IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON THEORY, ALGORITHMS, AND APPLICATIONS OF SPARSE RECOVERY

Received October 25, 2018, accepted November 13, 2018, date of publication November 21, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2882531

Deep Neural Network Structured Sparse Coding for Online Processing

HAOLI ZHAO, SHUXUE DING, (Member, IEEE), XIANG LI, AND HUAKUN HUANG
The School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-0005, Japan

Corresponding author: Shuxue Ding (sding@u-aizu.ac.jp)

This work was supported by the 2016 Grants-In-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science and Technology, Japan, under Project 16K00335.

ABSTRACT Sparse coding, which aims at finding appropriate sparse representations of data with an overcomplete dictionary set, has become a mature class of methods with good efficiency in various areas, but it faces limitations in immediate processing such as real-time video denoising. Unsupervised deep neural network structured sparse coding (DNN-SC) algorithms can enhance the efficiency of iterative sparse coding algorithms to achieve the goal. In this paper, we first propose a sparse coding algorithm by adding the idea "weighted" in the iterative shrinkage thresholding algorithm (ISTA), named WISTA, which can enjoy the benefit of the l_p norm ($0 < p < 1$) sparsity constraint. Then, we propose two novel DNN-SC algorithms by combining deep learning with WISTA and the iterative half thresholding algorithm (IHTA), which is the $l_{0.5}$ norm sparse coding algorithm. Furthermore, we present that by changing the loss function, the DNN can be learned supervisedly and unsupervisedly. Unsupervised learning is the key to ensure the DNN to be learned online during processing, which enables the use of the DNN-SC algorithms in applications lacking labels for signals. Synthetic data experiments show that WISTA can outperform ISTA and IHTA. Moreover, the DNN-structured WISTA can successfully achieve converged results of WISTA. In real-world data experiments, the procedure of utilizing DNN-SC algorithms in image denoising is first presented. All DNN-SC algorithms can accelerate at least 45 times while maintaining PSNR results compared with their corresponding sparse coding algorithms. Finally, the strategy of utilizing DNN-SC algorithms in real-time video denoising is presented. The video-denoising experiments show that the DNN-structured ISTA and WISTA can conduct real-time video denoising for 25 frames/s 360×480 pixels gray-scaled videos.

INDEX TERMS Sparse coding, deep neural network, weighted iterative shrinkage thresholding algorithm, unsupervised learning, real-time video denoising.

I. INTRODUCTION

The concept of sparse representation, which concentrates on presenting sparse estimations from underdetermined linear measurements, has proven its efficiency in signal processing for over ten years [1], [2]. Sparse coding considers a certain condition where one searches for sparse representations of a signal from a predefined overcomplete set of vector bases. The idea of sparse representation comes from the fact that most real-world signals and data can be represented by a linear combination of a few representative elements from a dictionary base in certain signal models. To construct effective sparse coding algorithms, many researches have made effort to design appropriate dictionary bases and develop efficient algorithms to reconstruct signals from noisy and incomplete measurements. As a consequence, various signal processing applications have benefited from employing sparse coding such as image denoising [3]–[5], inpainting [2], [6], super-resolution [7], [8], etc..

Focusing on the object of sparse coding, one is required to recover a sparse signal $\mathbf{z} \in \mathbb{R}^n$ from a lower-dimensional measured signal $\mathbf{x} \in \mathbb{R}^m$, $n > m$, in a linear relationship as $\mathbf{x} = \mathbf{D}\mathbf{z} + \mathbf{n}$, where $\mathbf{D} \in \mathbb{R}^{m \times n}$ is a dictionary matrix, and $\mathbf{n} \in \mathbb{R}^m$ is the measurement noise. Since $\mathbf{D}$ is overcomplete, this signal reconstruction task is ill-posed with infinite solutions if there is no restriction on $\mathbf{z}$. In some applications, to find a sparsest one in the infinite solutions can be a meaningful mode and can make the problem well-posed, which is the essence of sparse coding. Therefore, the optimization problem is commonly used to search for the optimal solution of the linear signal model,

$$\min_{\mathbf{z}} \quad \|\mathbf{z}\|_q \quad \text{subject to } \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_F^2 \leq \epsilon, \tag{1}$$

where $\|\mathbf{z}\|_q$ is the sparsity constraint. Among the various possible values for q, the sparsest solution can be guaranteed when $q = 0$, where the l_0 norm counts the number of nonzero entries in $\mathbf{z}$. However, this is a combinatorial optimization problem, where searching for an accurate sparse representation with large n is computationally prohibited [2]. One typical alternative method is using a greedy algorithm such as orthogonal matching pursuit (OMP) [9], which can always find sufficiently sparse solutions but faces the problem of high computational complexity, though it is not computationally prohibited, with large data size. Another idea to solve the optimization problem relies on relaxation-based approaches. For example, the l_1 norm, where $q = 1$, is often selected as the sparsity constraint for the optimization problem, e.g., the Iterative Shrinkage Thresholding Algorithm (ISTA) [10] and Basis Pursuit (BP) [11]. The l_1 norm is attractive because it is a convex problem and can achieve reasonable performance. Subband Adaptive Iterative Shrinkage Thresholding Algorithm (SISTA) [12], [13], a modified version of ISTA, can accelerate the convergence via processing of ISTA in subbands. The developed algorithm in the paper, as detailed in the following, works with the whole-band signal; but may be further modified to work in subbands. While searching for sparser and more accurate representations, the l_p norm, where $q = p \in (0, 1)$, can be an alternative choice as a relaxation approach. Although the l_p norm ($0 < p < 1$) makes the optimization problem nonconvex, it has proven its possibility in enhancing the sparsity and accuracy of solutions compared to the l_1 norm, e.g., the Iterative Half Thresholding Algorithm (IHTA) [14], and weighted l_1 norm [15]–[18].

Nevertheless, the efficiency and capacity of the current sparse coding algorithms have not reached the requirement of certain missions that require fast processing, such as real-time video denoising; thus, we propose our sparse coding algorithms, which use the DNN structure and corresponding learning procedure in sparse coding to achieve the goal. Recently, deep learning has shown to be a possible approach by combining with iterative sparse coding to resolve sparse representation problems in previous work (e.g., [19]–[24]). In 2010, Gregor and LeCun introduced the idea with ISTA, where the learned DNN can perform nearly 10 times faster than the original sparse coding algorithm, i.e., ISTA [19]. However, its learning procedures require one to know the true answers of the sparse representation. In 2015, Sprechmann et al. showed that it is possible to learn the DNN without requiring the true answers to serve as the training labels, i.e., the unsupervised learning is possible [22]. With unsupervised network learning, the Deep Neural Network structured Sparse Coding (DNN-SC) algorithms can be more suitable for online processing without prior training when the network can be efficiently quickly learned online to make the processing enjoy the acceleration introduced by the DNN structure. Then, we would like to give a brief review about DNN-structured sparse representation algorithms.

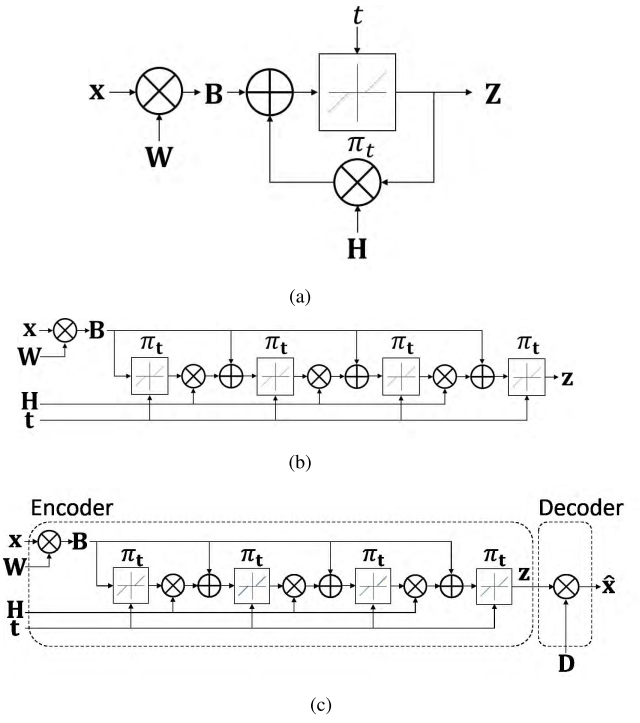


FIGURE 1. (a) Illustration of the ISTA algorithm for sparse coding. The optimal sparse representation can be obtained by the recursive structure $\mathbf{z}^{(k)} = \pi_t(\mathbf{W}\mathbf{x} + \mathbf{H}\mathbf{z}^{(k-1)})$, where $\mathbf{x}$ is the input signal, π_t is the soft thresholding function with threshold t, $\mathbf{W} = \frac{1}{\alpha}\mathbf{D}^{\mathbf{T}}$, $\mathbf{H} = \mathbf{I} - \frac{1}{\alpha}\mathbf{D}^{\mathbf{T}}\mathbf{D}$, and α is a restriction parameter for ISTA. (b) Network structure of the supervised learned DNN-ISTA, which is named LISTA, formed from unfolded ISTA and truncated to a fixed number of iterations (3 here). W, H, t are trainable parameters in the network to give an approximate sparse representation on a given dataset. (c) Network structure of the unsupervised learned DNN-ISTA, which is named TISTA. TISTA has a similar propagation structure to LISTA, and W, H, and t are targeted trainable parameters. The key difference is that TISTA uses a decoder to output x as the learning objective, where the original x is the known input. On the contrary, original z, which is required for supervised learning, is a priori knowledge.

A. PRIOR ART

All DNN-structured sparse representation algorithms are based on the idea of unfolding an iterative algorithm with shared parameters through specific layers based on the processing similarity between the iterative algorithm and the Recurrent Neural Network (RNN), which was developed by Domke and applied to the tree-reweighted belief propagation and mean-field inference [25], [26]. Gregor and LeCun were the first to implement this idea in the sparse coding algorithm [19]. They unfolded the structure of ISTA, as shown in Fig. 1(a), to form a feed-forward neural network named Learned ISTA (LISTA), which is illustrated in Fig. 1(b). LISTA is a supervised learned neural network that requires inputting a dataset of signals and corresponding sparse representations (as labels for training) pairs to a truncated unfolded ISTA structure to train the weights and bias. The learned DNN from LISTA has proven its efficiency in estimating spare representations of other signals besides the ones used in training, which can reach the converged result standard of ISTA with much fewer layers than the number of ISTA

convergent iterations. Sprechmann et al. proposed the unsupervised Trained ISTA (TISTA) [22]. TISTA has a similar neural network structure to LISTA because both form from a truncated unfolded ISTA structure, as shown in Fig. 1(c), but TISTA changes the learning procedure by adding a decoder at the end of the network. By requiring the output as close as possible to the input, it becomes possible to directly learn the weights and bias from the loss function of the original sparse representation problem. Consequently, TISTA avoids using the true sparse representations for a separated training and enables the online network learning and processing procedure simultaneously. The unfolded ISTA also shows that DNN-SC can help learning the optimal nonlinear threshold functions for iterative sparse coding to achieve better performances in known datasets [20], [21]. Kamilov and Mansour [20] proposed to learn the nonlinear activation function, which is modeled using cubic B-splines through DNN-structured ISTA. The learned nonlinear threshold can result in better accuracy than ISTA. Mahapatra *et al.* [21] proposed a more parsimonious representation of the thresholding function using a linear expansion of thresholds during learning. Furthermore, the DNN can be applied to other iterative spare coding algorithms such as Iterative Hard Thresholding (IHT) [23] and Approximate Message Passing (AMP) [24]. Both IHT and AMP have the key similarity to ISTA: All functions in these sparse coding algorithms are continuous and overall differentiable throughout; thus, they can be unfolded, and the parameters in their structures can be learned as a DNN. Moreover, Moreau and Bruna presented mathematical explanations about the acceleration of DNN-structured algorithms by analyzing the specific matrix factorization in the Gram kernel of dictionaries [27]. The findings show that the learning procedure in DNN-SC attempts to diagonalize the kernel with a basis, which produces a small perturbation of the original l_1 space, and the learning may fail if there is no factorization. Efforts are also made to combine sparse coding algorithms with different DNN structures. The convolutional neural networks (CNN), which have proven its superiority in image processing tasks, have been widely combined with sparse coding algorithms for image classification [28]–[30] and image restoration [31]–[33]. These deep learning architectures are significantly different from DNN-SC since they are not truncated iterative SC algorithms but using sparse coding structure to reconstruct signal for specific tasks. A notable difference is that the output signal is, e.g., class labels in the general deep learning architectures. In this paper, the output signal is $\mathbf{z}$, since the purpose is to present signal $\mathbf{x}$ with a sparse $\mathbf{z}$.

B. THIS PAPER

In contrast with previous work, the main contributions of this paper are as follows.

1) We show how to unfold IHTA and WISTA to form a feed-forward neural networks (section III), where the parameters can be learned by back-propagation. Both unfolded WISTA and IHTA have similar structures to ISTA and the recurrent neural network, where each layer is comprised by a differentiable combination of linear and nonlinear operators.
2) For the setting of the loss function, all DNN-SC algorithms can be learned through supervised and unsupervised schemes (section III.A). The unsupervised learning procedure is the key to apply DNN-SC in online learning processing.
3) Benefiting from the acceleration of the DNN structured processing, we show that DNN-SC algorithms can conduct real-time video denoising with only CPU for 25-FPS 360 × 480-pixel gray-scaled videos.

Subsequently, we give a experimental validation of proposed algorithms in section IV. Synthetic data experiments (section IV.A) present a performance comparison in terms of the relative norm error and accuracy among the sparse coding algorithms ISTA, IHTA and WISTA and their DNN-structured versions.

Real-world graphic experiments are shown in section IV.B. Both image and video denoising experiments concentrate on the acceleration of the DNN-SC algorithm in denoising while maintaining reasonably good performances. Furthermore, DNN-WISTA and DNN-ISTA have proven their possibility in conducting real-time video denoising.

TABLE 1. Notations and descriptions.

Notation	Description
ISTA	Iterative Shrinkage Thresholding Algorithm
IHTA	Iterative Half Thresholding Algorithm
WISTA	Weighted Iterative Shrinkage Thresholding Algorithm
WISTA0.9	The number 0.9 after WISTA stands for the p value used in WISTA, The same are WISTA0.7 and WISTA0.5
DNN	Deep Neural Network
DNN-SC	Deep Neural Network structured Sparse coding
DNN-ISTA	DNN-structured ISTA, the same are DNN-IHTA and DNN-WISTA
LISTA	Supervised Learned ISTA, 'L' stands for supervised learning, the same are LIHTA and LWISTA
TISTA	Unsupervised Trained ISTA, 'T' stands for unsupervised learning, the same are TIHTA and TWISTA
$\pi_t(x)$	Soft thresholding operator, $\pi_t(x) = \text{sign}(x)\max\{\|x\| - t, 0\}$
$h_t(x)$	Half thresholding operator, $h_t(x) = \frac{2}{3}x(1+\cos(\frac{2\pi}{3} - \frac{2}{3}\arccos(\frac{t}{8}(\frac{\|x\|}{3})^{-1.5})))$ $\text{sign}(\|x\| - \frac{\sqrt[3]{54}}{4}t^{\frac{2}{3}})$
$\mathbf{x}$	Input data vector, sized as $m \times 1$
$\mathbf{X}$	Input data set, sized as $m \times N$, $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)$
$\mathbf{D}$	Dictionary base for sparse representation, sized as $m \times n$
$\mathbf{z}$	Sparse representation vector, sized as $n \times 1$
$\mathbf{Z}$	Sparse representation set, sized as $n \times N$, $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_N)$
$\mathbf{z}^{(k)}$	k in the bracket stands for number of iterations or layers in algorithms

II. SPARSE CODING

The paper considers a sparse representation problem, where we tend to find a proper approach to obtain an optimal sparse solution $\mathbf{z} \in \mathbb{R}^n$ from given noisy data $\mathbf{x} \in \mathbb{R}^m$ based on the linear signal model described in the following equation:

$$\mathbf{x} = \mathbf{D}\mathbf{z} + \mathbf{n}, \tag{2}$$

Algorithm 1 ISTA

Input: data $\mathbf{x}$, dictionary $\mathbf{D}$, proper parameters λ and α.
Restriction: $\alpha >$ largest eigenvalue of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$
Initialization: $t = \frac{\lambda}{\alpha}$, $\mathbf{z}^{(0)} = \mathbf{0}$, $k = 0$.
Main iteration: increment k by 1
$\mathbf{z}^{(k)} = \pi_t(\mathbf{z}^{(k-1)} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}(\mathbf{D}\mathbf{z}^{(k-1)} - \mathbf{x}))$
Stopping rule: stop if $\mathbf{z}^{(k)}$ has converged
Output: $\mathbf{z} = \mathbf{z}^{(k)}$

where $\mathbf{D} \in \mathbb{R}^{m\times n}$ is an overcomplete dictionary matrix with $n > m$, and $\mathbf{n} \in \mathbb{R}^m$ is an additive white Gaussian distributed noise vector. The above equation (2) defines an underdetermined linear system. Because the dictionary in the equation is supposed to be a full row-rank matrix, this model should have infinite solutions. To achieve the required sparse answer, the sparse constraint is introduced. Therefore, the general minimization model with the square data fitting error and a sparse constraint is applied to solve the linear signal model,

$$\min_{\mathbf{z}} f(\mathbf{z}) = \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda d_q(\mathbf{z}), \tag{3}$$

where $\lambda > 0$ is a tuning parameter to adjust the effect of the sparse constraint; the function $d_q(\mathbf{z})$ is a sparse penalty term formulated as follows,

$$d_q(\mathbf{z}) = \|\mathbf{z}\|_q^q = \sum_{j=1}^{n} |z_j|^q, \tag{4}$$

A. ISTA

ISTA [10] is one of the best known iterative algorithms to solve the sparse linear problem. The q value in equation (3) of ISTA is 1, i.e., this is a convex problem where a local minimum is the global minimum. The detail of ISTA is shown in Algorithm 1, and the diagram is presented in Fig. 1(a). For input vector $\mathbf{x}$, ISTA iterates the following recursive equation to approach the optimal:

$$\mathbf{z}^{(k)} = \pi_t(\mathbf{W}\mathbf{x} + \mathbf{H}\mathbf{z}^{(k-1)}), \tag{5}$$

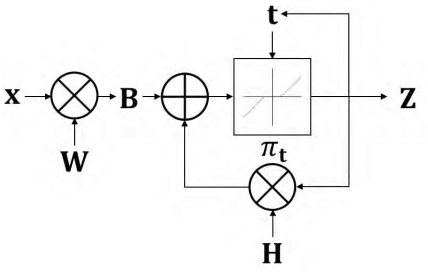
where $\mathbf{W} = \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}$, $\mathbf{H} = \mathbf{I} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}\mathbf{D}$ and α is a restriction parameter, which should be larger than the largest eigenvalue of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$. The operator π_t is a nonlinear soft thresholding operator, which is defined in equation (6).

$$[\pi_t(\mathbf{x})]_j = \mathrm{sign}(x_j)\max\{|x_j| - t, 0\} \tag{6}$$

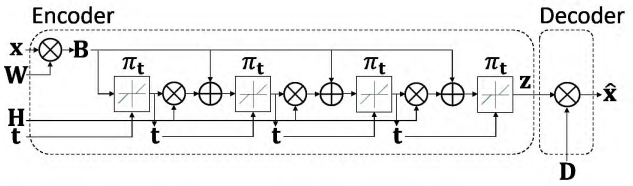
B. IHTA

IHTA [14] concentrates on the nonconvex and nonsmooth optimization model, where the q value of equation (3) is 0.5. The detail of IHTA is shown in Algorithm 2, and the diagram is presented in Fig. 2(a).

In Algorithm 2, h_t is the half thresholding operator, which is the key difference of IHTA compared to ISTA. Applying $q = 0.5$ may cause a convergence issue since it is nonconvex, whereas the lower $q \in (0, 1)$ value tends to more rapidly and efficiently achieve a sparser representation, meanwhile, the algorithm IHTA can lead to a converged result when



(a)



(b)

FIGURE 2. (a) Illustration of the IHTA structure for sparse coding. The optimal sparse representation can be obtained by the recursive structure $\mathbf{z}^{(k)} = h_t(\mathbf{W}\mathbf{x} + \mathbf{H}\mathbf{z}^{(k-1)})$, where $\mathbf{x}$ is the input signal, h_t is the half thresholding operator with threshold t, $\mathbf{W} = \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}$, $\mathbf{H} = \mathbf{I} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}\mathbf{D}$ and α is a restriction parameter for IHTA. (b) The network structure of DNN-IHTA is formed from unfolded IHTA and truncated to a fixed number of iterations (3 here). W, H, and t are trainable parameters in the network to provide an approximate sparse representation on a given dataset. The network can be trained supervisedly with only the encoder and unsupervisedly using both encoder and decoder.

Algorithm 2 IHTA

Input: data $\mathbf{x}$, dictionary $\mathbf{D}$, proper parameters λ and α.
Restriction: $\alpha >$ largest eigenvalue of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$
Initialization: $t = \frac{\lambda}{\alpha}$, $\mathbf{z}^{(0)} = \mathbf{0}$, $k = 0$.
Main iteration: increase k by 1
$\mathbf{z}^{(k)} = h_t(\mathbf{z}^{(k-1)} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}(\mathbf{D}\mathbf{z}^{(k-1)} - \mathbf{x}))$
Stopping rule: stop if $\mathbf{z}^{(k)}$ has converged
Output: $\mathbf{z} = \mathbf{z}^{(k)}$

λ is sufficiently small, and dictionary $\mathbf{D}$ satisfies a certain concentration assumption [34]. By applying the $l_{0.5}$ norm sparsity constraint, the half thresholding operator h_t can be formulated as an analytical expression of the well-defined resolvent operator on its loss function [14], which is defined in (7).

$$[h_t(\mathbf{x})]_j = \frac{2}{3}x_j\left(1 + \cos\left(\frac{2\pi}{3} - \frac{2}{3}\arccos\left(\frac{t}{8}\left(\frac{|x_j|}{3}\right)^{-1.5}\right)\right)\right) \times \mathrm{sign}\left(|x_j| - \frac{\sqrt[3]{54}}{4}t^{\frac{2}{3}}\right) \tag{7}$$

C. WISTA

The proposed Weighted Iterative Shrinkage Thresholding Algorithm (WISTA) in this paper also concentrates on the nonconvex and nonsmooth l_p regularization ($q = p \in (0, 1)$) optimization model considering the benefit in sparsity-inducing and efficiency. The word 'weighted' refers to the idea of restraining the l_1 sparsity constraint with information from the previous iteration, which can approximately

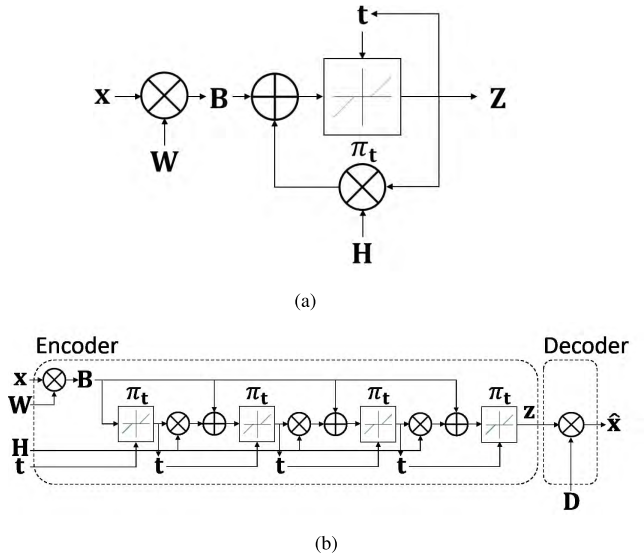


FIGURE 3. (a) Illustration of the WISTA structure for sparse coding. The optimal sparse representation can be recursively obtained in two steps: $\mathbf{z}^{(k)} = \pi_{\mathbf{t}(k-1)}(\mathbf{W}\mathbf{x} + \mathbf{H}\mathbf{z}^{(k-1)})$, $\mathbf{t}^{(k)} = \frac{\lambda}{\alpha}|\mathbf{z}^{(k)}|^{p-1}$, where $\mathbf{x}$ is the input signal, $\pi_{\mathbf{t}}$ is the soft thresholding operator with a changing threshold $\mathbf{t}$ during the iterations, $\mathbf{W} = \frac{1}{\alpha}\mathbf{D}^{\mathbf{T}}$, $\mathbf{H} = \mathbf{I} - \frac{1}{\alpha}\mathbf{D}^{\mathbf{T}}\mathbf{D}$, and α is a restriction parameter for WISTA. (b) The network structure of DNN-IHTA is formed from the unfolded WISTA and truncated to a fixed number of iterations (3 here). W, H, and t are trainable parameters in the network to provide an approximate sparse representation on a given dataset. The network can be trained supervisedly with only the encoder and unsupervisedly using both encoder and decoder.

Algorithm 3 WISTA

Input: data $\mathbf{x}$, dictionary $\mathbf{D}$, proper parameters λ and α.
Restriction: $\alpha >$ largest eigenvalue of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$
Initialization: $\mathbf{t} = \frac{\lambda}{\alpha}\mathbf{1}$, $\mathbf{z}^{(0)} = \mathbf{0}$, $k = 0$.
Main iteration: increase k by 1
$\mathbf{z}^{(k)} = \pi_{\mathbf{t}}(\mathbf{z}^{(k-1)} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}(\mathbf{D}\mathbf{z}^{(k-1)} - \mathbf{x}))$
$\mathbf{t} = \frac{\lambda}{\alpha}|\mathbf{z}^{(k)}|^{p-1}$
Stopping rule: stop if $\mathbf{z}^{(k)}$ has converged
Output: $\mathbf{z} = \mathbf{z}^{(k)}$

function as an l_p norm. Namely, the sparsity constraint can be reformed as follows:

$$\lambda\|\mathbf{z}\|_p^p = \lambda\sum_i^n |z_i|^p \approx \lambda\sum_i^n |z_i^{(\text{iteration}-1)}|^{p-1}|z_i|. \quad (8)$$

Because the component-wise weighted part from the previous iteration can be considered constant during the iterations, the algorithm is transformed to a sequence of weighted l_1 minimization problems from the l_p minimization problem [17], which is convex for the calculation. The detail of WISTA is shown in Algorithm 3, and the diagram is presented in Fig. 3(a).

Different from [17] and [18], where the gradient descent method is applied for the optimization, the proximal operator is applied in WISTA. Based on the difference in the sparse constraint, the form of soft thresholding operator $\pi_{\mathbf{t}}$ in WISTA is slightly different from the one in ISTA, in which the input vector $\mathbf{z}$ is handled with vector $\mathbf{t}$ component-wise during the iterations, as defined in equation (9).

$$[\pi_{\mathbf{t}}(\mathbf{x})]_j = \text{sign}(x_j)\max\ \{|x_j| - t_j, 0\} \quad (9)$$

III. DEEP NEURAL NETWORK STRUCTURED SPARSE CODING

In the conventional deep learning [35], the training data, which comprise of pairs of feature and label, are used to learn the parameters of a deep neural network to predict unknown labels using the new given features. Typically, a Recurrent Neural Network (RNN) receives features and subjects them to a deep structure of many layers for processing, where each layer consists of a linear transformation followed by a component-wise nonlinearity transformation. The unfolded ISTA is shown in Fig. 1 with unfolded IHTA (Fig. 2) and unfolded WISTA (Fig. 3); they hold similar structures to RNN, which enables the use of the parameter training methods in deep learning to train a DNN-SC algorithm. However, we should be clear about the differences between the two structures. The deep neural networks can be divided into two main types by their purposes, classification and regression. In a classification DNN, labels are generally discrete, e.g., when features are images, the labels will be their classes {dog, cat, ... , chicken}. In a contract, for a regression DNN and DNN-SC, the labels are numerical values, which are genearlly continuous and high-dimensional.

For the objective of building a DNN-SC algorithm, it is essential to ensure that all functinos in the encoders are continuous and overall differentiable throughout the network structure, which provides the possibility of using gradient-based learning methods to train network parameters. For example, an overall differentiable structure ensure back propagation provide gradients throughout.

In this paper, we would like to propose two novel DNN-SC algorithms: DNN-structured IHTA (DNN-IHTA) and DNN-structured WISTA (DNN-WISTA), which can be trained to compute approximate sparse codes. The two algorithms are based on IHTA and WISTA. We have applied two training modes in our proposed encoders: supervised and unsupervised.

A. SUPERVISED AND UNSUPERVISED LEARNING

To train a DNN that is formed from a truncated sparse coding algorithm, referring to Fig. 1(b), with training data $\mathbf{x}$ as the input, the encoder outputs $\mathbf{z}$ after the defined depth of the DNN. We use the gradient decent to train the parameters to minimize the loss function $L(\mathbf{z})$, which is defined as the squared error between the predicted code $\mathbf{z}$ from the DNN forward-propagation result and the corresponding optimal code $\mathbf{z}^*$ [19], as shown in equation (10). In practice, $\mathbf{z}^*$ can be obtained from the converged results of the corresponding sparse coding algorithms.

$$L(\mathbf{z}) = \|\mathbf{z} - \mathbf{z}^*\|_2^2 \quad (10)$$

Algorithm 4 DNN-IHTA Forward propagation

Input: data $\mathbf{x}$, dictionary $\mathbf{D}$, proper parameters λ and α, network layer T.
Restriction: $\alpha >$ largest eigenvalue of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$
Initialization: $\mathbf{t} = \frac{\lambda}{\alpha}\mathbf{1}$, $\mathbf{z}^{(0)} = \mathbf{0}$, $\mathbf{H} = \mathbf{I} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}\mathbf{D}$, $\mathbf{W} = \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}$, $\mathbf{B} = \mathbf{W}\mathbf{x}$.
For $k = 1$ to T
$\mathbf{c}^{(k-1)} = \mathbf{B} + \mathbf{H}\mathbf{z}^{(k-1)}$
$\mathbf{z}^{(k)} = h_{\mathbf{t}}(\mathbf{c}^{(k-1)})$
End
Output: $\mathbf{C} = \{\mathbf{c}^{(0)}, ..., \mathbf{c}^{(T-1)}\}$, $\mathbf{Z} = \{\mathbf{z}^{(0)}, ..., \mathbf{z}^{(T)}\}$

Therefore, as the true answer $\mathbf{z}^*$ is functioned as training label, the loss function results in a supervised learning procedure. Then, for a T-layer DNN-SC, the gradient of $\mathbf{z}^{(T)}$ in supervised learning is

$$\delta\mathbf{z}^{(T)} = \frac{\partial L(\mathbf{z}^{(T)})}{\partial \mathbf{z}} = \mathbf{z}^{(T)} - \mathbf{z}^*. \tag{11}$$

In comparison, unsupervised learning aims at training the parameters in the DNN without requiring the true answers to serve as the training labels, i.e., optimal code $\mathbf{z}^*$ in DNN-SC. By adding a decoder in the network, as shown in Fig. 1(c), we change the output of the DNN to $\hat{\mathbf{x}} = \mathbf{D}\mathbf{z}$. Therefore, we can use the general sparse representation loss function to avoid using $\mathbf{z}^*$ [22]. The loss function of unsupervised learning and the corresponding gradient of $\mathbf{z}^{(T)}$ is shown below.

$$L(\mathbf{z}) = \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda d_p(\mathbf{z}) \tag{12}$$

$$\delta\mathbf{z}^{(T)} = \frac{\partial L(\mathbf{z}^{(T)})}{\partial \mathbf{z}} = \mathbf{D}^{\mathrm{T}}(\mathbf{D}\mathbf{z}^{(T)} - \mathbf{x}) + \lambda\frac{\partial d_p(\mathbf{z}^{(T)})}{\partial \mathbf{z}} \tag{13}$$

Between two learning procedures resulted from different loss function settings, the key difference concerns that prior training is essential in supervised learning, which makes it difficult to implement the supervised DNN-SC in online processing and applications without training labels. On the contrary, by avoiding using $\mathbf{z}^*$, unsupervised learning enables the learned DNN for online processing when the learned DNN is efficient and learning procedure is fast enough.

B. DNN-STRUCTURED IHTA

By unfolding the iterations of IHTA from Algorithm 2, we can construct neural network structures. The algorithm can be rewritten as a network structure as follows, and the unfolded network structure is shown in Fig. 2(b).

In the network, $\mathbf{W}$, $\mathbf{H}$, and $\mathbf{t}$ are parameters to train. After the forward propagation with determined network layer T, $\mathbf{C}$ and $\mathbf{Z}$ of each layer in the IHTA network are saved for the back-propagation parameter training. The gradient-based parameter learning schedules of the network are shown in Algorithm 5. The learning procedure can be either adapted to supervised or unsupervised learning by respectively choosing $\delta\mathbf{z}^{(T)}$ from equation (11) or equation (13).

Algorithm 5 DNN-IHTA Back propagation

Input: $\mathbf{x}$, $\mathbf{D}$, $\delta\mathbf{z}^{(T)}$, $\mathbf{Z}$, $\mathbf{C}$, $\mathbf{B}$, $\mathbf{H}$, $\mathbf{t}$, λ and α.
Initialization: $\delta\mathbf{t}^{(T)} = \mathbf{0}$, $\delta\mathbf{B}^{(T)} = \mathbf{0}$, $\delta\mathbf{H}^{(T)} = \mathbf{0}$.
For $k = T - 1$ down to 0
$\delta\mathbf{t}^{(k)} = \delta\mathbf{t}^{(k+1)} + \frac{\delta h_{\mathbf{t}}((c)^{(k)})}{\delta \mathbf{t}}\delta\mathbf{z}^{(k+1)}$
$\delta\mathbf{c}^{(k)} = \frac{\delta h_{\mathbf{t}}(\mathbf{c}^{(k)})}{\delta \mathbf{c}}\delta\mathbf{z}^{(k+1)}$
$\delta\mathbf{B}^{(k)} = \delta\mathbf{B}^{(k+1)} + \delta\mathbf{c}^{(k)}$
$\delta\mathbf{H}^{(k)} = \delta\mathbf{H}^{(k+1)} + \delta\mathbf{c}^{(k)}\mathbf{z}^{(k)\mathrm{T}}$
$\delta\mathbf{z}^{(k)} = \mathbf{H}^{\mathrm{T}}\delta\mathbf{c}^{(k)}$
End
$\delta\mathbf{W} = \delta\mathbf{B}^{(0)}\mathbf{x}^{\mathrm{T}}$
Output: $\delta\mathbf{W}$, $\delta\mathbf{H}^{(0)}$, $\delta\mathbf{t}^{(0)}$

Algorithm 6 DNN-WISTA Forward propagation

Input: data $\mathbf{x}$, dictionary $\mathbf{D}$, proper parameters λ and α, network layer T.
Restriction: $\alpha >$ largest eigenvalue of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$
Initialization: $\mathbf{t}^{(0)} = \frac{\lambda}{\alpha}\mathbf{1}$, $\mathbf{z}^{(0)} = \mathbf{0}$, $\mathbf{H} = \mathbf{I} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}\mathbf{D}$, $\mathbf{W} = \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}$, $\mathbf{B} = \mathbf{W}\mathbf{x}$.
For $k = 1$ to T
$\mathbf{c}^{(k-1)} = \mathbf{B} + \mathbf{H}\mathbf{z}^{(k-1)}$
$\mathbf{z}^{(k)} = \pi_{\mathbf{t}^{(k-1)}}(\mathbf{c}^{(k-1)})$
$\mathbf{t}^{(k)} = \frac{\lambda}{\alpha}|\mathbf{z}^{(k)}|^{p-1}$
End
Output: $\mathbf{C} = \{\mathbf{c}^{(0)}, ..., \mathbf{c}^{(T-1)}\}$, $\mathbf{Z} = \{\mathbf{z}^{(0)}, ..., \mathbf{z}^{(T)}\}$

Algorithm 7 DNN-WISTA Back propagation

Input: $\mathbf{x}$, $\mathbf{D}$, $\delta\mathbf{z}^{(T)}$, $\mathbf{Z}$, $\mathbf{C}$, $\mathbf{B}$, $\mathbf{H}$, $\mathbf{T}$, λ and α.
Initialization: $\delta\mathbf{B}^{(T)} = \mathbf{0}$, $\delta\mathbf{H}^{(T)} = \mathbf{0}$.
For $k = T - 1$ down to 0
$\delta\mathbf{t}^{(k)} = \frac{\delta\pi_{\mathbf{t}^{(k)}}((c)^{(k)})}{\delta \mathbf{t}}\delta\mathbf{z}^{(k+1)}$
$\delta\mathbf{c}^{(k)} = \frac{\delta\pi_{\mathbf{t}^{(k)}}(\mathbf{c}^{(k)})}{\delta \mathbf{c}}\delta\mathbf{z}^{(k+1)}$
$\delta\mathbf{B}^{(k)} = \delta\mathbf{B}^{(k+1)} + \delta\mathbf{c}^{(k)}$
$\delta\mathbf{H}^{(k)} = \delta\mathbf{H}^{(k+1)} + \delta\mathbf{c}^{(k)}\mathbf{z}^{(k)\mathrm{T}}$
$\delta\mathbf{z}^{(k)} = \mathbf{H}^{\mathrm{T}}\delta\mathbf{c}^{(k)} + \frac{\lambda(p-1)|\mathbf{z}^{(k)}|^{p-2}\mathrm{sign}(\mathbf{z}^{(k)})}{\alpha}\delta\mathbf{t}^{(k)}$
End
$\delta\mathbf{W} = \delta\mathbf{B}^{(0)}\mathbf{x}^{\mathrm{T}}$
Output: $\delta\mathbf{W}$, $\delta\mathbf{H}^{(0)}$, $\delta\mathbf{t}^{(0)}$

C. DNN-STRUCTURED WISTA

Using the schedule in the previous section, by unfolding the iterations of WISTA from Algorithm 3, we can also construct a neural network structure with linear transformations and a simple nonlinear transformation in each layer, as shown in Fig. 3(b).

In this network, $\mathbf{W}$, $\mathbf{H}$, and $\mathbf{t}$ are targeted trainable parameters. After the forward propagation with determined network layer T, $\mathbf{C}$ and $\mathbf{Z}$ of each layer in the WISTA network are saved for the back-propagation parameter training. The gradient-based parameter learning schedules of the two networks are shown in Algorithm 7. The learning procedure can be either adapted to supervised learning

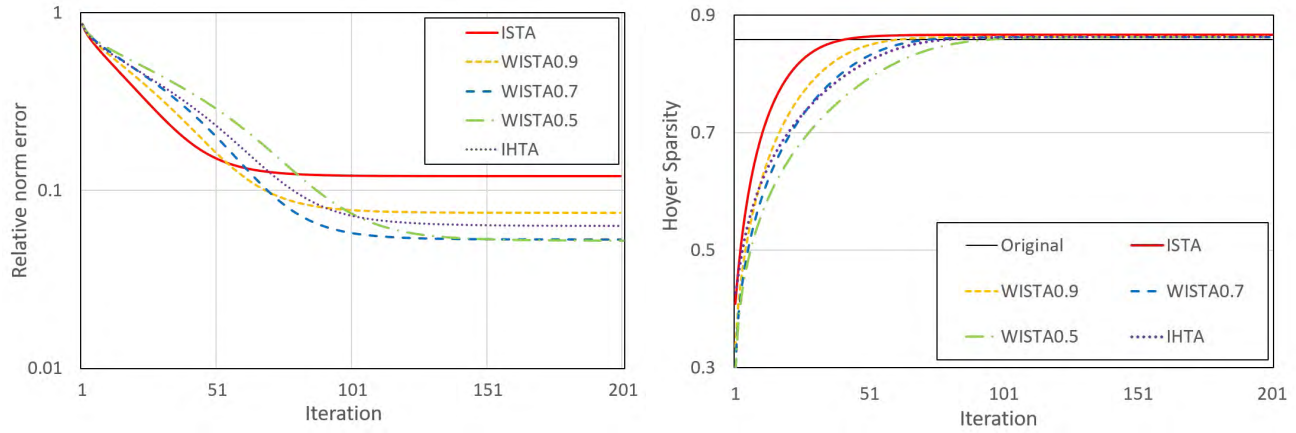


FIGURE 4. Average sparse representation errors and Hoyer sparsity convergence graph of different sparse coding algorithms.

by defining $\delta \mathbf{z}^{(T)}$ as equation (11) or unsupervised learning referring to equation (13).

IV. EXPERIMENTS

A. SYNTHETIC DATA EXPERIMENTS

In this section, we present our experiment results of the proposed algorithms in finding the ground truth sparse representations. Synthetically generated data were used throughout the experiments, which were built by randomly generated ground true dictionaries. All experiments were performed with Matlab R2016a, and the programs were run on a PC with a 3.4 GHz Intel core and 64 G of RAM.

There are two main parts that we would like to show in synthetic data experiments: comparison of performances among the ISTA, IHTA and WISTA and the performances of different DNN-SC algorithms. In the synthetic data simulation experiments, we first formed the dictionary $\mathbf{D}_{\text{orig}}$, which is sized as a 250×500 matrix, generated by randomly drawing value from a normal distribution $N(0, 1)$ and finally column-normalized. In every sparse representation vector, non-zero values were set in random positions determined by the Bernoulli distribution of possibility 0.05. The random values were selected from the normal distribution $N(0, 1)$. There were 100 ground truth-generated sparse representation vectors $\mathbf{Z}_{\text{orig}} = (\mathbf{z}_{\text{orig}1}, \ldots, \mathbf{z}_{\text{orig}100})$ in the experiments. Correspondingly, data set $\mathbf{X}_{\text{orig}}$ had 100 samples, which were generated by $\mathbf{D}_{\text{orig}}$ and $\mathbf{Z}_{\text{orig}}$ based on equation (2), The noise vectors $\mathbf{n}$ were added based on Gaussian random entries with 20 dB signal-to-noise ratios (SNR).

1) PERFORMANCES OF SPARSE CODING ALGORITHMS

The sparsity measurement used the Hoyer measure [36] based on the relationship between the l_1-norm and the l_2-norm, which can provide a well-defined sparsity. Hoyer sparsity measure is formulated as follows:

$$\text{Hoyer sparsity}(\mathbf{z}) = \frac{\sqrt{n} - (\sum |z_i|)/\sqrt{\sum z_i^2}}{\sqrt{n} - 1}, \qquad (14)$$

where n is the dimension of $\mathbf{z}$; when the value of the equation is closer to 1, the $\mathbf{z}$ vector approaches a sparse vector.

Fig. 4 and Fig. 5 show the ability of recovering accurate sparse representations using the three sparse coding algorithms. Three p values are used in our proposed algorithm WISTA, which is referred by the number after WISTA. The sparse representation error use the relative norm error

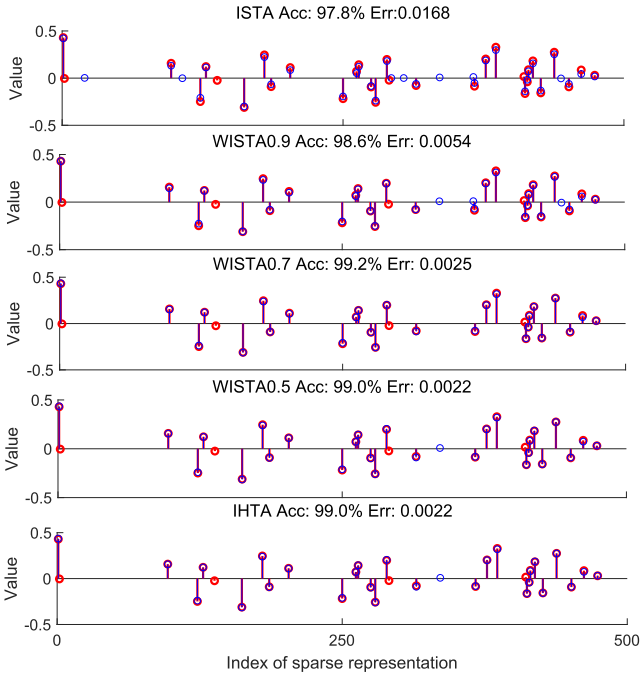


FIGURE 5. Index versus value of the recovered sparse representation (blue) compared with the original one (red) of ISTA, IHTA, WISTA0.9, WISTA0.7, and WITA0.5 (from top to bottom).

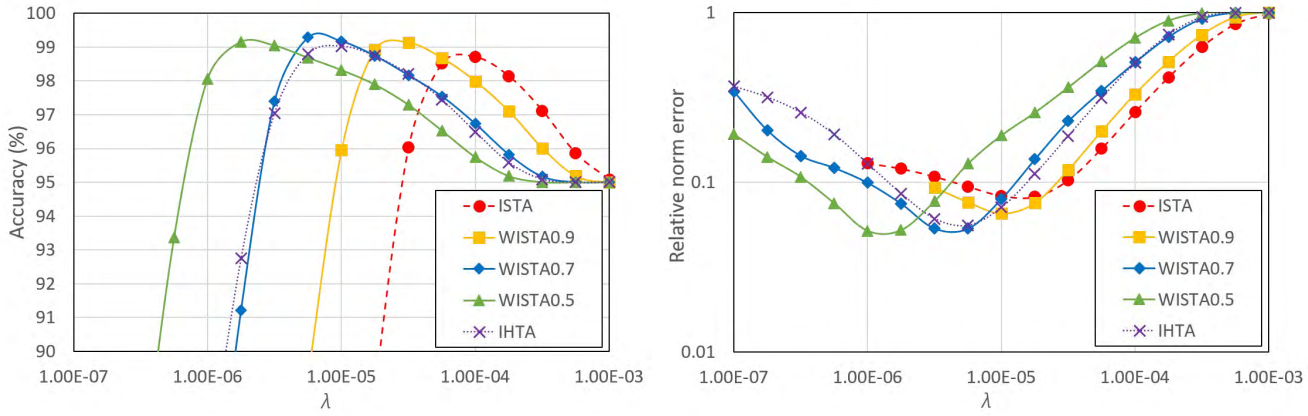


FIGURE 6. Average sparse representation accuracies and relative norm errors of different algorithms in a range of λ.

compared to $\mathbf{Z}_{\text{orig}}$, which is defined in equation (15).

$$\text{Relative norm error}(\mathbf{z}_i) = \frac{\|\mathbf{z}_{\text{orig}i} - \mathbf{z}_i\|_2^2}{\|\mathbf{z}_{\text{orig}i}\|_2^2} \tag{15}$$

Fig. 4 shows that all algorithms can achieve convergence results with small error and high sparsity in reasonable iterations. ISTA tends to converge most rapidly with a larger relative norm error than the other algorithms. With decreasing p, WISTA converges slower and result in a smaller relative error. When $p = 0.7$, WISTA surpasses IHTA in both converge speed and relative error. Fig. 5 shows that all algorithms can generally find accurate and sparse representations but ignore several small values compared to the ground true representation (red). ISTA has the largest amount of small value mistakes, so it has a larger relative error than the other algorithms. For WISTA, the number of support mistakes, where the blue points do not overlap with the red ones, tend to diminish when p decreases from 0.9 to 0.7; one small mistake in support reappears in WISTA0.5, which indicates that an appropriate lower p value can help finding sparser and more accurate results.

Fig. 6 shows the performance variation in a range of λ values. The sparse representation accuracy is defined in equations (16) and (17), which can indicate how accurate the sparse coding algorithms can recover the positions of those non-zero values.

$$S = \text{Support}\{\mathbf{z}\} \tag{16}$$

$$\text{Accuracy}(\mathbf{z}) = \frac{|S_{orig} \cap S|}{n} \times 100\% \tag{17}$$

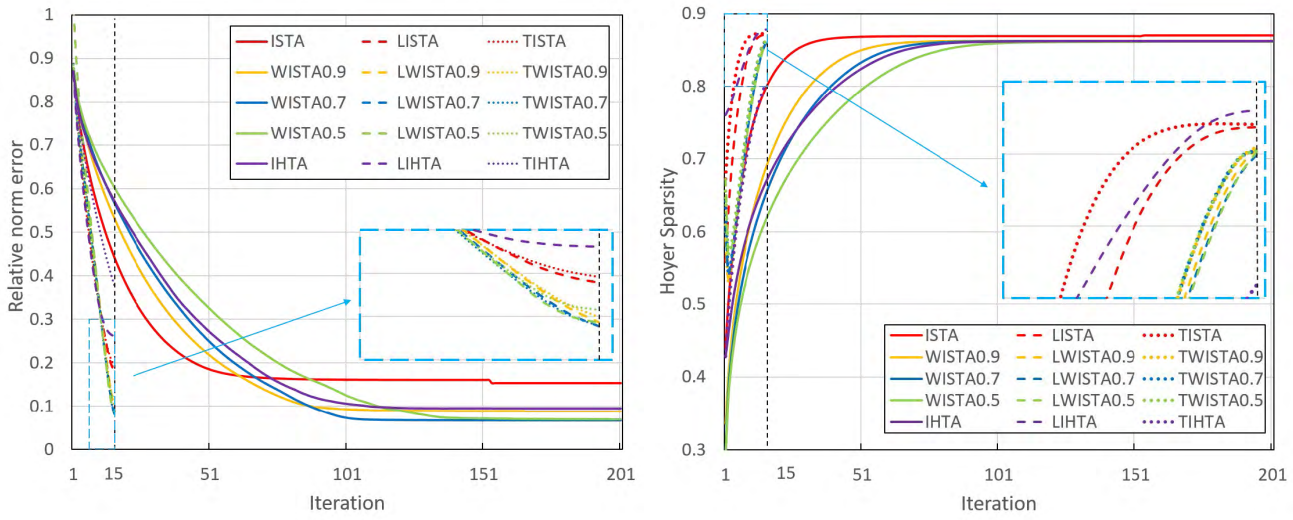
The definition of support is a set containing information of zero and non-zero positions in $\mathbf{z}$. Therefore, the accuracy is the percentage of how a given $\mathbf{z}$ meets the ground truth considering whether the values are zero. In the two figures, different algorithms have different ranges of effective λ. While deciding the optimal λ regions, we find that the λ value with the smallest relative norm error is always smaller than the λ with highest accuracy for all algorithms. In detail, ISTA has the worst performance in dislocating the optimal regions. The accuracy is smaller than 90% when ISTA reaches the lowest relative norm error, which results from the those support mistakes with small values in Fig. 5. Again, all other algorithms tend to perform better than ISTA in finding more accurate sparse representations. WISTA surpasses IHTA in both relative norm error and accuracy when $p = 0.7$ and 0.5. With decreasing p, WISTA tends to achieve smaller relative errors, whereas $p = 0.7$ likely results in the highest accuracy.

2) PERFORMANCES OF DNN-STRUCTURED SPARSE CODING ALGORITHMS

In the synthetic data experiment of DNN-SC algorithms, the DNN structures with various layers are used in different algorithms to show the performance of the combination of the sparse representation and DNN.

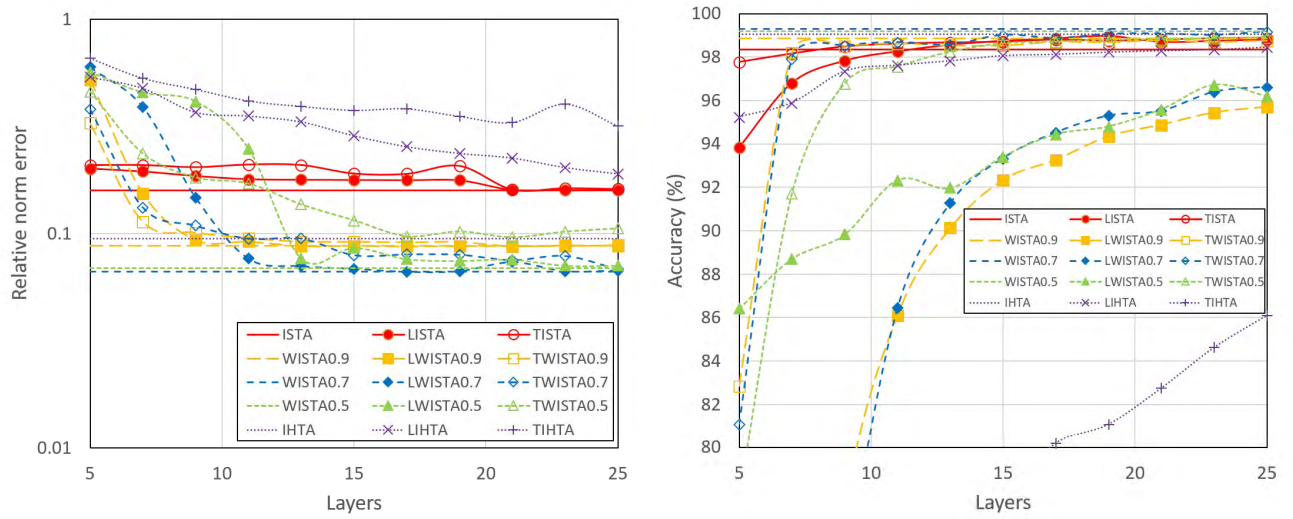
Fig. 7 shows the performances of different original sparse coding algorithms and their DNN-structured versions when there are 15 layers. The initial letter 'L' refers to supervised learned DNN, and the initial letter 'T' refers to unsupervised learned DNN. In general, except the DNN-structured IHTA, all other DNN-structured algorithms can reach similar performances to the converged results of their original sparse coding algorithms, which indicates that those learned DNN can accelerate 5-10 times in this experiment. Similar to the previous sparse coding experiment results, DNN-structured WISTA can perform better than DNN-structured ISTA when there are 15 layers.

Fig. 8 shows the result convergence performances of different original sparse coding algorithms and their DNN-structured versions in a range of DNN layers. In general, we find four points. 1) Except DNN-structured IHTA, all DNN-structured algorithms can nearly reach converged performances of their original algorithms when there are more than 13 layers in this simulation; 2) the performances of the DNN-structured algorithms tend to improve with the



(a) (b)

FIGURE 7. Comparison between 15-layer DNN-SC algorithms and the original sparse coding algorithms in terms of the average sparse representation relative norm errors and Hoyer sparsity.



(a) (b)

FIGURE 8. Comparison between the DNN-SC algorithms and the converged results of their original algorithms in terms of the average sparse representation relative norm errors and accuracy in a range of number of layers.

increase in number of layers; 3) DNN-structured WISTA can perform better than DNN-structured ISTA; and 4) the supervised DNN tends to obtain better relative norm error than the corresponding unsupervised ones.

Specifically, LISTA and TISTA have the best robustness against the decrease in number of layer in this data simulation. Through the selected range of number of layer, both LISTA and TISTA can reach similar relative norm errors compared to the converged ISTA, and LISTA and TISTA have better accuracy than ISTA when there are more than 15 layers. For WISTA, decreasing p in the DNN-structured WISTA algorithms tends to increase their minimum layer requirement to have similar performances to the converged WISTA, which may indicate that a smaller p value increases the difficulty to learn an appropriate network. Although supervised WISTAs have equivalent or better relative norm error than the unsupervised ones, the unsupervised versions have obvious advantages in finding more accurate positions in sparse representations, which implies that the supervised DNN can hardly avoid small support mistakes in sparse representations, which do not greatly affect the relative norm error but reduce the position accuracy.

B. GRAPHIC DENOISING EXPERIMENTS

In this section, we present the performance of our proposed algorithms with real-world data. There are two main parts. First, we want to show the performance of the algorithms in the image-denoising task. Then, we present the potential of

FIGURE 9. Image-denoising result: (a) Original image; (b) Noised: 20.17 dB; (c) ISTA: 29.13 dB; (d) WISTA0.9: 29.88 dB; (e) WISTA0.7: 30.79 dB; (f) WISTA0.5: 31.01 dB; (g) IHTA: 31.00 dB; and (h) OMP: 30.93 dB (from top left to bottom right)

applying DNN-SC algorithms in real-time video-denoising tasks.

1) IMAGE-DENOISING EXPERIMENTS

To present the denoising performance, we applied the aforementioned algorithms in image denoising in comparison with sparse coding algorithm OMP [3], [9], which performs well in this task. The details of the image-denoising experiment are described below. We selected the image named 'Lena' as the target, which is a 512 × 512-pixel gray-scaled portrait photograph. Random white noise was added to the image to generate a noised image of Peak signal-to-noise ratio (PSNR) 20.17 dB. Referring to the signal model in equation (2), we generate a 144×256 overcomplete Discrete Cosine Transform (DCT) distributed dictionary for the denoising task. The image was separated into 12 × 12-pixel small patches with an interval of 2 between patches to form the input data set $\mathbf{X} \in \mathbb{R}^{64\times 62009}$. All DNN-SC algorithms use 10-layer network structures in this experiment. The denoising results of the original sparse coding algorithms and their DNN-structured versions are presented below.

TABLE 2 and figure 9 show that all sparse coding algorithms can recover the noised image from 20.17 dB to approximately 30 dB, whereas all iterative shrinkage sparse coding algorithms tend to have several times higher computation time than OMP. In detail, IHTA and WISTA0.5 obtain the highest PSNR among these algorithms with PSNR over 31 dB. In this image-denoising task, we observe an obvious increase in PSNR and denoising time when p decreases among all iterative shrinkage sparse coding algorithms, which may indicate that a smaller p value is more suitable for the overcomplete sparse denoising model. To resolve the excessive computation cost of the iterative shrinkage sparse coding algorithms, the DNN-SC algorithm can be a solution that the learned DNN has shown 5-10 times acceleration in the previous synthetic data simulation.

TABLE 2. Denoising results of the sparse coding algorithms from a 20.17-dB noised image

Algorithms	PSNR (dB)	Denoising time (s)
ISTA	29.13	187.89
WISTA0.9	29.88	260.74
WISTA0.7	30.79	335.24
WISTA0.5	31.01	339.84
IHTA	31.06	445.65
OMP	30.93	77.94

TABLE 3. Denoising results of DNN-SC algorithms from a 20.17-dB noised image.

Algorithms	PSNR (dB)	Denoising time (s)	Learning time (s)
LISTA	29.71	2.52	0.05
TISTA	29.37	2.50	0.02
LWISTA0.9	29.89	5.82	0.03
TWISTA0.9	29.69	6.01	0.03
LWISTA0.7	30.45	5.90	0.03
TWISTA0.7	30.47	5.79	0.03
LWISTA0.5	30.68	5.84	0.03
TWISTA0.5	30.76	5.88	0.03
LIHTA	30.82	9.17	0.16
TIHTA	30.80	9.26	0.07

First, TABLE 3 and Fig. 10 show that all DNN-SC algorithms can recover the noised image from 20.17 dB to approximately 30 dB, which is the same level of their original sparse

FIGURE 10. Image-denoising result with DNN-SC algorithms: (a) LISTA: 29.71 dB; (b) TISTA: 29.37 dB; (c) LWISTA0.9: 29.89 dB; (d) TWISTA0.9: 29.69 dB; (e) LWISTA0.7: 30.45 dB; (f) TWISTA0.7: 30.47 dB; (g) LWISTA0.5: 30.68 dB; (h) TWISTA0.5: 30.76 dB; (i) LIHTA: 30.82 dB; and (j) TIHTA: 30.80 dB (from top left to bottom right).

coding algorithms. Furthermore, the learned DNNs can accelerate the denoising procedure, the sum of denoising and DNN learning time remains at least 45 times faster than the denoising time cost of their corresponding sparse coding algorithms. In detail, the unsupervised learned DNNs have similar PSNR values to their supervised versions, which proves that unsupervised learning can function in the graphic denoising task. Among all DNNs, DNN-structured WISTA and IHTA are better than the others: the PSNRs of the denoised image using their original sparse coding algorithms are approximately 0.25 dB slightly better, but the results remain reasonably good because DNN-structured WISTA0.5 can be over 10 times faster than OMP.

2) VIDEO-DENOISING EXPERIMENTS

In this section, we propose a procedure to apply the DNN-SC algorithm to real-time video denoising. The details of the video-denoising experiment are described below. We selected two videos from the dataset created by Gygli *et al.* [37] to test the denoising performance. The first video, which is named 'Fire Domino', is a 360 × 480-pixel gray-scaled video captured by a fixed camera; 'Fire Domino' is composed of 1612 frames at a rate of 25 frames/s (FPS). The second video, which is named 'Statue of Liberty', is a 360×480-pixel gray-scaled video captured by people in daily life; 'Statue of Liberty' has 1500 frames in total at a rate of 25 FPS. Random white noise was added to the two videos to generate noised videos with PSNR of approximately 20 dB. Referring to the signal model in equation (2), we generated a 225 × 256 overcomplete DCT distributed dictionary for this video-denoising task. Each video frame was separated into 15 × 15-pixel small patches with an interval of 10 between patches to form the input data set $\mathbf{X} \in \mathbb{R}^{225\times 1645}$. All DNN-SC algorithms used 4-layer network structures in this experiment. Video streaming was input into the DNN frame by frame, which implies that the DNNs learn from one frame and finish denoising before processing the next frame. The denoising results of unsupervised DNN-SC algorithms are presented below.

TABLE 4. Average denoising results of the DNN-SC algorithms from the first noised video 'Fire Domino' with an initial PSNR of 20.17 ± 0.02 dB

Algorithms	PSNR(dB)	Denoising time(s)	Learning time(s)
TISTA	30.35 ± 1.06	0.015 ± 0.009	0.004 ± 0.001
TWISTA0.9	31.15 ± 1.23	0.036 ± 0.021	0.004 ± 0.001
TWISTA0.7	31.74 ± 1.25	0.036 ± 0.021	0.004 ± 0.001
TWISTA0.5	31.83 ± 1.30	0.036 ± 0.021	0.004 ± 0.001
TIHTA	31.45 ± 1.06	0.071 ± 0.043	0.014 ± 0.005

TABLE 4, TABLE 5, Fig. 11 and Fig. 12 show that all unsupervised DNN-SC algorithms can successfully recover noised video streaming from 20 dB to approximately 30 dB with reasonable fast denoising time. More importantly, both TISTA and TWISTA can restrict the sum of denoising and DNN learning time to 0.04 s/frame, which implies that these two DNN-SC algorithms can conduct real-time video denoising for a 25-FPS 360 × 480-pixel gray-scaled video. In detail, we observe that TISTA continues being the fastest algorithm in processing, but its PSNR is relatively the worst. TIHTA achieves the highest PSNR among these algorithms, but its processing time is nearly twice that of TWISTA. TWISTA0.5 is the best algorithm, which restrains the processing time in the frame internal of a 25-FPS video. Furthermore, the denoising time of the video with identical resolution may vary depending on the video; thus, the average denoising time of different algorithms for the second video is approximately 75% compared to the time cost for the first video.

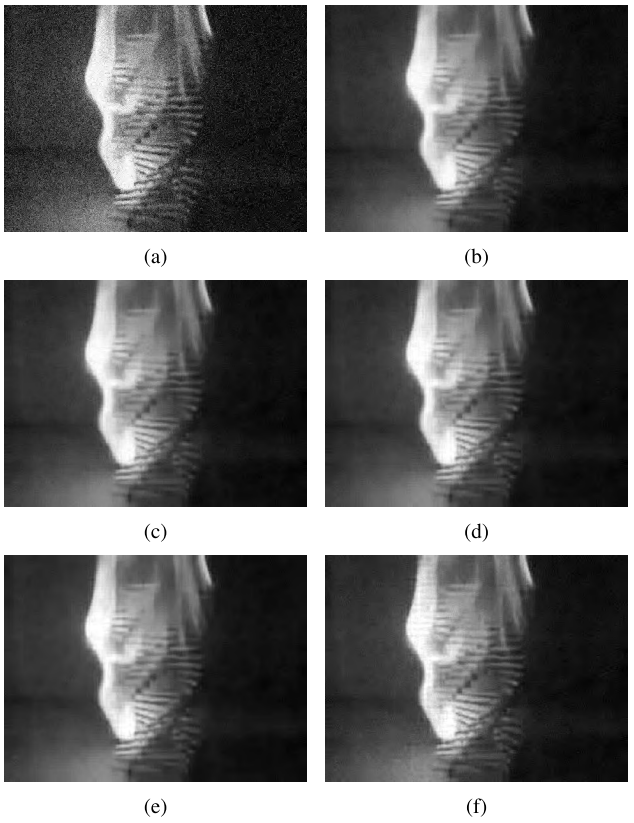


FIGURE 11. Denoising results of one frame in the video 'Fire Domino' from the initial PSNR of 20.19dB. (a) Noised:20.19dB. (b) TISTA:31.35dB. (c) TWISTA0.9:32.00dB. (d) TWISTA0.7:32.71dB. (e) TWISTA0.5:32.89dB. (f) TIHTA:32.44dB.

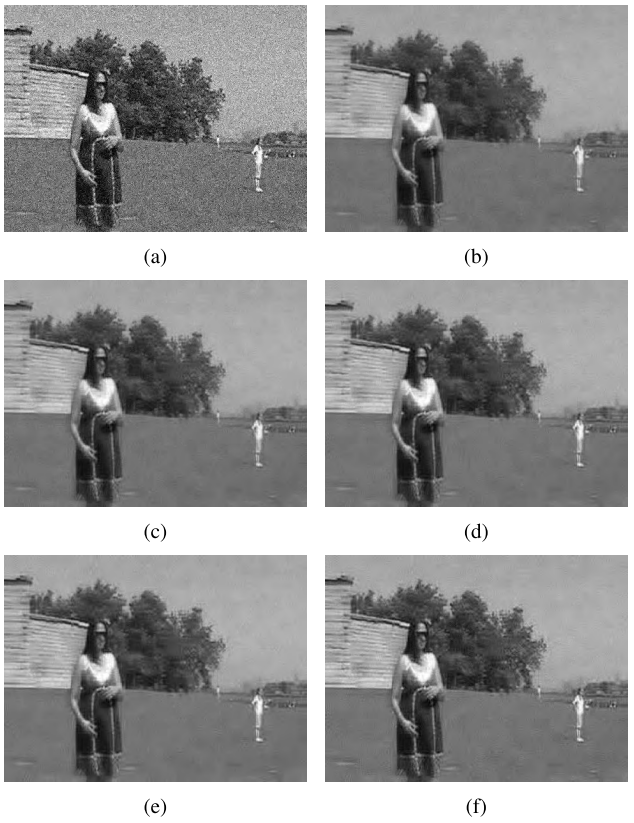


FIGURE 12. Denoising results of one frame in the video 'Statue of Liberty' from the initial PSNR of 20.17dB. (a) Noised:20.19dB. (b) TISTA:31.35dB. (c) TWISTA0.9:32.00dB. (d) TWISTA0.7:32.71dB. (e) TWISTA0.5:32.89dB. (f) TIHTA:32.44dB.

TABLE 5. Average denoising results of DNN-SC algorithms for the noised video 'Statue of Liberty' with the initial PSNR of 20.17 ± 0.01dB

Algorithms	PSNR (dB)	Denoising time (s)	Learning time (s)
TISTA	28.70 ± 1.53	0.012 ± 0.006	0.004 ± 0.001
TWISTA0.9	29.12 ± 1.58	0.029 ± 0.013	0.004 ± 0.001
TWISTA0.7	29.38 ± 1.63	0.028 ± 0.013	0.004 ± 0.001
TWISTA0.5	29.47 ± 1.71	0.028 ± 0.013	0.004 ± 0.002
TIHTA	29.81 ± 1.52	0.055 ± 0.025	0.008 ± 0.003

C. DISCUSSION

Throughout the conducted experiments, there are two points for discussion.

1) DNN-IHTA, which fails in giving a close approximation of the converged IHTA in synthetic data experiments, can function well in graphic denoising experiments.
   In Fig. 8, DNN-IHTA can hardly approach the results of IHTA in synthetic data experiments for all tested DNN layers, but TABLE 3, TABLE 4 and TABLE 5 show that TIHTA can work well and achieve similar results to the converged results of IHTA. The key difference is the data set. In the synthetic data experiments, data were randomly generated and obey the normal distribution $N(0, 1)$, but in denoising experiments, the data were extracted small patches from an image or a video, i.e., the data may have certain continuity and dependence among the data columns. Fig. 8 also shows that DNN-structured WISTA requires more DNN layers to achieve the converged results of WISTA as p decreases, whereas DNN-structured ISTA does not have this problem. One possibility is that the problem is caused by the nonconvex feature in IHTA and WISTA, which makes the back-propagation failed in finding the global optimal in random meaningless data set.
2) In both image and video denoising, the DNN learning time is small.
   TABLE 3, TABLE 4 and TABLE 5 show that all DNN-SC algorithms only require tens of millisecond to complete their learning procedure, which ensures that thee DNN update is sufficiently fast for online processing. The reason is that the DNNs in these experiments only use a small number of patches from the image until the learning procedure converges, which is not possible for the learning procedure of synthetic data experiments. In the synthetic data experiments, the DNN learning should pass several epochs of all input data to make the learned DNN converge and effective. The possible answer may again be the difference in data set. All image patches have identical standard white noise levels on a continuous and repeating graphic signal,

whereas the synthetic data put white noises on random generated signals, which are also the Gaussian distribution. Thus, the denoising procedure for each patch may have certain similarities that the learned DNN from a small number of patches can stand for the entire image.

V. CONCLUSION

In this paper, we have proposed two DNN-SC algorithms. The first algorithm applies deep learning approaches to WISTA, which is a modified sparse coding algorithm of ISTA proposed in this paper. WISTA considers to approximate the l_p norm sparse coding problem by joining the information of the sparse representation from the previous iteration. The 'weighted' idea enables one to enjoy the advantages of the l_p norm sparse constraint while maintaining the convex optimization model. The second approach combines IHTA [14] with deep learning, which is an $l_{0.5}$ norm sparse coding algorithm. We state the differences between two DNN learning schedules for DNN-SC algorithms: supervised and unsupervised. The benefit of unsupervised DNN learning is that it does not require input signals associated with labels, which enables one to apply DNN-SC algorithms to image denoising, video denoising and other applications that lack paired training samples.

The synthetic data experiments show that WISTA can outperform both ISTA and IHTA in terms of the relative norm error and accuracy. In addition, WISTA can retain the advantages of supervised and unsupervised DNN versions. We also find that unsupervised DNNs can achieve similar performance to their corresponding supervised ones. However, DNN-IHTA can hardly learn the appropriate parameters to yield a close approximation of the converged IHTA in synthetic data. Furthermore, it is difficult for DNN-WISTA to train the parameters when there are few layers, but all DNN-WISTAs can function well with at least 15 layers, which remains reasonable.

Then, we have applied the proposed algorithms to image and video denoising, which benefit from the unsupervised learning procedure and fast DNN learning time for graphic processing. The denoising results show that the DNN-SC algorithms can accelerate the denoising procedures at least 45 times while maintaining reasonably good performances from 20 dB to approximately 30 dB. Then, we conducted denoising experiments on two videos. Using TISTA and TWISTA, the total processing time for each frame can be restricted to 0.04 s/frame, which indicates that TISTA and our proposed TWISTA can be applied in real-time video denoising for a 25-FPS video with good denoising results. Although we only conduct experiments for 25-FPS 360 × 480-pixel gray-scaled videos, the future work may extend to higher FPS, higher resolution and colored videos.

REFERENCES

[1] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 5, pp. 2197–2202, 2003. [Online]. Available: http://www.pnas.org/content/100/5/2197

[2] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. New York, NY, USA: Springer-Verlag, 2010.

[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[4] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[5] X. Hu, F. Heide, Q. Dai, and G. Wetzstein, "Convolutional sparse coding for RGB+NIR imaging," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1611–1625, Apr. 2018.

[6] M. Hanif, A. Tonazzini, P. Savino, and E. Salerno, "Sparse representation based inpainting for the restoration of document images affected by bleed-through," *Multidiscipl. Digit. Publishing Inst. Proc.*, vol. 2, no. 2, p. 93, 2018. [Online]. Available: http://www.mdpi.com/2504-3900/2/2/93

[7] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[8] J. Jiang, J. Ma, C. Chen, X. Jiang, and Z. Wang, "Noise robust face image super-resolution through smooth sparse representation," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3991–4002, Nov. 2017.

[9] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[10] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[11] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001, doi: 10.1137/S003614450037906X.

[12] I. Bayram and I. W. Selesnick, "A subband adaptive iterative shrinkage/thresholding algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1131–1143, Mar. 2010.

[13] M. Guerquin-Kern, M. Häberlin, K. P. Pruessmann, and M. Unser, "A fast wavelet-based reconstruction method for magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 30, no. 9, pp. 1649–1660, Sep. 2011.

[14] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$L_{1/2}$ regularization: A thresholding representation theory and a fast solver," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1013–1027, Jul. 2012.

[15] M. A. T. Figueiredo and R. D. Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Sep. 2005, pp. II-782–II-785.

[16] M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak, "Majorization–minimization algorithms for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2980–2991, Dec. 2007.

[17] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 877–905, 2008, doi: 10.1007/s00041-008-9045-x.

[18] H. Zhao, S. Ding, Y. Li, Z. Li, X. Li, and B. Tan, "Dictionary learning for sparse representation using weighted ℓ_1-norm," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2016, pp. 292–296.

[19] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 399–406. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104374

[20] U. S. Kamilov and H. Mansour, "Learning optimal nonlinearities for iterative thresholding algorithms," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 747–751, May 2016.

[21] D. Mahapatra, S. Mukherjee, and C. S. Seelamantula. (2017). "Deep sparse coding using optimized linear expansion of thresholds." [Online]. Available: http://arxiv.org/abs/1705.07290

[22] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Learning efficient sparse and low rank models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1821–1833, Sep. 2015.

[23] Z. Wang, Q. Ling, and T. Huang, "Learning deep ℓ_0 encoders," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2194–2200.

[24] M. Borgerding and P. Schniter. (2016). "Onsager-corrected deep networks for sparse linear inverse problems." [Online]. Available: http://arxiv.org/abs/1612.01183

[25] J. Domke, "Parameter learning with truncated message-passing," in *Proc. CVPR*, Jun. 2011, pp. 2937–2943.

[26] J. Domke, "Learning graphical model parameters with approximate marginal inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2454–2467, 2013.

[27] T. Moreau and J. Bruna. (Sep. 2016). "Understanding trainable sparse coding via matrix factorization." [Online]. Available: https://arxiv.org/abs/1609.00285
[28] L. Guo and C. Guo, "A deep sparse coding method for fine-grained visual categorization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 632–639.
[29] S. Zhang, J. Wang, X. Tao, Y. Gong, and N. Zheng, "Constructing deep sparse coding network for image classification," *Pattern Recognit.*, vol. 64, pp. 130–140, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320316303466
[30] X. Sun, N. M. Nasrabadi, and T. D. Tran. (2017). "Supervised multilayer sparse coding networks for image classification." [Online]. Available: http://arxiv.org/abs/1701.08349
[31] K. Zhang *et al.*, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.
[32] J. Zhang and B. Ghanem. (2017). "ISTA-Net: Iterative shrinkage-thresholding algorithm inspired deep network for image compressive sensing." [Online]. Available: http://arxiv.org/abs/1706.07929
[33] J.-H. R. Chang *et al.*, "One network to solve them all—Solving linear inverse problems using deep projection models," in *Proc. ICCV*, 2017, pp. 5889–5898. [Online]. Available: http://arxiv.org/abs/1703.09912
[34] J. Zeng, S. Lin, Y. Wang, and Z. Xu, "$L_{1/2}$ regularization: Convergence of iterative half thresholding algorithm," *IEEE Trans. Signal Process.*, vol. 62, no. 9, pp. 2317–2328, May 2014.
[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[36] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.
[37] M. Gygli, H. Grabner, H. Riemenschneider, and L. van Gool, "Creating summaries from user videos," in *Computer Vision—ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 505–520.

HAOLI ZHAO received the B.Sc. degree in nuclear physics from the College of Physical Science and Technology, Sichuan University, China, in 2012, and the M.Sc. degree in nuclear environmental science and technology from the Department of Material, The Sheffield University, Sheffield, U.K., in 2014. He is currently pursuing the Ph.D. degree with the Graduate Department of Computer and Information System, The University of Aizu, Japan. His current research interests focus in signal processing, sparse representation, and deep neural network in detail.

SHUXUE DING (M'04) received the M.Sc. degree in physics from the Dalian University of Technology, China, in 1988, and the Ph.D. degree in physics from the Tokyo Institute of Technology, Japan, in 1996.

From 1989 to 1991 and from 1991 to 1992, respectively, he was an Assistant Professor and an Associate Professor with the School of Physics and Optoelectronic Technology, Dalian University of Technology. From 1996 to 1998 and from 1998 to 2003, he was with Fujisoft-ABC Inc. and Clarion Co., Ltd, Japan, respectively. From 2003 to 2005, he was a Visiting Faculty, and from 2005 to 2010 he was an Associate Professor, with The School of Computer Science and Engineering, The University of Aizu, Japan, where he is currently a Full Professor. He is a member of ACM and IEICE.

Dr. Ding has engaged himself in research in a wide range of areas of mathematical and physical engineering, such as statistical signal processing, optimization, neural computation, bioelectromagnetism, and information sciences. In particular, he has devoted himself to compressive sensing and sparse representation, machine learning, brain-style information processing, blind source separation, and independent component analysis. He is also interested in speech and image processing, quantum computation and optimization, quantum information, and other physical theories of information.

XIANG LI received the Ph.D. degree from the School of Mechanical Engineering, Xi'an Jiaotong University, China, in 2015, where he studied sparse representation and its application to inverse problem of structural health monitoring (SHM). He has held post-doctoral research appointment at the Cognitive Science Laboratory, School of Computer Science and Engineering, The University of Aizu, Japan, where he conducted the study of compressive sensing, deep neural network and non-convex optimization in inverse problem of Internet of Things (IoT) and SHM. He is currently an Associate Professor with The School of Computer Science and Engineering, The University of Aizu. His main research interests include sparse representation, optimization, machine learning, information theory and their applications, especially for inverse problems.

HUAKUN HUANG received the B.Sc. and M.Sc. degrees from Guangzhou University, Guangzhou, China, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with The School of Computer Science and Engineering, The University of Aizu, Japan. His current research interests include signal processing, sparse representation, optimization, machine learning, and their applications, such as localization, image processing, and pattern recognition.