Applied Soft Computing Journal 74 (2019) 747–759

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Hierarchical extreme learning machine based image denoising network for visual Internet of Things

Yifan Yang [a], Hong Zhang [a,*], Ding Yuan [a], Daniel Sun [b], Guoqiang Li [c,*], Rajiv Ranjan [d], Mingui Sun [e]

[a] Image Processing Center, Beihang University, China
[b] Data61, CSIRO, Australia
[c] School of Software, Shanghai Jiao Tong University, China
[d] Newcastle University, UK
[e] Department of Neurosurgery, University of Pittsburgh, Pittsburgh, PA 15213, USA

HIGHLIGHTS

- We address the heavy noise removing problem faced in visual Internet of Things by using the hierarchical extreme learning machine. The proposed framework contains a sparse auto-encoder and a supervised regression and a non-local aggregation.
- We provide an effective patch-to-patch image denoising networks which are robust for dealing with various noise levels in both clipped and unclipped noisy model. The key advantage of this denoising network is fast training.
- Experimental studies on images including both hand-written digits and natural scenes have shown that our method achieves excellent performance both in quality and efficiency. The nice performance can improve the compression ratio for data interactions in the visual Internet of Things.

ARTICLE INFO

Article history:
Received 16 November 2017
Received in revised form 9 August 2018
Accepted 24 August 2018
Available online 8 September 2018

Keywords:
Image denoising
Visual Internet of Things
Extreme learning machine
Supervised regression
Non-local
Heavy noise

ABSTRACT

In the visual Internet of Things (VIoT), imaging sensors must achieve a balance between limited bandwidth and useful information when images contain heavy noise. In this paper, we address the problem of removing heavy noise and propose a novel hierarchical extreme learning machine-based image denoising network, which comprises a sparse auto-encoder and a supervised regression. Due to the fast training of a hierarchical extreme learning machine, an effective image denoising system that is robust for various noise levels can be trained more efficiently than other denoising methods, using a deep neural network. Our proposed framework also contains a non-local aggregation procedure that aims to fine-tune noise reduction according to structural similarity. Compared to the compression ratio in noisy images, the compression ratio of denoised images can be dramatically improved. Therefore, the method can achieve a low communication cost for data interactions in the VIoT. Experimental studies on images, including both hand-written digits and natural scenes, have demonstrated that the proposed technique achieves excellent performance in suppressing heavy noise. Further, it greatly reduces the training time, and outperforms other state-of-the-art approaches in terms of denoising indexes for the peak signal-to-noise ratio (PSNR) or the structural similarity index (SSIM).

1. Introduction

In today's digital world, smart cameras are ubiquitously adopted in various areas such as security surveillance, automotives, and industry. Cameras have been widely connected to the Internet and managed by the infrastructure of the visual Internet of Things (VIoT). In recent years, the quality of imaging sensors has improved significantly, such that high-resolution cameras have become mainstream. The constant pursuit of more pixels integrated in a small chip results in low signal-to-noise ratio (SNR), which is related to the number of photons that are incident on a chip per unit area. A higher gain in signal amplification is necessary for some applications, such as low-light surveillance and dehazing enhancement [1–3]. The level of noise increases in proportion to the amplification factor. On one hand, under extreme conditions like low illumination or heavy noise, which substantially deteriorates image quality, object perception and recognition becomes difficult for both artificial observation and computer vision. Thus, image

* Corresponding authors.
E-mail addresses: stephenyoung@buaa.edu.cn (Y. Yang), dmrzhang@buaa.edu.cn (H. Zhang), dyuan@buaa.edu.cn (D. Yuan), daniel.sun@data61.csiro.au (D. Sun), li.g@sjtu.edu.cn (G. Li), raj.ranjan@ncl.ac.uk (R. Ranjan), drsun@pitt.edu (M. Sun).

https://doi.org/10.1016/j.asoc.2018.08.046
1568-4946/

noise limits applications over the VIoT [4]. On the other hand, the transmission bandwidth is sensitive for IoT applications [5]. However, a noisy image incurs a compression ratio bottleneck for network communication between sensor nodes and servers. Noise seriously affects compression for popular image compression methods, because it brings much contaminated and useless information into images [6]. Therefore, image denoising is an important factor that influences the quality of many imaging sensor nodes and the performance of VIoT systems.

Diverse denoising methods have been proposed to reduce noise in low-quality images in the past decade. Traditional denoising methods include non-local mean (NLM) [7], block matching with 3D filtering (BM3D) [8–11] and global denoising [12], low rank models including sparse representation denoising (k-SVD) [13,14], and robust principal component analysis (R-PCA) [15]. For example, the BM3D represents a milestone among the traditional denoising methods, which is based on the assumptions that noise is additive, white, and Gaussian (AWG) [16], and noisy natural images contain the appearance of similar patches. Though a method of this kind is well engineered, the generality is weakened due to these assumptions. Until recently, deep neural networks achieved desirable performance in various computer vision and image processing tasks, including image denoising algorithms, such as the plain multi-layer perceptron denoising (MLPD) [17], deep class aware denoising (DCAD) [18], deep convolutional neural network-based denoising [19], and deep Gaussian conditional random field network (DGCRF) denoising [20]. Most parametric denoising frameworks can be abstracted as multi-layer networks with certain connections. By learning a mapping from contaminated images to noise-free images on large-scale training dataset, deep neural networks have become the current state-of-the-art image denoising paradigm. However, training deep neural networks requires a large amount of data, and thus is often time-consuming. The extreme learning machine (ELM) [21] was proposed with the intrinsic feature that it can be trained quickly. Many improvements based on the typical ELM have been proposed to satisfy special applications, such as hierarchical ELM (HELM) [22], and online sequential ELM (OS-ELM) [23].

To overcome the deficiency in heavy and/or varying noise removal and improve the efficiency of model training, we propose a new denoising framework (Fig. 1) that consists of three modules: (1) patch decomposition and pre-processing; (2) patch-to-patch denoising based on hierarchical extreme learning machine (HELM); and (3) non-local aggregation. In the first module, a noisy image is partitioned into overlapping patches with a fix size and stride. Meanwhile, other necessary pre-processing is implemented. The second module HELM contains an auto-encoder and supervised regression, and is applied to image denoising. In the last module, fine denoising is performed by aggregating non-local patches according to their structural similarity. Our approach can be used to address heavy noise and to achieve competitive performance when compared to other methods. We designed a multi-channel embedded image-processing device that can bear our proposed denoising algorithm for VIoT-based surveillance systems. The architecture of the VIoT-based surveillance system is illustrated in Fig. 2. Raw images captured from distributed cameras can be transmitted to a multi-channel embedded image-processing device via standard interfaces. The on-board denoising algorithms are implemented to remove noise in the raw image according to a remote user's commands. The resulting noise-free images are then sent to a cloud server, where they can be retrieved by end users.

The rest of this paper is organized as follows. In Section 2, we briefly review previous research related to image denoising. Section 3 describes in detail the framework of a HELM-based denoising network with non-local aggregation. In Section 4, we introduce the embedded VIoT system for video surveillance. In Section 5, database and experimental setting are introduced. In Section 6, experimental results are presented, and comparisons with other methods are discussed. Finally, we draw conclusions and discuss future research directions in Sections 7 and 8.

2. Related work

In this section, we classify existing image denoising methods into traditional and neural network methods and review them separately. Then we introduce related research on embedded applications of image processing and machine learning for the IoT.

2.1. Traditional denoising methods

The first class of denoising methods assumes that an input image contains substantially repeated or similar regions that can be grouped to facilitate noise removal, including non-local mean (NLM) [7], block matching with 3D filtering (BM3D) [8–11], and global denoising [12]. The second class of denoising methods is based on the theory of low rank models, including sparse representation denoising (k-SVD) [13,14], robust principal component analysis (R-PCA) [15], low-rank matrix completion (LRMC) [24], and low rank representation (LRR) [25]. Another kind of method utilizes certain properties of the noise itself during algorithm design, such as its structural or statistical priors (e.g. gradients, structural similarity, and correlation) or its transformed noise spectrum (e.g. Fourier transform, discrete cosine transform [10], wavelets transform [26–28], and numerical optimization (e.g. k-SVD, R-PCA, and LRR). When the noise level is high, the performance of these algorithms usually decays significantly due to deterioration of similarities or correlations among various regions in the image. For instance, BM3D measures the patch similarity using a firm threshold. As the value of the SNR approaches the negative domain (measured in dB), it becomes difficult to make a decision regarding whether the neighborhood patches are similar. A high noise level has another effect in that numerous pixel values are clipped within the range of [0, 255], resulting in substantial distortion.

2.2. Deep denoising networks

In recent years, artificial neural networks (NNs) have been widely used to solve numerous image processing and computer vision problems [29–33], including image denoising. Burger et al. [17] proposed a plain multi-layer perceptron denoising (MLPD), which delivers competitive performance compared to BM3D. Li et al. [19] proposed a CNN-based denoising network that uses the deep neural network in TensorFlow, and Remez et al. [18] proposed deep class aware denoising (DCAD) without the AWG assumption, which is widely assumed in other methods. Vemulapalli et al. [20] proposed a deep Gaussian conditional random field (DGCRF) network for discriminative denoising. Despite these benefits, the adoption of deep learning within VIoT and embedded end-devices faces significant barriers due to the computational resource requirements and energy consumption of these algorithms [34].

In contrast to back-propagation based neural networks (BPNs) such as CNN, an extreme learning machine (ELM) was proposed by Huang et al. to train generalized single hidden layer feed-forward neural networks (SLFNs) [21]. The ELM trains an SLFN in two main stages: (1) random feature mapping and (2) solving linear parameters. During the first stage, ELM randomly initializes the hidden layer to map input data into a feature space by some nonlinear mapping function [35]. Specifically, SLFNs with randomly initialized hidden neurons and learnable output weights are trained to minimize regularized least square errors and can be computed efficiently. ELM has been successfully used to solve

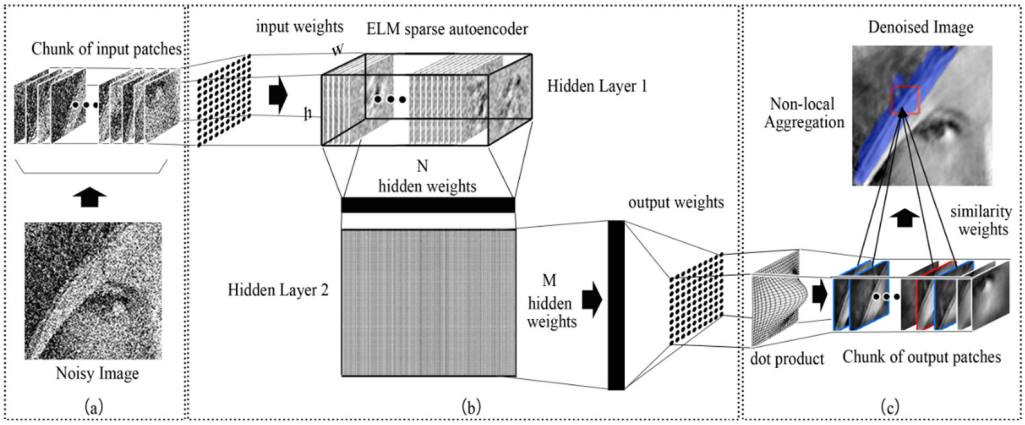


Fig. 1. Framework of the HELM-based denoising network with non-local aggregation. (a) Patch decomposition and preprocessing; (b) HELM-based patch-to-patch denoising network; (c) non-local aggregation.

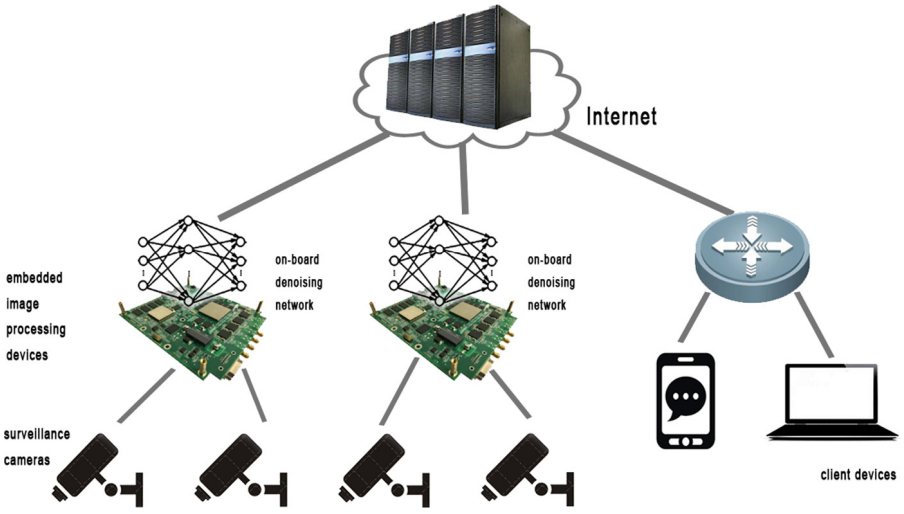


Fig. 2. VIoT architecture for a surveillance system using our denoising networks.

many practical problems [36], such as shape segmentation [37], facial recognition [38], wearable-based activity recognition [39], and histopathological image classification [40]. These applications motivate us to train an end-to-end denoising network based on ELM.

However, due to its shallow architecture, feature learning using ELM may not be effective for natural signals such as images [22]. To address this issue, a hierarchical ELM is proposed with a multi-layer perceptron framework [22], including an unsupervised auto-encoder module that can extract sparse and compact features. On the other hand, an incremental training strategy is demanded if a large-scale training dataset is to be used. Liang et al. [23] proposed an online sequential ELM (OS-ELM) that can learn sequentially from individual data points or from chunks of data.

2.3. Embedded applications for IoT

Many devices and applications of image processing and machine learning within the IoT ecosystem have emerged in recent years [41,42]. Drolia et al. [43] proposed a system called Precog, which accelerates image recognition by enabling caching and prefetching on the edge devices. Li et al. [3] proposed a road vehicle monitoring system based on intelligent VIoT, and they transplanted algorithms, e.g. image dehazing, license plate and vehicle type recognition, into an embedded processor for testing in practical environments. Ventura et al. [44] designed an IoT device using an Arduino MEGA board to reduce energy consumption, and they implemented a machine learning architecture on this platform. Dhote et al. [45] presented a multifunction image processing system that is accessible over the Internet using a system-on-chip (SoC) architecture with a field-programmable gate array (FPGA), and they used low-cost chips such as Raspberry-Pi and Spartan 6 FPGA in this system. Li et al. [46] proposed a data analysis approach for big data, where data was collected from IoT devices. They applied the deep convolutional computation model to learn hierarchical features of big data using a tensor representation model. Lane et al. [34] presented a preliminary measurement study of common deep learning models on representative mobile and embedded platforms. They compared targeted hard platforms, such as the Qualcomm Snapdragon 800, Intel Edison, and Nvidia Tegra K1, based on mainstream convolutional models or deep neural networks.

As aforementioned, the VIoT faced the challenge that heavy noise image captured in extreme environments and these image should be transmitted within limited bandwidth and computational resource. Existing traditional and deep NNs-based denoising methods are hard to make balance between denoising performance and computational burden. Our research provide an effective method that fills the gaps between the denoising quality and efficiency in width-sensitive VIoT edge devices. Our denoising method differs from previous methods in the following aspects: (1) we focus on cases with heavy noise that cannot be handled

effectively by traditional non-learning methods; (2) we improve the design of HELM denoising with desirable training time; (3) an online sequential HELM is proposed to enhance the learning capability of a patch-based denoising network; (4) a two-step denoising framework, including a neural network filter step and a structural aggregation step; and (5) an VIoT-based embedded image denoising device is designed for use as a video surveillance system.

3. HELM-based image denoising

In this section, we briefly review the theory of ELM and then describe details of the proposed framework, including data preprocessing, HELM-based auto-encoder learning, and the non-local aggregation procedure.

3.1. Recap of extreme learning machine

The output function of SLFNs with L hidden nodes can be computed as follows:

$$f_L(\boldsymbol{x}_k) = \sum_{i=1}^{L} G_i(x_k, \boldsymbol{a}_i, b_i) \cdot \boldsymbol{\beta}_i, \quad \boldsymbol{a}_i \in R^d, b_i, \boldsymbol{\beta}_i \in R, \quad k = 1, \ldots, N \tag{1}$$

where $\boldsymbol{x}_k$ is the input data; $\boldsymbol{a}_i$ and b_i denote the input weight vectors and bias of the ith hidden node, respectively; $\boldsymbol{\beta}_i$ is the output weight; and $G_i(\cdot)$ is the activation function (chosen from Gaussian, hyperbolic tangent, or hard limit functions) [35]. According to the universal approximation capability of ELM [47], SLFNs with additive or radial basis hidden nodes can be used to approximate any continuous target functions with randomly initialized parameters.

For a set of N training samples, Eq. (1) can be represented in the following compact form:

$$\boldsymbol{T} = \boldsymbol{H} \cdot \boldsymbol{\beta} \tag{2}$$

where $\boldsymbol{H}$ is the output matrix of the hidden layer

$$\boldsymbol{H} = \begin{bmatrix} G_1(\boldsymbol{x}_1, \boldsymbol{a}_1, b_1) & \cdots & G_L(\boldsymbol{x}_1, \boldsymbol{a}_L, b_L) \\ \vdots & \ddots & \vdots \\ G_1(\boldsymbol{x}_N, \boldsymbol{a}_1, b_1) & \cdots & G_L(\boldsymbol{x}_N, \boldsymbol{a}_i, b_i) \end{bmatrix} \tag{3}$$

The aim of ELM training is to minimize the error and norm of the output weights:

$$Minimize: \|\boldsymbol{\beta}\|_p^{\sigma_1} + C\|\boldsymbol{H} \cdot \boldsymbol{\beta} - \boldsymbol{T}\|_q^{\sigma_2} \tag{4}$$

where $\sigma_1 > 0$, $\sigma_2 > 0$, $p, q = 0, 1/2, 1, 2, \ldots, \infty$, and C is a parameter which controls the generalization ability of the model. Please refer to [21,48] for approaches that can solve for the output weights $\boldsymbol{\beta}$.

The ELM learning is implemented in the following three steps:
(1) Assign randomly hidden node parameters $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$;
(2) Calculate the hidden layer output matrix $\boldsymbol{H}$;
(3) Calculate the output weights $\boldsymbol{\beta}$.

The drawback of the original ELM is that its shallow architecture cannot effectively handle image contents, even with a large number of hidden nodes. Tang et al. [22] proposed HELM as a hierarchical learning framework with both unsupervised feature extraction and supervised feature regression/classification in a multi-layer manner, which is described as follows:

Let $\boldsymbol{H}_i$ be the output of the current ith hidden layer, and let $\boldsymbol{H}_{i-1}$ be the output from the previous hidden layer. The relationship between consecutive hidden layers can be represented by

$$\boldsymbol{H}_i = G(\boldsymbol{H}_{i-1} \cdot \boldsymbol{\beta}_i) \tag{5}$$

where $\boldsymbol{\beta}_i$ denotes the weights of the current hidden layer, and $G(\cdot)$ is the activation function. The layers are cascaded as a feature-learning network, and the extracted features become more powerful as the number of layers increases.

3.2. Data preparation for training and applying

Considering an 8-bit quantized image, an image with randomly generated noise according to an additive model could have pixel values below 0 or above 255, but the value of a practical noisy image may be clipped between 0 and 255. For analytical convenience, we convert the numerous pixel values from the range of [0, 255] to [0, 1] by normalization. The normalization model is defined as that pixel values of the input image are divided by 255, even if the original pixel value is outside the interval [0, 255]. Later, we decompose the noisy image into small patches, and they are denoised separately.

For training the network, white Gaussian noise is added to patches and then 0.5 is subtracted to generate the input data. Chunks of the corresponding clean patches will be used as a reference for training, thus the network learns how to map noisy patches to clean patches as expected.

To make the trained model more efficient, we apply the following procedures:

(1) The stride of cropped patches is set as small as possible in order to restore details in the output image.

(2) Training patches come from various type of images to train a generalized model.

For applying the network with a well-trained model, we crop an image into equal-sized patches as training data with a certain stride. The stride should be smaller than the patch size to ensure overlap between patches. Then, 0.5 was subtracted from each patch.

3.3. Improved HELM with online sequential training

Our target is to build a multi-output HELM regression network as an adaptive patch-to-patch filter that learns to reduce noise from the contaminated patch. As shown in Fig. 3, the proposed framework consists of an unsupervised auto-encoder and supervised regression. The number of input patches and the size of the training dataset are K and w^*h, respectively, where w^*h is also the number of input nodes. The first hidden layer of the network is regarded as an unsupervised feature extractor, which consists of input weights, the sparse auto-encoder, and hidden weights. The sparse auto-encoder generates a weight matrix as a group of adaptive convolutional kernels. The second hidden layer performs supervised regression and is implemented in the original ELM. Let N and M be the number of nodes in the first and second hidden layers, respectively. Randomly initialized mapping connects N nodes and M nodes between the two hidden layers.

The purpose of the auto-encoder is to extract sparse features from input data via certain hidden weights. The auto-encoder can learn the data structure adaptively and represent data efficiently [49]. However, directly training the auto-encoder with heavy noise patches is difficult in our experiment. Hence, a two-stage training strategy is employed. First, we train the auto-encoder using clean patches as both inputs and outputs to yield an initial auto-encoder. Second, the initial auto-encoder is fine-tuned with noisy training data to output effective features for the subsequent denoising module.

The basic HELM framework requires that the entire training dataset feed into the network in a one-shot manner. However, if there are millions of patches to be processed, learning is intractable due to memory limits. Motivated by OS-ELM [23], we separate the large training dataset into chunks and import them sequentially for training. Each chunk of data includes a batch of image patches. After a particular chunk is used for training, it will be discarded.

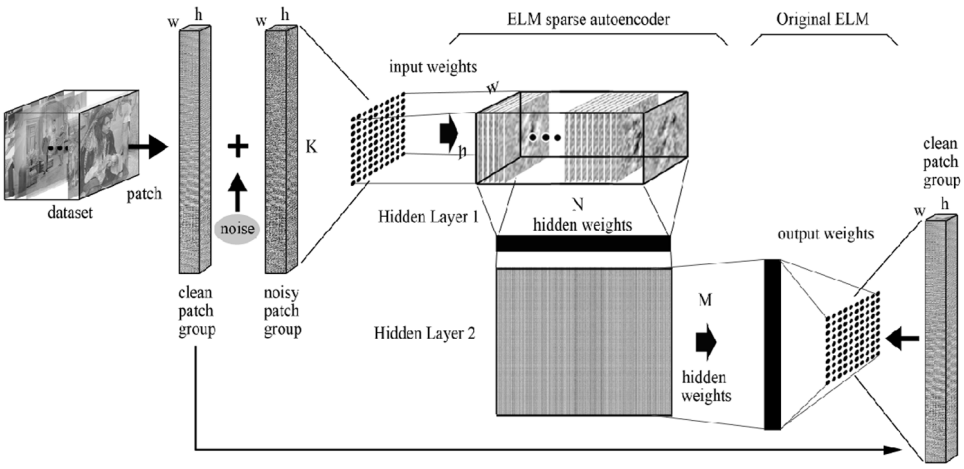


Fig. 3. Proposed HELM-based patch-to-patch denoising network.

3.4. Applying patch-based HELM for denoising

With a well-trained model for the improved HELM network, the decomposed patches are sequentially input to the neural network, either one-by-one or chunk-by-chunk. The denoised image patches are obtained from the network output. The input–output relationship can be expressed mathematically as follows:

$$\boldsymbol{H}_1 = tanh((\boldsymbol{X} - 0.5\boldsymbol{I}) \cdot \boldsymbol{\beta}_1) \tag{6}$$

$$\boldsymbol{H}_2 = tanh(\boldsymbol{H}_1 \cdot \boldsymbol{\beta}_2) \tag{7}$$

$$\boldsymbol{Y} = \boldsymbol{H}_2 \cdot \boldsymbol{\beta}_3 \tag{8}$$

where X and Y are the input and output data of 1-D extending image patch vectors. $0.5\boldsymbol{I}$ is an input bias. $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are matrices containing hidden weights, and $\boldsymbol{\beta}_3$ contains the output weights. We choose the hyperbolic tangent transfer function as the activation function *tanh* [50]. The most intensive operations in the above network are matrix multiplications and look-up table mapping. Due to the computational complexity, it is favorable to implement this network using MapReduce [51], graphics processing units (GPU) [52], or embedded field-programmable gate array (FPGA) devices [53], although common central processing units (CPUs) may also be used for small image patches.

3.5. Non-local aggregation

During inference, a test image is first decomposed into overlapping patches as the input to the trained neural network in order to obtain denoising patches. Here, we calculate the dot-product between each denoised patch with a 2-D Kaiser window and then aggregate the results to form a finer denoised image. In previous works, [17] proposed a method to reconstruct denoised patches at their corresponding locations, and then average them in the overlapping regions. Alternatively, the structural similarities between the current patch and its neighboring patches can be measured, and similar patches can be selected based on the non-local theory reported previously [7,16] to yield a fine result. A fast Lanczos singular value decomposition (LANSVD) [54] is used to decompose the current and neighboring patches to extract correlated features that easily describe structural similarity. The k largest singular values and the corresponding singular vectors of a unitary matrix are used. In our application, k is empirically assigned to 20. This yields

$$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \boldsymbol{P}_{\boldsymbol{n},\ \boldsymbol{w}*\boldsymbol{h}} \tag{9}$$

where $\boldsymbol{P}$ is a matrix representing a batch of patches (each 2D patch has been converted to a 1D vector here), n is the number of neighboring patches, and w and h are the weight and height of the patch, respectively. $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V}$ are decomposed matrices of $\boldsymbol{P}$ obtained via singular value decomposition.

$$\boldsymbol{A} = \boldsymbol{U}_{\boldsymbol{n},\boldsymbol{k}}\boldsymbol{\Sigma}_{\boldsymbol{k}*\boldsymbol{k}} \tag{10}$$

where A is the product of the k largest singular values $\boldsymbol{\Sigma}_{\boldsymbol{k}*\boldsymbol{k}}$ and the corresponding singular vectors $\boldsymbol{U}_{\boldsymbol{n},\boldsymbol{k}}$.

Define $S_{i,j}$ to be the structural similarity between the ith patch and the jth patch.

$$S_{i,j} = exp(-\gamma \left\| \boldsymbol{A}_{\boldsymbol{i}} - \boldsymbol{A}_{\boldsymbol{j}} \right\|_1) \tag{11}$$

According to the similarity determined with Eq. (11), the top m most similar patches are refined with a threshold τ and are averaged to obtain a mean patch. Then we place the mean back into those locations with corresponding similar patches. The purpose of this aggregation is to obtain cleaner and smoother results in a flat area while preserving clearer edges in the gradient area. In comparison with a simple weighted average used in [17], the non-local aggregation step can improve the peak signal-to-noise ratio (*PSNR*) by up to 0.3 dB on average.

4. VIoT-based embedded image processing system

To build a VIoT platform that can capture and analyze streaming video or images in-situ and connect the end device to the Internet, we designed a multi-channel video surveillance system using embedded image processing technology. This system can provide on-line processing and intelligent analysis, such as image denoising, image enhancement, and object detection and tracking, and can support 1000M Ethernet communication. The on-board system contains three serial digital interface (SDI) channels for video input, a field programmable gate array (FPGA) for pre-processing, a digital signal processor (DSPs) for data processing and analysis, peripheral interfaces, and power managers. As shown in Fig. 4(a), this hardware platform includes a high-performance Kintex XC7K325T FPGA and multi-core TMS320C6678 DSPs. Some necessary memory devices, such as static random-access memory (SRAM), double-data-rate synchronous dynamic random access memory generation 3 (DDR3 SDRAM), serial peripheral interface (SPI), and NOR flash memory, are included in this embedded system. A simplified chart of this system is shown in Fig. 4(b).

The SDI video was decoded from multi-channel surveillance cameras in parallel using the FPGA, and the raw image data was transmitted to the DSPs via serial rapid I/O (SRIO), which forms

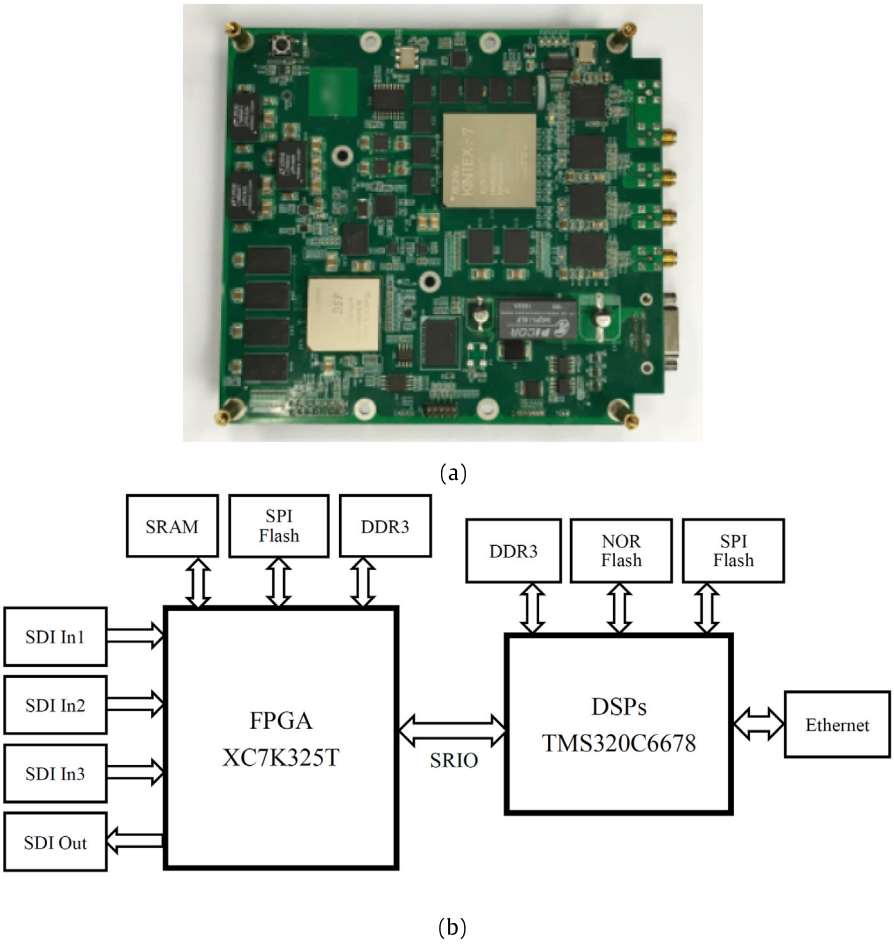


Fig. 4. (a) The embedded VIoT hardware platform and (b) its chart for multi-channel video surveillance.

the bridge between the FPGA and the DSPs. When the transplanted algorithm of HELM-based denoising and well-trained weights are loaded into the DSPs, an image of a video frame can be processed according to the received commands. Due to the matching algorithmic structure implemented on the DSPs or a general CPU, we can simulate and evaluate denoising performance on a CPU before transplanting to our device.

5. Database and experimental setting

For the patch-to-patch denoising network, the first auto-encoder hidden layer contained 900 nodes, and the output hidden layer contained 5000 hidden nodes.

We evaluated our methods by investigating two different tasks: denoising of hand-written digits and natural images, using public datasets. The first experiment implemented our patch-based denoising network using a dataset of hand-written digits, and the second experiment trained and applied our denoising framework to the natural image datasets.

(1) MNIST database

The MNIST handwritten digits dataset [55] contains 60,000 samples for training and 10,000 samples for testing. Each sample is a normalized image of a handwritten digit, with a fixed size of 28*28 pixels. We tested diverse additive Gaussian noise levels with standard deviation [44] ranging from 50 and 300 in increments of 10.

(2) Natural image training database

To collect suitable training data for image denoising, we downloaded high-quality photographs from the National Geographic website [56] to build our training dataset. There are 263 images in the dataset, which included landscapes, portraits, animals, architecture, and other images. We converted all color images to grayscale images and then extracted patches from random locations. Different sized patches were collected to train the model with different noise levels. Increasing noise levels require more neighbors to participate in the computation, therefore a larger patch size is carried with increasing noise. Assume that the patch size is $w*h$, where w is equal to h. For computational convenience, the patch width w and height h are assigned to be even integers. Thus, the relationship between patch size and noise level is shown in Fig. 5. The number of cropped patches for the auto-encoder was set to 200,000, and the number for supervised training data was set to 100,000. We found that these numbers were sufficiently large for the network to distinguish between image content and noise, even when the noise level was high.

(3) Natural image testing database

We utilized two datasets of natural images for experimental comparison. The first test dataset contained 11 standard images, including Lena, peppers, Barbara, camera man, hill, couples, house, fingerprint, montage, man, and boat, which were commonly used as benchmarks for denoising performance comparisons in previous studies [7,8,17]. The second test dataset was selected from the PASCAL VOC 2012 image database [57] containing the first 100 images named with the prefix '2012'.

When noise is added to an image, the pixel values could lie outside of the interval [0, 255]. To accommodate this, we allowed users to determine whether pixel value overflows should be retained or clipped to the range of [0, 255] before applying normalization,

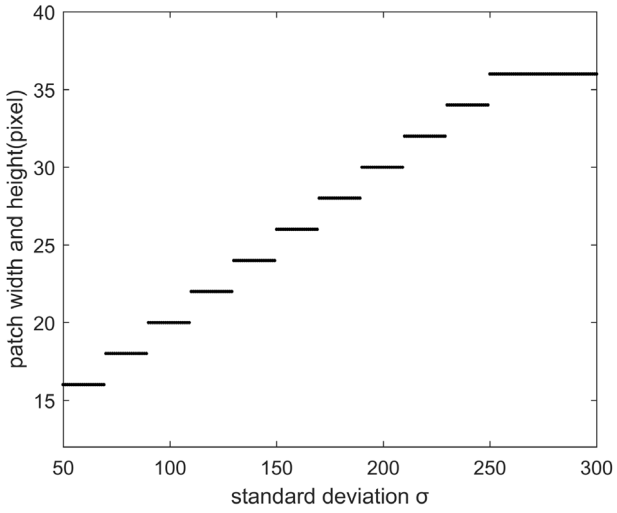


Fig. 5. Relationship between patch width and height (on the vertical axis), and noise level (on the horizontal axis).

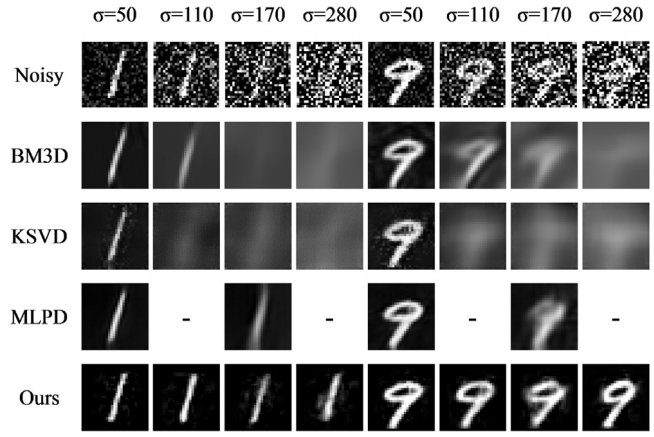


Fig. 6. Results from denoising of the MNIST test digit images using BM3D, KSVD, MLPD, and our method. In this example, we show the result of each method under different noise levels with standard deviation $\sigma = \{50, 100, 170, 280\}$ for the unclipped model (Eq. (12)). The results for the clipped model (Eq. (13)) were visually similar.

which was indicated in Section 3.2. We trained the network under both choices for different noise levels. The unclipped model is given by

$$y = x + n \tag{12}$$

where n, x, y are additive white Gaussian (AWG) noise, the noise-free image patch, and noise-contaminated image patch, respectively.

The clipped model is given by

$$y = \begin{cases} 0, & x+n < 0 \\ x+n, & 0 \le x+n \le 255 \\ 255, & x+n > 255 \end{cases} \tag{13}$$

In our experiments, we compared our approach with classical methods, such as block matching with 3D filtering [8] (BM3D, a non-local approach), image denoising via sparse and redundant representations over learned dictionaries [13] (KSVD, a sparse dictionary learning approach), and plain multi-layer perceptron denoising [17] (MLPD, a neural network based approach).

6. Experimental results

6.1. Handwritten digits denoising on MNIST dataset

Our patch-based denoising network was trained in approximately 2.5 min on an i7-5500U 2.7 GHz CPU. We subsequently tested this network with the images of hand written digits in the MNIST dataset. Our results, along with the results from the three conventional methods, are shown in Fig. 7. It can be seen that our method outperformed all three methods in all cases, except for two isolated points when $\sigma = 50$ and 70 in the unclipped model. The MLPD approach exhibited the best performance when unclipped images were used. Some examples of testing results are shown in Fig. 6, which demonstrates the high performance of our method compared to the other three methods.

To further apply the denoising network to real handwritten digits, we purposely captured an image with heavy noise using a smartphone camera (iPhone 7, ISO 640, shutter time 1/3200 s, file mode raw) in an extremely dark environment. A clear image is shown in Fig. 8(a) with 25 digits written by the first author, and the captured image is shown in Fig. 8(b) with a few recognizable details. We enhanced the image via linear grayscale stretching in mapping-form, as shown in Fig. 8(d). After applying this enhancement (Fig. 8(c)), the digits were faintly presented but remained unidentifiable due to corruption by heavy grain noise that is characteristic in images captured under extremely low illumination. After basic image processing steps consisting of grayscale-inversion, image cropping, and resizing, the cropped image patches (Fig. 9(a)) were fed into our patch-denoising network. The model was trained with a noise level of $\sigma = 300$. The results in Fig. 9(b) show that the proposed HELM-based denoising network exhibits high performance in terms of noise removal, and the model preserves the hand-written digits from an extremely-low-quality image. The results suggest a wide range of applications, such car license plate identification in extremely dark or hazy environments, and identifying hand-written documents while solving forensic cases involving low-quality, illegible document images.

6.2. General denoising on natural image dataset

In this experiment, we trained our denoising network on the aforementioned natural image dataset. We trained our denoising network with the same architecture and parameters, but different models (clipped and unclipped maximum intensity values) and noise levels ([51] between 50 and 300) were used. Each independent network required 5∼10 min for training on the same CPU mentioned in Section 6.1. During our tests, we implemented the image denoising in two ways. In the first method, noise was removed only with a patch-based network, and overlapping patches were combined via a simple weighted average as mentioned in [17]. In the second method, we utilized a complete denoising framework using the patch-based network and non-local aggregation, as described in Section 3.5.

Fig. 10 shows the results of our two patch-based network methods in comparison with the three conventional methods, in terms of the PSNR and SSIM. When the two versions of our method are compared, the non-local aggregation method outperforms the patch-based network for all noise levels based on a simple weighted average. The non-local aggregation improved PSNR by 0.2∼0.3 dB. Compared with BM3D and KSVD, our method is clearly superior when [54] is larger than 90 for the model without clipping. For the model with clipping, our method displays superior performance when [55] is larger than 80. When our method is compared to MPLD, the performance is nearly equal at high noise levels [56] = 170. However, our method slightly underperformed for [57] = {50, 75}. However, we achieved this performance using only several minutes of training time with a CPU. Fig. 11 shows the

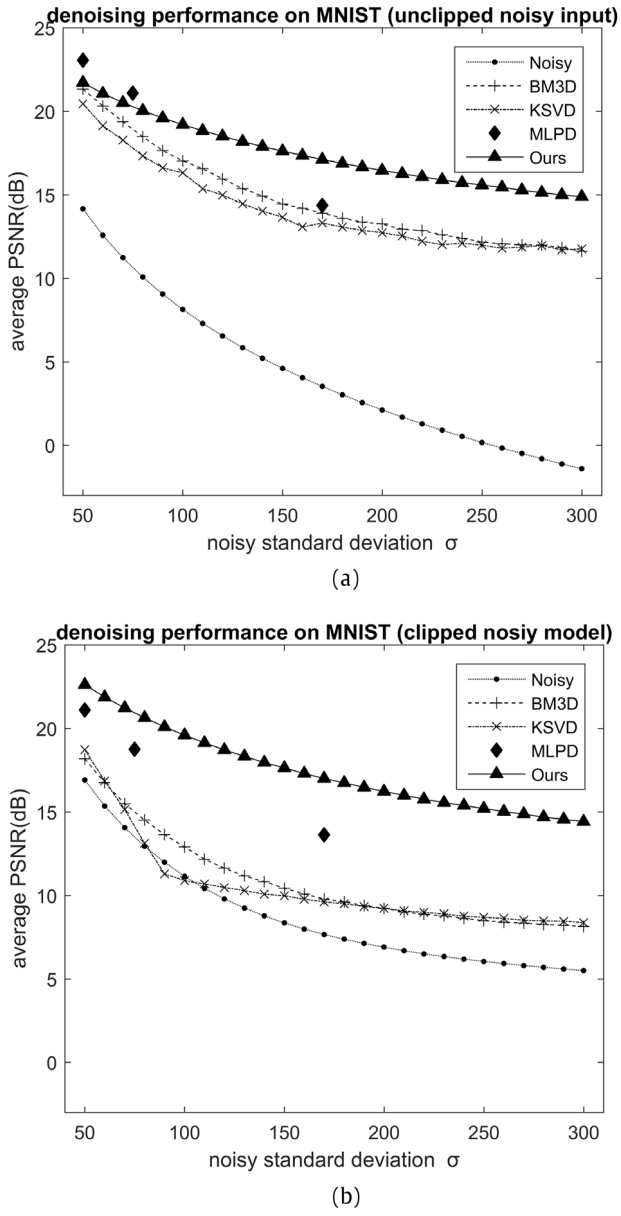


Fig. 7. Denoising performance on the MNIST dataset. (a) Unclipped model (Eq. (12)) and (b) clipped model (Eq. (13)).

denoised results of the image *Lena* with a noise of [59] = 170. It can be seen that the denoised images obtained via the MLPD and our method are superior to BM3D and KSVD, based on a subjective observation. There are fewer unreasonable textures in the flat area in our result compared to the MLPD result. However, the edge of our result is less sharp than the MLPD result.

To analyze the practical utility of our method in more diverse images, Table 1 shows a summary of the statistical results from test datasets (11 standard images and 100 images selected from PASCAL VOC) corrupted with white Gaussian noise ([60] = 170). In terms of the average PSNR, our method produces significantly better results than BM3D and KSVD and is competitive with MLPD in both datasets. On 11 standard test datasets, our method outperformed MLPD when applied to 7 of the 11 images. For the rest 4 underperformed images, our performance was only 0.1 dB lower than that of MLPD, except the 'Fingerprint' image for which the performance difference was 1 dB. It is likely that no similar fingerprint textures existed within our training data. Of the 100 images in

Table 1
Average denoising performance of two datasets in the clipped noisy model (noise standard deviation $\sigma = 170$).

Test dataset	11 standard images		100 images of PASCAL	
	PSNR	SSIM	PSNR	SSIM
BM3D	18.08	0.4945	16.24	0.4264
KSVD	17.89	0.4734	16.06	0.4068
MLPD	**21.23**	**0.5697**	20.58	0.5172
Ours	21.15	0.5501	**20.80**	**0.5192**

the PASCAL VOC 2012 dataset, our method over-performed MLPD by 0.2 dB on average in terms of PSNR, and the results displayed better performance in 58 of the 100 images.

As Levin and Nadler reported in [58], there is an inherent limit when we estimate a clean version from a noisy natural image contaminated by AWGN. A statistical limit has been proposed to tell how much more the natural image denoising algorithms can be improved when we use a certain patch size during implementation. This is measured by taking a lower bound on the optimal Bayesian minimum mean square error. We can then estimate the best possible bound of the optimal PSNR value ($PSNR^{op}$). We denote $p(x)$ as the probability density corresponding to the w^*h patches, $p(y)$ is the resulting density of the corresponding noisy patches, the noise level is σ, and x_c is the central pixel in each patch x. Using a sufficiently large set of $\mathcal{N}$ image patches and a set of $\mathcal{M}$ image patches, upper and lower bounds of MMSE can be determined as follows

$$MMSE^U = \frac{1}{\mathcal{M}} \sum_j \left(\widehat{\mu}\left(y_j\right) - x_{j,c}\right) \tag{14}$$

$$MMSE^L = \frac{1}{\mathcal{M}} \sum_j \widehat{\mathcal{V}}\left(y_j\right) \tag{15}$$

where $\widehat{\mu}\left(y_j\right)$ is an approximated mean and $\widehat{\mathcal{V}}\left(y_j\right)$ is an approximated variance given by

$$\widehat{\mu}\left(y_j\right) = \frac{\frac{1}{\mathcal{N}} \sum_i p(y_j|x_i) x_{i,c}}{\frac{1}{\mathcal{N}} \sum_i p(y_j|x_i)} \tag{16}$$

$$\widehat{\mathcal{V}}\left(y_j\right) = \frac{\frac{1}{\mathcal{N}} \sum_i p(y_j|x_i) (\widehat{\mu}(y_j) - x_{i,c})^2}{\frac{1}{\mathcal{N}} \sum_i p(y_j|x_i)} \tag{17}$$

For Gaussian noise,

$$p\left(y_j|x_i\right) = \frac{1}{(2\pi\sigma^2)^{w*h/2}} e^{-\frac{x_i - y_j{}^2}{2\sigma^2}} \tag{18}$$

When $\mathcal{N}$ is sufficiently large, the two $MMSE^U$ and $MMSE^L$ values are similar. The best possible optimal PSNR bound can be computed from

$$PSNR^{op} = 10log_{10}\left(\frac{1}{MMSE^L}\right) \tag{19}$$

We provide a reference of the best possible bound of PSNR based on the image *Lena* with three heavy noise levels ([80] = 170, 240, *and*300) using the corresponding patch size shown in Fig. 5; the $PSNR^{op}$ values are about 24.7, 23.6, and 22.5 dB, respectively, which are higher than the denoising results shown in Fig. 10(a). This shows that our denoising algorithm still has additional capacity to provide improvement when heavy noise is present.

Fig. 12 shows the real noisy images and their corresponding denoised results. The images were captured with an industry camera in small aperture mode and with an internal enhancement

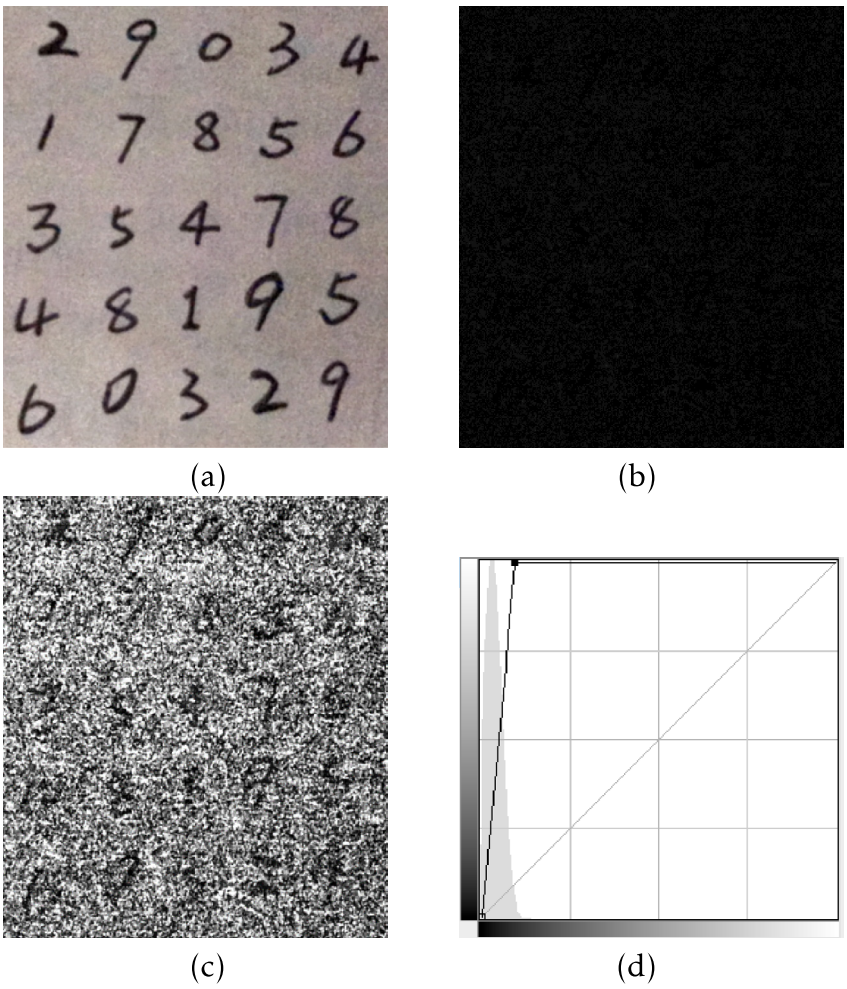


Fig. 8. Denoising test using real-world images. (a) Image of 25 digits captured using normal exposure in a bright room, (b) raw image captured in a dark room, (c) enhanced image by linear grayscale stretching, and (d) mapping curve (black solid) used to map the grayscale in (b) on the horizontal axis to the grayscale in (c) on the vertical axis by linear stretching. The gray area in (d) shows the histogram of the raw image in (b).

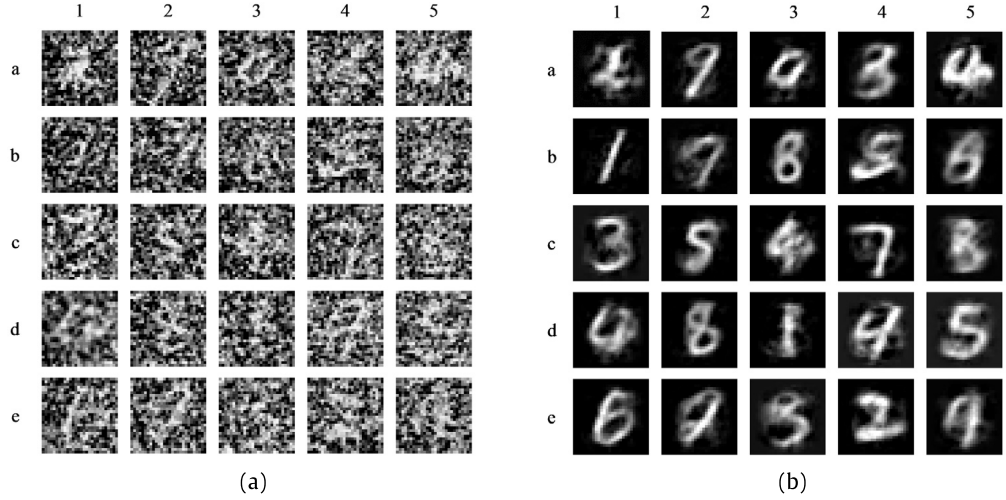


Fig. 9. Denoising results for Fig. 8(c). (a) Gray-scale inverted and cropped image patches containing hand-written digits and (b) results using our patch-based denoising network.

turned on. Thus, they contain heavy noise. After denoising, the visual quality is significantly improved.

The computational burden for applying the HELM-based denoising network is relatively low. For instance, to process a 512 $*$ 512 image cropped to 28 $*$ 28 patches, the patch-based network denoising step takes approximately 8 s with stride 4, or 2 s with stride 8; the non-local aggregation step takes up to 3 min. Our algorithm was simulated in Matlab scripts on a CPU (i7-5500U

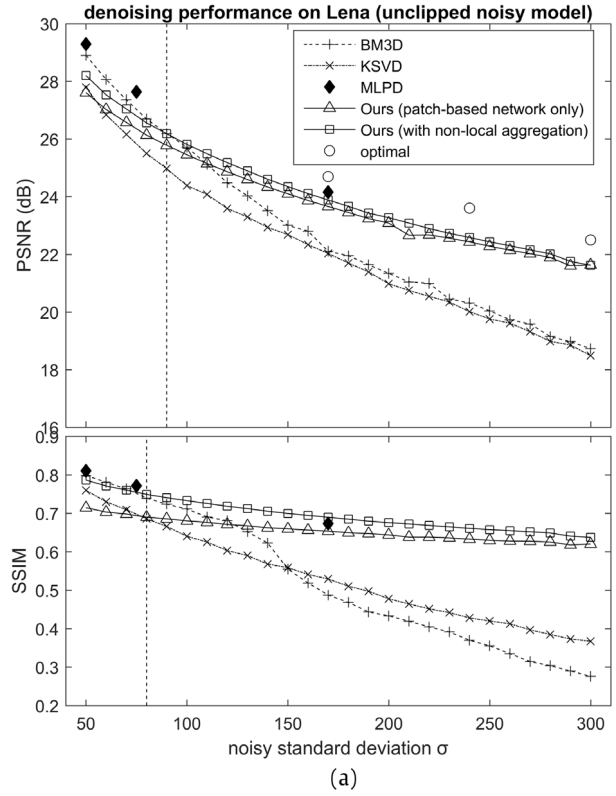


Fig. 10. Comparison of images with various noise levels.

2.7 GHz). Although the use of more complex computation in our case produced better performance, a proper trade-off is necessary and should be determined for particular applications.

Table 2
Comparison of the useful information ratio in the clipped noisy model (noise standard deviation $\sigma = 170$, JPEG compression format).

Approaches	Size of JPEG compressed file	*CR*	*PSNR* (dB)	*UIR* (1/dB)
Noisy	158 KB	0.617	8.05	0.0767
BM3D	15.6 KB	0.061	19.79	0.0031
K-SVD	18.5 KB	0.072	19.27	0.0038
MLPD	18.7 KB	0.073	23.35	0.0031
Ours	15.7 KB	0.061	23.60	**0.0026**
Noise-free	30.8 KB	0.120	∞	0

Table 3
Comparison of the useful information ratio in the clipped noisy model (noise standard deviation $\sigma = 170$, JPEG2000 compression format).

Approaches	Size of JPEG2000 compressed file	*CR*	*PSNR* (dB)	*UIR* (1/dB)
Noisy	292 KB	1.140	8.05	0.1417
BM3D	17.9 KB	0.070	19.79	0.0035
K-SVD	40.3 KB	0.157	19.27	0.0082
MLPD	41.7 KB	0.163	23.35	0.0070
Ours	19.3 KB	0.075	23.60	**0.0032**
Noise-free	128 KB	0.500	∞	0

The compression rate and useful information of denoised images are keys to communication efficiency over the VIoT. We give a definition to describe the useful information ratio (*UIR*) that is relevant to compression ratio (*CR*) of a denoised image and *PSNR* between the denoised and noise-free images:

$$CR = Sc/Sr \tag{20}$$

$$UIR = CR/PSNR \tag{21}$$

where *Sc* is the size of the compressed mono-denoised image file, and *Sr* is the size of the corresponding mono-raw image file.

A smaller *UIR* value indicates more efficient communication may be performed between end nodes and a server in the VIoT. As shown in Tables 2 and 3, the result is illustrated by the example of Lena, which contains Gaussian noise with standard deviation [88] = 170. The file size of the mono-raw data is 256 KB. The image compression approaches we chose are popular the JPEG and JPEG2000 methods [59]. The compression ratio of JPEG and JPEG2000 compressed files were called by using the '*imwrite*' function in Matlab. We compare denoising results with different approaches in the clipped model. Our approach archives the smallest (best) useful information ratio compared to the other three denoising approaches with both JPEG and JPEG2000 compression.

7. Discussion

The trained weights of the auto-encoder layer can be visualized as patches, as shown in Fig. 13. These weights are applied to the input patches with a dot product operation and can be interpreted as filtering kernels, which can be used to extract useful information like structural features from the noisy image. The outputs from this hidden layer are used as filtered elements to combine a denoised patch with the following hidden layer and the output layer.

The weights of the hidden layer and the output layer are trained efficiently without iteration. This is the reason why our denoising network can be implemented at a high computational speed. Meanwhile, the network can be trained to satisfy different noise models and levels (even if $SNR < 0$ dB) without extra effort. The aforementioned experimental results show that our method achieved high performance despite the presence of heavy noise compared with the other state-of-the-art methods. However, at present, our denoising approach cannot outperform others when

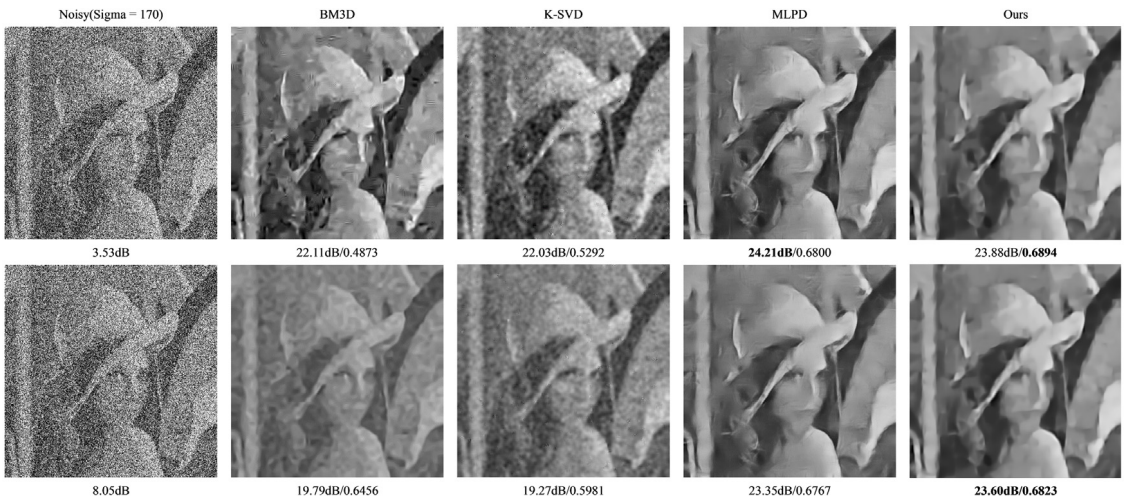


Fig. 11. A noisy image containing Gaussian noise with standard deviation $\sigma = 170$, and denoised images using BM3D, KSVD, MLP, and our method. The top and bottom rows show the results of the models without and with clipping, respectively.

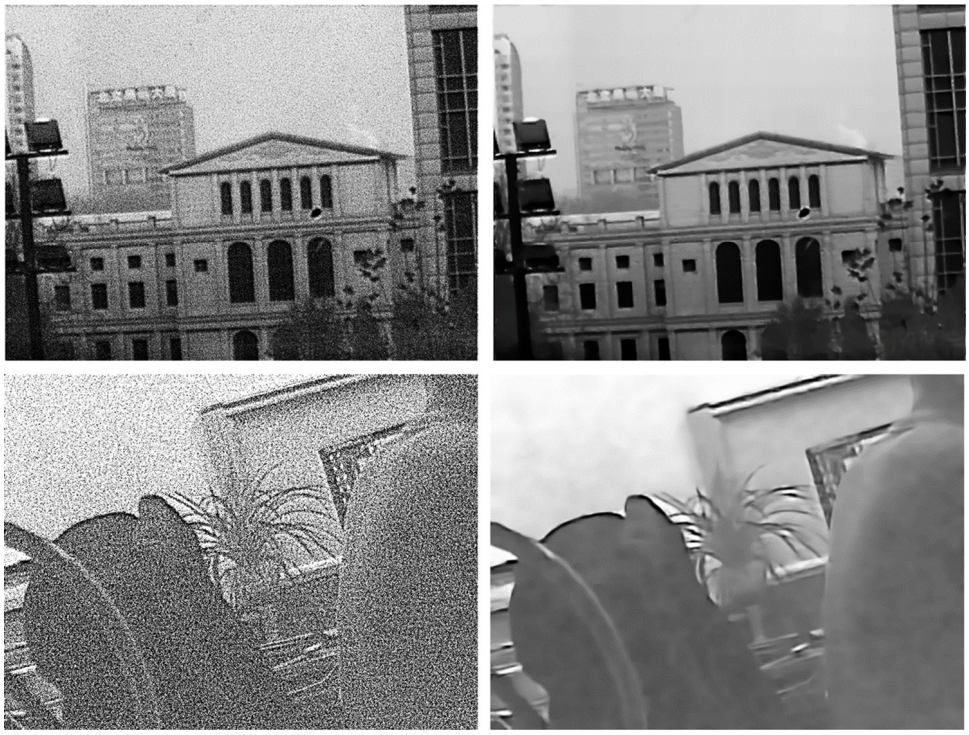


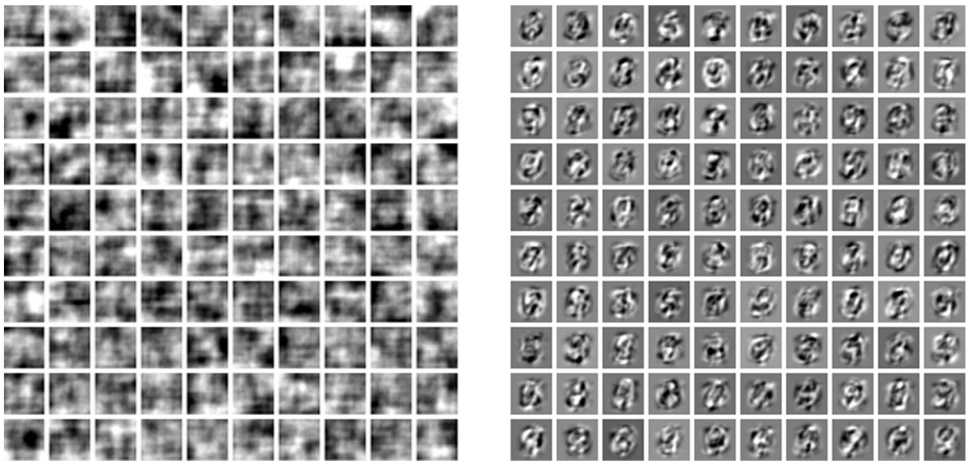
Fig. 12. Real noisy images (left) and denoised images using our method (right).

weak noise is present (i.e. $[92] < 70$). A possible reason is that the architecture of the network is not sufficiently complex to represent details in the image accurately. Increasing the number of nodes or layers may provide a potential solution, but over-fitting of the model should be taken into consideration.

A comparison of the useful information ratio (*UIR*) reveals that our approach is more efficient for transmitting useful information within a limited bandwidth for bandwidth-sensitive VIoT system. This indicates that communication cost can be reduced for data interaction between end devices and servers.

8. Conclusion

To address the problem of removing heavy noise in the visual Internet of Things, we proposed a novel denoising method that is based on hierarchical extreme learning machine. The framework of our method consists of a patch-to-patch image-denoising network and non-local aggregation. Fast training is a key feature of our method compared to other approaches, e.g., our denoising network can be well trained within several minutes on a single CPU. Experimental results show that our method can effectively and efficiently address various noise levels, in both clipped and unclipped noisy models. This method can benefit VIoT applications in surveillance situations, which suffer from heavy noise and limited bandwidth. Our approach can provide efficiently transmit useful information within a limited-bandwidth. For engineering towards practical applications, we have designed an embedded VIoT system that can accommodate our denoising algorithm for multi-camera video surveillance.



(a) (b)

Fig. 13. Random selection of weights in the auto-encoder layer. (a) Weights learned from the natural images dataset. (b) Weights learned from the MNIST dataset.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China [Grant No. 61571026]; the National Key Project of Research and Development Plan, China [No. 2016YFE0108100]; and the National Institutes of Health, the United States [Grant No. R01CA165255 and R21CA172864].

References

[1] A. Boukhayma, A. Peizerat, C. Enz, Temporal readout noise analysis and reduction techniques for low-light CMOS image sensors, IEEE Trans. Electron Devices 63 (1) (2015) 72–78.
[2] J. Han, J. Yue, Y. Zhang, L. Bai, Local sparse structure denoising for low-light-level image, IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 24 (12) (2015) 5177.
[3] Q. Li, H. Cheng, Y. Zhou, G. Huo, Road vehicle monitoring system based on intelligent visual internet of things, J. Sensors 2015 (3) (2015).
[4] D.K. Yadav, K. Singh, S. Kumari, Challenging issues of video surveillance system using Internet of Things in cloud environment, in: International Conference on Advances in Computing and Data Sciences, 2016, pp. 471–481.
[5] N. Bessis, C. Dobre, Big Data and Internet of Things: A Roadmap for Smart Environments, Springer International Publishing, 2014.
[6] W.D. Pence, R. Seaman, R.L. White, Lossless astronomical image compression and the effects of noise, Publ. Astron. Soc. Pac. 121 (878) (2009) 414–427.
[7] A. Buades, B. Coll, J.M. Morel, Nonlocal image and movie denoising, Int. J. Comput. Vis. 76 (2) (2008) 123–139.
[8] K. Dabov, A. Foi, K. Egiazarian, Image denoising with block-matching and 3d filtering, Proc. SPIE - Int. Soc. Opt. Eng. 6064 (2006) 354–365.
[9] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D Transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.
[10] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image restoration by sparse 3D transform-domain collaborative filtering, 6812 (8) (2008) 681207.
[11] K. Dabov, A. Foi, K. Egiazarian, Video denoising by sparse 3d transform-domain collaborative filtering, in: Signal Processing Conference, 2007 European, 2015, pp. 145–149.
[12] H. Talebi, P. Milanfar, Global image denoising, IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 23 (2) (2014) 755–768.
[13] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.
[14] J. Mairal, G. Sapiro, M. Elad, Learning multiscale sparse representations for image and video restoration, SIAM J. Multiscale Model. Simul. 7 (1) (2008) 214–241.
[15] D.D. Muresan, T.K. Parks, Adaptive principal components and image denoising, in: Proc. Int. Conf. Image Processing, vol. 1, 2003, pp. I–101–4.
[16] V. Katkovnik, A. Foi, K. Egiazarian, J. Astola, From local kernel to nonlocal multiple-model image denoising, Int. J. Comput. Vis. 86 (1) (2010) 1.
[17] H.C. Burger, C.J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with BM3D? in: Computer Vision and Pattern Recognition, 2012, pp. 2392–2399.
[18] T. Remez, O. Litany, R. Giryes, A.M. Bronstein, Deep class aware denoising, 2017.
[19] C. Li, P. Qin, J. Zhang, Research on image denoising based on deep convolutional neural network, Comput. Eng. (2017).
[20] R. Vemulapalli, O. Tuzel, M.Y. Liu, Deep Gaussian conditional random field network: A model-based deep network for discriminative denoising, in: Computer Vision and Pattern Recognition, 2016, pp. 4801–4809.
[21] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[22] J. Tang, C. Deng, G.B. Huang, Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2017) 809–821.
[23] N. Liang, G. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411–1423.
[24] N. Barzigar, A. Roozgard, S. Cheng, P. Verma, An efficient video denoising method using decomposition approach for low-rank matrix completion, in: Signals, Systems and Computers, 2013, pp. 1684–1687.
[25] Q. Lu, Z. Lu, X. Tao, H. Li, A new non-local video denoising scheme using low-rank representation and total variation regularization, in: IEEE International Symposium on Circuits and Systems, 2014, pp. 2724–2727.
[26] H. Sadreazami, M.O. Ahmad, M.N.S. Swamy, A Study on Image Denoising in Contourlet Domain Using the Alpha-Stable Family of Distributions, Elsevier North-Holland, Inc., 2016, pp. 459–473.
[27] R. Charnigo, J. Sun, R. Muzic, A semi-local paradigm for wavelet denoising, IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 15 (3) (2006) 666–677.
[28] H.S. Bhadauria, M.L. Dewal, Medical Image Denoising Using Adaptive Fusion of Curvelet Transform and Total Variation, Pergamon Press, Inc., 2013, pp. 1451–1460.
[29] Y. Lecun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, in: IEEE International Symposium on Circuits and Systems, 2010, pp. 253–256.
[30] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: International Conference on International Conference on Machine Learning, 2010, pp. 807–814.
[31] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: International Conference on Machine Learning, 2009, pp. 609–616.
[32] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1646–1654.
[33] X. Zhou, L. Xie, P. Zhang, Y. Zhang, An ensemble of deep neural networks for object tracking, in: IEEE International Conference on Image Processing, 2015, pp. 843–847.
[34] N.D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, F. Kawsar, An early resource characterization of deep learning on wearables, smartphones and Internet-of-Things devices, in: International Workshop on Internet of Things Towards Applications, 2015, pp. 7–12.
[35] G. Huang, G.B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. Off. J. Int. Neural Netw. Soc. 61 (C) (2015) 32.
[36] M. Eshtay, H. Faris, N. Obeid, Metaheuristic-based extreme learning machines: a review of design formulations and applications, Int. J. Mach. Learn. Cybern. (2018).
[37] Z. Xie, X. Kai, L. Liu, Y. Xiong, 3D shape segmentation and labeling via extreme learning machine, in: Computer Graphics Forum, 2014, pp. 85–95.

[38] S.J. Wang, H.L. Chen, W.J. Yan, Y.H. Chen, X. Fu, Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine, Neural Process. Lett. 39 (1) (2014) 25–43.
[39] L. Hu, Y. Chen, J. Wang, C. Hu, X. Jiang, OKRELM: online kernelized and regularized extreme learning machine for wearable-based activity recognition, Int. J. Mach. Learn. Cybern. (6256) (2017) 1–14.
[40] J. Fang, X. Xu, H. Liu, F. Sun, Local receptive field based extreme learning machine with three channels for histopathological image classification, Int. J. Mach. Learn. Cybern. (2018) 1–11.
[41] X. Luo, J. Liu, D. Zhang, X. Chang, A large-scale web QoS prediction scheme for the Industrial Internet of Things based on a kernel machine learning algorithm, Comput. Netw. 101 (2016) 81–89.
[42] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, J. Qin, A survey on application of machine learning for internet of things, 2018.
[43] U. Drolia, K. Guo, P. Narasimhan, Precog: prefetching for image recognition applications at the edge, in: ACM/IEEE Symposium on Edge Computing, 2017, pp. 1–13.
[44] D. Ventura, D. Casado-Mansilla, J. Lopezdearmentia, P. Garaizar, D. Lopezdeipina, V. Catania, ARIIMA: A Real IoT Implementation of a Machine-Learning Architecture for Reducing Energy Consumption, Springer International Publishing, 2014, pp. 444–451.
[45] S. Dhote, P. Charjan, A. Phansekar, A. Hegde, S. Joshi, J. Joshi, Using FPGA-SoC interface for low cost IoT based image processing, in: International Conference on Advances in Computing, Communications and Informatics, 2016, pp. 1963–1968.
[46] P. Li, Z. Chen, L.T. Yang, Q. Zhang, M.J. Deen, Deep convolutional computation model for feature learning on big data in Internet of Things, IEEE Trans. Ind. Inf. PP (99) (2017) 1–1.
[47] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[48] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. B 42 (2) (2012) 513–529.
[49] L. Meng, S. Ding, Y. Xue, Research on denoising sparse autoencoder, Int. J. Mach. Learn. Cybern. 8 (5) (2017) 1719–1729.
[50] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Alkon, Accelerating the convergence of the back-propagation method, Biol. Cybernet. 59 (4–5) (1988) 257–263.
[51] Q. He, T. Shang, F. Zhuang, Z. Shi, Parallel extreme learning machine for regression based on mapreduce, Neurocomputing 102 (2) (2013) 52–58.
[52] M.V. Heeswijk, Y. Miche, E. Oja, A. Lendasse, GPU-accelerated and parallelized ELM ensembles for large-scale regression, Neurocomputing 74 (16) (2011) 2430–2437.
[53] S. Decherchi, P. Gastaldo, A. Leoncini, R. Zunino, Efficient digital implementation of extreme learning machines for classification, IEEE Trans. Circuits Syst. II Express Briefs 59 (8) (2012) 496–500.
[54] M. Brand, Fast low-rank modifications of the thin singular value decomposition, Linear Algebra Appl. 415 (1) (2006) 20–30.
[55] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[56] NationalGeographic, National Geographic website, http://www.nationalgeographic.com/.
[57] M. Everingham, S.M.A. Eslami, L.V. Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (1) (2015) 98–136.
[58] A. Levin, B. Nadler, Natural image denoising: optimality and inherent bounds, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2833–2840.
[59] T. Acharya, P.S. Tsai, JPEG - Still Image Compression Standard, John Wiley and Sons, Inc., 2005, pp. 61–80.