Xiang *et al.*
*Journal of Engineering and Applied Science* (2024) 71:93
https://doi.org/10.1186/s44147-024-00425-7

Journal of Engineering and Applied Science

RESEARCH Open Access

# Multi-task learning with self-learning weight for image denoising

Qian Xiang[1], Yong Tang[2*] and Xiangyang Zhou[1]

*Correspondence:
xq_new21@163.com

[1] College of Information Science and Engineering, Wuchang Shouyi University, Wuhan, People's Republic of China
[2] School of Artificial Intelligence, Hubei Business College, Wuhan, People's Republic of China

## Abstract

**Background:** Image denoising technology removes noise from the corrupted image by utilizing different features between image and noise. Convolutional neural network (CNN)-based algorithms have been the concern of the recent progress on diverse image restoration problems and become an efficient solution in image denoising.

**Objective:** Although a quite number of existing CNN-based image denoising methods perform well on the simplified additive white Gaussian noise (AWGN) model, their performance often degrades severely on the real-world noisy images which are corrupted by more complicated noise.

**Methods:** In this paper, we utilized the multi-task learning (MTL) framework to integrate multiple loss functions for collaborative training of CNN. This approach aims to improve the denoising performance of CNNs on real-world images with non-Gaussian noise. Simultaneously, to automatically optimize the weights of individual sub-tasks within the MTL framework, we incorporated a self-learning weight layer into the CNN.

**Results:** Extensive experiments demonstrate that our approach effectively enhances the denoising performance of CNN-based image denoising algorithms on real-world images. It reduces excessive image smoothing, improves quantitative metrics, and enhances visual quality in the restored images.

**Conclusion:** Our method shows the effectiveness of the improved performance of denoising CNNS for real-world image denoising processing.

**Keywords:** Self-learning weight, Multi-objective optimization, Non-Gaussian noise model, Image denoising, Multi-task learning, Convolutional neural network

## Introduction

The digital image is an essential source of information in many fields, such as image surveillance, target tracking, and magnetic resonance images (MRI) [1, 2]. However, the digital image is inevitable to be corrupted by various types of noise in the procedure of capture and transmission, which decreases image quality. A noisy image is usually formulated as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{v}$$

where $\boldsymbol{y}$ denotes the noisy image, $\boldsymbol{x}$ the noise-free image, and $\boldsymbol{v}$ inductive noise. The noise $\boldsymbol{v}$ is often assumed to be subject to some kind of distribution.

In past decades, numerous image denoising techniques have been proposed, such as non-local self-similarity methods, partial differential equations (PDEs) algorithms, threshold algorithms, sparse representation algorithms[3], and hybrid method [4]. With the development of deep learning, CNN-based image denoising method has become the focus of image denoising [5]. As proposed in [6], a feed-forward denoising convolutional neural network (DnCNN) is introduced, consisting of a cascaded structure that includes convolution layers, rectified linear unit (ReLU), batch normalization (BN) layers, and residual learning (RL) introduced at network output. Although most of these image denoisers mentioned above perform well for noisy images polluted by additive white Gaussian noise (AWGN), their performance usually suffers degrading dramatically when removing noise in real-world noisy images captured by digital cameras which introduce more sophisticated noise. In view of this problem, Wei et al. [7] aim to establish a more accurate simulation of image noise models in real-world scenarios, intending to generate target data for improving the denoising capabilities of algorithms on images captured in real scenes. Guo et al. [8] proposed a more realistic noise model that considers signal-dependent noise and the influence of the image signal processing (ISP) pipeline on noise. They also proposed a convolutional blind denoising network (CBDNet) to restore a clean image from a realistic noisy image. This is achieved by designing a noise estimation sub-network based on a more realistic noise model. Chen et al. [9] contends that conventional training methods involve overfitting to the noise in the training set and has devised a masking training approach. It involves applying a random and substantial masking to the input image, compelling the model to learn the reconstruction of the obscured image content, thus improving the model's generalization capability. However, due to the influence of various factors on real camera noise, existing noise models still struggle to fully match the complexity of real-world noise. As a result, these methods have not significantly enhanced the generalization performance of denoising networks and still face challenges when dealing with mismatched noise distributions [10].

Moreover, the MSE loss used in the training of traditional denoising CNN is also designed based on the assumption of Gaussian noise and with the aim of enhancing the peak signal-to-noise ratio (PSNR) index. However, it has been indicated that the PSNR index does not effectively reflect human visual perception features, leading to evaluation results that often differ from human visual perception. In other words, even though the PSNR index of the image is improved, denoising results in excessive smoothing of image details. Therefore, for images with non-Gaussian noise, when CNN-based image denoisers are using only MSE loss, the denoised image actually contains additional information introduced by various denoising methods [11], resulting in artifacts. Simultaneously, the excessive smoothing of the image leads to the loss of texture details.

Multi-task learning (MTL) is a learning paradigm that aims at taking advantage of knowledge contained in multiple related tasks to promote the generalization performance for each task [12, 13]. It can effectively leverage information provided by different learning tasks more efficiently than single-task learning [14] and facilitate knowledge sharing between tasks, thereby reducing the risk of overfitting for each

individual task and improving overall performance [15]. However, the performance of a MTL model relies heavily on the weight selection among tasks, while searching for an optimal weight using manual adjustment is time-consuming and difficult [16, 17]. In this paper, we propose a method to enhance the existing image denoising convolutional neural network (DCNN) within the framework of multi-task learning (MTL) for non-Gaussian noise image denoising and design a data-driven sub-task weight self-learning method.

(1) Through the MTL framework, different image quality assessment metrics and image features (including MSE, SSIM, statistical characteristics of image residuals) are utilized as sub-tasks to achieve collaborative training for the DCNN. This approach gradually transforms non-Gaussian image noise towards Gaussian noise, thereby enhancing denoising performance and improving the visual quality of denoised images.
(2) We designed a network layer that, in conjunction with the collaborative training of the aforementioned image convolutional neural network (DCNN), automatically and rapidly learns the weights for each subtask.
(3) Two training strategies are researched, one for optimal performance and the other for obtaining the most suitable shared features for multiple tasks.

The experiments demonstrate that our approach enhances the image denoising performance of convolutional neural networks (DCNN) on two types of networks and four image datasets, under both Gaussian and non-Gaussian noise conditions. This improvement is observed in terms of both quantitative metrics and visual perception of the images.

## Related work

### Image DCNN

The image DCNN has achieved great improvement in Gaussian noise denoising. To deal with more complex noises, a fast and flexible denoising convolutional neural network (FFDNet) [18] has been presented by introducing noise level graph as an additional input of the network based on DnCNN. In view of the difficulty to obtain noisy/noise-free images sample pairs, the Noise2Noise (N2N) [19] method uses samples pairs composed of independent noisy images from the same background to train DCNN and reach comparable performance of training with noisy/noise-free pairs. Its training strategy is derived from the statistical observation that the loss function only requires the target signal to be "clean" on some statistical values while not needing to be "clean" on every target signal. CBDNet [8] consists of two sub-networks. One is a noise estimation sub-network which has a symmetric structure and the total variation losses and outputs a noise level graph of the same size as the input image, and the other is a non-blind denoising sub-network to obtain the latent clean image with noise level graph and noisy image as input. The synthetic structure and real-world noisy images are merged for CBDNet training to achieve a more robust performance even though the noise model is slightly different from the real-world

noise. Experiments demonstrated the crucial role of the image noise model in real noisy images.

#### MTL

MTL aims at improving the performance of each task by inductive knowledge transfer to share domain information between tasks and has been successfully applied in machine learning and deep learning. Tang. et al. [20] designed a face recognition network with multi-task learning for better performance by jointing optimization on the face recognition loss and the face classification loss. Gao et al. [21] applied the MTL framework to integrate target recognition and image noise reduction in the defect recognition of railway insulator images, which carried out coordinated training for CNN by alternately freezing one task and optimizing the other. Considering that manual adjustment of the weight coefficient of each task is time-consuming and laborious, Kendall et al. [17] adopted the homoscedastic uncertainty of each task to weigh each loss and showed their method superior to individual models trained respectively on each task in per-pixel depth regression and other problems. Ozan et al. [22] transformed the (MTL problem into a multi-objective optimization (MOO) problem to optimize a set of potentially conflicting multiple objectives. Thus, the objective of MTL is converted into finding the Pareto optimal solutions for the corresponding MOO problem. Ozan et al. use the multiple-gradient descent algorithm (MGDA) to solve the weight coefficients for potentially conflicting targets. They demonstrated that their method produces a solution that is either a Pareto stationary point or provides a descent direction that can improve each task objective. This method has been successfully applied in scene understanding and multi-label classification.

## Proposed method

#### MTL framework and auxiliary tasks

To leverage the MTL framework for denoising convolutional neural network (DCNN) training, we contemplate introducing loss functions based on different principles for DCNN. The optimization of these loss functions is treated as sub-tasks within the MTL framework, thereby transforming MTL into the following MOO problem:

$$\operatorname{Min}\mathbf{L}(\boldsymbol{x},\boldsymbol{y}) = \underset{n=1,2,\cdots,N}{\left\{\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta^s},\boldsymbol{\theta}^1\right),\cdots,L_n\left(x,y,\theta^s,\theta^n\right)\right\}} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ denote input space, and $\boldsymbol{y} \in \mathbb{R}^N$ denote a set of objective space. $N$ is the total number of objectives, $\boldsymbol{\theta}^n$ are objective-specific parameters, $\boldsymbol{\theta}^s$ are shared parameters, and $L_n\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}^s,\boldsymbol{\theta}^n\right) : \boldsymbol{x} \rightarrow \boldsymbol{y}^n$ is $n^{\text{th}}$ sub-tasks or loss function of DCNN.

As mentioned in Introduction, when the DCNN solely uses mean square error (MSE) as the loss function, although it suppresses the amplitude of image residuals, the distribution of image residuals is influenced by non-Gaussian noise and the denoising algorithm, leading to a reduction in the quality of visual perception [11]. Therefore, we introduce the distribution distance metric as an auxiliary task to make the image residuals $\widehat{\boldsymbol{e}}$ approximate Gaussian white noise. Simultaneously, the structural similarity index (SSIM) is introduced to enhance the structural similarity

between the denoised image and the target image. The integration of these tasks under a MTL framework aims to not only suppress the amplitude of image residuals but also reduce redundant information in order to remove noise and improve denoising effectiveness.

#### Distribution distance loss

If the image residuals $\widehat{\boldsymbol{e}}$ are close to zero-mean Gaussian white noise, it indicates that the geometric structure or texture features have been effectively removed from the noisy image. Therefore, within the MTL framework of the DCNN, we introduce a sub-task aimed at making the image residuals in the denoised image approximate white Gaussian noise. This is intended to align with the traditional DCNN noise model, thereby improving the denoising performance of the DCNN under non-Gaussian noise conditions. There are several methods to evaluate how closely image residuals $\widehat{\boldsymbol{e}}$ approximate white Gaussian noise, with one of these being the auto-correlation coefficient of the residuals. This coefficient can be calculated using the following formula:

$$\rho = \frac{E\left[(X-\mu_x)\left(Y-\mu_y\right)\right]}{\sigma_x \sigma_y} \tag{2}$$

where $\mu_x$, $\mu_y$ denote the mean of $x$ and $y$, and $\sigma_x$ and $\sigma_y$ denote the standard deviation of $x$ and $y$ respectively. Then, the auto-correlation coefficient can be examined through a randomness test. Another method is the Kullback–Leibler divergence (KLD), which can be used directly to calculate the difference between two different distributions. The formula of KLD is

$$KL(p \parallel q) = -\int p(x) \ln\left\{\frac{q(x)}{p(x)}\right\} dx \tag{3}$$

where $x \sim q(x)$ is the distribution of $\widehat{\boldsymbol{e}}$, and $p(x) \sim N(0,1)$ is distribution of white Gaussian noise, and we calculate the KLD between $\widehat{\boldsymbol{e}}$ and white noise as an auxiliary task.

#### SSIM loss

Denoised images reconstructed by DCNN-based methods that minimize MSE loss often lose important details, such as over-smoothing artifacts in texture-rich regions [12], leading to a degradation in image quality. In contrast, the SSIM measures the structural similarity between images by comparing image brightness, contrast, and structure. Its evaluation results are considered to be more consistent with how humans measure the differences between two images. In [23], training DCNN with the joint loss function of SSIM and $L_1$ indeed achieved better image denoising results. Therefore, we introduce the SSIM index as another subtask in the MTL framework. Therefore, we use SSIM index as another sub-task in the MTL framework. Let $C_l\left(I,\widehat{I}\right)$ and $C_c\left(I,\widehat{I}\right)$ denote respectively the difference between two images in luminance and contrast and $\mu_I$ and $\sigma_I$ the mean and standard deviation of image; then, the formula is

$$\begin{aligned} C_l(I,I) &= \frac{2\mu_I\mu_{\widehat{I}}+C_1}{{\mu_I}^2+{\mu_{\widehat{I}}}^2+C_1} \\ C_c\left(I,\widehat{I}\right) &= \frac{2\sigma_I\sigma_{\widehat{I}}+C_2}{{\sigma_I}^2+{\sigma_{\widehat{I}}}^2+C_2} \end{aligned} \tag{4}$$

where $C_1$ and $C_2$ are constants for stability. Let $C_s\left(I,\widehat{I}\right)$ denote the difference between two images in structure; then, the SSIM index is calculated as follows.

$$\begin{aligned} C_s\left(I,\widehat{I}\right) &= \frac{\sigma_{I\widehat{I}}+C_3}{\sigma_I\sigma_{\widehat{I}}+C_3} \\ SSIM\left(I,\widehat{I}\right) &= [C_l\left(I,\widehat{I}\right)]^{\alpha}[C_c\left(I,\widehat{I}\right)]^{\beta}[C_s\left(I,\widehat{I}\right)]^{\gamma} \end{aligned} \tag{5}$$

where $\alpha$, $\beta$, and $\gamma$ are adjustable parameters.

#### Non-Gaussian noise model

As acquiring noisy/noise-free image pairs is not easy, DCNNs usually have to utilize simulated noise image data to perform training, and the proper noise model has great influence on the training effect and denoising performance of DCNN. Existing CNN denoisers [24], BM3D-Net [25] or DnCNN, generally occur performance degradation on real-world noisy images, which is owing to that they adopt a simple AWGN model while the real noise is usually non-Gaussian. In this section, Poisson-Gaussian model [26] is introduced as the real noise distribution model. The Poisson-Gaussian model has 0 mean value, and its variance varies with the actual pixel value, which is signal-dependent and changes with different cameras and camera settings. The Poisson-Gaussian noise model can be further simplified to be the heterogeneous Gaussian noise model that is made up of a stationary noise and a signal-dependent noise. It has been proved that heterogeneous Gaussian noise model is more suitable than AWGN for noise modeling in real-world image. In the heterogeneous Gaussian (HG) noise model, each observed sample $y$ is regarded as a random variable with a signal-depended variance which is formulated as follows.

$$y \sim N\left(\mu = x, \sigma^2 = \lambda_r + \lambda_s x\right) \tag{6}$$

where $x$ is the signal, and $\lambda_r$ and $\lambda_s$ are parameters which depend on sensor's gains. Moreover, other image processing procedures (such as color correction, and tone mapping) are also considered in generating the simulated noisy images, which are synthesized by adding noise to raw sensor measurements and used as training data.

#### Network structure

Depending on the optimization process, we propose two strategies for the training of DCNN with MSE loss, SSIM loss and distribution distance loss as tasks in the MTL framework. Network structure of the first strategy is illustrated in Fig. 1, where the DCNN contains all the shared parameters, and each loss is computed as an objectives-specific task. Then, we convert the MOO problem to the following SOO problem by weighting all these losses.

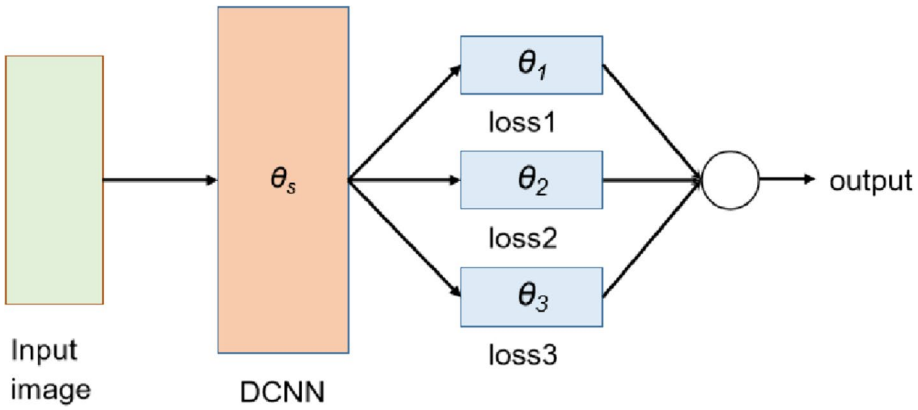


**Fig. 1** Network architecture of the first strategy

$$\underset{\alpha^1,\cdots,\alpha^T,\boldsymbol{\theta}^s}{\text{Argmin}} \sum_{t=1}^{T} \alpha^t \nabla_{\boldsymbol{\theta}^s} L_t(\boldsymbol{\theta}^s, \boldsymbol{\theta}^t) \tag{7}$$

The weight $\alpha$ in the optimization problem (7) can be considered as a group of hyper-parameters. For a small number of hyper-parameters, Bayesian optimization [27] can be employed for parameter search. In this paper, we designed a linear layer output to perform a weighted sum for different tasks, and the weights $\alpha$ are automatically optimized through network training. The shared parameters contained in the DCNN are also optimized according to the gradient descent algorithm. The algorithm based on this image DCNN training strategy is formulated in Algorithm 1.

**Algorithm 1**. Training according to the first strategy.

1: **Input**:

- $\boldsymbol{X}$: training data-set, including noisy images
- $\boldsymbol{Y}$: target data-set, including ground truth images

2: **for** epochs of training **do**

Sample $m$ batch from $\boldsymbol{X}$ and $\boldsymbol{Y}$.

Calculate task-related loss function $\text{loss}_n$

Calculate weight $\alpha_n$ with Bayesian optimization

Update the parameters of the DCNN by minimizing the total $\sum_{n=1}^{3} \alpha_n \text{loss}_n$ .

**end for**

3. **Output**: denoised images

The second strategy employs a task-switching multi-task learning (MTL) framework for training the DCNN. The network structure of this strategy is illustrated in Fig. 2, where each output corresponds to the DCNN utilizing a different loss. In this strategy, an alternate optimization method is used to optimize the network in turn according to

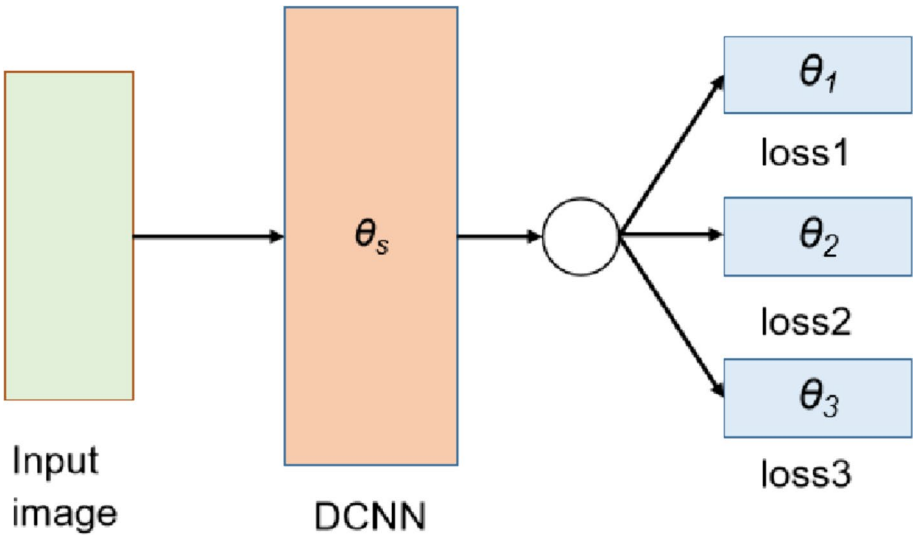


**Fig. 2** Network architecture of the second strategy based on feature transformation MTL framework

all sub-tasks. Through this training strategy, the DCNN can acquire shared features that are most suitable for multiple sub-tasks.

**Algorithm 2.** Training according to the second strategy.

1: **Input**:

- $\boldsymbol{X}$: training data-set, including noisy images
- $\boldsymbol{Y}$: target data-set, including ground truth images

2: **for** epochs of training **do**

Sample $m$ batch from $\boldsymbol{X}$ and $\boldsymbol{Y}$.

**for** n = 1 to 3 **do**

Calculate the task-related loss function $\text{loss}_n$

Update the DCNN by minimizing $\text{loss}_n$

**end for**

**end for**

3. **Output**: denoised images

## Results and discussion

### Experiments data preparation

To evaluate our method, we selected two denoising convolutional neural network (DCNN) models: denoising autoencoder (DAE) and CBDNet. We applied the training

methods described in the "Network structure" section to study the improvement in denoising performance of the DCNN. Evaluation was conducted using the PSNR and SSIM metrics. The algorithm was implemented in Python.

For the DAE, we clip images from STL dataset [28] to patches with size $96 \times 96$. Various levels of Gaussian or non-Gaussian noise were added to these patches to create sample pairs. Images from PolyU [29] and RENOIR [30] are used for testing. The testing data pair are generated on PolyU and RENOIR by the same way with the training data pair.

For the CBDNet, we kept its network architecture and training parameters unchanged. We extracted 1200 images from the DND dataset [31] and synthesized training noisy images using the heterogeneous Gaussian noise model and image processing pipeline (ISP) proposed in literature [8]. Testing was conducted using images from the PolyU dataset, BSDS500 [32], and RENOIR.

### Experiments on DAE

The DAE used in our experiments is made up of two parts: one is encoder $z = f(\widetilde{x}_i)$, and the other is decoder $y = g(z)$, and both are CNNs (Table 1). $z$ denotes the low-dimensional hidden layer feature vector extracted from input $x$. On the generation of training data, we utilize three different noise models: Gaussian noise, heterogeneous Gaussian noise, and heterogeneous Gaussian noise with ISP. The test data are generated in the same way on PolyU and RENOIR datasets. In the procedure of training, we use stochastic optimization algorithm with a learning rate $3 \times 10^{-4}$ and set the training epoch as 100.

Figures 3 and 4 show the image denoised results of a PolyU image respectively on the AWGN model and heterogeneous Gaussian noise model. Compared with the improved

**Table 1** Network structure of the tested DAE

| **Encoder** | | | **Decoder** | | |
|---|---|---|---|---|---|
| Encoder layer | Layer shape | Kernel | Decoder layer | Layer shape | Kernel |
| Conv2d | [, 64, 96, 96] | (3, 64, 3, 1, 1) | ConvTranspose2d | [, 256, 24, 24] | (256, 128, 3,1, 1) |
| BatchNorm2d | [, 64, 96, 96] | 64 | ReLU | [, 256, 24, 24] | - |
| Conv2d | [, 64, 96, 96] | (64, 64, 3, 1, 1) | BatchNorm2d | [, 256, 24, 24] | 128 |
| ReLU | [, 64, 96, 96] | - | ConvTranspose2d | [, 128, 48, 48] | (128, 128, 3,2,1, 1) |
| MaxPool2d | [, 64, 96, 96] | (2, 2) | ReLU | [, 128, 48, 48] | - |
| BatchNorm2d | [, 64, 48, 48] | 64 | BatchNorm2d | [, 128, 48, 48] | 128 |
| Conv2d | [, 64, 48, 48] | (64, 64, 3, 1, 1) | ConvTranspose2d | [, 64, 48, 48] | (128, 64, 3,1, 1) |
| ReLU | [, 64, 48, 48] | - | ReLU | [, 64, 48, 48] | - |
| BatchNorm2d | [, 64, 48, 48] | 64 | BatchNorm2d | [, 64, 48, 48] | 64 |
| Conv2d | [, 64, 48, 48] | (64, 128, 3, 1, 1) | ConvTranspose2d | [, 32, 48, 48] | (64, 32, 3,1, 1) |
| ReLU | [, 128, 48, 48] | - | ReLU | [, 32, 48, 48] | - |
| BatchNorm2d | [, 128, 48, 48] | 128 | BatchNorm2d | [, 32, 48, 48] | 32 |
| Conv2d | [,128, 48, 48] | (128, 128, 3, 1, 1) | ConvTranspose2d | [, 32, 48, 48] | (32, 32, 3,1, 1) |
| ReLU | [, 128, 48, 48] | - | ConvTranspose2d | [, 16, 96, 96] | (32, 16, 3,2,1, 1) |
| BatchNorm2d | [, 128, 48, 48] | 128 | ReLU | [, 16, 96, 96] | - |
| Conv2d | [,256 48, 48] | (128, 256, 3, 1, 1) | BatchNorm2d | [, 16, 96, 96] | 16 |
| ReLU | [, 256, 48, 48] | - | ConvTranspose2d | [, 16, 96, 96] | (16, 3, 3,1, 1) |
| MaxPool2d | [, 256, 24, 24] | (2, 2) | Sigmoid | [, 3, 96, 96] | - |
| BatchNorm2d | [, 256, 24, 24] | 256 | | | |

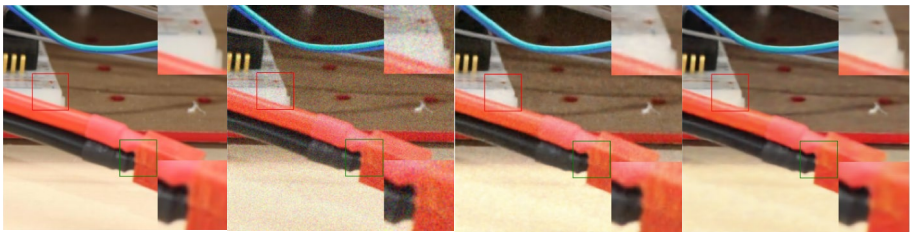


**Fig. 3** Restored results of a PolyU image over AWGN ($\sigma = 30$). **a** Target image. **b** Noisy image PSNR/SSIM: 19.16/0.33. **c** Traditional DAE PSNR/SSIM: 30.17/0.94. **d** Improved DAE PSNR/SSIM: 32.07/0.98

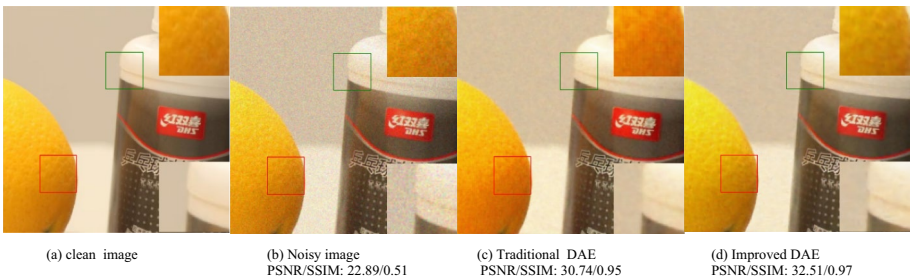


**Fig. 4** Restored results of a PolyU image over HG noise. **a** Clean image. **b** Noisy image PSNR/SSIM: 22.89/0.51. **c** Traditional DAE PSNR/SSIM: 30.74/0.95. **d** Improved DAE PSNR/SSIM: 32.51/0.97

DAE, the traditional DAE methods generate more artifacts in the denoised images and lose some details in the image structures. The improved DAE performs better in preserving image detail structures and achieves positive gains in PSNR and SSIM metrics when removing noise.

Figure 5 shows the variation of MSE loss and validation result respectively about the traditional DAE and the DAE improved by the two strategies with the training on the RENOIR dataset. As is seen from Fig. 5, the improved DAES1 has the fastest rate of decline speed on MSE loss curve and validation curve. All the two improved DAEs outperform traditional DAE on the decline speed of MSE loss curve and validation curve, demonstrating an improvement in denoising effectiveness.

#### Experiments on CBDNet

CBDNet has demonstrated effective noise reduction capabilities on real-world images. In order to enhance the denoising performance and improve the generalization to non-Gaussian noise of CBDNet, we applied the MTL framework to its training process. It takes about 2 days to train the improved CBDNet on a Nvidia GeForce GTX 1060 GPU.

Figure 6 provides the denoised result on a PolyU image in heterogeneous Gaussian noise model with ISP. The improved CBDNet demonstrates positive gains over the traditional CBDNet in preserving image edges and achieving relative increases in PSNR and SSIM on the RENOIR dataset. Figure 7 shows the denoising results of different methods on RENOIR images under the Gaussian noise model. The original DAE method produces over-smoothing results, while the improved DAE method restores more local details and has better denoising effect than DnCNN and FFDNet.

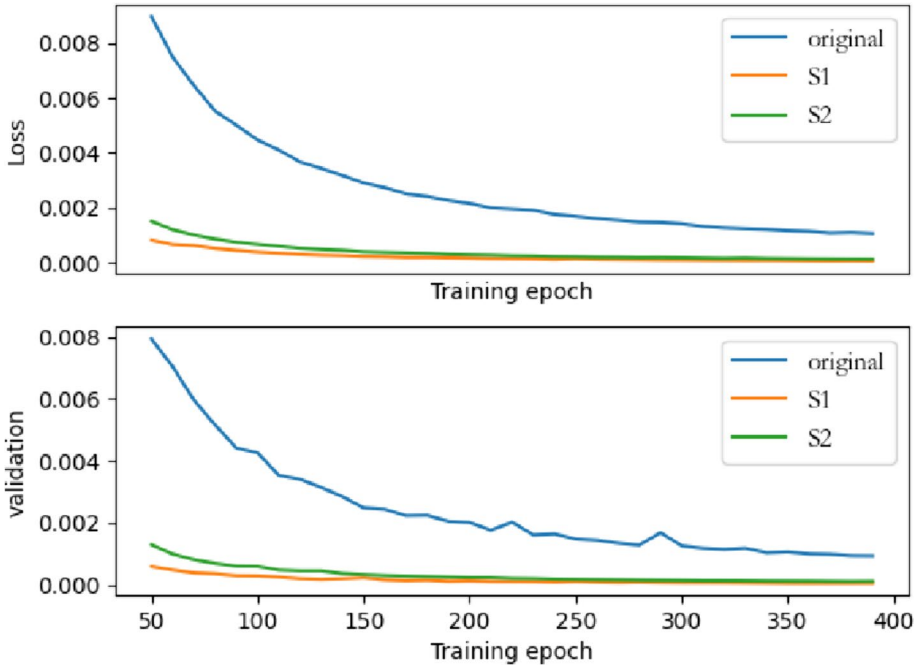


**Fig. 5** The variation in MSE loss and validation

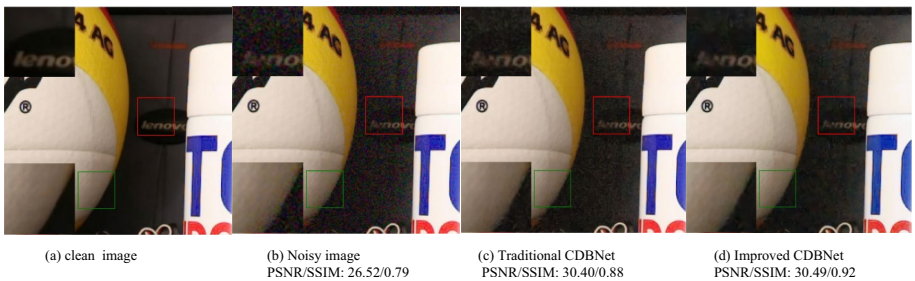


**Fig. 6** Denoising results of a PolyU image over HG noise + ISP. **a** Clean image. **b** Noisy image PSNR/SSIM: 26.52/0.79. **c** Traditional CDBNet PSNR/SSIM: 30.40/0.88. **d** Improved CDBNet PSNR/SSIM: 30.49/0.92

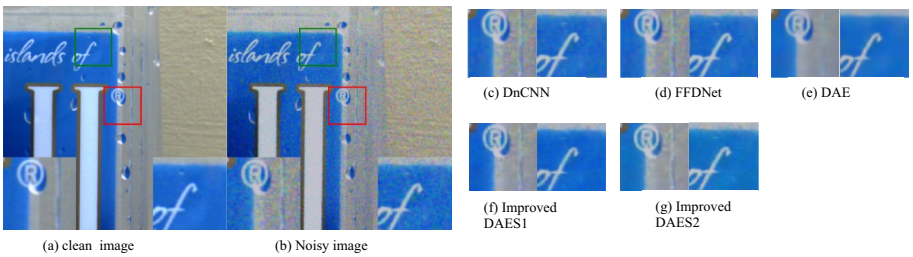


**Fig. 7** Denoising results of a RENOIR image. **a** Clean image. **b** Noisy image. **c** DnCNN. **d** FFDNet. **e** DAE. **f** Improved DAES1. **g** Improved DAES2

Table 2 exhibits the denoising performance comparison on the RENOIR dataset between the improved versions using the two strategies proposed in the "Network structure" section and the original DAE, DnCNN, FFDNet, and CBDNet. The improved

**Table 2** The denoising results on RENOIR dataset

| Method | PSNR | SSIM | Time(s) |
|---|---|---|---|
| DnCNN | 28.77 | 0.6399 | 0.3782 |
| FFDNet | 28.61 | 0.6882 | 0.3653 |
| CBDNet | 29.70 | 0.9016 | 0.3745 |
| DAE | 28.54 | 0.8460 | 0.3431 |
| Improved DAES1 | 28.99 | 0.8562 | 0.3415 |
| Improved DAES2 | 28.85 | 0.8594 | 0.3221 |
| Improved DAES1 + HG | 29.52 | 0.8710 | 0.3082 |
| Improved DAES2 + HG | 29.36 | 0.8781 | 0.3182 |
| Improved CBDNetS1 | 31.49 | 0.9055 | 0.3762 |
| Improved CBDNetS2 | 31.67 | 0.9075 | 0.3826 |

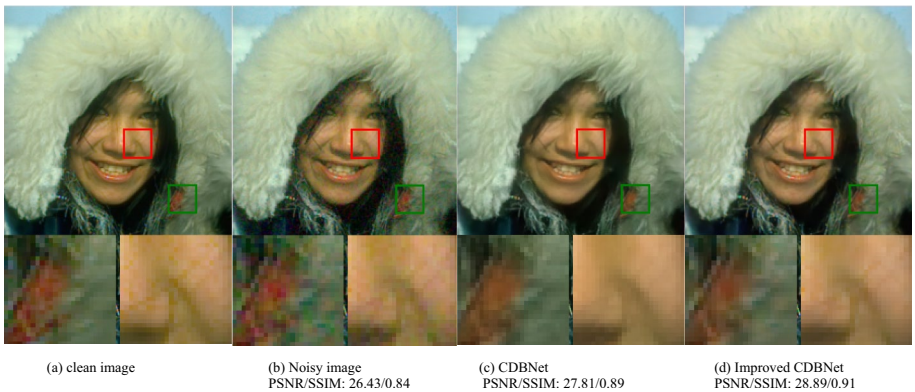


**Fig. 8** Denoising results of a BSDS500 image generated over the HG noise model + ISP. **a** Clean image. **b** Noisy image PSNR/SSIM: 26.43/0.84. **c** CDBNet PSNR/SSIM: 27.81/0.89. **d** Improved CDBNet PSNR/SSIM: 28.89/0.91

DAE is trained with image generated through the AWGN model and the heterogeneous Gaussian noise model respectively. In the heterogeneous Gaussian noise model with ISP, the improved DAE demonstrates significantly enhanced denoising performance on noisy images. Its PSNR/SSIM results outperform those of DnCNN and FFDNet, indicating that the proposed method can enhance the denoising performance of DCNN for non-Gaussian noisy images. The improved CBDNetS2 has the highest PSNR/SSIM results in all methods.

Figure 8 shows the denoising results on a BSDS500 image using CBDNet and its improved version under the ISP's heterogeneous Gaussian noise model. Compared to the original method, the improved approach preserves more local details in the denoised image, resulting in a better visual effect.

## Conclusion

Due to the limitations of training DCNNs solely using MSE loss, which cannot fully match non-Gaussian noise in images and may introduce additional information during the denoising process, resulting in a decrease in the visual quality of the denoised image. To address this, we explore various image evaluation metrics that describe image characteristics from different angles, such as residual statistical properties and image structural

similarity. These metrics are then employed as loss functions to improve the training of DCNN. This approach aims to enhance the generalization ability of DCNNs for non-Gaussian noise, improve the recovery of details in denoised images, and reduce the generation of artifacts. Additionally, we introduced novel training strategies to address the issue of automatically selecting appropriate weight coefficients for each task. These measures effectively enhanced the image denoising performance of the original DCNNs. Future research will explore the introduction of more reasonable image evaluation metrics, applying the MTL framework to new network architectures, such as non-reference image quality evaluation metrics and denoising networks based on Transformer. We will also consider integrating these metrics with self-supervised denoising methods for noisy images to reduce dependence on noise-free training data.

**Abbreviations**

| | |
|---|---|
| CNN | Convolutional neural network |
| AWGN | Additive white Gaussian noise |
| SSIM | Structural similarity metric |
| MTL | Multi-task learning |
| MRI | Magnetic resonance images |
| PDEs | Partial differential equations |
| DnCNN | Denoising convolutional neural network |
| ReLU | Rectified linear unit |
| BN | Batch normalization |
| RL | Layers and residual learning |
| CBDNet | Convolutional blind denoising network |
| SNRs | Signal-to-noise-ratios |
| MSE | Minimize mean square error |
| MOO | Multi-objective optimization |
| PSNR | Peak signal-to-noise ratio |
| DCNN | Denoising CNN |
| N2N | Noise2Noise |
| BM3D | Block matching and 3D collaborative filtering |
| DAE | Denoising autoencoder |
| FFDNet | A fast and flexible denoising convolutional neural network |

**Acknowledgements**
Not applicable.

**Authors' contributions**
All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Qian Xiang and Yong Tang. The first draft of the manuscript was written by Qian Xiang, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

**Funding**
This work is supported by Science and Technology Project of Education Department of Hubei Province (No. B2022396), and Natural Science Foundation of Hubei Province (2022CFB488).

**Availability of data and materials**
The data that support the findings of this study are available upon request from the authors.

## Declarations

**Ethics approval and consent to participate**
The human photos used in the paper are from the publicly available facial image datasets (https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html) that do not contain personally identifiable information or violate any privacy rights. Thus, ethics approval was not required for this research.

**Competing interests**
The authors declare no competing interests.

Received: 5 June 2023 Accepted: 8 April 2024
Published online: 18 April 2024

**References**

1. Chen S, Sun P, Song Y, Luo P (2022) DiffusionDet: diffusion model for object detection. arXiv preprint arXiv:2211.09788
2. Arathi T, Rahul C (2022) MRI denoising: a sparse ICA-based dictionary learning approach. Int J Med Eng Informatics 14(4):347–357
3. Mokari A, Ahmadyfard A (2017) Fast single image SR via dictionary learning. IET Image Proc 11(2):135–144
4. He L, Wang Y, Xiang Z (2019) Wavelet frame-based image restoration using sparsity, non-local, and support prior of frame coefficients. Vis Comput 35(2):151–174
5. Tian C, Fei L, Zheng W, Xu Y, Zuo W, Lin CW (2020) Deep learning on image denoising: an overview. Neural Netw 131:251–275
6. Zhang K, Zuo W, Chen Y et al (2017) Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Processing 26(7):3142–3155
7. Wei K, Fu Y, Yang J, Huang H (2020) A physics-based noise formation model for extreme low-light raw denoising. In Proceedings of the IEEE/CVF Conferenceon Computer Vision and Pattern Recognition, p 2758–2767
8. Guo S et al (2019) Toward convolutional blind denoising of real photographs. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
9. Chen H, Gu J, Liu Y et al (2024) Masked image training for generalizable deep image denoising [C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), p 1692–1703
10. Feng H, Wang L, Wang Y, Fan H, Huang H (2024) Learnability enhancement for low-light raw image denoising: a data perspective. IEEE Trans Pattern Anal Mach Intell 46(1):370–387.
11. Xiang Q, Peng LK et al (2020) Image denoising auto-encoders based on residual entropy maximum. IET Image Process 14(6):1164–1169
12. Liu D et al (2018) When image denoising meets high-level vision tasks: a deep learning approach. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, p 842–848
13. Yu Z, Qiang Y (2021) A survey on multi-task learning. arXiv:1707.08114v3
14. Jin X, Xu J, Tasaka K, Chen Z (2021) Multi-task learning-based all-in-one collaboration framework for degraded image super-resolution. ACM Transactions on Multimedia Computing Communications, and Applications (TOMM) 17(1):1–21
15. Ko JU, Jung JH, Kim M, Kong HB, Lee J, Youn BD (2021) Multi-task learning of classification and denoising (MLCD) for noise-robust rotor system diagnosis. Comput Ind 125:103385.
16. Marvasti-Zadeh SM, Ghanei-Yakhdan H, Kasaei S, Nasrollahi K, Moeslund TB (2021) Effective fusion of deep multi-tasking representations for robust visual tracking. The Visual Computer, p 1–21
17. Kendall A, Gal Y, Cipolla R (2018) Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition, p 7482–7491
18. Zhang K, Zuo W, Zhang L (2018) FFDNet: toward a fast and flexible solution for CNN-based image denoising. IEEE Trans Image Process 27(9):4608–4622.
19. Moran N, Schmidt D, Zhong Y, Coady P (2020) Noisier2noise: learning to denoise from unpaired noisy data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, p 12064–12072
20. Sun Y, Chen Y, Wang X, Tang X (2014) Deep learning face representation by joint identification-verification. Advances in neural information processing systems, 27, MIT Press, p 1988-1996
21. Kang G, Gao S, Yu L, Zhang D (2018) Deep architecture for high-speed railway insulator surface defect detection: denoising autoencoder with multitask learning. IEEE Trans Instrum Meas 68(8):2679–2690
22. Ozan S, Vladlen K (2019) multi-task learning as multi-objective optimization. arXiv: 1810.04650v2
23. Hang et al (2017) Loss functions for image restoration with neural networks. IEEE Trans Comput Imaging 3(1):47–57
24. Gu S, Li Y, Gool LV, Timofte R (2019) Self-guided network for fast image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision, p 2511–2520
25. Yang D, Sun J (2018) BM3D-Net: a convolutional neural network for transform-domain collaborative filtering. IEEE Signal Process Letters (25):55–59
26. Brooks T et al (2019) Unprocessing images for learned raw denoising. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), p 11036–11045
27. Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. Adv Neural Inform Process Syst 25:2951–2959
28. Adam et al (2011) An analysis of single-layer networks in unsupervised feature learning. J Mach Learn Res 15:215–223
29. Xu J, Li H, Liang Z, Zhang D, Zhang L (2018) Real-world noisy image denoising: a new benchmark. arXiv preprint arXiv:1804.02603
30. Anaya J, Barbu A (2014) RENOIR - a dataset for real low-light image noise reduction. J Vis Commun Image Represent 51:144–154
31. Plotz T, Roth S (2017) Benchmarking denoising algorithms with real photographs. In Proceedings of the IEEE conference on computer vision and pattern recognition, p 1586–1595
32. Martin, David et al (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001 2:416–423

## Publisher's Note

**Qian Xiang** received his Ph.D from the PLA Naval University of Engineering in 2005. He is currently a senior engineer of Wuchang Shouyi University. His research interests include digital image processing and computer vision.

**Yong Tang** received his Ph.D from the PLA Naval University of Engineering in 2010. He is currently a professor of Hubei Business College. His research interests include applied research of power electronics and information processing.

**Xiangyang Zhou** received his master degree from Wuhan University of Technology Communication Information System major in 2004. He is currently a lecturer of Wuchang Shouyi University. His research interests include remote sensing image processing and artificial intelligence.