IET Image Processing

Research Article

# Deep convolutional neural network for mixed random impulse and Gaussian noise reduction in digital images

ISSN 1751-9659
Received on 23rd July 2019
Revised 30th September 2020
Accepted on 6th November 2020
E-First on 18th February 2021
doi: 10.1049/iet-ipr.2019.0931
www.ietdl.org

*Mehdi Mafi*[1] ✉, *Walter Izquierdo*[1], *Harold Martin*[1], *Mercedes Cabrerizo*[1], *Malek Adjouadi*[1]
[1]*Center for Advanced Technology and Education, Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA*
✉ *E-mail: mmafi002@fiu.edu*

**Abstract:** This study utilises a deep convolutional neural network (CNN) implementing regularisation and batch normalisation for the removal of mixed, random, impulse, and Gaussian noise of various levels from digital images. This deep CNN achieves minimal loss of detail and yet yields an optimal estimation of structural metrics when dealing with both known and unknown noise mixtures. Moreover, a comprehensive comparison of denoising filters through the use of different structural metrics is provided to highlight the merits of the proposed approach. Optimal denoising results were obtained by using a 20-layer network with 40 × 40 patches trained on 400 180 × 180 images from the Berkeley segmentation data set (BSD) and tested on the BSD100 data set and an additional 12 images of general interest to the research community. The comparative results provide credence to the merits of the proposed filter and the comprehensive assessment of results highlights the novelty and performance of this CNN-based approach.

## 1 Introduction

Noise remains a challenge in many image acquisition and transmission systems. The presence of noise, regardless of its nature or source of origin, often leads to image degradation that is as difficult to overcome as it is to formulate its underlying effects. This study focuses on images containing impulse noise (salt and pepper, and random value) and additive Gaussian noise. Impulse noise is caused by analogue-to-digital (A/D) converter saturation, transmission and/or memory errors, and faulty pixels in camera sensors [1, 2]. Gaussian noise, on the other hand, is caused by three common factors: amplifier noise, shot noise, and film grain noise [3, 4]. In order to remove or attenuate significantly these sources of noise, smoothing filters are often used to decrease the noise variance while attempting to preserve the important image details as much as possible. However, noise removal is further complicated when the images are degraded by mixtures of impulse and Gaussian noise, whose combination significantly alters the structural metrics of any given image. It thus becomes necessary to find a reliable process by which the challenging effects of such mixed noise could be attenuated. Therefore, a standard course of action is to perform a suitable smoothing technique to the image before some form of gradient could be applied to preserve the finer image details. Given that derivatives could amplify the effect of noise, a trade-off must be negotiated between the task of decreasing the noise variance and that of keeping all the relevant image details. This trade-off should consider the difficult challenge faced in any ensuing edge detection that needs to be performed on the resulting denoised image.

Consequently, edge detection remains a difficult nontrivial problem and yet an essential preprocessing step for object identification, image segmentation, feature extraction, pattern recognition, and other relevant image processing tasks. When dealing with images, pertinent details can be useful for all types of real-world applications, but the concern has always been in delineating what really constitutes an edge with a high degree of similarity in contrast to other background pixels and noise data that could be misidentified as edges.

The current literature provides two comprehensive reviews on the use of neural networks for image denoising [5, 6] along with several studies proposing different convolutional neural networks (CNNs) with various merits and different application domains such as hyperspectral image classification [7, 8], and Gaussian denoising [9–12]. While the authors of [13] produce other significant work on convolutional networks for eye fixation prediction by using a three-layer network with two convolutional layers and one fully-connected layer. Multi-label image classification is offered in [14] through an AlexNet-like network with two auxiliary layers and one additional input layer. Fine-grained image classification is discussed in [15] and multi-view multitask gaze estimation using *n*- and 1-fully connected layers is provided in [16].

Several impulse and Gaussian denoising methods have also been proposed, such as switching adaptive median and fixed weighted mean filter (SAMFWMF) [17], the proposed filter in [18] and CNN filters [11, 12]. Moreover, there are also several mixed impulse-Gaussian denoising methods, among them mixed noise filter (MNF) [19] (a combination of non-local mean [20] and bilateral filters [21]), adaptive median filter [22], total variation-based filters [23, 24], fuzzy-based filters [25–30], sparsity-based filter [31–33], low-rank approximation filter [34, 35], non-local similarity filter [36], maximum likelihood estimation filter (PARIGI method) [37], genetic programming filter [38], finite element filter [39], morphological-based filter [40], and the so-called framelet filter [41].

The methods uniquely used for either impulse or Gaussian denoising have shown good performance in the presence of the targeted noise, but such performance is degraded when the two noises are combined (i.e. present in a given image at the same time). Previously reported methods that have considered mixed impulse and Gaussian denoising filters are based on traditional methods that could not properly confront such problem when in the presence of high noise levels, leading to loss of image details and excessive blurring burdening further the edge detection process.

For a fair assessment of the proposed method against others that have focused on mixed impulse and Gaussian noise under different intensity levels, a comparative study is provided contrasting the results obtained using the proposed deep learning algorithm against those from the most recent and effective denoising filters. In this comparative assessment, the focus is placed on the low-rank approximation algorithm (LSM-NLR) [35] which has proven to be effective for the removal of mixed impulse and Gaussian noise in images. The LSM-NLR method is based on Laplacian scale mixture (LSM) modelling and non-local low-rank regularisation. In

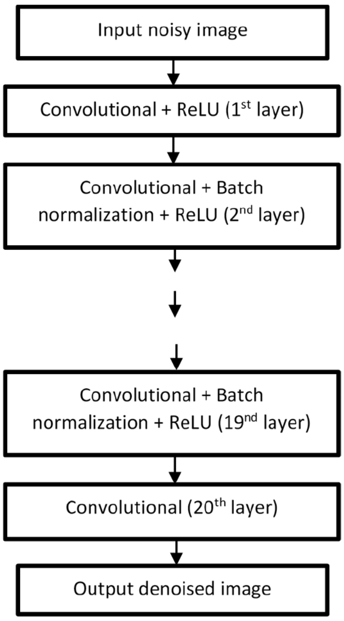
Input noisy image
Convolutional + ReLU (1st layer)
Convolutional + Batch normalization + ReLU (2nd layer)
Convolutional + Batch normalization + ReLU (19nd layer)
Convolutional (20th layer)
Output denoised image

**Fig. 1** *Network model*

order to model the impulse noise in LSM, a MAP estimator is defined by the authors in [35]. For non-local rank regularisation, a combination of the LSM model, the MAP estimator, and a low-rank regularisation model was used with the assumption that similar patches are interdependent, consequently resolving the denoising process by using an optimisation algorithm. The authors of this effective approach claim to obtain better result than the two-phase deblurring/denoising (TPD) method (or Cai1) [42], sparse and low-rank regularisation denoising (SLR) [43], the well-known BM3D [44], the non-locally centralised sparse representation [45], and the weighted encoding with sparse non-local regularisation (WESNR) [33].

It should be noted that the WESNR method does not have an impulse noise detection step due to its generated artefacts when in the presence of high noise levels. It defines the image as the product of sparse coding and a dictionary and in order to eliminate the noise, an optimal estimation of sparse code is calculated by encoding the noisy image over the dictionary. Because of the two different noise categories, the weight (close to 1 for pixels corrupted by Gaussian noise and smaller weights ($w$) for pixels corrupted by impulse noise) is assigned to residuals; and therefore, an optimal estimation for sparse coding is defined in the presence of mixed noise based on sparse regularisation. This method is claimed to outperform ROR-NLM [46], Cai [47], $l_1 - l_0$ [31], TF [21], and M + BM3D [44].

With all these challenges, the proposed study is motivated by three objectives: (i) To determine new ways for overcoming the persisting problems experienced by previously reported denoising methods in order to attenuate the effects of noise as much as possible while retaining more image details. (ii) To deblur the image in such a way as to yield an enhanced noise-free image were the absence of noise is visually appreciable. (iii) To preserve edge information and therefore yield sharper and continuous edge boundaries (one of the main aims of this study).

Herein, a novel denoising filter is introduced to combine the ability to preserve more edge details with better prospects for achieving high structural similarity to the original (noise-free) image even in the presence of high levels of mixed impulse and Gaussian noise. The obtained results, as presented and discussed later, are contrasted to well-known denoising filters to gauge the performance of the proposed denoising method.

## 2 Proposed method

The proposed method, as illustrated in Fig. 1, uses an end-to-end deep CNN to achieve optimal denoising of mixed impulse and Gaussian noise while, consequently, directly estimating the original noise-free image. Thereafter, batch normalisation is applied to speed up and improve the denoising process. Finally, the network is trained for both known and unknown noise levels. Known noise level is presumed when the training and testing data have similar noise levels. However, for unknown noise level, the network is trained based on a specific high noise intensity level and the testing data is assumed to have different levels of the noise intensity, which are presumed to be lower than those used in the training data.

### 2.1 Noise models

The normalised salt and pepper impulse noise model is expressed as follows:

$$x_c = \begin{cases} X_{\min} & \text{Probability } P_{\rm p} \\ X_{\max} & \text{Probability } P_{\rm s} \\ X_{\min} < c < X_{\min} & \text{Probability } 1 - P_{\rm p} - P_{\rm s} \end{cases} \tag{1}$$

In this type of representation, also used in [48], $c$ denotes the uncorrupted value of a given pixel that could be affected by salt noise with probability $P_{\rm s}$ and by pepper noise with probability $P_{\rm p}$. This normalised representation of the image assigns $X_{\min}$ to be 0 and $X_{\max}$ as 1. When using random value impulse noise as in [49], any given corrupted pixel has a random value between $X_{\min}$ and $X_{\max}$.

Gaussian noise is additive and independent. It can be the result of amplifiers noise, shot noise, film grain noise [50], and others. The model of Gaussian noise can thus be expressed as follows:

$$x_{\rm G}(i, j) = x(i, j) + n(i, j) \tag{2}$$

where $x_{\rm G}$ represents the noisy image containing Gaussian noise, $x$ is the original noise-free image, and $n$ is the additive Gaussian noise.

### 2.2 Evaluation metrics

Standard structural metrics are computed to compare the performance of multiple filters against the proposed method and gauge the quality of the denoised image. The following are the metrics used in this study:

- Feature similarity index (FSIM) [51] measures the quality of the denoised image based on the human visual system (HVS).
- Peak signal-to-noise ratio (PSNR) measures the level of noise remaining in the denoised image.

Equations (3)–(8) provide the formulations for FSIM and PSNR measures together with their respective related components. Hereafter, $x(i, j)$ represent a pixel $(i, j)$ in the original noise-free image, and $y(i, j)$ represent a pixel $(i, j)$ in the denoised image resulting from the filtering process.

The FSIM between the noisy and the denoised images is measured as follows:

$$\mathrm{FSIM} = \frac{\sum_{i \in \Omega} S_L(i) \cdot \mathrm{PC}_m(i)}{\sum_{i \in \Omega} \mathrm{PC}_m(i)} \tag{3}$$

where $\Omega$ is a partial domain of the whole image with $\mathrm{PC}_m(i) = \max(\mathrm{PC}_1(i), \mathrm{PC}_2(i))$, and where $S_L(i)$ is the similarity at the location $i$ as given below:

$$S_L(i) = S_{\mathrm{PC}}(i) S_{\mathrm{G}}(i) \tag{4}$$

$S_{\mathrm{PC}}(i)$ is the similarity measure between $\mathrm{PC}_1$ and $\mathrm{PC}_2$ as the extracted PC maps from the noisy and denoised images, respectively, as defined in the following equation:

$$S_{\mathrm{PC}}(i) = \frac{2\mathrm{PC}_1(i) \cdot \mathrm{PC}_2(i) + T_1}{\mathrm{PC}_1(i)^2 + \mathrm{PC}_2(i)^2 + T_1} \tag{5}$$

where $T_1$ is a positive constant used to increase the stability of $S_{\mathrm{PC}}(i)$. $S_{\mathrm{G}}(i)$, defined in (6), represents the similarity measure between $G_1$ and $G_2$ representing the $GM$ maps extracted from the noisy and denoised images, respectively,

$$S_{\mathrm{G}}(i) = \frac{2G_1(i) \cdot G_2(i) + T_2}{G_1(i)^2 + G_2(i)^2 + T_2} \tag{6}$$

$GM = \sqrt{G_h^2 + G_v^2}$, where $G_h$ and $G_v$ are the partial derivatives in the horizontal and vertical directions. $T_2$ is a positive constant dependent on the dynamic range of $GM$.

The PSNR measure is given as follows:

$$\mathrm{PSNR} = 10\log\frac{(\max(x))^2}{\mathrm{MSE}} \tag{7}$$

where the MSE defines the mean square error as in (8) with $\max(x)$ expressing the maximum pixel intensity of image $x$.

$$\mathrm{MSE} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x(i, j) - y(i, j))^2}{M \times N} \tag{8}$$

Here $M$ and $N$ define the $x$ and $y$ dimensions of the image.

### 2.3 Related work on the use of CNN for denoising images

The following are some notable research endeavours that have used CNNs for image denoising. The work in [52] proposes a combination of sparse coding and deep neural networks, pre-trained with denoising auto-encoders (DAs), as an alternative to training pure DAs. The study in [53] uses a multilayer perceptron (MLP) for image denoising. While study in [11] presents a trainable non-linear reaction-diffusion (TRND) method and uses supervised training to construct a dynamic non-linear reaction-diffusion model with time-dependent parameters (linear filters and influence functions) for Gaussian denoising. This method in [11] is similar to the feedback CNN presented in [54], and in both cases, each iteration (stage) of the proposed diffusion process uses convolutional operations of a set of linear filters. The work in [55] introduced a super-resolution algorithm and CNN for mixed noise removal. The work in [12] introduces a deep CNN method (DnCNN) that can be seen as a generalisation of TRND [11] that (i) is easier to train, by replacing the influence function with a rectified linear units (ReLUs) [56]; (ii) increases architectural depth (number of convolution layers) to improve image modelling capacity; and (iii) incorporates batch normalisation [57] to improve performance.

All the aforementioned networks are used solely for Gaussian denoising and all of them, except for [12], were used to remove known/predefined noise levels. As Gaussian noise is additive, the network in [12] removes the noise by using residual learning, i.e. learning the noise. However, when Gaussian and impulse noise are mixed the resulting interference is no longer additive and the network in [12] no longer provides adequate denoising results. It is for this reason, that the proposed approach did not use residual learning to directly estimate the denoised image.

### 2.4 Batch normalisation and network parameters

During training, any change to a deep neural network layer's parameters causes a change in the distribution of the subsequent layer's input, referred to as an internal covariate shift. Batch normalisation [57] can alleviate internal covariate shift by learning the normalisation parameters of each part of the model and applying it to each mini-batch training.

Empirically, it can be observed that the network's depth and patch size are dependent on the type and level of noise present. Specifically, larger patch sizes are shown to exhibit better performance in the presence of higher noise levels [58]. The network presented in [11] used ten convolutional layers (or five stages) with patches of 61 × 61 to remove a predefined level of Gaussian noise. The network in [12] uses 17 layers of 40 × 40 patches for known-noise-level denoising and 20 layers of 50 × 50 patches for unknown noise-level Gaussian denoising.

### 2.5 Network model

The proposed CNN model is a modified version of the VGG [59] and DnCNN [12] models aimed at overcoming the challenge of mixed impulse and Gaussian noise removal. The network's input is a noisy image $y_i$, produced by artificially injecting noise to a clean original image $(x_i)$, and the network's output $f(y_i)$ is an estimate of the original noise-free image. The network's loss function is the summation of the squared error between the estimated and original noise-free images as formulated in (9). Finally, the network's parameters are updated by minimising this loss function as in [6, 60].

$$L = \sum_{i=1}^{N} \| f(y_i) - x_i \|_2^2 \tag{9}$$

where $N$ is the number of training image sets $(\{y_i, x_i\})$.

As previously mentioned, the network used in this study resembles that used in [12] with few changes. It is composed of three different types of layers, where the first is a convolutional layer of 64 $3 \times 3 \times 1$ filters with ReLU non-linear activation functions [9, 10, 13, 56] used to create 64 feature maps, the second through second-to-last layers are batch normalised [10, 57] convolutional layers of 64 $3 \times 3 \times 64$ filters with ReLU activations [56]; and the last layer is composed of a single convolutional kernel of shape $3 \times 3 \times 64$ used to output the reconstructed image. The use of ReLU activations [56] on convolutional layers separates the mixed noise from the noisy observations through the hidden layers. Finally, the input images are directly padded with zeros to reduce boundary artefact [12] resulting from size mismatches between different input images. Fig. 1 illustrates the proposed CNN-based network model.

In contrast to [12], this network attempts to predict the noise-free image directly instead of obtaining it from subtracting the predicted residuals of the noisy image. This is a consequence of the non-additive nature of the noise types considered in this study.

Although batch normalisation is used to prevent overfitting, additional steps are also considered to ensure the desired outcome. For example, every time the network starts a new epoch (a run through the training data) a new random seed is used to regenerate the noisy images. This extra step has proved to be a very helpful regulation technique as it prevents the network from seeing the same input image twice, or at least assign a very low probability to such event, allowing the network to better generalise.

Through empirical evaluations, it is noted that optimal denoising results are obtained by using 20 layers with 40 × 40 patches for both known and unknown noise-level denoising. Although either, stochastic gradient descent (SGD) [61] and Adam gradient-based optimisation [9, 62] could have been used, previous CNNs [7, 10, 12, 14–16] have used SGD, and the performance of networks with and without batch normalisation for both SGD and Adam are assessed in [12]. This demonstrates that batch normalisation can significantly improve the PSNR for SGD by increasing the number of epochs. Therefore, in this study SGD [61] is used.

Finally, all the source code developed and modified for this study is available at: 'https://github.com/wizquierdo/DnCNN' for other researchers to perform comparative assessments and explore any potential improvements.

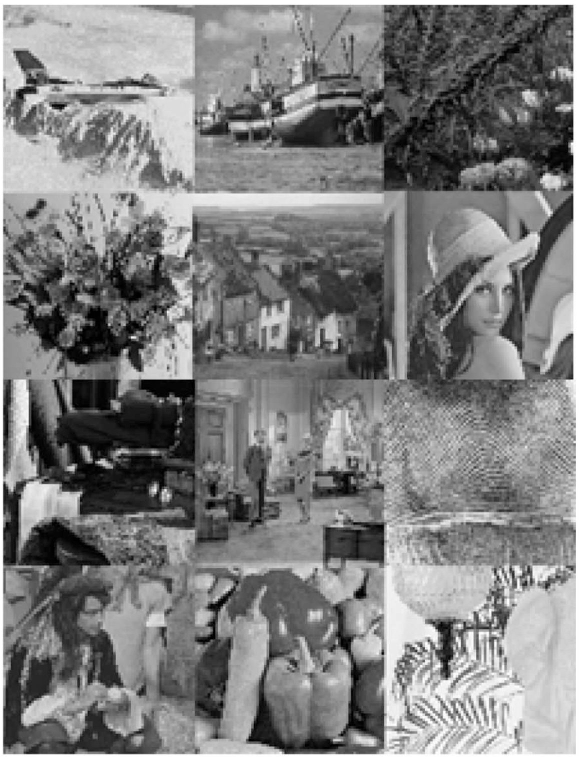

**Fig. 2** *12 test images considered*

## 3 Results

In the implementation phase, 400 180 × 180 pixels images from the Berkeley segmentation dataset (BSD) are used to train the described network for both known and unknown noise-level removal, similar to [11, 12, 35]. While for testing, the data sets used include the BSD100 dataset and the additional 12 images shown in Fig. 2 not seen in the training. The study in [35] uses these same data sets.

As previously stated, the optimal results were obtained by using a 20-layer network with 40 × 40 patches for any noise-level. The SGD method is used with an additional learning rate of 0.1 (which decreased over subsequent epochs), weight decay of 0.0001, momentum of 0.09, and mini-batches of size 128 similar to studies reported in [12–14, 16, 63]. It should be noted that the type of noise mixture affected the numbers of epochs that the model needed to complete the training phase.

All implementations were carried out in MATLAB 2017b using the MatConvNet package [12, 20] for CNNs on a PC with Nvidia Quadro M6000 GPU. The time required to train the network varied between 24 and 48 h depending on the noise mixture (as different mixtures required a different number of epochs).

Once the network was trained, the results obtained from the proposed denoising method are compared to the results obtained from the WESNR [33] and the LSM-NLR [35] methods on different images and under the same mixed impulse and Gaussian noise intensities.

Fig. 3 shows the comparative results when removing Gaussian noise with a standard deviation of 20 and 50% salt and pepper impulse noise from the test image 'Vase'. Fig. 4 shows the denoising comparison for Gaussian noise with a standard deviation of 20 and 30% random value impulse noise the from test image 'Flower'. Fig. 5 presents the performance in the presence of Gaussian noise with a standard deviation of 10, with 40% salt and pepper impulse noise, and 10% random value impulse noise from the test image 'Boat'. As seen from these figures, the proposed CNN attains better performance at preserving relevant image details than all other filters, with the highest similarity measure, and resulting in the least amount of noise residue. Moreover, these improvements lead to better edge tracking, especially when dealing with high-intensity mixtures of impulse and Gaussian noise.

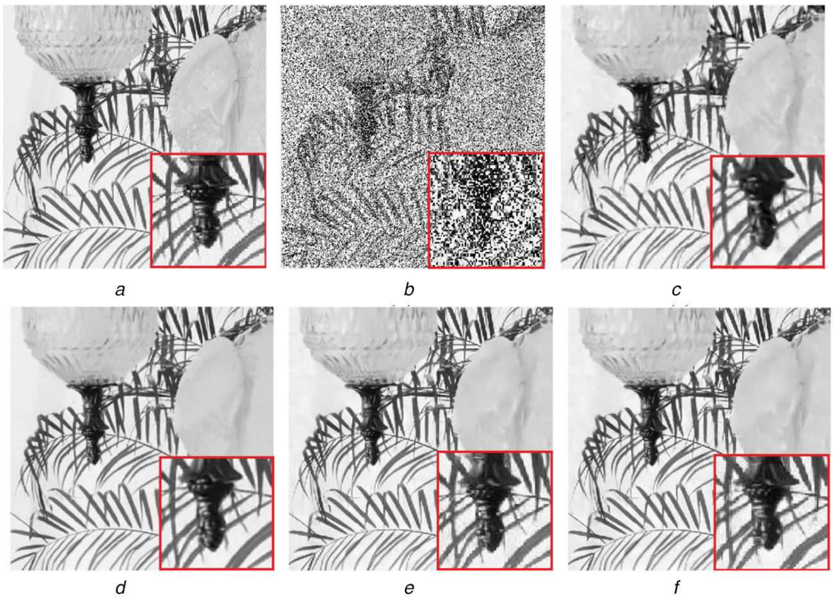

**Fig. 3** *Denoising results of different filters on test image 'Vase' corrupted with Gaussian with standard deviation 20 and salt and pepper impulse noise with 50%*

***(a)*** Original image, ***(b)*** Noisy image; images denoised by, ***(c)*** WESNR [33] (PSNR = 24.43 dB, FSIM = 0.9235), ***(d)*** LSM-NLR [35] (PSNR = 29.24 dB, FSIM = 0.9556), ***(e)*** Unknown noise-level proposed CNN (PSNR = 29.17 dB, FSIM = 0.9532), ***(f)*** Known noise-level proposed CNN (PSNR = 29.58 dB, FSIM = 0.9586)

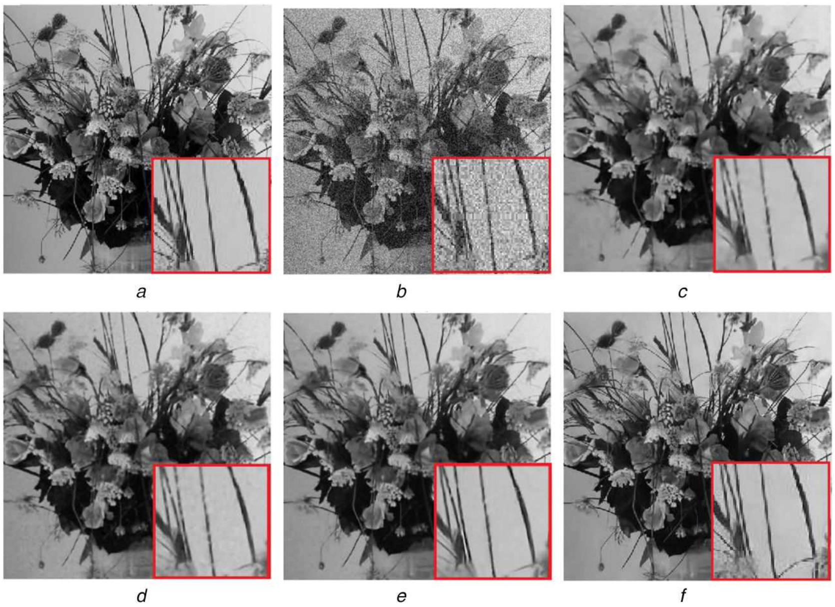

**Fig. 4** *Denoising results of different filters on test image 'Flower' corrupted with Gaussian with standard deviation 20 and random value impulse noise with 30%*
***(a)*** Original image, ***(b)*** Noisy image; images denoised by, ***(c)*** WESNR [33] (PSNR = 23.04 dB, FSIM = 0.8956), ***(d)*** $l_0$-NLR [35] (PSNR = 23.51 dB, FSIM = 0.9071), ***(e)*** LSM–NLR [35] (PSNR = 24.36 dB, FSIM = 0.9156), ***(f)*** Unknown noise-level proposed CNN (PSNR = 27.07 dB, FSIM = 0.9482)

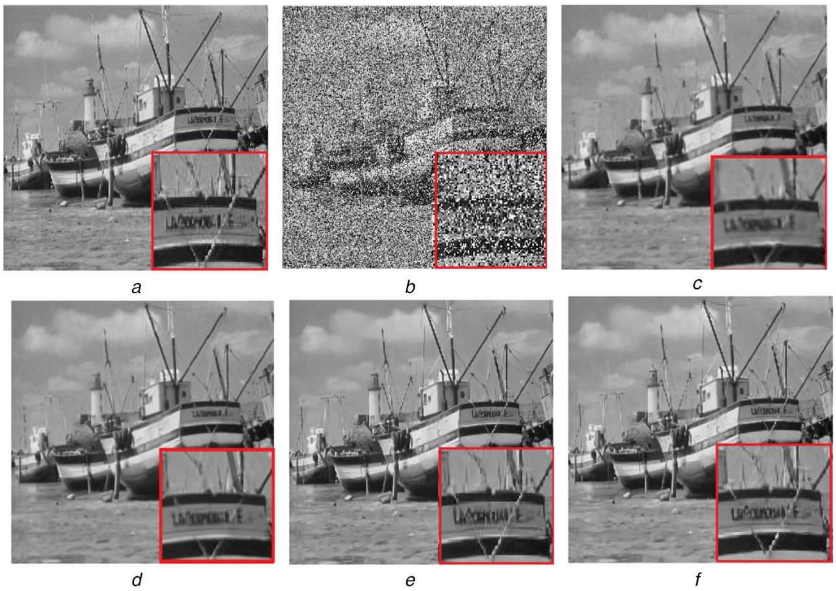

**Fig. 5** *Denoising results of different filters on test image 'Boat' corrupted with Gaussian with standard deviation 10, salt and pepper impulse noise with 40% and random value impulse noise with 10%*
***(a)*** Original image, ***(b)*** Noisy image; images denoised by, ***(c)*** WESNR [33] (PSNR = 27.32 dB, FSIM = 92.75), ***(d)*** LSM-NLR [35] (PSNR = 28.89 dB, FSIM = 0.9482), ***(e)*** Unknown noise-level proposed CNN (PSNR = 30.97 dB, FSIM = 0.9620), ***(f)*** Known noise-level proposed CNN (PSNR = 30.82 dB, FSIM = 0.9646)

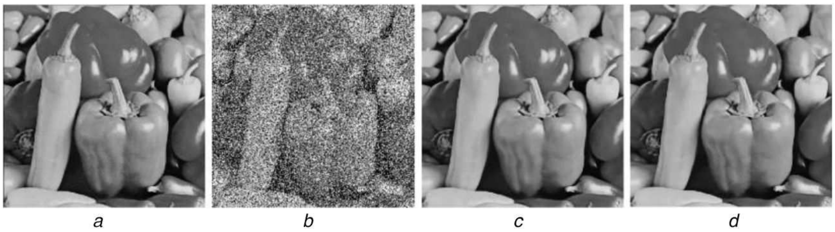

**Fig. 6** *Denoising results of proposed CNN on test image 'Fruits'*
***(a)*** Original test image, ***(b)*** Corrupted image with mixed Gaussian (s.d. = 10) and salt and pepper (50%), ***(c)*** Unknown noise-level denoising (PSNR = 33.28 dB, FSIM = 0.9736), ***(d)*** Known noise-level denoising (PSNR = 33.53 dB, FSIM = 0.9743)

Finally, Figs. 6–11 show the case the denoising results of the proposed CNN trained with varying degrees of known and unknown noise-levels in the presence of different intensity mixtures of Gaussian and impulse noises on the testing images.

Tables 1 and 2 show the results of known and unknown noise-levels removal for mixed Gaussian and impulse noise. They, respectively, show the results obtained from the average PSNR and the average FSIM [51] metrics from the 12 test images shown in

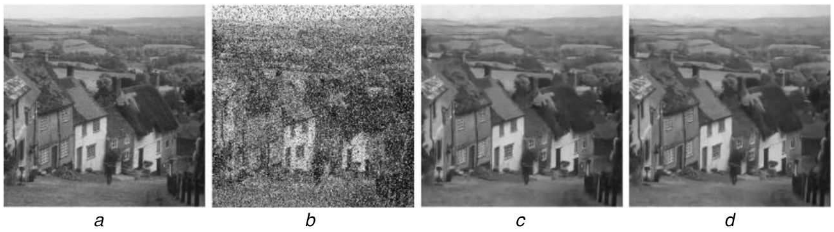

**Fig. 7** *Denoising results of proposed CNN on test image 'Hill'*
***(a)*** Original test image, ***(b)*** Corrupted image with mixed Gaussian (s.d. = 30) and salt and pepper (20%), ***(c)*** Unknown noise-level denoising (PSNR = 28.57 dB, FSIM = 0.9200), ***(d)*** Known noise-level denoising (PSNR = 28.77 dB, FSIM = 0.9236)

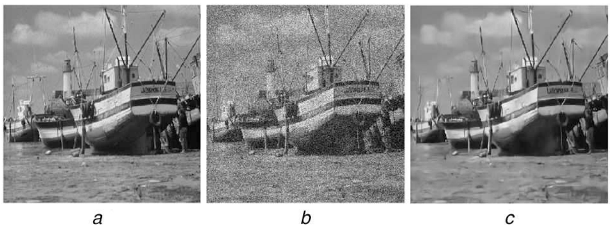

**Fig. 8** *Denoising results of proposed CNN on test image 'Boat'*
***(a)*** Original test image, ***(b)*** Corrupted image with mixed Gaussian (s.d. = 30) and random value impulse noise (30%), ***(c)*** Unknown noise-level denoising (PSNR = 28.24 dB, FSIM = 0.9211)

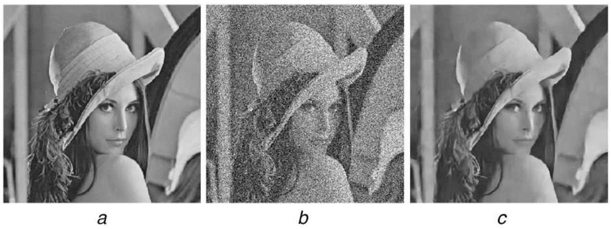

**Fig. 9** *Denoising results of proposed CNN on test image 'Lena'*
***(a)*** Original test image, ***(b)*** Corrupted image with mixed Gaussian (s.d. = 50) and random value impulse noise (50%), ***(c)*** Unknown noise-level denoising (PSNR = 27.31 dB, FSIM = 0.9107)

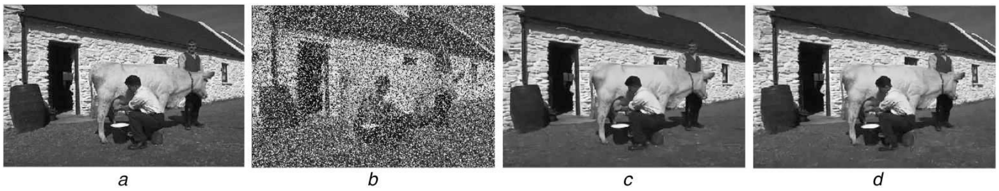

**Fig. 10** *Denoising results of proposed CNN on test image '385039 of BSD100 dataset'*
***(a)*** Original test image, ***(b)*** Corrupted image with mixed Gaussian (s.d. = 10) and salt and pepper (40%) random value impulse noise (10%), ***(c)*** Unknown noise-level denoising (PSNR = 27.18 dB, FSIM = 0.8902), ***(d)*** Known noise-level denoising (PSNR = 27.09 dB, FSIM = 0.8962)

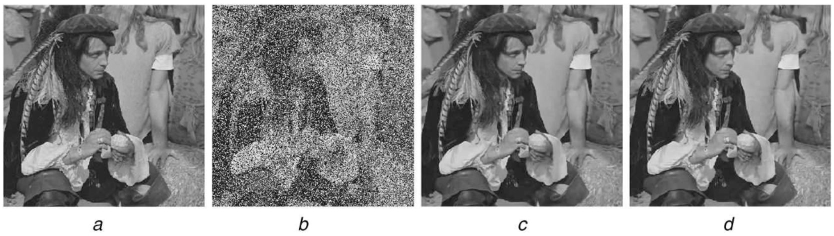

**Fig. 11** *Denoising results of proposed CNN on test image 'Couple'*
***(a)*** Original test image, ***(b)*** Corrupted image with mixed Gaussian (s.d. = 20) and salt and pepper (10%) random value impulse noise (30%), ***(c)*** Unknown noise-level denoising (PSNR = 28.16 dB, FSIM = 0.9266), ***(d)*** Known noise-level denoising (PSNR = 28.55 dB, FSIM = 0.9303)

Fig. 2 after the denoising process. Gaussian noise with a standard deviation of 10, 20, 30, and 50, and salt and pepper impulse noises with standard deviation 10, 20, 30, 40, and 50% were injected. Known noise-level denoising required 125 epochs while unknown noise-level denoising required 150; further increments of the number of training epochs yielded better results in both cases. The network was run 50 times for each noise level over the testing set and the means and standard deviations of the results were calculated.

Tables 3 and 4 show the results obtained for the average PSNR and the average FSIM measures on the BSD100 data set images. The collection method, number of epochs, and noise levels injected are equal to those of Tables 1 and 2.

Tables 5 and 6 show the average PSNR and the average FSIM measures for the 12 test images for unknown noise-level of mixed Gaussian and random value impulse noise. The collection method, number of epochs, and noise levels injected are equal to those of Tables 1 and 2.

Tables 7 and 8 show the average PSNR and average FSIM for the same 12 test images after performing both unknown and known noise-level denoising for mixed Gaussian, salt and pepper, impulse, and random value impulse noise. Gaussian noise at 10 and 20, salt and pepper impulse noises at 10 and 40%, and random value impulse noise at 10 and 30% are used. The number of epochs for known noise-level denoising is 30 and for unknown noise-level denoising is 35.

Tables 9 and 10 show the PSNR and FSIM of the results from unknown noise-level denoising for mixed Gaussian and random value impulse noise on the BSD100 data set images. The combinations of Gaussian noise with standard deviations 10, 20, 30, and 50 and random value impulse noise at 10, 20, 30, 40, and 50% are used. The network was trained for 150 epochs while further training showed added improvement.

Tables 11 and 12 show the PSNR and FSIM of the results of both known and unknown noise-level denoising from mixed Gaussian, salt and pepper impulse, and random value impulse noise images of the BSD100 data set. Gaussian noise with standard deviations 10 and 20, salt and pepper impulse noises at 10 and 40%, and random value impulse noise at 10 and 30%. The network was trained for 30 epochs for known noise-level denoising and for 35 epochs for unknown noise-level denoising.

**Table 1** PSNR (DB) comparison on 12 test images for unknown versus known noise-level denoising for mixed Gaussian and salt and pepper

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 50 | 28.95 | 29.75 | 29.89 | 30.60 | **31.0823 ± 0.0095** | **31.4845 ± 0.0127** |
| $\sigma = 20$ | 50 | 26.73 | 27.50 | 27.75 | 28.51 | **28.6711 ± 0.0096** | **28.9127 ± 0.0112** |
| $\sigma = 30$ | 20 | 26.80 | 26.59 | 27.09 | 28.31 | **28.3351 ± 0.0067** | **28.5869 ± 0.0836** |
| | 50 | 24.52 | 25.85 | 26.13 | 26.70 | **26.9530 ± 0.1371** | **27.1718 ± 0.0138** |
| $\sigma = 50$ | 10 | 20.80 | 24.44 | 24.83 | 26.00 | **26.1794 ± 0.0063** | **26.4956 ± 0.0063** |
| | 50 | 14.43 | 23.35 | 23.56 | 24.36 | **24.5109 ± 0.0136** | **24.8086 ± 0.0116** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 2** FSIM (%) comparison on 12 test images for unknown versus known noise-level denoising for mixed Gaussian and salt and pepper

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 50 | 95.29 | 96.49 | 96.39 | 96.63 | **$0.9677 \pm 1.1773 \times 10^{-4}$** | **$0.9703 \pm 1.0412 \times 10^{-4}$** |
| $\sigma = 20$ | 50 | 91.99 | 93.32 | 93.38 | 93.76 | **$0.9409 \pm 1.8120 \times 10^{-4}$** | **$0.9430 \pm 2.1311 \times 10^{-4}$** |
| $\sigma = 30$ | 20 | 91.55 | 93.02 | 93.12 | 93.21 | **0.9346 ± 0.0014** | **$0.9367 \pm 1.6123 \times 10^{-4}$** |
| | 50 | 88.80 | 90.02 | 90.04 | 91.06 | **$0.9133 \pm 3.1004 \times 10^{-4}$** | **$0.9154 \pm 4.0265 \times 10^{-4}$** |
| $\sigma = 50$ | 10 | 82.29 | 89.31 | 89.36 | 90.04 | **$0.9000 \pm 2.0388 \times 10^{-4}$** | **$0.9005 \pm 2.1699 \times 10^{-4}$** |
| | 50 | 66.08 | 83.69 | 83.50 | 85.44 | **$0.8637 \pm 4.0668 \times 10^{-4}$** | **$0.8661 \pm 4.4334 \times 10^{-4}$** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 3** PSNR (DB) comparison on BSD100 data set for unknown vs. known noise-level denoising for mixed Gaussian and salt and pepper

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 50 | 26.62 | 27.54 | 27.36 | 28.17 | **29.0404 ± 0.0046** | **29.4035 ± 0.0055** |
| $\sigma = 20$ | 50 | 24.81 | 25.86 | 26.12 | 26.88 | **27.1435 ± 0.0049** | **27.3621 ± 0.0053** |
| $\sigma = 30$ | 20 | 24.94 | 24.61 | 25.27 | 26.94 | **27.0714 ± 0.0031** | **27.2842 ± 0.0021** |
| | 50 | 22.92 | 24.43 | 24.81 | 25.44 | **25.7770 ± 0.0060** | **25.9552 ± 0.0044** |
| $\sigma = 50$ | 10 | 19.82 | 22.13 | 22.80 | 24.05 | **25.2762 ± 0.0031** | **25.5064 ± 0.0033** |
| | 50 | 14.44 | 22.22 | 22.66 | 23.96 | **23.9800 ± 0.0058** | **24.0689 ± 0.0048** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 4** FSIM (%) comparison on BSD100 data set for unknown and known noise-level denoising for mixed Gaussian and salt and pepper

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 50 | 86.45 | 89.96 | 90.06 | 89.87 | **$0.8979 \pm 1.2083 \times 10^{-4}$** | **$0.9085 \pm 9.5743 \times 10^{-5}$** |
| $\sigma = 20$ | 50 | 80.61 | 83.25 | 83.79 | 83.83 | **$0.8528 \pm 2.1385 \times 10^{-4}$** | **$0.8638 \pm 1.4583 \times 10^{-4}$** |
| $\sigma = 30$ | 20 | 80.05 | 82.79 | 83.00 | 83.00 | **$0.8500 \pm 1.4967 \times 10^{-4}$** | **$0.8551 \pm 1.0677 \times 10^{-4}$** |
| | 50 | 78.45 | 80.43 | 80.66 | 80.75 | **0.8136 ± 0.200** | **$0.8170 \pm 1.8009 \times 10^{-4}$** |
| $\sigma = 50$ | 10 | 73.45 | 76.71 | 78.05 | 80.51 | **$0.8060 \pm 1.3329 \times 10^{-4}$** | **$0.8083 \pm 1.8824 \times 10^{-4}$** |
| | 50 | 63.90 | 75.22 | 75.60 | 75.66 | **$0.7627 \pm 3.1885 \times 10^{-4}$** | **$0.7578 \pm 2.5994 \times 10^{-4}$** |

Bold values emphasize the obtained results of the proposed filter in this paper.

## 4 Conclusion

**Table 5** PSNR (DB) comparison on 12 test images for unknown noise-level denoising for mixed Gaussian and random value impulse

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) |
|---|---|---|---|---|---|---|
| $\sigma = 10$ | 10 | 30.24 | 31.25 | 31.36 | 32.30 | **33.0696 ± 0.0049** |
| | 20 | 29.36 | 29.46 | 29.86 | 30.82 | **31.2961 ± 0.0100** |
| | 30 | 28.40 | 27.74 | 28.55 | 29.37 | **30.3377 ± 0.0116** |
| | 40 | 27.02 | 26.74 | 26.92 | 27.24 | **29.4561 ± 0.0080** |
| | 50 | 25.30 | 24.72 | 25.18 | 25.36 | **28.6942 ± 0.0423** |
| $\sigma = 20$ | 10 | 27.69 | 28.62 | 28.90 | 29.22 | **29.8603 ± 0.3333** |
| | 20 | 27.09 | 27.50 | 27.78 | 28.27 | **29.8453 ± 0.0082** |
| | 30 | 26.42 | 26.21 | 26.74 | 27.28 | **28.9769 ± 0.0097** |
| | 40 | 25.24 | 24.83 | 25.37 | 26.08 | **28.4833 ± 0.0084** |
| | 50 | 23.86 | 23.19 | 24.00 | 24.62 | **27.9477 ± 0.0236** |
| $\sigma = 30$ | 10 | 26.11 | 26.56 | 26.95 | 26.98 | **27.3427 ± 0.0097** |
| | 20 | 25.55 | 25.49 | 25.92 | 26.20 | **28.2655 ± 0.0115** |
| | 30 | 24.92 | 24.30 | 25.00 | 25.39 | **27.8864 ± 0.0104** |
| | 40 | 23.74 | 23.11 | 23.74 | 24.33 | **27.4426 ± 0.0100** |
| | 50 | 22.30 | 21.62 | 22.52 | 23.18 | **27.0158 ± 0.0112** |
| $\sigma = 50$ | 10 | 23.16 | 23.83 | 24.14 | 24.27 | **26.5025 ± 0.0031** |
| | 20 | 22.59 | 22.73 | 23.20 | 23.61 | **26.5615 ± 0.0055** |
| | 30 | 21.84 | 21.59 | 22.31 | 22.85 | **26.3021 ± 0.0084** |
| | 40 | 20.78 | 20.73 | 21.14 | 21.95 | **25.6629 ± 0.0086** |
| | 50 | 19.35 | 19.34 | 20.08 | 20.73 | **24.7693 ± 0.0160** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 6** FSIM (%) comparison on 12 test images for unknown noise-level denoising for mixed Gaussian and random value impulse

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) |
|---|---|---|---|---|---|---|
| $\sigma = 10$ | 10 | 96.65 | 97.48 | 97.57 | 97.63 | **0.9777 ± 8.6987 × $10^{-5}$** |
| | 20 | 96.06 | 96.63 | 96.78 | 96.99 | **0.9712 ± 1.0816 × $10^{-4}$** |
| | 30 | 95.25 | 95.32 | 95.78 | 96.17 | **0.9677 ± 1.6653 × $10^{-4}$** |
| | 40 | 93.69 | 93.25 | 93.93 | 93.88 | **0.9480 ± 2.0952 × $10^{-4}$** |
| | 50 | 90.99 | 90.23 | 90.96 | 91.00 | **0.9386 ± 2.1346 × $10^{-4}$** |
| $\sigma = 20$ | 10 | 92.82 | 94.29 | 94.77 | 94.78 | **0.9608 ± 1.0206 × $10^{-4}$** |
| | 20 | 92.12 | 93.31 | 93.56 | 93.80 | **0.9543 ± 1.1349 × $10^{-4}$** |
| | 30 | 91.27 | 91.91 | 92.15 | 92.67 | **0.9451 ± 2.2174 × $10^{-4}$** |
| | 40 | 89.44 | 89.05 | 89.78 | 90.83 | **0.9366 ± 2.0672 × $10^{-4}$** |
| | 50 | 87.07 | 85.86 | 86.70 | 87.92 | **0.9278 ± 2.7104 × $10^{-4}$** |
| $\sigma = 30$ | 10 | 90.19 | 91.06 | 91.67 | 91.68 | **0.9414 ± 9.0921 × $10^{-4}$** |
| | 20 | 89.32 | 89.66 | 89.86 | 90.42 | **0.9382 ± 1.6833 × $10^{-4}$** |
| | 30 | 88.30 | 87.91 | 87.96 | 89.17 | **0.9303 ± 0.0020** |
| | 40 | 85.95 | 84.47 | 85.06 | 86.59 | **0.9223 ± 2.1602 × $10^{-4}$** |
| | 50 | 82.93 | 81.13 | 81.93 | 83.78 | **0.9136 ± 4.1212 × $10^{-4}$** |
| $\sigma = 50$ | 10 | 82.71 | 85.19 | 85.54 | 86.31 | **0.9112 ± 8.6603 × $10^{-4}$** |
| | 20 | 80.92 | 83.09 | 82.66 | 84.60 | **0.9075 ± 1.7635 × $10^{-4}$** |
| | 30 | 78.94 | 80.82 | 79.86 | 82.72 | **0.9012 ± 2.2361 × $10^{-4}$** |
| | 40 | 76.17 | 76.79 | 76.39 | 79.66 | **0.8927 ± 2.9754 × $10^{-4}$** |
| | 50 | 72.51 | 73.53 | 73.74 | 76.80 | **0.8827 ± 3.8588 × $10^{-4}$** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 7** PSNR (DB) comparison on 12 test images for unknown and known noise-level denoising for mixed Gaussian, salt and pepper impulse and random value impulse

| Gaussian noise | Impulse noise, % | Random value noise level, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 40 | 10 | 27.25 | 28.00 | 28.36 | 29.20 | **31.9400 ± 0.0179** | **30.7781 ± 0.0104** |
| $\sigma = 20$ | 10 | 30 | 25.27 | 25.74 | 25.87 | 26.10 | **28.2207 ± 0.0089** | **28.6746 ± 0.0102** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 8** FSIM (%) comparison on 12 test images for unknown and known noise-level denoising for mixed Gaussian, salt and pepper impulse and random value impulse

| Gaussian noise | Impulse noise, % | Random value noise level, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 40 | 10 | 93.53 | 95.06 | 95.14 | 96.09 | **0.9738 ± 1.0498 × $10^{-4}$** | **0.9665 ± 1.1537 × $10^{-4}$** |
| $\sigma = 20$ | 10 | 30 | 90.30 | 90.56 | 90.56 | 91.32 | **0.9348 ± 2.6162 × $10^{-4}$** | **0.9379 ± 2.0485 × $10^{-4}$** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 9** PSNR (DB) comparison on BSD100 data set for unknown noise-level denoising for mixed Gaussian and random value impulse

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) |
|---|---|---|---|---|---|---|
| $\sigma = 10$ | 10 | 27.66 | 29.13 | 29.15 | 30.12 | **31.9660 ± 0.0020** |
| | 20 | 27.06 | 27.67 | 27.79 | 28.54 | **30.1583 ± 0.0029** |
| | 30 | 26.44 | 26.27 | 26.75 | 27.27 | **29.0402 ± 0.0029** |
| | 40 | 25.40 | 25.22 | 25.43 | 27.27 | **28.1507 ± 0.0040** |
| | 50 | 24.09 | 23.61 | 23.92 | 24.37 | **27.4191 ± 0.0023** |
| $\sigma = 20$ | 10 | 25.67 | 26.97 | 27.22 | 27.64 | **28.9212 ± 0.0033** |
| | 20 | 25.47 | 26.07 | 26.21 | 26.73 | **28.6438 ± 0.0031** |
| | 30 | 25.04 | 25.03 | 25.35 | 25.86 | **27.7893 ± 0.0055** |
| | 40 | 24.13 | 23.85 | 24.24 | 24.85 | **27.2935 ± 0.0043** |
| | 50 | 22.98 | 22.43 | 23.01 | 23.45 | **26.7856 ± 0.0021** |
| $\sigma = 30$ | 10 | 24.86 | 25.38 | 25.71 | 25.87 | **26.6653 ± 0.0171** |
| | 20 | 24.46 | 24.35 | 24.65 | 24.96 | **27.2172 ± 0.0164** |
| | 30 | 23.97 | 23.53 | 24.01 | 24.51 | **26.7719 ± 0.0057** |
| | 40 | 23.03 | 22.56 | 23.01 | 23.73 | **26.3448 ± 0.0062** |
| | 50 | 21.74 | 21.21 | 21.89 | 22.69 | **25.9463 ± 0.0054** |
| $\sigma = 50$ | 10 | 22.67 | 23.09 | 23.19 | 23.40 | **25.6711 ± 0.0012** |
| | 20 | 22.21 | 22.38 | 22.60 | 23.08 | **25.6309 ± 0.0023** |
| | 30 | 21.47 | 21.36 | 21.79 | 22.48 | **25.3034 ± 0.0030** |
| | 40 | 20.50 | 20.58 | 20.82 | 21.85 | **24.7095 ± 0.0056** |
| | 50 | 19.13 | 19.14 | 19.65 | 20.62 | **23.8956 ± 0.0050** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 10** FSIM (%) comparison on BSD100 data set for unknown noise-level denoising for mixed Gaussian and random value impulse

| Gaussian noise | Impulse noise, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) |
|---|---|---|---|---|---|---|
| $\sigma = 10$ | 10 | 87.25 | 91.82 | 92.51 | 92.70 | **0.9358 ± 5.9362 × $10^{-5}$** |
| | 20 | 86.12 | 90.00 | 90.38 | 90.83 | **0.9116 ± 7.4322 × $10^{-5}$** |
| | 30 | 84.91 | 87.72 | 88.20 | 88.94 | **0.8897 ± 0.0016** |
| | 40 | 82.60 | 81.63 | 82.61 | 88.94 | **0.8938 ± 1.7099 × $10^{-4}$** |
| | 50 | 80.20 | 79.34 | 80.36 | 79.62 | **0.8463 ± 2.8516 × $10^{-4}$** |
| $\sigma = 20$ | 10 | 79.44 | 87.89 | 87.24 | 87.44 | **0.9003 ± 8.3381 × $10^{-5}$** |
| | 20 | 78.62 | 83.82 | 85.23 | 85.38 | **0.8859 ± 1.7674 × $10^{-4}$** |
| | 30 | 77.81 | 82.49 | 82.98 | 83.35 | **0.8674 ± 2.0273 × $10^{-4}$** |
| | 40 | 46.03 | 78.05 | 80.20 | 80.44 | **0.8508 ± 1.7974 × $10^{-4}$** |
| | 50 | 74.59 | 75.84 | 76.26 | 76.83 | **0.8345 ± 2.1202 × $10^{-4}$** |
| $\sigma = 30$ | 10 | 77.30 | 80.96 | 83.46 | 82.70 | **0.8644 ± 7.9881 × $10^{-5}$** |
| | 20 | 76.59 | 79.44 | 80.27 | 80.20 | **0.8558 ± 9.7496 × $10^{-5}$** |
| | 30 | 76.01 | 78.46 | 78.82 | 78.87 | **0.8422 ± 1.8048 × $10^{-4}$** |
| | 40 | 74.27 | 75.05 | 75.72 | 75.74 | **0.8277 ± 2.4142 × $10^{-4}$** |
| | 50 | 72.78 | 73.04 | 73.19 | 71.98 | **0.8129 ± 2.2593 × $10^{-4}$** |
| $\sigma = 50$ | 10 | 74.93 | 76.17 | 76.15 | 76.19 | **0.8259 ± 4.5774 × $10^{-5}$** |
| | 20 | 73.78 | 74.19 | 74.38 | 74.88 | **0.8128 ± 1.4083 × $10^{-4}$** |
| | 30 | 72.65 | 73.77 | 73.51 | 73.34 | **0.7997 ± 9.9043 × $10^{-5}$** |
| | 40 | 70.52 | 70.27 | 70.34 | 70.00 | **0.7866 ± 3.1593 × $10^{-4}$** |
| | 50 | 67.97 | 69.01 | 69.13 | 68.66 | **0.7734 ± 3.2558 × $10^{-4}$** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 11** PSNR (DB) comparison on BSD100 data set for unknown and known noise-level denoising for mixed Gaussian, salt and pepper impulse and random value impulse

| Gaussian noise | Impulse noise, % | Random value noise level, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 40 | 10 | 26.20 | 26.13 | 26.45 | 27.11 | **29.0286 ± 0.0044** | **29.0522 ± 0.0044** |
| $\sigma = 20$ | 10 | 30 | 23.91 | 24.58 | 24.55 | 24.70 | **27.0155 ± 0.0057** | **27.3811 ± 0.0042** |

Bold values emphasize the obtained results of the proposed filter in this paper.

**Table 12** FSIM (%) comparison on BSD100 data set for unknown and known noise-level denoising for mixed Gaussian, salt and pepper impulse and random value impulse

| Gaussian noise | Impulse noise, % | Random value noise level, % | WESNR [33] | $l_1$-NLR [35] | $l_0$-NLR [35] | LSM-NLR [35] | Proposed CNN (unknown) | Proposed CNN (known) |
|---|---|---|---|---|---|---|---|---|
| $\sigma = 10$ | 40 | 10 | 86.23 | 86.60 | 86.09 | 88.44 | **$0.8962 \pm 1.1804 \times 10^{-4}$** | **$0.9055 \pm 7.0711 \times 10^{-5}$** |
| $\sigma = 20$ | 10 | 30 | 80.65 | 79.24 | 80.27 | 79.72 | **$0.8592 \pm 1.4142 \times 10^{-4}$** | **$0.8551 \pm 1.5470 \times 10^{-4}$** |

Bold values emphasize the obtained results of the proposed filter in this paper.

This study introduces a deep feed-forward CNN filter for the removal of mixed Gaussian and impulse noise, assuming both known and unknown noise level denoising. Empirical evaluations demonstrate that the proposed method does overcome the presence of mixed Gaussian and impulse noise and is capable of addressing different intensity levels, which prior state-of-the-art methods had difficulties at addressing both issues simultaneously (mixed noise with different intensity levels).

The proposed CNN-based network is composed of 20 layers and directly estimates the clean noise-free image instead of the residuals due to the non-additive nature of the targeted noise. Batch normalisation along with random noise seeds are used to overcome the internal covariate shift problem, to allow for higher learning rates, and achieve greater training data variance.

Consequently, the results of the proposed filter show the best (i.e. highest) structural metrics in comparison to other well-known denoising filters, which are proven to be quite effective in their own merits when tasked with removing mixed Gaussian and impulse noise. These results also prove that the proposed filter preserves larger amounts of edge details, as reflected by the highest structural similarity measure, proving the closeness of the denoised images to their respective original noise-free images.

The CNN-based network model can be extended for use to the denoising of colour images. For this model to work with colour images, a modification is needed to transform the convolutional layers to handle 3D filters. However, you could get around this modification by applying the CNN to each RGB channel independently and then recombining them into the original colour image. This model is also amenable for the removal of other types of noise as long as their respective transfer functions can be formulated. Moreover, the results obtained through such a model can be used as an initial first step for other application domains such as edge tracking, object identification, pattern recognition, segmentation, and registration, among others.

## 5 Acknowledgments

The authors were grateful for the continued support from the National Science Foundation (NSF) under grants CNS-1920182, CNS-1532061, CNS-1551221, CNS-1338922, CNS-2018611, and the NIA/NIH 1P30AG066506-01 with the 1Florida ADRC. They remain grateful to the support of the Ware Foundation. Additional support was provided through the National Science Foundation Graduate Research Fellowship Program (NSF-GRFP) for Mr Harold Martin and the FIU-University Graduate School (UGS) through the dissertation year fellowship (DYF) provided to Dr Mehdi Mafi.

## 6 References

[1] Bovik, A.C.: '*Handbook of image and video processing*' (Academic press, Cambridge, MA, USA, 2000, 2nd edn.)

[2] Healey, G.E., Kondepudy, R.: 'Radiometric CCD camera calibration and noise estimation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1994, **16**, (3), pp. 267–276

[3] Liu, C., Szeliski, R., Kang, S.B*., et al.*: 'Automatic estimation and removal of noise from a single image', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008, **30**, (2), pp. 299–314

[4] Liu, W., Lin, W.: 'Additive white Gaussian noise level estimation in SVD domain for images', *IEEE Trans. Image Process.*, 2013, **22**, (3), pp. 872–883

[5] Mafi, M., Martin, H., Andrian, J.*, et al.*: 'A comprehensive survey on impulse and Gaussian denoising filters for digital images', *Signal Process.*, 2019, **157**, pp. 236–260

[6] Xu, Y., Zhou, X., Chen, S.*, et al.*: 'Deep learning for multiple object tracking: a survey', *IET Comput. Vis.*, 2019, **13**, (4), pp. 355–368

[7] Shamsolmoali, P., Zareapoor, M., Yang, J.: 'Convolutional neural network in network (CNNiN): hyperspectral image classification and dimensionality reduction', *IET Image Proc.*, 2019, **13**, (2), pp. 246–253

[8] Hamouda, M., Ettabaa, K.S., Bouhlel, M.S.: 'Hyperspectral imaging classification based on convolutional neural networks by adaptive sizes of windows and filters', *IET Image Proc.*, 2019, **13**, (2), pp. 392–398

[9] Zhang, F., Cai, N., Wu, J.*, et al.*: 'Image denoising method based on a deep convolution neural network', *IET Image Proc.*, 2018, **12**, (4), pp. 485–493

[10] Khaw, H.Y., Soon, F.C., Chuah, J.H.*, et al.*: 'High-density impulse noise detection and removal using deep convolutional neural network with particle swarm optimisation', *IET Image Proc.*, 2019, **13**, (2), pp. 365–374

[11] Chen, Y., Pock, T.: 'Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, **39**, (6), pp. 1256–1272

[12] Zhang, K., Zuo, W., Chen, Y.*, et al.*: 'Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising', *IEEE Trans. Image Process.*, 2017, **26**, (7), pp. 3142–3155

[13] Liu, N., Han, J., Liu, T.*, et al.*: 'Learning to predict eye fixations via multiresolution convolutional neural networks', *IEEE Trans. Neural Netw. Learn. Syst.*, 2018, **29**, (2), pp. 392–404

[14] Shi, W., Gong, Y., Tao, X.*, et al.*: 'Training DCNN by combining max-margin, max-correlation objectives, and correntropy loss for multilabel image classification', *IEEE Trans. Neural Netw. Learn. Syst.*, 2018, **29**, (7), pp. 2896–2908

[15] Shi, W., Gong, Y., Tao, X.*, et al.*: 'Fine-grained image classification using modified DCNNs trained by cascaded softmax and generalized large-margin losses', *IEEE Trans. Neural Netw. Learn. Syst.*, 2019, **30**, (3), pp. 683–694

[16] Lian, D., Hu, L., Luo, W.*, et al.*: 'Multiview multitask gaze estimation with deep convolutional neural networks', *IEEE Trans. Neural Netw. Learn. Syst.*, 2018, **30**, (10), pp. 3010–3023

[17] Mafi, M., Rajaei, H., Cabrerizo, M.*, et al.*: 'A robust edge detection approach in the presence of high impulse noise intensity through switching adaptive median and fixed weighted mean filtering', *IEEE Trans. Image Process.*, 2018, **27**, (11), pp. 5475–5490

[18] Mafi, M., Tabarestani, S., Cabrerizo, M.*, et al.*: 'Denoising of ultrasound images affected by combined speckle and Gaussian noise', *IET Image Proc.*, 2016, **12**, (12), pp. 2346–2351

[19] Bing, L., QuanSheng, L., JiaWei, X.*, et al.*: 'A new method for removing mixed noises', *Sci. Chin. Inf. Sci.*, 2011, **54**, (1), pp. 51–59

[20] Buades, A., Coll, B., Morel, J.-M.: 'A non-local algorithm for image denoising'. Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005, vol. 2, pp. 60–65

[21] Garnett, R., Huegerich, T., Chui, C.*, et al.*: 'A universal noise removal algorithm with an impulse detector', *IEEE Trans. Image Process.*, 2005, **14**, (11), pp. 1747–1754

[22] Su, T.J., Li, C.I.: 'An adaptive filtering method for mixed noise of images'. IEEE Int. Symp. on Computer, Consumer and Control, Taichung, Taiwan, 2012

[23] Huang, Y.M., Ng, M.K., Wen, Y.W.: 'Fast image restoration methods for impulse and Gaussian noises removal', *IEEE Signal Process. Lett.*, 2009, **16**, (6), pp. 457–460

[24] Rodríguez, P., Rojas, R., Wohlberg, B.: 'Mixed Gaussian-impulse noise image restoration via total variation'. IEEE Int. Conf. Acoustics, Speech, and Signal processing, Kyoto, Japan, 2012

[25] Arnal, J., Sanchez, M.G., Vidal, V.: 'Parallel filter for mixed Gaussian-impulse noise removal'. IEEE Int. Conf. Signal Processing: Algorithms, Architectures, Arrangements, and Applications, Poznan, Poland, 2013

[26] Morillas, S., Gregori, V., Hervás, A.: 'Fuzzy peer groups for reducing mixed Gaussian-impulse noise from color images', *IEEE Trans. Image Process.*, 2009, **18**, (7), pp. 1452–1466

[27] Jayasree, M., Narayanan, N.K.: 'A novel fuzzy filter for mixed impulse Gaussian noise from color images'. Proc. Int. Conf. Signal, Networks, Computing, and Systems, New Delhi, India, 2016, pp. 53–59

[28] Sun, Y., Junwei, H., Jun, L.: 'An information-fusion edge preserving method in image filtering'. 16th IEEE Conf. Wireless Communications Networking and Mobile Computing, Chengdu, People's Republic of China, 2010

[29] Guo, X., Guo, B.: 'A fuzzy filter for color images corrupted by mixed noise'. IEEE Int. Conf. Identification, Information and Knowledge in the Internet of Things, Beijing, People's Republic of China, 2014

[30] Chankhachon, S., Intajag, S.: 'Resourceful method to remove mixed Gaussian-impulse noise in color images'. 12th IEEE Int. Conf. Computer Science and Software Engineering, Songkhla, Thailand, 2015

[31] Xiao, Y., Zeng, T., Yu, J.*, et al.*: 'Restoration of images corrupted by mixed Gaussian-impulse noise via l1–l0 minimization', *Pattern Recognit.*, 2011, **44**, pp. 1708–1720

[32] Filipović, M., Jukić, A.: 'Restoration of images corrupted by mixed Gaussian-impulse noise by iterative sost-hard thresholding', *Pattern Recognit.*, 2011, **44**, (8), pp. 1708–1720

[33] Jiang, J., Zhang, L., Yang, J.: 'Mixed noise removal by weighted encoding with sparse nonlocal regularization', *IEEE Trans. Image Process.*, 2014, **23**, (6), pp. 2651–2662

[34] Jiang, J., Yang, J., Cui, Y.*, et al.*: 'Mixed noise removal by weighted low rank model', *Neurocomputing*, 2015, **151**, pp. 817–826

[35] Huang, T., Dong, W., Xie, X.*, et al.*: 'Mixed noise removal via Laplacian scale mixture modeling and nonlocal low-rank approximation', *IEEE Trans. Image Process.*, 2017, **26**, (7), pp. 3171–3186

[36] Eslahi, N., Mahdavinataj, H., Aghagolzadeh, A.: 'Mixed Gaussian-impulse noise removal from highly corrupted images via adaptive local and nonlocal statistical priors'. 9th IEEE Iranian Conf. Machine Vision and Image Processing, Tehran, Iran, 2015

[37] Delon, J., Desolneux, A., Guillemot, T.: 'PARIGI: a patch-based approach to remove impulse-Gaussian noise from images', *Image Process. On Line*, 2014, **5**, pp. 130–154

[38] Aher, R.P., Jodhanle, K.C.: 'Removal of mixed impulse noise and Gaussian noise using genetic programming'. Proc. IEEE Int. Conf. Signal Processing, Beijing, People's Republic of China, 2012

[39] Lamichhane, B.P.: 'Finite element techniques for removing the mixture of Gaussian and impulsive noise', *IEEE Trans. Signal Process.*, 2009, **57**, (7), pp. 2538–2547

[40] Mendiola-Santibañez, J.D., Terol-Villalobos, I.R.: 'Filtering of mixed Gaussian and impulsive noise using morphological contrast detectors', *IET Image Process.*, 2014, **8**, (3), pp. 131–141

[41] Shen, Y., Han, B., Braverman, E.: 'Removal of mixed Gaussian and impulse noise using directional tensor product complex tight framelets', *J. Math. Imag.*, 2016, **54**, (1), pp. 64–77

[42] Cai, J., Chan, R.H., Nikolova, M.: 'Two-phase methods for deblurring images corrupted by impulse plus Gaussian noise', *Inverse Probl. Imag.*, 2008, **2**, pp. 187–204

[43] Ji, H., Huang, S., Shen, Z.*, et al.*: 'Robust video restoration by joint sparse and low rank matrix approximation', *SIAM J. Imag. Sci.*, 2011, **4**, (4), pp. 1122–1142

[44] Dabov, K., Foi, A., Katkovnik, V.*, et al.*: 'Image denoising by sparse 3-d transform-domain collaborative filtering', *IEEE Trans. Image Process.*, 2007, **16**, (8), pp. 2080–2095

[45] Dong, W., Zhang, L., Shi, G.*, et al.*: 'Nonlocally centralized sparse representation for image restoration', *IEEE Trans. Image Process.*, 2013, **22**, (4), pp. 1620–1630

[46] Xiong, B., Yin, Z.: 'A universal denoising framework with a new impulse detector and nonlocal means', *IEEE Trans. Image Process*, 2012, **21**, (4), pp. 1663–1675

[47] Cai, J.F., Chan, R., Nikolova, M.: 'Fast two-phase image deblurring under impulse noise', *J. Math. Imag. Vis.*, 2010, **36**, pp. 46–53

[48] Mafi, M., Martin, H., Adjouadi, M.: 'High impulse noise intensity removal in MRI images'. IEEE Symp. on Signal Processing in Medicine and Biology Symp., Philadelphia, PA, USA, 2017

[49] Dong, Y., Chan, R.H., Xu, S.: 'A detection statistic for random-valued impulse noise', *IEEE Trans. Image Process.*, 2007, **16**, (4), pp. 1112–1120

[50] Healey, G.E., Kondeputy, R.: 'Radiometric CCD camera calibration and noise estimation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1994, **16**, (3), pp. 267–276

[51] Zhang, L., Zhang, L., Mou, X.*, et al.*: 'FSIM: A feature similarity index for image quality assessment', *IEEE Trans. Image Process.*, 2011, **20**, (8), pp. 2378–2386

[52] Xie, J., Xu, L., Chen, E.: 'Image denoising and inpainting with deep neural networks'. Proc. 25th Int. Conf. Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 2012, pp. 341–349

[53] Burger, H.C., Schuler, C.J., Harmeling, S.: 'Image denoising: can plain neural networks compete with BM3D?'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 2392–2399

[54] Jain, V., Seung, S.: 'Natural image denoising with convolutional networks'. Proc. Conf. Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 2009, vol. 21, pp. 769–776

[55] Ding, L., Zhang, H., Xiao, J.*, et al.*: 'An improved image mixed noise removal algorithm based on superresolution algorithm and CNN', *Neural Comput. Appl.*, 2019, **31**, (Suppl 1), pp. S325–S336

[56] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'Image net classification with deep convolutional neural networks'. Proc. Conf. Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 2012, pp. 1097–1105

[57] Ioffe, S., Szegedy, C.: 'Batch normalization: accelerating deep network training by reducing internal covariate shift'. Proc. Conf. Machine Learning (ICML), Lille, France, 2015, pp. 448–456

[58] Levin, A., Nadler, B.: 'Natural image denoising: optimality and inherent bounds'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, RI, USA, 2011, pp. 2833–2840

[59] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition'. Conf. Learning Representations, San Diego, CA, USA, 2015

[60] Zhao, H., Gallo, O., Frosio, I.*, et al.*: 'Loss functions for image restoration with neural networks', *IEEE Trans. Comput. Image.*, 2017, **3**, (1), pp. 47–57

[61] Duchi, J., Hazan, E., Singer, Y.: 'Adaptive subgradient methods for online learning and stochastic optimization', *J. Mach. Learn. Res.*, 2011, **12**, pp. 2121–2159

[62] Kingma, D., Ba, J.: 'Adam: A method for stochastic optimization'. Conf. Learning Representations, San Diego, CA, USA, 2015

[63] He, K., Zhang, X., Ren, S.*, et al.*: 'Delving deep into rectifiers: surpassing human-level performance on imagenet classification'. Proc. IEEE Conf. Computer Vision, Las Condes, Chile, 2015, pp. 1026–1034