Hierarchical Residual Learning for Image Denoising

Wuzhen Shi[a], Feng Jiang[a], Shengping Zhang[a], Rui Wang[b], Debin Zhao[a], Huiyu Zhou[c]

[a]*School of Computer Science and Technology, Harbin Institute of Technology, China*
[b]*School of Architecture, Harbin Institute of Technology, China*
[c]*Department of Informatics, University of Leicester, University Road, Leicester, LE1 7RH, United Kingdom*

Abstract

In recent years, residual learning based convolutional neural networks have been applied to image restoration and achieved some success. To avoid network degradation, deep layers in these methods are identity mappings, which are not easy to be learned as observed in recent image recognition work. In this paper, we propose a novel residual learning based CNN framework for image denoising, which does not need to learn identify mappings while avoiding network degradation. The proposed CNN network contains three kinds of sub-networks: feature extraction sub-network, inference sub-network and fusion sub-network. The feature extraction sub-network is first used to densely extract patches and represent them as high dimensional feature maps. Multiple inference sub-networks are then cascaded to learn noise maps by exploiting multi-scale information in a hierarchical fashion, which makes our method have a strong ability of toleraing errors in noise estimation. Finally, the fusion sub-network fuses the noise maps to obtain the final noise estimation. The proposed hierarchical residual learning network can tackle with multiple general image denoising tasks such as Gaussian denoising and single image super-resolution. Experimental results on several datasets show that our hierarchical residual learning based image denoising method outperforms many state-of-the-art ones.

Keywords: Image Denoising, convolutional neural network, residual learning, hierarchical residual learning, multi-scale information

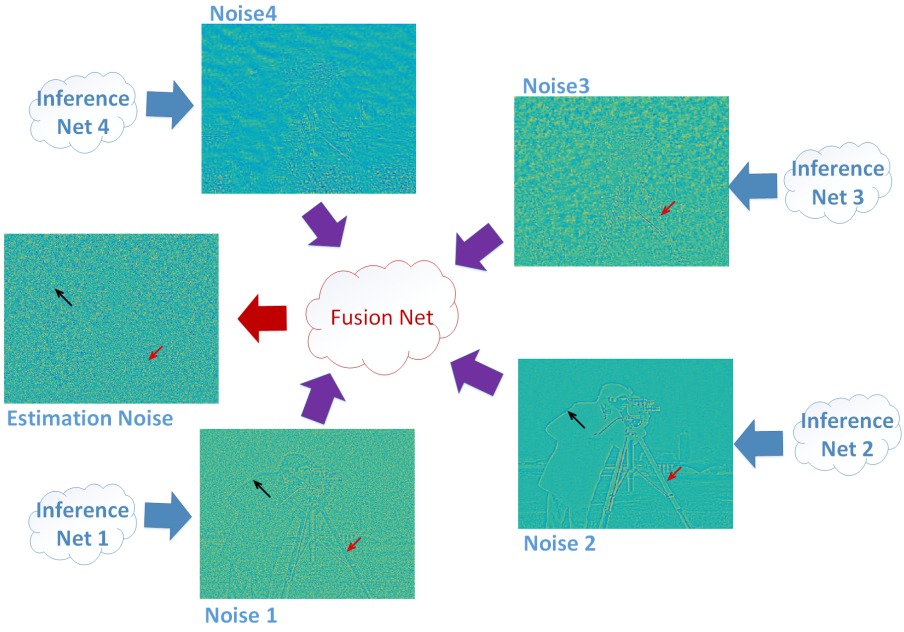

Figure 1: Illustration for hierarchical residual learning. Each inference net estimates a noise map. Then multiple noise maps will be fused to get the final estimated noise map by a fusion net, which makes our method have a strong ability of toleraing errors in noise estimation. It is a hierarchical learning process that the later inference net will learn less noise than the previous one.

1. Introduction

Image denoising aims to remove noises from a noisy image, which makes the recovered image not only approximate to the undegraded one but also comply with human visual system, so that it can benefit the subsequent analysis and processing. A lot of popular denoising methods [1, 2, 3] are based on sparse representation model. They generally consider the property of both self-similarity and sparsity of the image. BM3D [1] stacks some similar image patches to constitute a three dimensional array and sparsely represents them with a three dimensional wavelet basis or DCT basis. On the basis of each image patch having the property of sparsity, LSSC [2] proposes similar image patches should

have similar sparse decomposition. CSR [3] proposes a centralized sparse representation model based on image patch self-similarity to get more accurate sparse decomposition. However, the same as the traditional sparse representation model applied to other tasks [4, 5, 6, 7, 8], sparse representation based image denoising usually contain process of solving a very complex optimization problem that causes conflicts between good performance and running time.

Recently, deep learning method has got much attention and it is successfully applied in many computer vision problems [9, 10, 11, 12, 13]. Specifically, some deep learning based methods have also been explored for the low level tasks. Dong et al. [14, 15] demonstrate that a convolutional neural network (CNN) can learn a mapping from low resolution image to high resolution one in an end-to-end manner. It does not require any engineered features that are typically necessary in traditional methods. Soon after, they expand this work for JPEG compressive image restoration [16]. An effective method [17] to reduce the amount of weights and speed it up has been proposed. For image denoising, Burger et al. [18] attempt to learn the mapping from a noisy image to a noise-free image directly with a plain multi-layer perceptron (MLP). In [19], Chen et al. describe a flexible learning framework based on the concept of nonlinear reaction diffusion models. By embodying recent improvements in nonlinear diffusion models, they propose a dynamic nonlinear reaction diffusion model with time-dependent parameters. Different from [14, 15, 16, 17, 18, 19] that use the undegraded image as ground true for training, some works try to learn image residual. Kim et al. [20] propose a very deep network to learn residual to fast the convergence speed. In [21], residual learning and batch normalization are utilized to speed up the training process as well as boost the denoising performance. However, they simply stack multiple convolution layers to construct a plain network which needs the deep layers to be an identity mapping to get good performance. But an identity mapping is hard to train in deep layers, which has been mention in recent image recognition work [22]. Therefore, it is necessary to explore a better residual learning method.

Investigating an effective ways to use the multiple scale information is also

important. The degraded image can be successful recovered is mainly based on the assumption that patches in a natural image tend to redundantly recur many times inside the image. However, it is not only exist in the same scale but also across different scales. Make full use multiple scales information can improve the restoration result has been proved in traditional method [23]. However, the multiple scale information has been little investigated in deep learning methods. In [20], Kim et al. try to train a multi-scale model for different magnification super resolution. It is a very rough tactics to explore the scale information since they just put different scale image as input for training. Its successful can be attribute to the powerful learning ability of CNN instead of the multiple scale information being considered in the network structure itself.

In this paper, we propose an CNN based image denoising method. Firstly, we propose a novel hierarchical residual learning method. Different from existing residual learning method, which uses a plain network to predict residual, that our hierarchical residual learning method iteratively increase different level residual to get the final residual estimation. Making full use of the relationship between identity mapping and zero mapping, our hierarchical residual learning method can make the network deeper is better. Secondly, based on the idea of hierarchical residual learning, we design a convolutional neural network for image noise estimation. It contains three kinds of sub-network, i.e. feature extraction, inference and fusion sub-network. We cascade multiple inference sub-network to estimate different level noise. Each inference sub-network has different scale receptive field. They constitute a receptive field pyramid that makes our network has the natural attributes of learning multiple scale information. A fusion sub-network is also design to fuse these different level noise maps to increase our networks fault tolerance. Figure 1 gives a glimpse of the hierarchical residual learning process of our proposed network and how it corrects these error components. More details will be given in the later suction.

In short, the contributions of this work are mainly in three aspects:

- We present a hierarchical residual learning method, which gradually in-

crease the residual and remove these error components. It makes full use of the relationship between identity mapping and zero mapping that guarantees our network deeper is better.

- We design a hierarchical residual learning based image noise estimation network. The receptive fields of multiple sub-networks form a pyramid, which makes our network can learn multiple scale information.

- The proposed hierarchical residual learning based image noise estimation network can handle multiple image denoising tasks. Experiments on Gaussian denosing and single image super-resolution show that our method outperforms many state-of-the-art methods.

2. Related Work

Residual learning. Residual learning actually has been widely used in traditional image restoration methods. A lot of sparse representation based image super-resolution methods try to learn the image high frequency components. Specifically, in [24], Zhang et al. propose dual-dictionary to learn residual iteratively. Part of our work inspired by this multiple residual learning method. After the success of ResNet [22] in image recognition, some works try to learn the residual instead of the undegraded image. Very recently, DnCNN [21] proposed to learn a CNN network with residual learning for image denoising. Compared with DnCNN, our proposed method has three very significant advantages. On the one hand, there is no need of any operation of batch normalization that makes our network having fewer parameters that speed up the running time. On the other hand, hierarchical residual learning makes our method more robust.

Multiple scale information. In order to avoid the tedious matter of training different model for different magnification image super resolution, Kim et al. [20] train a multi-scale model. With this approach, parameters are shared across all predefined scale factors. However, their network structure does not

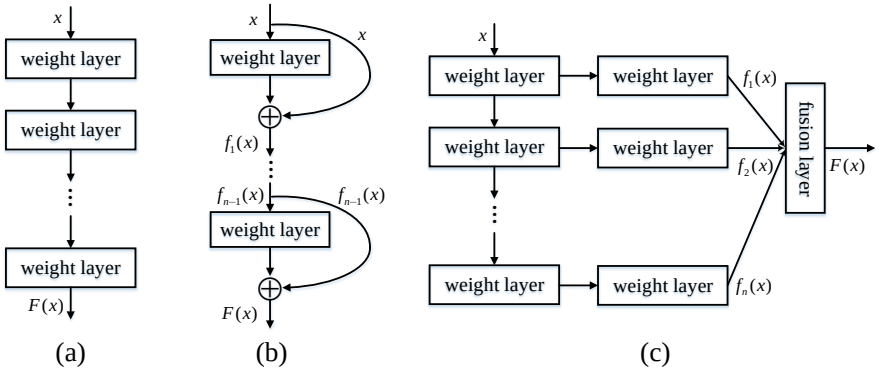

Figure 2: Comparison between the existing residual learning method and our proposed hierarchical residual learning. (a) is the network proposed by Kim et al. [20]. (b) is the identify mapping based residual learning that learns residual progressively. (c) is the proposed hierarchical residual learning.

contain any scale information. Their success maybe should attribute to the powerful learning ability of the convolutional neural network. In [25], Hui et al. propose a Multi-Scale Guided convolutional network for depth map super resolution. But it can be classed as cascade operation because it does not learn the scale information from the input low resolution image.

3. The Proposed CNN based Image Denoising Method

In this section, we first propose our hierarchical residual learning framework, then give detail introduction to the proposed image denoising method. An insight into how our method does work well is also given.

3.1. Hierarchical Residual Learning

In [20], Kim et al. propose a very deep network to learn residual to fast the convergence speed. It is a plain convolutional neural network as shows in Figure 2 (a), which stacks multiple convolution layers with the same size filters. These network can be expressed as:

$$F_i(x) = \max(0, W_i * x + B_i), i \in \{1, 2, \cdots, n-1\} \quad (1)$$

$$F(x) = W_n * F_{n-1}(x) + B_n \tag{2}$$

where W_i and B_i represent the filters and biases of the i^{th} layer respectively, and $*$ is a convolution operation. While it successfully introduced residual learning into the image restoration problem, we find its limitations in two aspects: first, it needs the high layers to be an identity mapping to guarantee the network not degrade since it is a very deep network. Second, the output of the network just relates to single receptive field. In this paper, we propose hierarchical residual learning to solve these two problems.

Intuitively, if the shallow network has got the best performance, the increased layers being an identity mapping will not destroy the good performance that result in the shallow network. However, in reality, when the network reaches a certain depth, the deeper the network is not necessarily the better. In [22], He et al. give an example that a 56 layers network get worse performance that the 26 layers one. This is in conflict with the identity mapping theory. We think this can be explained as the identity mapping is hard to train with a convolution neural network. To solve this problem, He et al. explicitly add an identity mapping into the network. Then the output of the network with input x becomes $f(x) + x$. They explain x is the identity mapping and $f(x)$ is the residual. Obviously, the identity mapping based network can also be used to learn image residual. As show in Figure 2 (b), the identify mapping based residual learning network can be formulated as:

$$F_i(Y) = \max(0, (W_i * Y + B_i) + Y), i \in \{1, 2, \cdots, n-1\} \tag{3}$$

$$F(Y) = (W_n * F_{n-1}(Y) + B_n) + F_{n-1}(Y) \tag{4}$$

The identify mapping based residual learning network lands back on the basic idea of original residual learning proposed in [22] for image recognition. The difference has two aspects: first, the input x is residual and $f(x)$ is the residual's residual. Second, we focus on low level task instead of the high level one. The identify mapping based residual learning network can progressively refine the residual map. However, the output of this kind of network is related

to single receptive field. In this paper, we propose hierarchical residual learning framework that optimizes the residual map progressively as the identify mapping based residual learning network does. But different with identify mapping based residual learning network, our hierarchical residual learning makes full use of different receptive field information. The graphical representation of our hierarchical residual learning framework is Figure 2 (c) and it can be formulated as:

$$f_i(Y) = \max(0, W_i * Y + B_i), i \in \{1, 2, \cdots, n\} \tag{5}$$

$$F(Y) = fusion(f_1(Y), f_2(Y), \cdots, f_n(Y)) \tag{6}$$

Where $fusion(\cdot)$ is a fusion method. Obviously, different fusion method can be used to fuse these residual maps. $fusion(\cdot)$ is a Concat layer follows by a convolution layer in our proposed hierarchical residual learning based image denoising method. Different from the identify mapping based network structure, the hierarchical one no need to explicitly use the identity mapping. If the fusion output of the front i residual maps is the desired result, we just need to train the remains to be zero. This idea is consistent with the finding in [22]. This design results in some very interesting characters that we will analysis it in the later subsection.

3.2. Hierarchical Residual Learning based Image denoising

Based on our hierarchical residual learning framework introduced in the above subsection, we design an end-to-end network to estimate the image noise. The network framework outlined in Figure 3, consists of three kinds of sub-networks: feature extraction, inference and fusion networks. We use F_E, F_I and F_R to denote these three kinds of sub-network respectively. The feature extraction network learns the effective feature maps ready for inference. Then multiple inference sub-networks cascade to learn noise. Finally, all these learned noise maps are fused to get the final estimated noise by a fusion net.

The feature extraction sub-network takes the noise image as input and learns a set of feature maps. It is a single layer network with d_1 filters of size $f_e \times f_e \times c$

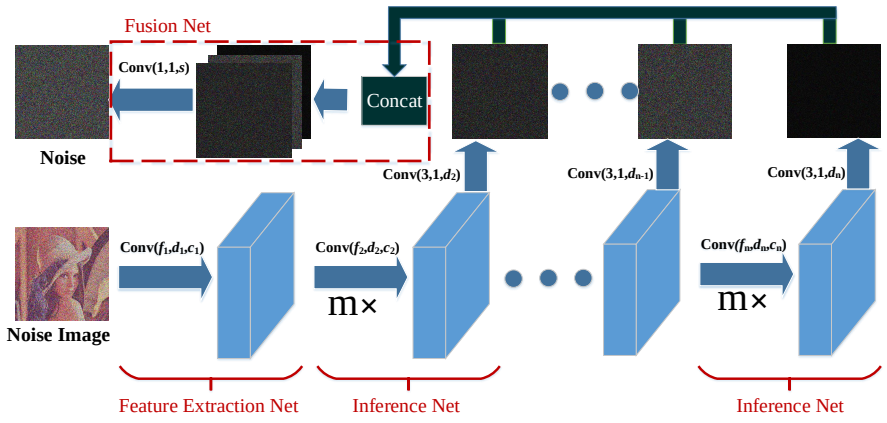

Figure 3: Our Network Structure. Given a noise image, the feature extraction sub-network extracts a set of feature maps. Then multiple inference sub-network cascade to predict different scale noise. These noise maps are grouped together into a high dimensional noise feature map. The fusion sub-network fuses these noise maps to produce the final noise estimation.

as in SRCNN [14, 15]. f_e is the spatial size of a filter and c depends on the channel number of the noise image, i.e. $c = 1$ for gray image and $c = 3$ for the color one. By doing convolution with these filters, each patch of the input noise image, which has the same size with the receptive field of a neuron, is represented as a high-dimensional feature vector. Then, given the noise image x, the formula for feature extraction sub-network is as follows:

$$F_E(x) = \max(0, w_e * x + b_e) \tag{7}$$

where w_e, b_e are the filter and bias respectively and $*$ denotes a convolution. $\max(0, \cdot)$ corresponds to the Rectified Linear Unit (ReLU). In Figure 3, it is marked as $Conv(f_1, d_1, c_1)$.

The inference sub-network contains m convolution cascade operation that results in a large receptive field to takes more image context into account for detail recovery. In the existing residual learning method proposed by Jim et al. [20], it mainly relies on the identity mapping to guarantee deeper being better. In addition to the identity mapping, based on hierarchical residual learning, we cascade

multiple inference sub-networks that introduces zero mapping to provide double protection for the goal of deeper being better. One inference sub-network can be expressed as:

$$F_I^j(Y) = \max\left(0, w_I^j * Y + b_I^j\right), j \in \{1, 2, \cdots, m\} \tag{8}$$

$$F_I^{m+1}(Y) = w_I^{m+1} * Y + b_I^{m+1} \tag{9}$$

where w_I^j and b_I^j represent the filters and biases of the j^{th} layers respectively. Each layer has d_I filters and biases for layers from 1 to m, while the last layer has only one filter and bias for estimating a noise map. One inference sub-network can only takes one scale information into account that make it not flexible enough to the complex image detail. The cascade structure is a good way to learn multiple scale information. Very interesting, this design conforms to the traditional multiple residuals learning idea. More importantly, it introduces a zero mapping relationship to guarantee our network deeper being better. The input Y is different for different inference sub-network. The input Y of the first inference sub-network is the high dimensional feature maps outputted by the feature extraction sub-network, i.e. $Y_1 = F_E(x)$. To the other inference sub-network, the input Y is the m^{th} layer output of the previous sub-network, i.e. $Y_i = F_{I_{i-1}}^m(Y_{i-1})$. In Figure 3, the i^{th} inference sub-network is marked as $m \times Conv(f_i, d_i, c_i)$.

The fusion sub-network fuses all noise maps, which learns from multiple inference sub-networks, to get final image noise estimation. It contains a concat and convolution operation. The concat layer group these noise maps together into a high dimensional noise feature map, then the convolution layer fuses them. It can be expressed as

$$F_R^1\left(F_{I_1}^{m+1}, F_{I_2}^{m+1}, \cdots, F_{I_N}^{m+1}\right) = Concat\left(F_{I_1}^{m+1}, F_{I_2}^{m+1}, \cdots, F_{I_N}^{m+1}\right) \tag{10}$$

$$F_R^2\left(F_{I_1}^{m+1}, F_{I_2}^{m+1}, \cdots, F_{I_N}^{m+1}\right) = w_R * F_R^1\left(F_{I_1}^{m+1}, F_{I_2}^{m+1}, \cdots, F_{I_N}^{m+1}\right) + b_R \tag{11}$$

where Concat operation concatenates the inputs along the feature channel dimension, w_R is a filter of size of $1x1$ and b_R is the biases, $F_R^2(\cdot)$ is the final

estimated noise.

3.3. Training

We now describe the objective to minimize in order to find optimal parameters of our model. Following most of CNN based image restoration methods, the mean square error is adopt as the cost function of our network. Since we have N inference sub-networks and a fusion one to estimate noise maps, we have $N+1$ objectives to minimize. Given a training dataset $\{x_i, y_i\}_{i=1}^{n}$, where x_i and y_i is a noise image its corresponding noise map respectively. The optimization objective of the j^{th} inference sub-network is represented as:

$$L_I^j(\theta_j) = \frac{1}{2n}\sum\nolimits_{i=1}^{n}\left\|F_I^j(x_i;\theta_j) - y_i\right\|_F^2 \tag{12}$$

where θ_j and $F_I^j(x_i;\theta_j)$ denotes the parameter set and the estimated noise map of the j^{th} inference sub-network. For the fusion sub-network, we have

$$L_R(\theta) = \frac{1}{2n}\sum\nolimits_{i=1}^{n}\|F_R(x_i;\theta) - y_i\|_F^2 \tag{13}$$

where θ is the network parameters needed to be trained and $F_R(x_i;\theta)$ is the final estimated noise map with respect to noise image x_i. Note that parameter set θ_j is part of θ. The final loss function can be represented as:

$$L(\theta) = \alpha_R L_R(\theta) + \sum\nolimits_{j=1}^{N}\alpha_j L_I(\theta_j) \tag{14}$$

where α_R and α_j denote the importance of corresponding loss functions. Rectified Linear Unit (ReLU) is used as activation function after each convolution layer. We use the adaptive moment estimation (Adam) [26] to optimize all network parameters.

3.4. Insights

Deeper is better. As discuss in the previous section, if the shallow network has got the best performance, the increase layers being an identity mapping will not degrade the network. In our hierarchical residual, we dont give the identity mapping explicitly. However, if the fusion result of the front i inference sub-networks get the desired output, they will become an identity mapping and

the later inference sub-networks all become a zero mapping. Our experimental results prove our reason. Figure 1 is an illustration for hierarchical residual learning and fusion process. Noise map 1 is the noise estimation of the first inference sub-network. It is the similar interpretation to other noise maps. The figure shows that the noise information become less and less from noise map 1 to noise map 4. There is hardly any information in the last noise map. In the other words, it become a really zero mapping. In summary, the ideas of identity mapping and zero mapping provide double protection for the goal of deeper being better for our proposed hierarchical residual based image denoising network.

Furthermore, the fusion sub-network is not a simple add operation. It makes these hierarchical learned noise maps form a complementary relationship to amend the error estimations. That is, they are not only provide different level noise estimation but also point out the errors existing in each other. For example, in Figure 1, the noise map 1 mistakenly regard the image texture as noise, then the noise map 2 tell our network that there are something wrong in noise map 1. We mark these obvious errors in each noise map in arrows. Through our fusion sub-network, all these errors are amended. This is a significant advantages of hierarchical learning than adder tree structure and the plain one.

Multiple Scale information learning. Another important advantage of our hierarchical residual learning method is that it can take multiple scale information into account. The multiple scale information has been widely used in traditional image restoration methods and has demonstrated to be conducive to improve restoration result. They often sample or interpolate the image to different scales that constitutes an image pyramid that providing much more useful information for detail recovery. In our method, each inference sub-network has different size of receptive field. It results in a receptive field pyramid that each scale provides different detail information like the traditional image pyramid does. In other words, our hierarchical residual learning has a natural attributes of multiple scale information learning. It is another reason that our method has

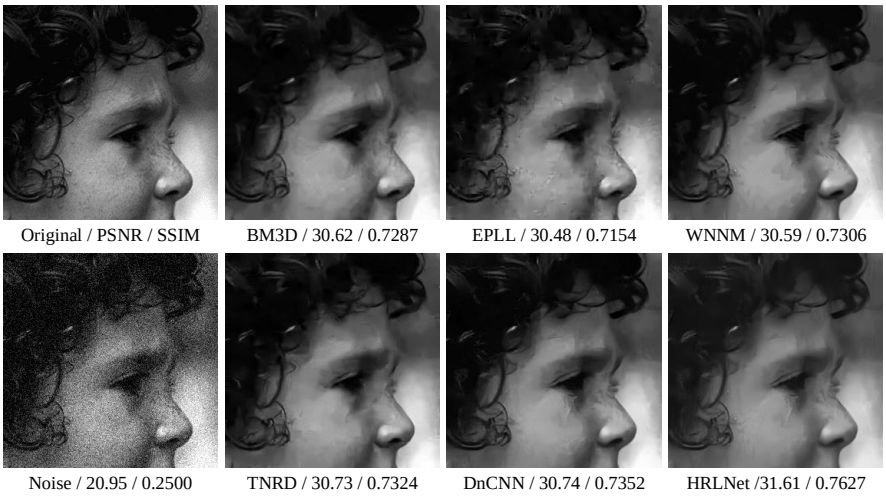

Figure 4: Visual quality comparison of Gaussian noise removal on image "head" from Set5 [27] in the case of sigma = 25.

very strong fault tolerance.

4. Experimental Results

In this section, we evaluate the performance of our method on both Gaussian denoising and single image super-resolution. Firstly, the training and testing datasets are introduced. Next, some training details are given. Finally, we show the quantitative and qualitative comparisons with four state-of-the-art methods. We name the proposed method as HRLNet.

4.1. Datasets for Training and Testing

It is well known that training dataset is very important for the performance of learning based image restoration methods. A lot of training dataset can be found in the literature. For example, SRCNN [14] uses a 91 images dataset and VDSR [20] uses 291 images dataset. For a fare comparison with TNRD [19] and DnCNN [21], which are two very new image restoration methods in the literature, we use the same 400 images of size 180×180 for training. We set the

Table 1: The PSNR/SSIM results of Gaussian noise removal by various algorithms on Set5 [27]

Image	Level	BM3D [1]		EPLL [28]		WNNM [29]		TNRD [19]		DnCNN [21]		HRLNet	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
baby	5	39.42	0.9634	39.29	0.9632	39.51	0.9637	39.44	0.9631	-	-	**40.01**	**0.9650**
	25	31.75	0.8613	31.51	0.8561	31.83	0.8626	31.76	0.8625	31.85	0.8647	**32.59**	**0.8751**
	50	28.98	0.7897	28.51	0.7660	28.89	0.7909	28.81	0.7861	28.99	0.7891	**29.72**	**0.8062**
bird	5	40.65	0.9800	40.20	0.9778	41.10	0.9816	40.74	0.9791	-	-	**41.58**	**0.9812**
	25	31.80	0.8946	31.37	0.8770	32.21	0.9034	31.88	0.8907	32.17	0.8995	**32.85**	**0.9055**
	50	28.30	0.8112	27.77	0.7824	27.83	0.7930	28.10	0.7953	28.37	0.8077	**28.96**	**0.8182**
butterfly	5	37.36	0.9759	37.54	0.9764	38.38	0.9786	38.00	0.9772	-	-	**39.01**	**0.9802**
	25	28.35	0.9133	28.41	0.9106	29.40	0.9265	29.30	0.9192	29.72	0.9268	**30.33**	**0.9283**
	50	24.75	0.8412	25.10	0.8465	25.08	0.8622	25.90	0.8393	26.28	0.8695	**26.78**	**0.8701**
head	5	36.63	0.9203	36.66	0.9264	36.64	0.9221	36.50	0.9174	-	-	**37.25**	**0.9225**
	25	30.62	0.7287	30.48	0.7277	30.59	0.7306	30.73	0.7324	30.74	0.7352	**31.61**	**0.7627**
	50	28.46	0.6576	28.37	0.6395	28.32	0.6454	28.68	0.6517	28.53	0.6475	**29.42**	**0.6977**
woman	5	39.19	0.9764	38.91	0.9749	39.29	0.9764	39.29	0.9760	-	-	**39.68**	**0.9765**
	25	30.74	0.9008	30.31	0.8896	30.96	0.9072	30.76	0.9004	30.98	0.9077	**31.65**	**0.9115**
	50	27.17	0.8301	26.99	0.8096	27.43	0.8312	26.97	0.8280	27.67	0.8434	**28.25**	**0.8497**
Avg.		32.28	0.8696	32.09	0.8616	32.50	0.8717	32.46	0.8679	-	-	**33.31**	**0.8834**

Table 2: Average PSNR/SSIM of Gaussian noise removal by various algorithms on three datasets

Image	Level	BM3D [1]		EPLL [28]		WNNM [29]		TNRD [19]		DnCNN [21]		HRLNet	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	5	38.65	0.9632	38.52	0.9637	38.98	0.9645	38.79	0.9625	-	-	**39.51**	**0.9651**
	25	30.65	0.8597	30.42	0.8522	31.00	0.8661	30.89	0.8610	31.09	0.8668	**31.80**	**0.8766**
	50	27.53	0.7860	27.33	0.7688	27.42	0.7793	27.78	0.7853	27.97	0.7914	**28.63**	**0.8084**
Set12	5	37.98	0.9568	37.55	0.9572	38.03	0.9583	37.80	0.9566	-	-	**38.45**	**0.9590**
	25	29.56	0.8227	29.24	0.8209	29.85	0.8282	29.62	0.8256	29.75	0.8305	**30.46**	**0.8368**
	50	26.39	0.7195	26.03	0.7053	26.43	0.7276	26.45	0.7210	26.62	0.7253	**27.29**	**0.7369**
BSD68	5	37.56	0.9635	37.54	0.9647	37.78	0.9650	37.70	0.9645	-	-	**37.95**	**0.9667**
	25	28.56	0.8011	28.67	0.8121	28.83	0.8156	28.91	0.8151	29.02	0.8190	**29.14**	**0.8238**
	50	25.61	0.6860	25.67	0.6877	25.78	0.7036	25.96	0.7019	26.10	0.7076	**26.16**	**0.7143**
Avg.		31.39	0.8398	31.22	0.8370	31.57	0.8454	31.54	0.8437	-	-	**32.15**	**0.8542**

patch size as 40×40, and use data augmentation (rotation or flip) to prepare training data. For Gaussian denoising, we test on three datasets, i.e. Set5 [27] (5 images), Set14 [30] (14 images) and BSD68 [31] (68 images), which are widely used for benchmark in other works [14, 15, 19, 20]. For single image super-resolution, we test on Set5 [27] (5 images) and Urban100 [32]. It is needed to be noted that we only consider the luminance channel (in YCrCb color space) in our experiments following most image restoration works like [14, 15]. However, our method can be extend to directly training/testing on color images by setting the appropriate channel number.

4.2. Training Details

For feature extraction sub-network, we set $f_e = 3$, $d_e = 80$, and $c_e = 1$ for gray image. For each inference sub-network, we uniformly set $f_i = 64$, $d_i = 64$, and $m = 5$. We cascade 4 inference sub-networks for estimating multiple levels noise maps. To fusion sub-network, $s = 4$ that is consistent with the amount of inference sub-network to be cascaded. Because the input and the output of the proposed network should have the same resolution, so we will up-sample the low resolution image to the desired resolution by bicubic interpolation for single image super-resolution. For weight initialization, we use the method described in He et al. [33]. This is a theoretically sound procedure for networks utilizing rectified linear units (ReLU). For the other hyper-parameters of Adam, we set the exponential decay rates for the first and second moment estimate to 0.9 and 0.999, respectively. We train all our experiments only over 50 epochs and each epoch iterate 1600 with patch size 128. The learning rate of the first 30 epochs is 0.001 while that of the other 20 epochs is 0.0001. We implement our model using the MatConvNet package [34].

4.3. Experiments on Gaussian Denoising

We compare our proposed HRLNet with four state-of-the-art methods, namely BM3D [1], EPLL [28], WNNM [29], TNRD [19], and DnCNN [21]. All the test experiments are implemented in Matlab 2015a on Windows 7 system, and

Figure 5: Visual quality comparison of Gaussian noise removal on image *Lenna* from Set14 [30] in the case of sigma = 50.

runs on desktop computer with 4 cores CPU at 3.4 GHz and 12 GB RAM. To TNRD, we use its 7x7 filter model for comparison. The implementation codes are downloaded from the authors' websites and the default parameter settings are used in our experiments. We compare these methods in three noise levels (sigma value), i.e. 5, 25 and 50. To DnCNN, the authors do not release the model in noise level of 5, so we just compare with DnCNN in noise level of 25 and 50. Both quantitative and qualitative comparisons are given. Table 1 shows the PSNR and SSIM results of Gaussian noise removal by various algorithms on each image of Set5. To this five images, our proposed HRLNet outperforms all the other five methods on all noise level with respect to both PSNR and SSIM assessment criteria. On this dataset, our HRLNet can improve roughly 0.85 dB, 0.81 dB, 1.03 dB, 1.22 dB on average, in comparison with TNRD, WNNM, BM3D and EPLL, respectively. When we use the SSIM as the assessment criteria, the average gains achieved by our HRLNet are 0.0117, 0.0138, 0.0155 and 0.0218 in comparison with WNNM, BM3D, TNRD and EPLL, respectively. We note that Set5 may not be a conclusive test set due to the limited number of test

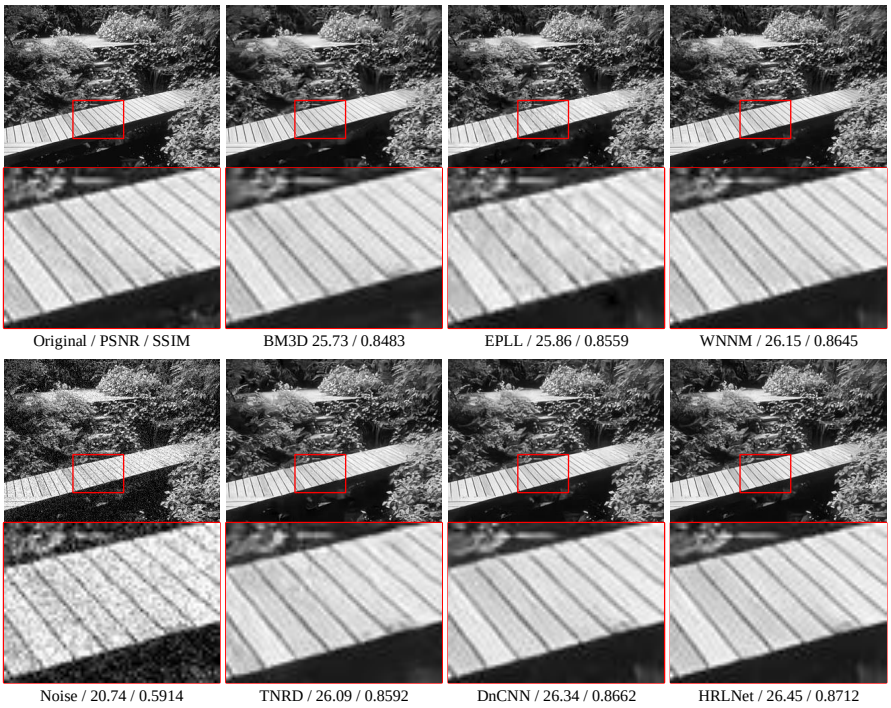

Figure 6: Visual quality comparison of Gaussian noise removal on image *Test021* from BSD68 [31] in the case of sigma = 25.

samples, but the results show that the proposed HRLNet can handle different noise level well. To further validate the performance of our proposed model, we test it on the other two larger dataset. In Table 2, we provide a summary of average PSNR and SSIM results of Gaussian noise removal by various algorithms on Set5, Set14 and BSD68. We highlight the best results with bold fonts. Both Table 1 and Table 2 show that our method outperforms all the five compared state-of-the-art methods by a large margin with respect to PSNR and SSIM. Specially, compare to the recent proposed TNRD, our method get average 0.41 dB gain over these three dataset on three noise levels.

Since all the compared methods get almost no difference in visual effect on noise level 5, we just show the high level noise (i.e. sigma is 25 and 50) removal

Table 3: Average CPU running time (in seconds) of different methods on images of size 256×256, 512×512, and 1024×1024 with noise level 25.

Method	BM3D	EPLL	WNNM	TNRD	DnCNN	HRLNet
256×256	0.65	25.40	203.10	0.45	0.74	0.55
512×512	2.85	45.50	773.20	1.33	3.41	2.17
1024×1024	11.89	422.10	2536.40	4.61	12.10	9.21

results. Figure 4 gives an example of qualitative comparison on noise level 25. It shows the visual quality comparison of Gaussian noise removal on image *head* from Set5. Our method can recover much more image detail information such as the eyelid part, while the results of the other methods either over smooth or can't get clean output. Figure 5 shows the very high level noise removal performance, where the sigma value is 50 and the PSNR and SSIM of the noise image are only 14.53 dB and 0.1026, respectively. Obviously, Figure 5 shows the result of our proposed HRLNet produces much sharper edges than other approaches without any obvious artifacts across the image. Figure 6 shows another example of visual quality comparison. The enlarged portion shows the the result of HRLNet is the clearest and the most visually pleasant.

4.4. Running Time

The running time comparison of various methods are shown in Table 3. The running time of the compared method are taken from [21]. The running time of HRLNet is the implementation time on the platform of Matlab 2015a on Windows 7 system with an Intel Core i7-3770 CPU. The average CPU running time comparisons on three different size images shows that HRLNet obtains comparable running speed with the state-of-the-art methods.

4.5. Experiments on Single Image Super-resolution

The compared single image super-resolution methods include: SRCNN [15], TNRD [19], VDSR [20], and DnCNN [21]. Bicubic interpolation results are also listed for comparison. All the codes are downloaded from the authors' websites.

Table 4: Average PSNR and SSIM comparisons of different methods for single image super-resolution with upsampling factors of 3 and 4 on Set5 and Urban100 datasets. The best results are highlighted in bold.

Dataset	Scale	Bicubic		SRCNN [15]		TNRD [19]		VDSR [20]		DnCNN [21]		HRLNet	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	x3	30.39	0.8682	32.39	0.9033	33.18	0.9152	33.67	0.9220	**33.75**	**0.9222**	**33.75**	0.9219
	x4	28.42	0.8104	30.09	0.8503	30.85	0.8732	31.35	0.8845	31.40	0.8845	**31.49**	**0.8849**
Urban	x3	24.46	0.7349	26.03	0.7973	26.42	0.8076	27.13	0.8283	27.15	0.8276	**27.19**	**0.8292**
	x4	23.14	0.6577	24.32	0.7183	24.61	0.7291	25.17	0.7528	25.20	0.7521	**25.23**	**0.7522**
Average		26.60	0.7678	28.21	0.8173	28.77	0.8313	29.33	0.8469	29.38	0.8466	**29.42**	**0.8471**

All the tests use the models released by the corresponding author. Table 4 shows the average PSNR and SSIM comparisons of different methods for single image super-resolution with upsampling factors of 3 and 4 on Set5 and Urban100 datasets. As shown, HRLNet gets the highest average PSNR and SSIM in comparison with the four state-of-the-art single image super-resolution methods. Figure 7 shows an example of visual quality comparison of different single image super-resolution methods on image *Woman* from Set5 [27] with scale factor $\times 4$. The enlarged portion shows that the texture of the reconstructed image of HRLNet is clearer than that of the compared methods.

5. Conclusion

In this paper, we propose a hierarchical residual learning convolutional neural network (HRLNet) for image noise estimation. It contains three kinds of sub-networks, i.e. feature extraction, inference and fusion sub-network. Such a hierarchical learning strategy makes the residual map be refined progressively. Furthermore, our network receptive fields constitute a receptive field pyramid that make it has a natural attributes to make full use of multiple scale information. Experimental results show that the proposed HRLNet outperforms many state-of-the-art methods on Gaussian denoising and single image super-resolution.

Figure 7: Visual quality comparison of single image super-resolution on image *Woman* from Set5 [27] with scale factor $\times 4$.

6. Acknowledgement

This work has been supported in part by the Major State Basic Research Development Program of China (973 Program 2015CB351804), the National Science Foundation of China under Grant No. 61572155. H. Zhou was supported by UK EPSRC under Grant EP/N011074/1, Royal Society-Newton Advanced Fellowship under Grant NA160342, and European Unions Horizon 2020 research and innovation program under the Marie-Sklodowska-Curie grant agreement No 720325.

References

[1] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, IEEE Transactions on image processing 16 (8) (2007) 2080–2095.

[2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 2272–2279.

[3] W. Dong, L. Zhang, G. Shi, Centralized sparse representation for image restoration, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1259–1266.

[4] D. Tao, X. Li, Z. He, X. You, Y. Y. Tang, Connected component model for multi-object tracking, IEEE Trans Image Process 25 (8) (2016) 3698–3711.

[5] Z. He, S. Yi, Y. M. Cheung, X. You, Y. Y. Tang, Robust object tracking via key patch sparse representation, IEEE Transactions on Cybernetics 47 (2) (2017) 354–364.

[6] S. Yi, Z. He, Y. M. Cheung, W. S. Chen, Unified sparse subspace learning via self-contained regression, IEEE Transactions on Circuits and Systems for Video Technology PP (99) (2017) 1–1.

[7] S. Yi, Z. Lai, Z. He, Y. M. Cheung, L. Yang, Joint sparse principal component analysis, Pattern Recognition 61 (Complete) (2017) 524–536.

[8] S. Yi, Y. Liang, Z. He, Y. Li, W. Liu, Y. M. Cheung, Dual pursuit for subspace learning, IEEE Transactions on Multimedia.

[9] L. Zhang, A. A. Mohamed, R. Chai, B. Zheng, S. Wu, Automated deep-learning method for whole-breast segmentation in diffusion-weighted breast mri, in: Medical Imaging, SPIE, 2019.

[10] L. Zhang, R. Chai, S. W. Dooman Arefan, Jules Sumkin, Deep-learning method for tumor segmentation in breast dce-mri, in: Medical Imaging, SPIE, 2019.

[11] D. Xie, L. Zhang, L. Bai, Deep learning in visual computing and signal processing, Applied Computational Intelligence and Soft Computing.

[12] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, J. Liang, Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 7340–7349.

[13] L. Zhang, F. Yang, Y. D. Zhang, Y. J. Zhu, Road crack detection using deep convolutional neural network, in: International Conference on Image Processing, IEEE, 2016.

[14] C. Dong, C. C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: European Conference on Computer Vision, Springer, 2014, pp. 184–199.

[15] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE transactions on pattern analysis and machine intelligence 38 (2) (2016) 295–307.

[16] C. Dong, Y. Deng, C. Change Loy, X. Tang, Compression artifacts reduction by a deep convolutional network, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 576–584.

[17] C. Dong, C. C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: European Conference on Computer Vision, Springer, 2016, pp. 391–407.

[18] H. C. Burger, C. J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with bm3d?, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2392–2399.

[19] Y. Chen, T. Pock, Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[20] J. Kim, J. Kwon Lee, 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.

[21] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising., IEEE Transactions on Image Processing PP (99) (2017) 1–13.

[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[23] D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 349–356.

[24] J. Zhang, C. Zhao, R. Xiong, S. Ma, D. Zhao, Image super-resolution via dual-dictionary learning and sparse representation, in: Circuits and Systems (ISCAS), 2012 IEEE International Symposium on, IEEE, 2012, pp. 1688–1691.

[25] T.-W. Hui, C. C. Loy, X. Tang, Depth map super-resolution by deep multi-scale guidance, in: European Conference on Computer Vision, Springer, 2016, pp. 353–369.

[26] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[27] M. Bevilacqua, A. Roumy, C. Guillemot, M. L. Alberi-Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding.

[28] D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 479–486.

[29] S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2862–2869.

[30] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: International conference on curves and surfaces, Springer, 2010, pp. 711–730.

[31] S. Roth, M. J. Black, Fields of experts, International Journal of Computer Vision 82 (2) (2009) 205.

[32] J. B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Computer Vision and Pattern Recognition, 2015.

[33] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[34] A. Vedaldi, K. Lenc, Matconvnet: Convolutional neural networks for matlab, in: Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 689–692.