FormResNet: Formatted Residual Learning for Image Restoration

[1]Jianbo Jiao, [2]Wei-Chih Tu, [3]Shengfeng He, [1]Rynson W. H. Lau
[1]Department of Computer Science, City University of Hong Kong;
[2]National Taiwan University; [3]South China University of Technology
jianbjiao2-c@my.cityu.edu.hk, wctu@media.ee.ntu.edu.tw, hesfe@scut.edu.cn, rynson.lau@cityu.edu.hk

Abstract

In this paper, we propose a deep CNN to tackle the image restoration problem by learning the structured residual. Previous deep learning based methods directly learn the mapping from corrupted images to clean images, and may suffer from the gradient exploding/vanishing problems of deep neural networks. We propose to address the image restoration problem by learning the structured details and recovering the latent clean image together, from the shared information between the corrupted image and the latent image. In addition, instead of learning the pure difference (corruption), we propose to add a "residual formatting layer" to format the residual to structured information, which allows the network to converge faster and boosts the performance. Furthermore, we propose a cross-level loss net to ensure both pixel-level accuracy and semantic-level visual quality. Evaluations on public datasets show that the proposed method outperforms existing approaches quantitatively and qualitatively.

1. Introduction

A lot of imaging algorithms/applications assume the input images to be clean and of high-resolution. However, in practice, these images may suffer from corruption, *e.g.*, noise, or low resolution due to the limitation of digital imaging. The image restoration task is to handle this problem and recover the latent clean image, including image denoising, super-resolution, artifact removal, *etc*. In general, a corrupted image I_C can be modeled as the latent clean image I added with a certain type of corruption C. Image restoration aims to recover clean image I by separating it from corruption C. Hence, if C can be accurately estimated, I can then be well recovered. Notwithstanding the demonstrated success, most of the traditional image restoration methods are problem-specific and cannot be easily adapted to different problems.

In recent years, deep convolutional neural networks (CNNs) have become very popular in solving many high-level vision problems. There are also some emerging works applying CNNs to low-level vision tasks like image denoising [18, 5], by directly learning the mapping function from a noisy image to its clean version. However, learning such a dense mapping is prone to the gradient vanishing/exploding problems of deep CNNs [2, 16]. Besides, most existing CNN-based methods train the networks based on the pixel-level ℓ_2 norm (MSE) objective, which can easily produce blur artifacts in the final inference.

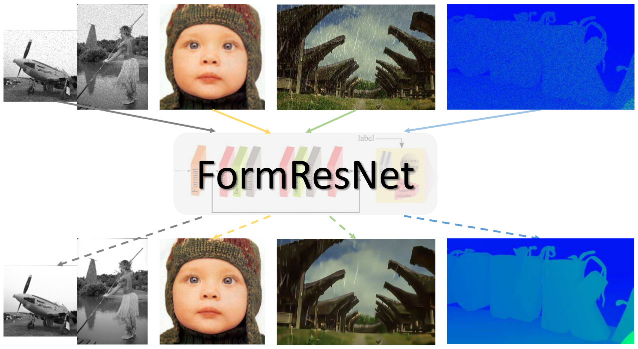


Figure 1. Application illustration of the proposed FormResNet on image restoration tasks such as denoising, super-resolution, rain removal, and depth enhancement (left to right).

To resolve the above problems, we model the image restoration problem as learning the residual by CNN, in which the corruption is considered as "residual information". In addition, we observe that the clean image and the corrupted image share similar information in most homogeneous regions, but differ more in highly-structured (*e.g.*, texture) regions. Since both the structured regions and corruptions are high-frequency signals in most cases, directly learning the high-frequency residual is similar to approximating a low-pass filter, and the highly-structured details in the latent image are also filtered out (see Section 3.2). Thus, we propose to extend the network to learn the formatted residual information. To this end, we add a "residual formatting" layer to format the residual to sparsely distributed and more structured information, which is favored by deep residual learning [16]. The highly structured details can then be reconstructed in the following layers.

We further introduce a cross-level loss net to reduce the artifacts caused by the conventional pixel-level ℓ_2 norm. Two gradient layers are added to model the loss in the gradient domain. Besides, high-level similarity measured in the feature domain is taken into consideration, which helps improve the visual quality of the final result. We refer to the final framework as *FormResNet*. Fig. 1 shows some applications of the FormResNet. Extensive evaluation on public datasets shows that the proposed framework outperforms the state-of-the-art denoising and other image restoration methods.

The main contributions of this work include:

1. We design a new deep neural network to learn the formatted residual information to reconstruct the structural details in image restoration.
2. We propose a cross-level loss net that supervises the network based on both pixel-level and high-level similarities, resulting in better visual quality compared to traditional MSE-based loss.
3. We achieve state-of-the-art performance. Specifically, the proposed method outperforms existing methods across different noise levels and noise types in a single model, and is shown to be able to handle other image restoration problems.

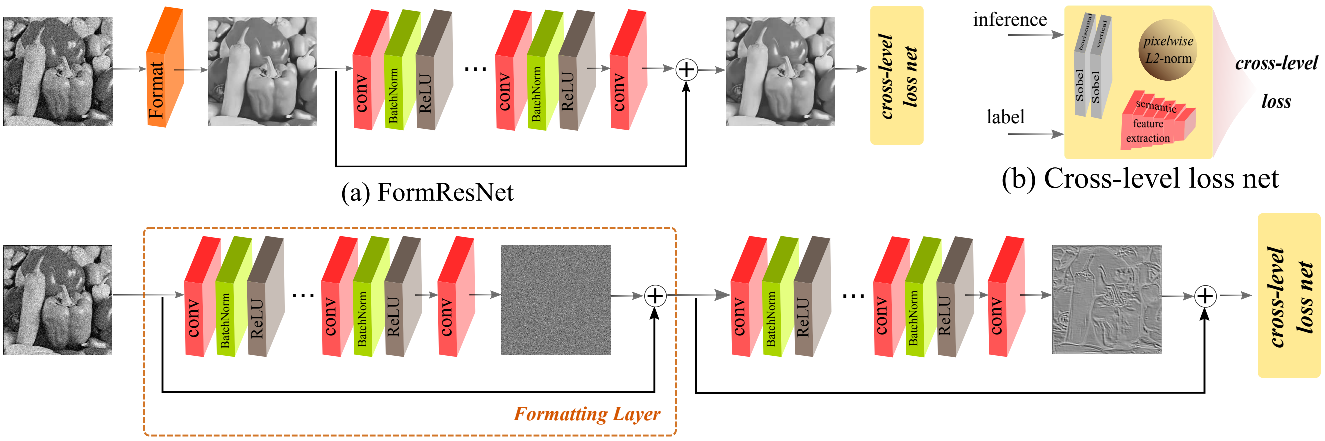
Although significant improvement has been achieved, most of these methods focus on learning the dense mapping from observed image to the target one directly, while for many image restoration problems such mapping is close to an identical mapping, which is difficult to learn and prone to the gradients vanishing/exploding problems [2, 16, 19]. The recent proposed residual learning scheme [16] aims to solve these problems for deep neural networks and achieve superior performance on various high-level problems like classification, detection, segmentation, *etc*. In low-level problems, residual learning has also shown its effectiveness in single image super-resolution [19], in which a very deep network is learned efficiently with the help of residual learning. A recent work proposes a deployed CNN [39] with similar residual structure [19] for image denoising and achieves promising results. Unlike previous residual learning that either stacks several blocks or directly learns the difference, the proposed method uses a simple architecture by introducing a residual formatting layer to model stochastic residual information into a more structured one. It can handle different noise types and levels in a single network and generalizes well to other image restoration tasks. To our knowledge, this is the first approach to tackle multiple noise types and noise levels in a single model.

2. Related Work

Image Restoration. Image restoration is a widely studied problem in computer vision and remains an active area. Extensive studies have been conducted to solve the problem in the past decades. We refer the readers to a survey [27] on image restoration for more details. Generally, these methods can be categorized into single image based methods and multiple image based methods. Single image based methods like BM3D [7] utilizes non-local information from the corrupted image itself to remove artifact like noise. As image restoration is an ill-posed problem, image priors learned from external dataset are also widely used [11, 30, 40, 9, 20] to reconstruct the latent clean image. Usually the above methods focus on a specific kind of corruptions and the result image tend to be over-smoothed.

Deep Learning for Image Restoration. In recent years, CNN has been applied to some low-level vision problems, including image filters [36, 21, 24], super-resolution [8, 19], denoising [5, 34], deconvolution [35], stereo matching [38], optical flow [10], among others. Xie *et al*. [34] combine sparse coding and deep networks to handle problems like complex pattern removing in image inpainting and denoising. Burger *et al*. [5] learn a plain multi-layer perceptron based on a large dataset for denoising, and obtain competitive results to BM3D. Since then, other multi-layer models [31, 6, 33] are also proposed for image restoration.

Objective Function. As CNN based methods are data-driven, an objective/loss function is needed to constrain the training process. Usually the objective is to minimize a ℓ_2 norm (or MSE) loss $L = \|T - I\|^2$ which is used to construct the loss between the network inference I and the target label T. For regression problems like image restoration, such kind of ℓ_2 norm has been widely used in the literature [36, 21, 19, 39]. However, the ℓ_2 norm is prone to cause over-smoothed result. While most deep learning methods focus on crafting the network structure, little attention has been paid on the design of loss function. In [12], Gatys *et al*. use the feature maps extracted from a basic CNN to model the loss function for image style transfer. As in the application of style transfer pixel-wise accuracy is not that important, the feature map based objective function leads to a good visual quality. The recent popular GAN (Generative Adversarial Network) [13, 28] directly uses a CNN which named discriminator to supervise the training process of the front generator network. Such ingenious structure not only supervises the generator training but also improves the objective part (discriminator) simultaneously. However, a stable training is not easy to achieve for the GAN. In this paper, we propose a new stable cross-level loss net that integrates pixel-level and semantic-level similarities, so that both pixel-wise accuracy and high-level visual quality are guaranteed.

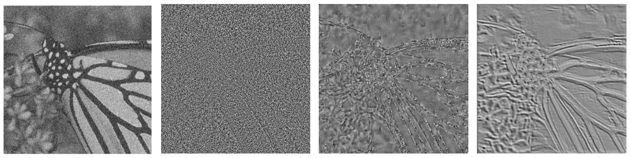


(a) FormResNet

(b) Cross-level loss net

(c) RecursiveFormResNet

Figure 2. Proposed network structure. (a) is the general framework of FormResNet, in which the orange block represents the formatting layer; (b) is the cross-level loss net that incorporate pixel-wise ℓ_2 norm, gradient consistency, and semantic high-level features, to better describe the similarity between network inference and ground truth label; (c) is the RecursiveFormResNet that takes convolutional layers as the formatting layer in (a). This structure can be performed in a recursive fashion. $\oplus$ denotes pixel-wise subtraction/summation.



(a) Noisy image

(b) Inference by DiffResNet

(c) Noise difference

(d) Inference by FormResNet

Figure 3. Different residual information. (a) is a noisy image corrupted by Gaussian noise with $\sigma = 25$; (b) is the inference output of DiffResNet, which includes both the noise and high-structured regions; (c) is the difference between the ground truth noise and (b); (d) shows the inference from FormResNet. High-structured details (c) are also removed when doing denoising (subtract (b) from (a)) by DiffResNet while (d), after the formatting layer, well recovers the structured details.

3. Proposed Method

In many image restoration problems, the observed image is similar to the target latent one. Taking denoising as an example, the "difference" between noisy image and clean image is the pure noise itself. We observe that most homogeneous regions in the corrupted image and clean image share similar low-frequency information, while the highly-structured (high-frequency) regions between them are relatively different. Due to the inherently different properties in these two regions, learning the difference map only cannot well reconstruct the high-frequency regions, which is shown in Fig. 3(c). As a result, we bias the learning process to structured regions, while the homogeneous regions are mainly handled by a formatting layer. In this way, the residual after formatting layer refers to the structure or fine details of the image (Fig. 3(d)).

3.1. Learning the Difference

Conventional CNN-based methods usually learn the mapping from corrupted image to clean image directly [8, 35, 5]. Whereas for deep neural network, during the training all the image details require to be preserved through many layers. This is prone to the gradient vanishing and exploding problems [2, 16, 19]. Thus we propose to learn the residual mapping $\hat{C} = f(I_C)$ that only sparse residual information needs to be learned. We name this network as DiffResNet which consists of fully convolutional layers and a skip connection from the network input to the inference. A similar structure is proposed in a concurrent work [39]. D layers are used in DiffResNet: the first layer is a *conv.* layer with 64 filters of size $3 \times 3 \times c$ ($c = 1$ for gray-scale image and $c = 3$ for color-scale image) followed by a ReLU (rectified linear unit); the following layers except the last layer are of the same type consist of 64 filters of size $3 \times 3 \times 64$ and followed by ReLUs; the last layer which is used for reconstruction, is a single filter of size $3 \times 3 \times 64$. The input of the network is the corrupted image, and the inference is the residual, *i.e.*, corruption. The inference is subtracted from the corrupted input to form the loss function as $\frac{1}{2}\|I - (I_C - f(I_C))\|^2$. By minimizing this objective over the training set $\{I_C^{(i)}, I^{(i)}\}_{i=1}^N$ we can learn the parameters for the model.

3.2. Learning the Formatted Residual

Due to learning the residual instead of dense mapping, the above DiffResNet architecture is shown to achieve better performance and converge faster than previous "direct learning" (Fig. 4). Such DiffResNet can be considered as approximating a low-pass filter. The advantage of low-pass filter is that the high-frequency artifact (*e.g.* noise) can be

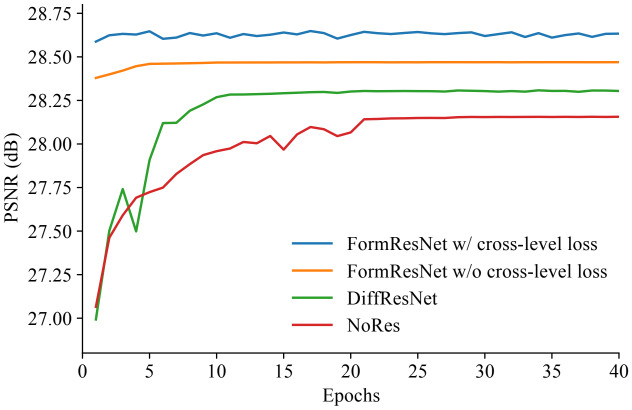


Figure 4. Performances for residual learning. We compare FormResNet with and without the proposed cross-level loss net, DiffResNet, and NoRes (training without residual learning).

filtered, whereas the drawback is also the "low-pass" property. Besides the artifact, other high-frequency information (structures, edges, *etc*.) is also filtered out. Thus the latent highly-structured regions are difficult to be recovered, as shown in Fig. 3(c). This is because the high-frequency structured regions show inherently different properties to the homogeneous regions.

Residual Formatting Layer. We propose a new structure to handle this problem, and the network architecture is shown in Fig. 2. Unlike DiffResNet, which learns the pure difference, we add a "residual formatting" layer (orange part in Fig. 2) to format the residual to be more structured information. The proposed formatting layer aims to reduce the corruption on the input image. It is a non-linear operator, which can be constructed by conventional method (*e.g*. BM3D) or neural network (Fig. 2 (c)). Through this formatting layer, the residual map lies more on the image details, instead of random distributed noise. As shown in Fig. 3(d) and Fig. 2, the formatted residual is much sparser than the previous random one, with most regions closer to zero and residual lies in highly-structured regions. The rest part of the network is similar to DiffResNet with several weight layers. The proposed formatting layer removes high-frequency corruption in homogeneous regions well, while the structured regions are left to the remaining part of the network. In this way, we take advantage from both low-pass filter and high-pass filter. When taking neural network as the formatting layer, the FormResNet can be represented in a recursive fashion (Fig. 2(c)): $y[k] = x[k] + y[k-1]$ where $y[k]$ is the output of k^{th} formatting layer and $x[k]$ is the learned formatted residual. Fig. 2(c) shows the structure when $k = 2$. In this fully convolutional version (Fig. 2(c)), the formatting layer is jointly trained with other layers end-to-end.

In order to avoid the resolution reduction problem [21, 8] and predict a dense output with the same size as the input, we pad zeros before each *conv.* layer and it turns out to work

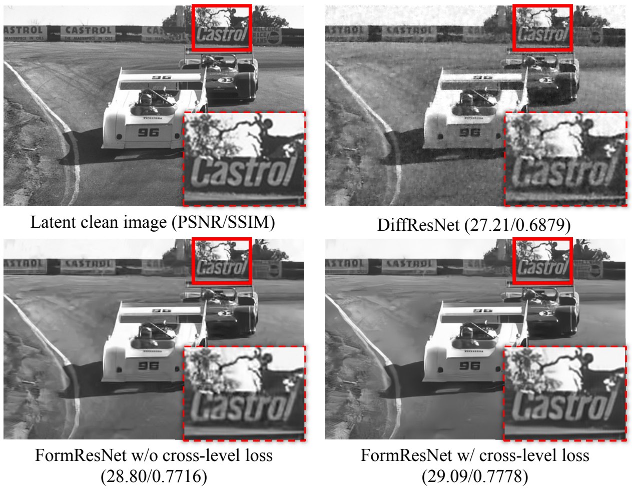

Latent clean image (PSNR/SSIM)
DiffResNet (27.21/0.6879)
FormResNet w/o cross-level loss
(28.80/0.7716)
FormResNet w/ cross-level loss
(29.09/0.7778)

Figure 5. Visual comparison between DiffResNet and FormResNet. The experiment setups are the same as in Fig. 4.

well. In addition, we find the batch normalization (BN) [17] is benefit to the convergence speed, then we simply add a BN layer between each of the *conv.* and ReLU layer.

Cross-level Loss Net. Computer sees images in a manner of "pixel-to-pixel", while we humans see more semantic information. In most CNN-based methods, when judging the quality of image, a pixel-wise similarity (*e.g*. ℓ_1 norm, MSE) is adopted as the loss function. Whereas in practice we not only count on pixel-wise performance, but care more about the visual quality in many situations. In addition, using MSE loss only can usually get blurry images, as shown in Fig. 5. Thus in this paper we also consider high-level visual information for the loss description, and propose a cross-level loss function that combines both the pixel-level information and high-level semantic features, as shown in Fig. 2(b).

Let x be the corrupted image and y the latent clean image and $F()$ as the formatting function in the residual formatting layer. Then the pixel-level loss can be defined as:

$$L_{pix}(\Theta) = \frac{1}{2N}\sum_{i=1}^{N}\left\|r - R(x^i, \Theta)\right\|^2, \qquad (1)$$

where $r = y - F(x)$ is the residual image, $R(x^i, \Theta)$ is the estimated residual by the network, and N is the number of training pairs.

For high-level loss, we first leverage the feature map extracted from a stack of convolutional layers ϕ, which is part of a pre-trained network used for high-level vision. These convolutional layers are concatenated to the end of our FormResNet. The feature-level loss part (pink block in Fig. 2(b)) is inspired by [12] which optimizes a style transfer problem by minimizing the difference between feature maps. As ϕ is only used to extract feature maps for loss computation, all the parameters in ϕ are fixed instead of simultaneous learning with the main body as in [12]. Denote

ϕ_l as the feature map after the l-th ReLU layer of ϕ, and the dimension of ϕ_l as $W_l \times H_l \times C_l$ where W, H, C are the width, height, and number of channels respectively. Then the feature-level loss is defined as:

$$L_{feat}(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \frac{1}{W_l^i H_l^i C_l^i} \left\| \phi_l(y^i) - \phi_l(\hat{y}^i) \right\|^2, \quad (2)$$

where $\hat{y} = F(x) + \hat{r}$ is the recovered image. In addition, information in gradient domain is also leveraged as a high-level loss term as:

$$L_{grad}(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \left| \nabla_h(y^i) - \nabla_h(\hat{y}^i) \right| + \left| \nabla_v(y^i) - \nabla_v(\hat{y}^i) \right|, \quad (3)$$

where ∇_h and ∇_v indicate the horizontal and vertical gradients. The gradient loss term is achieved by two Sobel layers (gray block in Fig. 2(b)) concatenated to the end of FormResNet. By combining the above pixel-level and high-level loss components together, we get the final cross-level loss net representation:

$$L_{cross}(\Theta) = (1-\alpha-\beta) \cdot L_{pix}(\Theta) + \alpha \cdot L_{feat}(\Theta) + \beta \cdot L_{grad}(\Theta), \quad (4)$$

where α, β are balancing weights for the corresponding components.

4. Network Properties

In this section, we study the properties of the proposed network, including the effectiveness of formated residual learning, loss components, network depth and the extension to learn multiple corruptions in a single model.

4.1. Formatted Residual Learning

Residual learning is suitable for image restoration as in many restoration problems the corrupted image and its corresponding latent image is highly correlated. However, the difference between the corrupted and latent images varies for different problems. It is not that easy to directly apply the same structure (*e.g.* [19]) to different tasks. As a result, we show the effect of FormResNet compared to learning the difference (DiffResNet). In addition, the influence of the cross-level loss net is also included for the comparison.

In this experiment, image denoising is taken as an example. We use 10 layers (each layer consists of *conv.*, BN, and ReLU except the first and last layer) for the study on BSD100 (Section 5.2) and the corrupted noise is an additive Gaussian noise with zero mean and standard deviation of 25. A conventional method (BM3D [7]) is used as the formatting layer in FormResNet (other methods like EPLL [40], WNNM [14] can also be taken, BM3D is just used for simplicity when considering the accuracy and efficiency). The VGG-16 net [32] pre-trained for classification is utilized as the function ϕ, and $l = 4$ (feature map after $ReLU2\_2$). The performance curve is shown in Fig. 4.

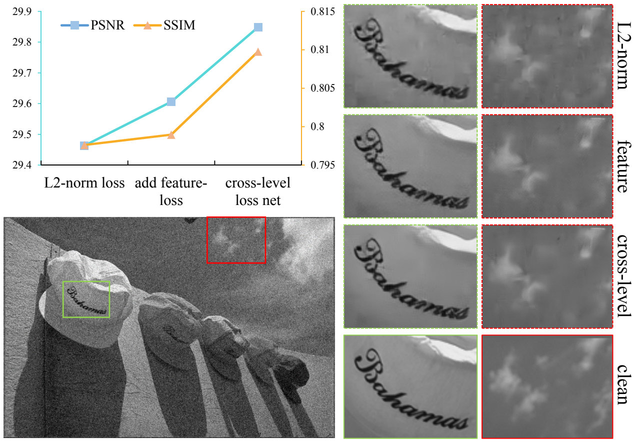


Figure 6. Effectiveness of different loss terms. Quantitative performance (left corner) and qualitative results (right) are shown for comparison.

We can see by using residual learning the network converges faster than without residual (NoRes) learning. After adding the residual formatting layer, the network converges in fewer iterations and result in a higher PSNR. When replacing the MSE loss with the proposed cross-level loss, the performance boosts further. The added formatting layer together with the cross-level loss function are powerful for residual learning. A visual comparison is shown in Fig. 5, in which FormResNet shows a better visual quality compared to others, and more details are recovered by the cross-level loss.

4.2. Loss Components

In order to evaluate the effectiveness of the proposed cross-level loss net, in this study we compare the performance of different loss terms in the loss net. The FormResNet with $k = 1$ *i.e.* DiffResNet is taken as the testbed, after which different loss terms are concatenated. Three parts of ℓ_2 norm (MSE) loss, ℓ_2 norm added feature-loss, and the final cross-level loss are concatenated respectively. The restoration task of image denoising is used for the evaluation with added Gaussian noise ($sigma = 25$) on Kodak dataset. The comparison result is shown in Fig. 6. From the quantitative result we can see with each added loss term, the performance boosts continually. We speculate this is due to the local convexity and smoothness properties of different measures: ℓ_2-only may has many local minima that prevents a global (or better local) minimum, while for the combination with ℓ_1-gradient and feature-level constrains perceptually plausible solutions may lead to a much better minimum. In the visual comparison (right side of Fig. 6), the blur artifact caused by ℓ_2 norm is noticeable on the sky region, while for the characters our cross-level loss recovers more details.

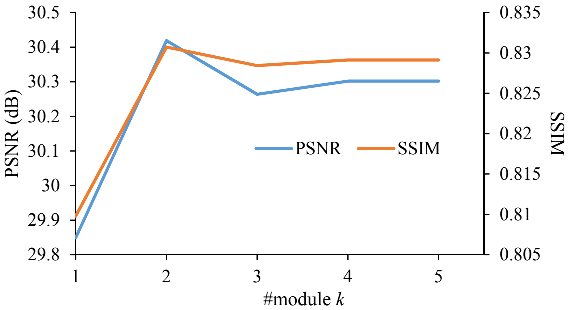


Figure 7. Performance (PSNR/SSIM) with respect to the network depth (module number).

4.3. Network Depth

Here we study the network performance with respect to the depth of the network. Taking the formatting layer as a 10-layer convolutional module, the performance of different number of modules are evaluated as shown in Fig. 7 (experimental settings are the same as in Section 4.2). We can see that the network performance boosts when using the formatting layer and almost converges after the 2nd module, *i.e.* structure in Fig. 2(c). It indicates that the performance not always increases with the depth when the network achieves its capacity.

4.4. Multiple Corruptions in a Single Model

Usually for CNN-based methods, each kind of corruptions (*e.g.* noise level or type) corresponds to a single model, which is not flexible for real application. With the proposed formatting layer, different kinds of corruptions can be formatted to an analogous representation, which can be jointly learned for the corresponding residual map.

In this study, we consider a general blind denoising problem. The training data consists of different noise levels and types: the Gaussian noise with different noise levels, salt&pepper noise and speckle noise. The recursive version of FormResNet with $k = 2$ (Fig. 2(c)) is trained on the above multi-type data as a single model for all noise types. We denote this network as FormResNet-m. Due to the various noise types, median filter (*medfilt2* in Matlab with default parameters) is used as a baseline method for comparison since many applications use median filter as preprocessing. The state-of-the-art CNN-based denoising method DnCNN [39] is also included for the comparison. As the DnCNN has a blind Gaussian denoising version (DnCNN-B), we finetune their model on our multi-type training data for a fair comparison. Experiment is performed on Kodak dataset for example and the result is shown in Table 1. We can see that by training a single model, image corrupted with different noise levels/types can be improved to a large extent. We also test the Poisson noise which is unseen in the training data. Our FormResNet-m also performs well for Poisson noise. Example visual results are shown in Fig. 8.

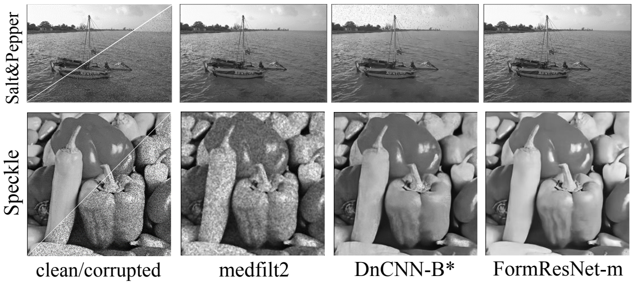


Figure 8. Visual results on multiple corruptions. *kodim06* and *Peppers* are corrupted with Salt&Pepper and Speckle noise.

5. Experiments and Applications

In this section, we show the detail setups for the proposed network and the performance on several image restoration applications.

5.1. Network Training

The proposed network is implemented by the *MatConvNet* package on a server equipped with a Nvidia Tesla K40 GPU card and an Intel Core i7-4790 CPU.

We use the fully convolutional FormResNet in our experiment (*i.e.* RecursiveFormResNet in Fig. 2(c)) and set $k = 2$. The final network depth is set to 20, which is corresponding to the receptive field of the training patch size. Training is carried out by using the stochastic gradient descent (SGD) [29] on mini-batch. The mini-batch size is 128 for denoising and 64 for other applications. A weight decay for the SGD learning is set to 10^{-4} and the momentum is 0.9. For all experiments, the number of training iterations is under 40 epochs (some converge in 10 epochs). The learning rate decreases gradually and is initialized to 0.1. The balancing weights for the cross-level loss net are set to $\alpha = \beta = 0.3$. The weights for the network are initialized according to the method proposed in [15], which is shown to be better than random initialization when using non-linear ReLU as the activation function.

5.2. Applications

Image Denoising Image denoising is an fundamental problem for many computer vision problems. Theoretically, synthetic training data can be infinitely generated. Whereas in this paper, the training set is generated from a small dataset covers 400 natural images: the BSD500 [1] (*train* and *test* subsets). For testing, we use three datasets: 14 commonly used benchmark images (Set14) [7, 4, 14, 23] as shown in Fig. 10, BSD100 (the *val* subset of BSD500), and the Kodak Lossless Image Suite [1]. In this experiment, only gray-scale images are shown for example (for color images, we can simply adjust the number of input channels to 3). Training images are added Gaussian noise or other types of

[1] http://r0k.us/graphics/kodak/index.html

Methods	Gs15	Gs25	Gs45	Salt&Pepper	Speckle	Poisson	Average
medfilt2	27.73/0.6987	25.40/0.8745	21.73/0.3515	30.08/0.8682	23.98/0.5184	30.60/0.8745	26.59/0.6976
DnCNN-B*	31.93/0.8624	29.81/0.8037	27.64/0.7297	28.57/0.7876	28.81/0.8087	33.59/0.9019	30.06/0.8156
FormResNet-m	**32.61/0.8842**	**30.34/0.8301**	**27.82/0.7489**	**43.76/0.9945**	**31.01/0.8667**	**38.80/0.9682**	**34.06/0.8821**

Table 1. Performances on different corruptions, including Gaussian (Gs), salt&pepper, speckle, and Poisson noise (unseen in the training set). Average PSNR/SSIM values are reported for quantitative evaluation. DnCNN-B* means the finetuned result of DnCNN-B [39].

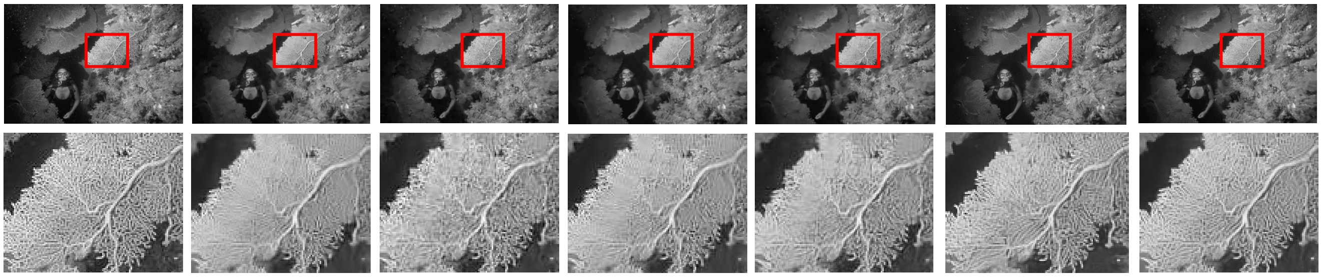

Clean (PSNR/SSIM)
BM3D (25.52/0.6541)
EPLL (25.97/0.6972)
WNNM (25.76/0.6681)
MLP (25.92/0.6826)
DnCNN-S (26.18/0.7067)
FormResNet (**26.28**/**0.7129**)

Figure 9. Visual results on *156065* (BSD100) with Gaussian noise level 25. The proposed FormReNet recovers sharp contours and more details, compared to other methods.

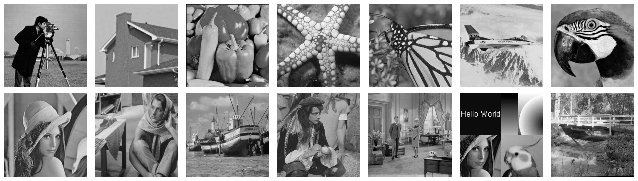


Figure 10. Commonly used test images (Set14).

noises in corresponding experiments. Data augmentation including flipping and rotation are used on the training set. For testing, we use the whole image as input without cropping.

We show both quantitative (Table 2) and qualitative (Fig. 9) performance for the proposed network and give a comparison with other state-of-the-art image denoising methods including: BM3D [7], EPLL [40], WNNM [14], MLP [5], and DnCNN [39]. The implementation of these methods are all from the authors' codes. Metrics of PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural SIMilarity) are calculated for the evaluation. Following [33], the noisy images are quantized to range [0-255] for realistic evaluation. From the result we can see the proposed FormResNet outperforms other methods and recovers more details and structure, especially on high noise levels.

Single Image Super-resolution Our proposed network can also be applied to single image super-resolution. We use 91 images from [37] as our training set, which is smaller than the final training set (291 images) of [19]. Multiple scaling factors of 2, 3, 4 are trained together for the network. Evaluation is performed on Set5 [3] and the results are shown in Table 3. The results of state-of-the-art VDSR [19] training on 91/291 images and the basic bicubic interpolation are included for comparison. From the table we can see that, even training with much fewer images, our FormResNet outperforms VDSR training on the same 91 image set. Our method even performs better than the VDSR training on 291 images.

Single Image Rain Removal We apply our method to the problem of rain removal as a illustration to artifact removal. As there is no large public rain dataset, we use the 12 rain image dataset from [22] for our evaluation. Training is performed on randomly selected 10 images from the 12 rain image and the rest 2 images are used for evaluation. Similar to the denoising application, image patches are extracted with data augmentation for the training process. For comparison, a single image rain removal method DSC [25], and DnCNN-B finetuned on the 10 training images are used here. Results are shown in Fig. 11. We can see that most of the rain artifact on the input image is removed by our FormResNet. Far fewer rain streaks can be observed compared to other methods.

Other Applications The powerful capacity of our network can also benefits other image restoration applications. Examples on natural image inpainting/completion, single depth image enhancement are shown in Fig. 12. For natural image inpainting 50% of the total pixels are randomly removed from the original image, while for depth image enhancement both a downsample (with scale=3) and pixel removal (with 50%) are performed on the clean sharp depth map. The training set for inpainting is the BSD400 [1] while 344 random selected images from [26] are used to train the depth enhancement.

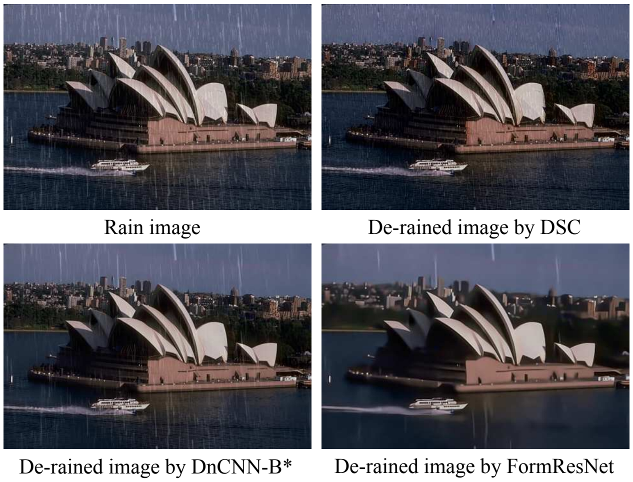
Table 4 compares the computation time of different methods. Image sizes of 256×256 and 512×512 are included, with Gaussian noise level 25. Computation time on GPU is shown if available. Overall, our running time is comparable to BM3D on CPU. However, our method on GPU is fast, and comparable to the state-of-the-art DnCNN.

Testsets	σ	BM3D	EPLL	WNNM	MLP	DnCNN-S	FormResNet
Set14	15	32.31/0.8959	32.03/0.8952	32.62/0.8981	-	32.75/0.9034	**32.77/0.9036**
	25	29.79/0.8471	29.48/0.8436	30.02/0.8506	29.70/0.8455	30.22/0.8584	**30.30/0.8599**
	45	26.55/0.7663	26.33/0.7556	26.76/0.7693	26.60/0.7603	26.91/0.7747	**27.38/0.7873**
	75	23.41/0.6766	22.80/0.6443	23.03/0.6622	23.88/0.6829	23.18/0.6637	**24.89/0.7093**
BSD100	15	30.79/0.8641	30.92/0.8763	31.01/0.8684	-	31.39/0.8831	**31.51/0.8848**
	25	28.14/0.7842	28.29/0.7979	28.31/0.7893	28.46/0.7960	28.71/0.8106	**28.98/0.8153**
	45	25.16/0.6680	25.28/0.6743	25.31/0.6707	25.61/0.6656	25.56/0.6883	**26.34/0.7114**
	75	22.56/0.5703	22.20/0.5481	22.23/0.5446	23.25/0.5771	22.29/0.5515	**24.31/0.6154**
Kodak	15	32.19/0.8738	32.12/0.8792	32.45/0.8770	-	32.76/0.8883	**32.87/0.8890**
	25	29.69/0.8112	29.57/0.8134	29.90/0.8145	29.84/0.8142	30.19/0.8300	**30.42/0.8307**
	45	26.76/0.7207	26.60/0.7148	26.95/0.7220	26.95/0.7129	27.05/0.7329	**27.76/0.7457**
	75	23.95/0.6413	23.36/0.6126	23.70/0.6295	24.51/0.6461	23.57/0.6273	**25.58/0.6702**

Table 2. Comparison results on Set14, BSD100, and Kodak. We compare different methods on the average PSNR/SSIM values. Best performance is shown in **bold**. The proposed FormResNet consistently outperforms other methods on each dataset.

Scale	Bicubic	VDSR-91	VDSR-291	FormResNet-91
x2	33.66	37.06	37.53	**37.55**
x3	30.39	33.27	33.66	**33.75**
x4	28.42	30.95	31.35	**31.40**

Table 3. Performance comparison on single image super-resolution on Set5. Numbers in the table are PSNR values.


Rain image

De-rained image by DSC

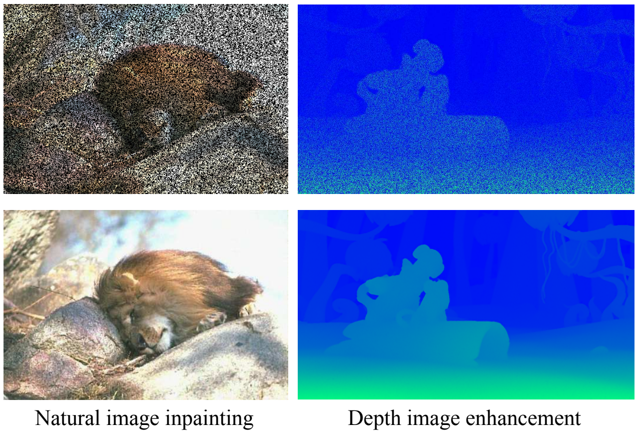
De-rained image by DnCNN-B*

De-rained image by FormResNet

Figure 11. Comparison result on rain removal. Fewer rain streaks can be seen on the result of our FormResNet, compared to those of DSC and DnCNN-B.

size	BM3D	EPLL	WNNM	MLP	DnCNN-B	FormResNet
256^2	0.54	30.67	146.42	2.37	1.05/**0.01**	1.11/**0.01**
512^2	2.24	124.54	599.16	6.56	4.60/**0.04**	4.62/**0.05**

Table 4. Comparison on computation time in seconds. Time on CPU/GPU (if available) is reported.


Natural image inpainting

Depth image enhancement

Figure 12. Other applications on natural image inpainting (left) and depth image enhancement (right). Top row: corrupted images; Bottom row: restored images by our FormResNet.

6. Conclusion

In this paper, we have presented a formatted residual learning framework for image restoration. A residual formatting layer is proposed to format the residual information to structured details. The proposed cross-level loss net contributes to high visual quality by leveraging high-level similarity. Evaluations on multiple public datasets show that the proposed FormResNet outperforms existing image restoration methods both quantitatively and qualitatively, while being very efficient. FormResNet is also able to handle different corruptions (noise types and noise levels) in a single model. By applying different operations to the residual formatting layer, we believe the proposed FormResNet can be easily extended to more other low-level vision problems.

Acknowledgments

This work was supported by the Hong Kong PhD Fellowship Scheme from the Research Grants Council (RGC) of Hong Kong. We acknowledge the support of NVIDIA Corporation with the donation of the GPU card used for this work.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, 2011. 6, 7

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 1, 2, 3

[3] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012. 7

[4] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 6

[5] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, 2012. 1, 2, 3, 7

[6] Y. Chen, W. Yu, and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. In *CVPR*, 2015. 2

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE TIP*, 16(8):2080–2095, 2007. 2, 5, 6, 7

[8] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 2, 3, 4

[9] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE TIP*, 20(7):1838–1857, 2011. 2

[10] A. Dosovitskiy, P. Fischery, E. Ilg, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, T. Brox, et al. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2

[11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE TIP*, 15(12):3736–3745, 2006. 2

[12] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 2, 4, 5

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[14] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 5, 6, 7

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 3

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[18] V. Jain and S. Seung. Natural image denoising with convolutional networks. In *NIPS*, 2009. 1

[19] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 2, 3, 5, 7

[20] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE TPAMI*, 32(6):1127–1133, 2010. 2

[21] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep joint image filtering. In *ECCV*, 2016. 2, 4

[22] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *CVPR*, 2016. 7

[23] H. Liu, R. Xiong, J. Zhang, and W. Gao. Image denoising via adaptive soft-thresholding based on non-local samples. In *CVPR*, 2015. 6

[24] S. Liu, J. Pan, and M.-H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, 2016. 2

[25] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, 2015. 7

[26] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 7

[27] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, Jan 2013. 2

[28] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[29] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 6

[30] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005. 2

[31] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR*, 2014. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[33] R. Vemulapalli, O. Tuzel, and M.-Y. Liu. Deep gaussian conditional random field network: A model-based deep network for discriminative denoising. In *CVPR*, 2016. 2, 7

[34] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *NIPS*, 2012. 2

[35] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014. 2, 3

[36] L. Xu, J. S. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *ICML*, 2015. 2

[37] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE TIP*, 19(11):2861–2873, 2010. 7

[38] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32, 2016. 2

[39] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP*, PP(99):1–1, 2017. 2, 3, 6, 7

[40] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 2, 5, 7