i=1}^M \) by compressing with different quality factor \( Q \in (0, 40) \). The parameters of EFDN are \( T = 4, m = 64 \) and \( m' = 128 \). If not specified, all retrained networks in our experiment adopt the settings above.

The ADAM optimizer [29] is adopted to optimize the loss function, and the hyper-parameters are \( \beta_1 = 0.9, \beta_2 = 0.99, \) and \( \epsilon = 10^{-8} \). The weights are initialized following DnCNN [3]. The training epoch is 100 and the batch size is 16. The learning rate is set by \( 10^{-3} \) and multiplies 0.1 every 30 epochs. We implement our networks with the PyTorch framework. We set the same random seed when the training process begins. The training process takes about 5 hours for EFDN.

4.2 | Ablation study

In this subsection, we compare the effect of up-and-down and down-and-up feedback strategies in EFDN and D-DBPN. D-DBPN is originally designed for super-resolution. Therefore, we modify D-DBPN for denoising by replacing its last deconvolutional layer with a convolution layer. The original projection sequence of D-DBPN [8] is up-and-down. We also verify the effect of our proposed compression block. We implement two compression methods including 1 convolution, and computing time (Time). The computing time is the average run-time of running 100 times on a 256 \( \times \) 256 image. Because the difference between UD and DU is only the sequence of projection layers, the number of parameters of DU and UD are the same. The larger feature size of UD leads to about quadruple computing time. The size of the receptive field of DU is twice as large as UD’s. The receptive field of D-DBPN is larger than EFDN but more computationally expensive. Moreover, the PSNR of our EFDN is comparable with D-DBPN with fewer parameters.

Table 1 also shows that the proposed compression block both enlarges the receptive field and improves the reconstruction quality.

Figure 2 shows PSNRs curves during the training process of networks with different feedback strategies and depth. All curves show an upward trend in the first 30 epochs and improve a lot in the 30th epoch. After 30 epochs, PSNR fluctuates around a relatively high value. The left picture shows the training process of different feedback blocks, in which the UD block with CB has the highest PSNR value and the plain block is the lowest. In the right figure, the PSNR of \( T = 4 \) is relatively lower. The PSNRs of \( T = 8 \) and 12 are similar.
4.3 Experiment results

In this subsection, we test our proposed networks on denoising and JPEG image deblocking.

The results of PSNR and SSIM [30] are reported. Some result pictures are displayed to compare the visual quality. PSNR or SSIM values are cited from the original papers if available; otherwise, we use their open codes to test.

**Synthetic image denoising:** We test our proposed networks on gray images with AWGN, from Set12 [3], BSD68 [32] and Urban100 [33] datasets. To verify the efficiency of our implementation of the feedback mechanism, we compare the PSNR between different methods including BM3D [1], WNNM [2], TNRD [34], DnCNN [3], FFDNet [4], and N^3Net [35]. The PSNR values are reported in Table 2. It shows that our proposed EFDN has the highest PSNR. On Set12, Set68, and Urban100, the PSNR of EFDN outperforms other methods by 0.19 dB, 0.1 dB, and 0.45 dB at least. We also compare the visual result for gray image denoising, as shown in Figure 3.

Figure 3 shows the restoration of the gray image. The noisy images are corrupted by the AWGN with the standard deviation of \( \sigma = 50 \). The reference regions are enlarged and located on the bottom. In Figure 3, the left enlarged image contains the trees and the building, and the right local image includes the edge of the building. WNNM and FFDNet can restore the sharper edge of the building, while there are some obvious artefacts. The results of our EFDN are cleaner and have fewer artefacts than others.

For the colour image, we add AWGN on images from Set5, LIVE1, and CBSD100 and report the average PSNR of the results. The compared method is DnCNN and FFDNet. As the PSNR results in Table 3, the average PSNR of our CEFDN (EFDN for colour images) outperforms CDnCNN and FFDNet by 0.21 dB and 0.33 dB respectively. The visual results are shown in Figures 4 and 5.

Figures 4 and 5 show the results for AWGN denoising on colour images. In Figure 4, EFDN can restore obvious textures in the enlarged beak of the parrot. Moreover, the outline of the eye is clearer than others. In the enlarged local image of Figure 5, EFDN also keeps more details.

**Real image blind denoising:** We train CEFDN using AWGN with \( \sigma \in [0, 40] \) for blind denoising and test it on NC12 [31] and Nam [36] real-world dataset. The results are shown in Table 4. In Table 4, our CEFDN outperforms the second best method MC-WNNM by 0.53 dB in PSNR and 0.0043 in SSIM.
FIGURE 3  Visual quality comparison for gray image denoising on noise level $\sigma = 50$. Two different regions are marked in red and green and enlarged on the bottom. The PSNR values (dB) are given in the parentheses. The tested picture is the 11-th image in BSD68.

TABLE 3  Average PSNR (dB) for denoising on colour images. The noise level is 25

| Dataset   | CDnCNN [3] | FFDNet [4] | CEFDN |
|-----------|------------|------------|-------|
| Set5      | 32.16      | 32.05      | 32.44 |
| LIVE1     | 31.49      | 31.36      | 31.69 |
| CBSD100   | 31.12      | 30.99      | 31.26 |

Figures 6 and 7 show the denoising results of two real-world images in NC12 [31]. Figure 6 is in a dark environment. In the left enlarged image, the result of our CEFDN is smoother than CBDNet. The boundary between the hair and the background is not much clear in CBDNet, while ours can reserve a rough outline. The right enlarged image is a part of the background wall, which should be clean. The result of CEFDN is cleaner than CBDNet in the local bottom image. In the bottom left enlarged image of Figure 7, the result of CBDNet is over-smooth and the edges are vague. The dog's whisker is sharper in the result of EFDN. In comparison, the result of CEFDN is more natural. In the bottom right images, the reconstruction of CEFDN is cleaner and has a bolder outline.

JPEG deblocking: In JPEG deblocking task, we generate the low-quality image with different quality factor $Q \in [0, 40]$ to train EFDN. The trained model in this experiment is named EFDN-DB. We compare our EFDN-DB with AR-CNN [38],

FIGURE 4  Visual results on colour images in LIVE1 with noise level $\sigma = 25$. The reference regions are marked and enlarged. The PSNR values (dB) are given in the parentheses.
HOU AND LI

**FIGURE 5** Visual results on colour images in BSD100 with noise level $\sigma = 25$. The PSNR values (dB) are given in the parentheses.

**TABLE 4** The average PSNR and SSIM for denoising on 15 randomly cropped images in Nam [36]

| Method       | PSNR/SSIM  | PSNR/SSIM  | PSNR/SSIM  | PSNR/SSIM  |
|--------------|------------|------------|------------|------------|
| Index        |            |            |            |            |
| Average      | 37.71/0.9851 | 33.86/0.9594 | 36.50/0.9795 | 38.24/0.9894 |

TNRD [34], and DnCNN-3 [3] on JPEG deblocking task. There are four AR-CNNs trained for the JPEG quality factors 10, 20, 30, and 40, respectively. For TNRD, three models for JPEG quality factors 10, 20, and 30 are trained. DnCNN-3 is a single model trained for JPEG quality factors from 5 to 99. As the results in Table 5, our EFDN-DB outperforms the three compared methods and surpasses DnCNN-3 by 0.29-0.41 dB in PSNR on all quality factors. Averagely, our EFDN outperforms the second-best method DnCNN-3 by 0.32 dB in PSNR and 0.0063 in SSIM.

**Figure 8** shows the results of JPEG image deblocking. The quality factor is $Q = 10$. The details are better reconstructed by our proposed EFDN than others; see the textures in the enlarged regions.

**Computing time:** Figure 9 shows the computing time and the corresponding PSNR values of the compared CNN-based methods. The vertical axis is the average PSNR results on Set12 when images are contaminated by AWGN with $\sigma = 50$, and the horizontal axis is the average time on the same 256 $\times$ 256 image running 100 times. The computational speed of EFDN is slightly slower than DnCNN and FFDNet, while the PSNR is higher than them about 0.2 dB. Our EFDN is dominantly faster than N$^3$Net.

**5 | CONCLUSION**

This paper examines the error feedback strategy to the image denoising problem and design a down-and-up feedback...
mechanism to denoise effectively. The down-projection can extract more abstract feature and remove the noise, and the up-projection reconstruct the clean structure. Then by the residual connection, the clean structure can be removed from the estimated noise feature step by step. This down-and-up feedback sequence is essentially different from the up-and-down sequence of other existing feedback networks. By this sequence, the computing time can be saved because of the smaller middle features. Moreover, the well-designed compression block can improve the expression ability compared with the single convolution layer. Experimental results on image denoising verify the good visual quality and the leading PSNR results of our EFDN.

| Dataset | Quality factor | AR-CNN [38] PSNR/SSIM | TNRD [34] PSNR/SSIM | DnCNN-3 [3] PSNR/SSIM | EFDN-DB PSNR/SSIM |
|---------|----------------|------------------------|----------------------|-----------------------|-------------------|
| Classic5 | 10             | 29.03/0.7929           | 29.28/0.7992         | 29.40/0.8026          | 29.81/0.8140      |
|         | 20             | 31.15/0.8517           | 31.47/0.8576         | 31.63/0.8610          | 31.99/0.8673      |
|         | 30             | 32.51/0.8806           | 32.78/0.8837         | 32.91/0.8861          | 33.24/0.8908      |
|         | 40             | 33.34/0.8953           | -                    | 33.77/0.9003          | 34.07/0.9040      |
|         | 10             | 28.96/0.8076           | 29.15/0.8111         | 29.19/0.8123          | 29.49/0.8219      |
| LIVE1   | 20             | 31.29/0.8733           | 31.46/0.8769         | 31.59/0.8802          | 31.84/0.8862      |
|         | 30             | 32.67/0.9043           | 32.84/0.9059         | 32.98/0.9090          | 33.27/0.9136      |
|         | 40             | 33.63/0.9198           | -                    | 33.96/0.9247          | 34.25/0.9285      |

**FIGURE 7** Denoising results in a real-world noisy image from NC12 [31]

**FIGURE 8** Visual results of JPEG image deblocking. One different region is marked in red and enlarged on the top left. The PSNR values (dB) are given in the parentheses.
Though the computing speed is slightly slower than DnCNN, the denoising quality is better. So the loss of computing speed is acceptable. The proposed EFDN has a weakness that some details or textures are over-smoothed. To overcome this drawback, we will consider redesigning the basic block of the network, which is left as our future work.

ACKNOWLEDGEMENTS
This work is supported in part by the National Natural Science Foundation of China (NSFC) (No. 61731009, No. 11671002), the Fundamental Research Funds for the Central Universities, and Science and Technology Commission of Shanghai Municipality (No. 19JC1420102, No. 18dz2271000).

REFERENCES
1. Kostadin, et al.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Process. 16 (8), 2080–2095 (2007)
2. Gu, S. et al.: Weighted nuclear norm minimization and its applications to low level vision. Int. J. Comput. Vision 121 (2), 183–208 (2017)
3. Zhang, K. et al.: Beyond a gaussian denoiser: residual learning of deep cnns for image denoising. IEEE Trans. Image Process. 26 (7), 3142–3155 (2016)
4. Zhang, K., Zuo, W., Zhang, L.: Fidnet: toward a fast and flexible solution for cnn based image denoising. IEEE Trans. Image Process. 27 (9), 4608–4622 (2018)
5. Chen, R. et al.: Persistent memory residual network for single image super resolution. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops pp. 809–816 (2018)
6. Jia, X. et al.: Focnet: A fractional optimal control network for image denoising. 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6054–6063 (2019)
7. Guo, S. et al.: Toward convolutional blind denoising of real photographs. 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1712–1722 (2019)
8. Haris, M., Shakhnarovich, G., Utkin, N.: Deep back-projection networks for super-resolution. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1664–1673 (2018)
9. Carreira, J. et al.: Human pose estimation with iterative error feedback. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4733–4742 (2016)
10. Zhang, Y. et al.: Residual dense network for image super-resolution. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2472–2481 (2018)
11. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with bm3d? 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2392–2399 (2012)
12. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks, advances in neural information processing systems (nips). pp. 341–349 (2012)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. pp. 1097–1105. Curran Associates, Inc. (2012)
14. Ren, S. et al.: Faster r-cnn: towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems (NIPS) pp. 91–99 (2015)
15. Karpathy, A. et al.: Large-scale video classification with convolutional neural networks. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1725–1732 (2014)
16. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. 37 448–456 (2015)
17. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. 39 2481–2495 (2017)
18. He, K. et al.: Identity mappings in deep residual networks. 2016 european conference on computer vision (eccv). pp. 630–645 (2016)
19. Isola, P. et al.: Image-to-image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition pp. 4700–4708 (2017)
20. Huang, G. et al.: Densely connected convolutional networks. The IEEE Conference on Computer Vision and Pattern Recognition pp. 3147–3155 (2017)
21. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. 2017 IEEE Conference on Computer Vision and Pattern Recognition pp. 3147–3155 (2017)
22. Li, K., Hanhuran, B., Malik, J.: Iterative instance segmentation. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3659–3667 (2016)
23. Zamir, A.R. et al.: 2017 ieee conference on computer vision and pattern recognition. pp. 1308–1317 (2017)
24. Cao, C. et al.: Look and think twice: capturing top-down visual attention with feedback convolutional neural networks. 2015 IEEE International Conference on Computer Vision (ICCV) pp. 2956–2964 (2015)
25. Li, Z. et al.: Feedback network for image super-resolution. 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3867–3876 (2019)
26. Xie, S. et al.: Aggregated residual transformations for deep neural networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition pp. 5987–5995 (2017)
27. Dong, C., Loy, C.C., Tang, X.: Springer, Accelerating the super-resolution convolutional neural network. 2016 European Conference on Computer Vision (ECCV) pp. 391–406 (2016)
28. Argustsson, E., Timofte, R.: Nitre 2017 challenge on single image super-resolution: dataset and study. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops pp. 126–135 (2017)
29. Kingma, D.P. , Ba, J.: Adam: a method for stochastic optimization. arXiv preprint, (2014)
30. Wang, Z. et al.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), (2004)
31. Lebrun, M., Colom, M., Morel, JM: The noise clinic: a universal blind denoising algorithm. 2014 IEEE International Conference on Image Processing pp. 2674–2678 (2014)
32. Arbelaez, P. et al.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (5), 898–916 (2011)
33. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5197–5206 (2015)
34. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. 39 1256–1272 (2017)
35. Plötz, T., Roth, S.: Neural nearest neighbors networks. pp. 1095–1106 (2018)
36. Nam, S. et al.: A holistic approach to cross-channel image noise modeling and its application to image denoising. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
37. Xu, J. et al.: Multi-channel weighted nuclear norm minimization for real color image denoising. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 1105–1113 (2017)
38. Dong, C. et al.: Compression artifacts reduction by a deep convolutional network. 2015 IEEE International Conference on Computer Vision (ICCV) pp. 576–584 (2015)

How to cite this article: Hou R, Li F. Error feedback denoising network. IET Image Process. 2021;15:1508–1517.
https://doi.org/10.1049/ipr.12121