Stacked Convolutional Denoising Auto-Encoders for Feature Representation

Bo Du, *Senior Member, IEEE,* Wei Xiong, *Student Member, IEEE,* Jia Wu, *Member, IEEE,* Lefei Zhang, *Member, IEEE,* Liangpei Zhang, *Senior Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—Deep networks have achieved excellent performance in learning representation from visual data. However, the supervised deep models like convolutional neural network require large quantities of labeled data, which are very expensive to obtain. To solve this problem, this paper proposes an unsupervised deep network, called the stacked convolutional denoising auto-encoders, which can map images to hierarchical representations without any label information. The network, optimized by layer-wise training, is constructed by stacking layers of denoising auto-encoders in a convolutional way. In each layer, high dimensional feature maps are generated by convolving features of the lower layer with kernels learned by a denoising auto-encoder. The auto-encoder is trained on patches extracted from feature maps in the lower layer to learn robust feature detectors. To better train the large network, a layer-wise whitening technique is introduced into the model. Before each convolutional layer, a whitening layer is embedded to sphere the input data. By layers of mapping, raw images are transformed into high-level feature representations which would boost the performance of the subsequent support vector machine classifier. The proposed algorithm is evaluated by extensive experimentations and demonstrates superior classification performance to state-of-the-art unsupervised networks.**

***Index Terms*—Convolution, deep learning, denoising auto-encoders, unsupervised learning.**

Manuscript received September 7, 2015; revised December 6, 2015; accepted February 18, 2016. Date of publication March 16, 2016; date of current version March 15, 2017. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB719905, in part by the National Natural Science Foundation of China under Grant 61471274, 91338202, 41431175, U1536204, and 60473023, in part by the Natural Science Foundation of Hubei Province under Grant 2014CFB193, and in part by the Fundamental Research Funds for the Central Universities under Grant 2042014kf0239. This paper was recommended by Associate Editor J. Basak. *(Corresponding author: Lefei Zhang.)*

B. Du, W. Xiong, and L. Zhang are with the State Key Laboratory of Software Engineering, Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Computer, Collaborative Innovation Center of Geospatial Information Technology, Wuhan University, Wuhan 430072, China (e-mail: gunspace@163.com; wxiong@whu.edu.cn; zhanglefei@whu.edu.cn).

J. Wu and D. Tao are with the Centre for Quantum Computation and Intelligent Systems and the Department of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: jia.wu@uts.edu.au; dacheng.tao@gmail.com).

L. Zhang is with the Collaborative Innovation Center of Geospatial Technology, State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China (e-mail: zlp62@whu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2016.2536638

I. Introduction

FEATURE learning is an efficient way to discriminately represent image data [1]–[5]. Shallow feature learning algorithms such as bag of visual words [6] and spatial pyramid matching [7] can learn determinative features (e.g., edges and color) from images with large variation but belonging to similar categories. When stacking shallow models into deeper models in a proper way, more abstract features, such as contours, which are combinations of low level features following certain principles, are learned automatically [2], [8], [9]. One typical type of deep models is convolutional neural network [10]–[12] (CNN). It is a hierarchal model that outperforms most algorithms on visual recognition tasks. One property that makes CNN work well is its deep structure, which allows the model to learn layers of filters and transform the input data to good representation [13]–[15], boosting the performance of the subsequent classifier. The other property is convolution [16] and pooling [17] structures. The convolution structure shares weights and keeps the relative location of features, thus reserves spatial information of the input data. However, deep CNNs usually have a huge number of parameters to train, which requires a tremendous amount of labeled data and considerable expenditure of computing resource.

The other type of deep model is unsupervised deep networks, which attempt to learn deep representation from the data itself without knowing its label. One typical unsupervised feature learning algorithm is stacked auto-encoders (SAEs) [18]–[20]. It is a stack of the shallow auto-encoder model, which learns features by first encoding the vector-form input data then reconstructing it. Shin *et al.* [21] applied stacked sparse auto-encoders (SSAEs) to medical image recognition and made notable promote in recognition accuracy. Vincent *et al.* [22], [23] introduced a novel type of auto-encoder called the denoising auto-encoder (DAE), which corrupts the input with random noise at the training stage to make the algorithm robust to data with noise or large variation. And it stacks the DAEs into a deep unsupervised network to learn deep representation [23].

Though the SAEs are capable to learn hierarchical features from complicated visual data, they reshape the high dimension input data into vectors due to the network structures, discarding the inherent structures. To solve this problem, Masci *et al.* [24] proposed a kind of deep network called convolutional auto-encoders, which directly takes the 3-D image data as the input and trains the auto-encoder convolutionally. Adjacent convolutional auto-encoders are combined by the

convolution and pooling operations. The convolutional kernels are learned to convolve the input feature maps of each layer into more abstract features. Compared to SAEs, the hierarchical convolutional auto-encoders reserve more structural information.

Besides auto-encoders, other types of unsupervised deep networks with convolution structure have achieved sound performance for learning representation in visual tasks. Norouzi *et al.* [25] proposed stacked convolutional restricted Boltzmann machines (RBMs), which trains a convolutional RBM modified from the conventional RBM (CRBM) [26], [27] to include spatial locality as well as weight sharing and stacks the CRBM to construct deep models. Lee *et al.* [28] proposed convolutional deep belief network (CDBN), which replaces the RBM in each layer with CRBM, and uses convolution structure to combine the layers to construct hierarchical models [29], [30]. Compared to the traditional DBN [31], CDBN reserves information of local relevance and improves the capability of feature representation. With the similar ideas, Zeiler *et al.* [32], [33] proposed a deconvolutional network based on traditional sparse coding algorithm [34]. The deconvolutional network is based on convolutional decomposition of input data under a sparsity constraint. It is a modification of conventional sparse coding algorithm. Compared to sparse coding, the deconvolutional network can learn richer feature sets and build mid-level representations. Kavukcuoglu *et al.* [35] developed the stacked convolutional sparse coding to make the inference procedure faster for real-time applications.

While the convolutional deep models have been successfully applied to various areas, two problems prevent the further development of these algorithms. One problem is that the deep convolutional models are hard to train. When training the deep network convolutionally, such as CNN, CDBN, the large network is hard to optimize, as the optimization methods used in deep networks without convolution structure (we call them nonconvolution networks) such as SAEs and deep belief networks do not fit properly on the deep networks with convolution structures. Some efficient optimization and regularization techniques perform better on the nonconvolution models, like sparse constraint on the response of the auto-encoder. Though some regularization methods have been proposed to optimize the deep CNN, they are not proved to be suitable for the unsupervised networks.

The second problem comes from one of the intrinsic properties of a deep network—sensitiveness to the input. Large networks are sensitive to small perturbance of the input images, though they are capable to learn robust features. The network may be misled to misclassify an image by a certain imperceptible perturbation on the image [36]. So when there is noise in the input data, or we have to deal with image sets with large variation, features learned by current networks may not be robust.

For the first problem, we use patch-wise training to optimize the weights of an auto-encoder in place of convolutional training. We first extract patches from the 3-D input images or feature maps, then trains a basic auto-encoder without convolution operation to learn weights. In this way, the optimization methods like sparse regularization and whitening can be used efficiently to improve the performance of feature learning and obtain better weights. The weights are then reorganized to convolutional kernels. At the inference stage, we introduce the convolution structure to the model. The convolutional kernels are used to convolve the 3-D input data to more abstract 3-D feature maps, thus still reserves local relevance. We still call this structure the convolutional auto-encoder. Then the convolutional auto-encoders are stacked to learn high-level feature representation.

For the second problem, to wipe off the noise in the input data as possible, we adopt the theory and network framework of DAEs introduced in [22]. The DAE can automatically denoise the input images to learn robust features. It has been given thorough proof in theory and in practice [23], [37]. Our improvement on the second problem mainly lies on that we use DAEs to replace the conventional auto-encoders. At the inference stage, we stack the DAEs with convolution structure, then construct the proposed model—stacked convolutional denoising auto-encoder (SCDAE). Compared to the stacked DAEs, our model transforms the vector-form layers to high dimensional convolutional layers. The combination of DAE and convolution structure forces our model to learn more robust and abstract hierarchical features, which will help to improve our model's representation learning performance.

The proposed unsupervised deep network is optimized through layer-wise training. To learn better convolutional kernels for each layer, a whitening layer is embedded before each convolutional layer. We call it the layer-wise whitening optimization technique. Specifically, in the whitening layer, ZCA whitening technique [38] is utilized to sphere the input features, which can remove the correlations of features inside a local area to allow the DAEs to learn better weights. Then the processed data are sent to the convolutional layer. In addition, dropout [39] is utilized in the hidden layer of the auto-encoders to achieve model averaging.

In summary, the main contributions of this paper are that we propose an unsupervised deep network, the SCDAEs, which stacks DAEs in a convolutional way to generate a hierarchical model. For better training performance, the parameters of the DAE are optimized through patch-wise training. The SCDAE model can learn robust and abstract hierarchical feature representations from raw visual data in an unsupervised manner. To better optimize the large number of parameters, the network is trained with layer-wise whitening technique. Before each convolutional layer, a whitening layer is embedded to sphere the input feature maps.

The remainder of this paper is organized as follows. Related work is introduced in Section II and details of the SCDAE algorithm are provided in Section III. Experimental results and analysis on five popular datasets are presented in Section IV, followed by our main conclusions in Section V.

II. Related Work

A. Auto-Encoder

The conventional auto-encoder is a three-layer symmetrical neural network that constrains the output to be equal to the

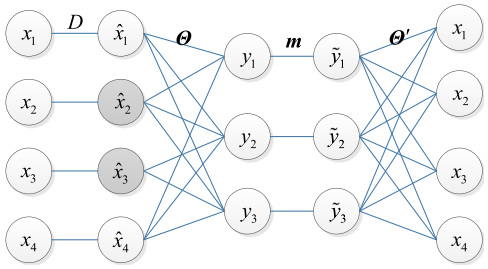


Fig. 1. Illustration of DAE with dropout. Neurons in gray denote the corrupted input neural units.

input. Given an input vector $\boldsymbol{x}$, the auto-encoder first maps $\boldsymbol{x}$ into a latent representation $\boldsymbol{y}$ through a nonlinear mapping $\boldsymbol{y} = f(\boldsymbol{\Theta}_1\boldsymbol{x} + \boldsymbol{\beta}_1)$, Where $\boldsymbol{\Theta}_1$ is a mapping matrix to be learned, and $\boldsymbol{\beta}_1$ is a bias vector that controls the activations of the neural units. The feature representation $\boldsymbol{y}$ is then mapped back to a reconstructed vector $\boldsymbol{z}$ via backward mapping $\boldsymbol{z} = g(\boldsymbol{\Theta}_2\boldsymbol{y} + \boldsymbol{\beta}_2)$. $\boldsymbol{z}$ is constrained to approximate the original input $\boldsymbol{x}$, i.e., $\boldsymbol{x} \approx \boldsymbol{z}$. The construction error is defined as the Euclidian distance between $\boldsymbol{x}$ and $\boldsymbol{z}$. The parameters of auto-encoder are optimized by minimizing the construct error. The latent feature $\boldsymbol{y}$ mapped from $\boldsymbol{x}$ by well learned parameters is the final representation of $\boldsymbol{x}$.

B. Dropout

Dropout is a technique applied to the fully connected layers to prevent overfitting. In each training iteration, each hidden unit in a layer is randomly omitted from the network with a certain probability, and in this way the hidden units do not rely on other hidden units to change their states. Dropout is a model averaging technique, since different networks are trained in different training iterations.

III. Stacked Convolutional Denoising Auto-Encoder

A. Denoising Auto-Encoder With Dropout

In the training stage of the proposed network, convolutional kernels of each layer are trained by DAE with dropout technique optimizing the large network. The DAE is a simple but effective variant of the basic auto-encoder. The main idea of this approach is to train an auto-encoder which could reconstruct the input data from a corrupted version that has been manually added with random noise. The optimized model is then capable to automatically denoise the input data and thus generates better feature representations for the subsequent classification tasks.

The structure of the DAE with dropout is demonstrated in Fig. 1. A DAE takes a vector $\boldsymbol{x} \in \mathrm{R}^d$ as the input and corrupts $\boldsymbol{x}$ into vector $\hat{\boldsymbol{x}}$ with a certain probability λ by means of a stochastic mapping

$$\hat{\boldsymbol{x}} \sim D\big(\hat{\boldsymbol{x}} \mid \boldsymbol{x}, \lambda\big). \tag{1}$$

D is a type of distribution determined by the original distribution of $\boldsymbol{x}$ and the type of random noise added to $\boldsymbol{x}$. Then $\hat{\boldsymbol{x}}$ is mapped to a latent vector representation $\boldsymbol{y}$ using a deterministic function φ

$$\boldsymbol{y} = \varphi\big(\boldsymbol{\Theta}\hat{\boldsymbol{x}} + \boldsymbol{\beta}\big). \tag{2}$$

When dropout technique is applied to the network to optimize the training process, neural units in the hidden layers are randomly omitted with a probability q. The representation $\boldsymbol{y}$ is then transformed into a dropped representation $\tilde{\boldsymbol{y}}$ by a scalar product with a masking vector $\boldsymbol{m}$. $\boldsymbol{m} \sim$ Bernoulli$(1 - q)$, "$\cdot$" denotes to scalar product

$$\tilde{\boldsymbol{y}} = \boldsymbol{m} \cdot \boldsymbol{y}. \tag{3}$$

Dropout technique is extremely helpful for the optimization of large neural networks. Since the network is updated iteratively, a unique network is trained in each iteration as a result of randomly dropping the neurons in the hidden layer. When the training process converges, the network gets an average representation of $2^{|\boldsymbol{m}|}$ networks, which greatly improves the subsequent classification performance.

The dropped hidden feature vector $\tilde{\boldsymbol{y}}$ is then reversely mapped to a final feature $\boldsymbol{z}$ used to reconstruct the original input $\boldsymbol{x}$ by another mapping function

$$\boldsymbol{z} = \varphi'\big(\boldsymbol{\Theta}'\tilde{\boldsymbol{y}} + \boldsymbol{\beta}'\big). \tag{4}$$

The expected result will be that $\boldsymbol{z}$ equals $\boldsymbol{x}$. So we define the constructing error using $\boldsymbol{z}$ to represent $\boldsymbol{x}$ as its Euclidean distance $\Gamma(\boldsymbol{x}, \boldsymbol{z}) = \|\boldsymbol{x} - \boldsymbol{z}\|_2^2$. By minimizing Γ, the optimized parameters is obtained

$$\boldsymbol{\Theta}_{\text{opt}}, \boldsymbol{\Theta}'_{\text{opt}}, \boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\beta}'_{\text{opt}} = \underset{\boldsymbol{\Theta}, \boldsymbol{\Theta}', \boldsymbol{\beta}, \boldsymbol{\beta}'}{\arg\min}\ \Gamma(\boldsymbol{x}, \boldsymbol{z}). \tag{5}$$

In our model, to make the learned features more discriminative, sparse constraint is included to the hidden representation [40], [41]. $\tilde{\boldsymbol{y}}$ is trained to be sparse, and the average value of elements in $\tilde{\boldsymbol{y}}$ is expected to approximate zero. Then the objective function is revised to

$$\boldsymbol{\Theta}_{\text{opt}}, \boldsymbol{\Theta}'_{\text{opt}}, \boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\beta}'_{\text{opt}} = \underset{\boldsymbol{\Theta}, \boldsymbol{\Theta}', \boldsymbol{\beta}, \boldsymbol{\beta}'}{\arg\min}\ \Gamma(\boldsymbol{x}, \boldsymbol{z}) + \text{sparse}(\tilde{\boldsymbol{y}}) \tag{6}$$

where sparse() represents a type of sparse constraint, which is expressed with KL [42] distance in our model.

Three basic types of noise are commonly utilized to corrupt the input of the DAE, and the zero masking noise [22] is employed in the proposed model.

B. Overall Architecture

The proposed SCDAE is an unsupervised deep network that stacks well-designed DAEs in a convolutional way to generate high-level feature representation. The overall architecture is optimized by layer-wise training.

Fig. 2 illustrates the whole structure of the proposed method. An input image is first sphered by a whitening layer and sent to the first convolutional layer to be convolved into feature maps with filters learned by the DAE of this convolutional layer and sub-sampled by pooling operation to get smaller feature maps. Then the feature maps are further processed by the next whitening layer and passed to the next convolutional layer. By layers of mapping, final convolutional feature maps are

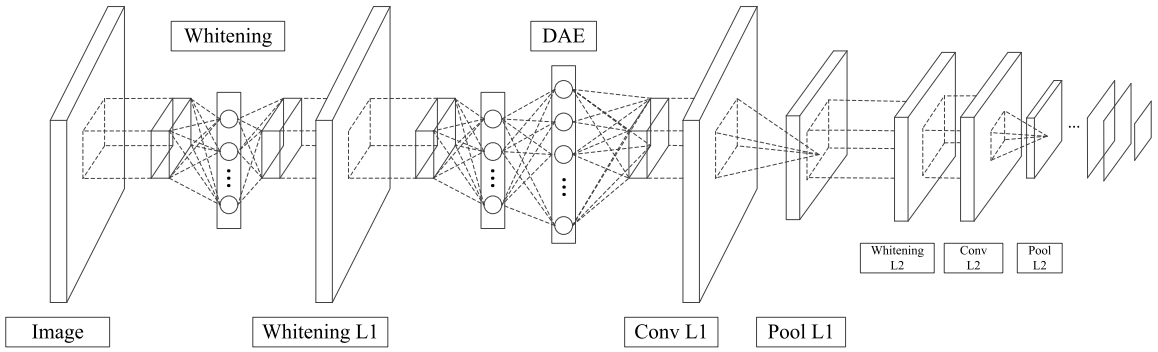


Fig. 2. Illustration of the proposed SCDAE.

generated, which are reshaped into discriminative input feature vectors of the subsequent support vector machine (SVM) classifier.

The training samples of the DAE in a convolutional layer are vector-form patches extracted from the feature maps of the last convolutional layer. Dropout is used in DAE of each layer.

C. Convolutional Layer

The input of each convolutional layer is 3-D feature maps, a mid-level representation of the input image. In each convolutional layer, the convolutional DAE transforms the input features into more robust and abstract feature maps through the learned denoising filters.

Given that $S^{(l)} \in R^{I_1^i \times I_2^i \times C^i}$ is the whole input feature maps of layer l, where C^i is the number of channels, I_1^i and I_2^i denotes the height and width of each input feature map. To learn latent features from the input, adequate 2-D patches $\boldsymbol{P} \in R^{(K_1 \times K_2 \times C^i) \times N}$ are first extracted from $S^{(l)}$ to compose the training set of a DAE, where $K_1 \times K_2 \times C^i$ denotes the convolutional kernel size, i.e., each kernel is a 3-D array with size $K_1 \times K_2 \times C^i$. N denotes the number of patches. Each sample is reshaped to $(K_1 \times K_2 \times C^i) \times 1$ vector for the convenience of training the DAE. The number of neurons in the hidden layer C^o can be manually designed.

Patches extracted from the input feature maps are then normalized by subtracting the mean and dividing the standard deviation. By contrast normalization, input patches are mapped to a canonical form ranging from [0.05, 0.95] to reduce redundancy and make the network converge faster

$$\boldsymbol{P} = \frac{\boldsymbol{P} - \text{mean}(\boldsymbol{P})}{\text{var}(\boldsymbol{P})}. \tag{7}$$

Following the contrast normalization, patches are sent to the whitening layer, where ZCA whitening is applied to the patches to sphere features of the input in the training process:

$$\boldsymbol{P} = \boldsymbol{W}\boldsymbol{P} = \boldsymbol{E}\boldsymbol{T}^{-1/2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{P} \tag{8}$$

where $\boldsymbol{E}$ and $\boldsymbol{T}$ are the eigenvectors and eigenvalues of the covariance of $\boldsymbol{P}$, respectively. $\boldsymbol{W}$ denotes the whitening matrix.

The parameters of the large network are updated using stochastic gradient decent [43]. For instance, the weights connecting the auto-encoder layers are updated as follows:

$$\boldsymbol{\Theta} = \boldsymbol{\Theta} - \alpha \frac{\partial \Gamma(\boldsymbol{\Theta}, \boldsymbol{\beta}, \boldsymbol{x})}{\partial \boldsymbol{\Theta}} \tag{9}$$

where α is the learning rate and decreases as the number of iteration increases. The updating rule of the learning rate is demonstrated

$$\alpha = \alpha \cdot (1 + \gamma \cdot n)^{-t} \tag{10}$$

where n is the number of iterations, γ and t are scalar hyper parameters predefined.

Robust and discriminative weights are learned when the training process converges. We reshape the weights learned by the DAE to 4-D form convolutional kernels $\boldsymbol{k} \in R^{K_1 \times K_2 \times C^i \times C^o}$, meaning C^o kernels are learned in the hidden layer of the DAE and each kernel is a $K_1 \times K_2 \times C^i$ array. The lth layer's feature maps $S^{(l)}$ are then convolved by the kernels and subsampled by the pooling operation to form the $(l+1)$th layer's feature maps $S^{(l+1)}$

$$S_j^{(l+1)} = \varphi\left(\sum\nolimits_{i=1}^{l} S_i^{(l)} * \boldsymbol{k}_{ij} + \beta_j\right) \tag{11}$$

where $S_i^{(l)}$ is the ith feature map in layer l, $\boldsymbol{k}_{ij}$ denotes the ith channel of the jth kernel, "*" denotes to convolution operation. After convolution, pooling is conducted to the $(l+1)$th layer's feature maps with pooling size $s_1 \times s_2$.

The pooling operation can select significant features and reduce the parameters of the network. Several forms of pooling method have been proposed to subsample the features, e.g., max pooling, average pooling [44], stochastic pooling [45], and spatial pyramid pooling [46]. We utilize two typical pooling methods—max pooling and average pooling. Different pooling methods are selected facing datasets with different distributions.

D. Analysis

Compared with stacked DAEs and SSAEs, the proposed model stacks DAEs in a convolutional way in the inference stage. The spatial position of features in the lower layer are kept in the feature maps of the higher layer in this way. This structure greatly reserves the local relevance of features inside the neighborhood.

Besides the convolution structure, the DAE contributes to the representation learning in our model. Compared to the conventional auto-encoder, the DAE can automatically denoise the input data. This property is especially helpful when tackling datasets with noise or large variation. Moreover, DAE can be

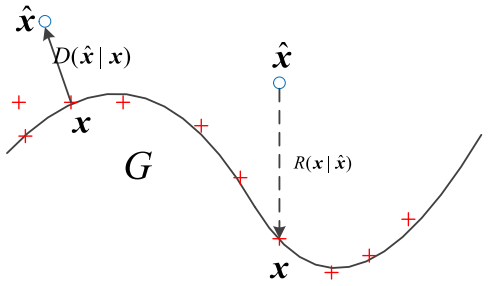


Fig. 3. Illustration of learning stochastic mapping R. During training $\boldsymbol{x}$ is mapped to $\hat{\boldsymbol{x}}$ by $D(\hat{\boldsymbol{x}}|\boldsymbol{x}, \lambda)$, then $\hat{\boldsymbol{x}}$ is mapped back to $\boldsymbol{x}$ via the learned mapping $R(\boldsymbol{x}|\hat{\boldsymbol{x}})$. Then data even far away from the curve can be mapped onto the right surface through learned R.

interpreted from a probabilistic perspective. Suppose the original data $\boldsymbol{x}$ (here $\boldsymbol{x}$ can be seen as a random variable that follows distribution G, not a point) is in a hyperplane defined by distribution G. During denoising training, we learn a stochastic operator $R(\boldsymbol{x}|\hat{\boldsymbol{x}})$ that maps $\hat{\boldsymbol{x}}$ back to $\boldsymbol{x}$

$$R(\boldsymbol{x}|\hat{\boldsymbol{x}}) = \varphi'\left(\boldsymbol{\Theta}'(\varphi(\boldsymbol{\Theta}\boldsymbol{x} + \boldsymbol{\beta})) + \boldsymbol{\beta}'\right). \tag{12}$$

The corrupted version $\hat{\boldsymbol{x}}$ usually differs from the original input data $\boldsymbol{x}$ that follows the distribution G, as Fig. 3 illustrates. During training, $\boldsymbol{x}$ is mapped to $\hat{\boldsymbol{x}}$ by $D(\hat{\boldsymbol{x}}|\boldsymbol{x}, \lambda)$, then $\hat{\boldsymbol{x}}$ is mapped back to $\boldsymbol{x}$ to learn a mapping $R(\boldsymbol{x}|\hat{\boldsymbol{x}})$. At the inference stage, data even far away from the original feature space can be mapped onto the right surface through the learned mapping R. The latent representation $\boldsymbol{y}$ can thus be regarded as a coordinate that is capable to capture the main variations of the input data.

The above theoretical analysis gives a conclusion that though DAE is a simple variant of the basic auto-encoder, the principles and functionality differ a lot. Stacking DAEs in a convolutional way can learn robust features and accumulate the robustness layer by layer, thus can obtain much more robust feature representations than merely stacking conventional auto-encoders.

To learn the convolutional kernels, Masci *et al.* [24] trained a convolutional auto-encoder which forwardly propagates the features and backwardly propagates the gradients all in a convolutional way. Though the convolution structure can preserve local relevance of the input data, training the auto-encoder convolutionally is not easy. To the best of our knowledge, there are not many optimization method for convolutional training in the unsupervised networks. To solve this problem, we use the convolutional structure only in the inference stage. In the training stage, we optimize a DAE in each layer through patch-wise training. From the perspective of the number of training samples, they nearly see the same number of samples. The difference lies in that when training the auto-encoder convolutionally, the auto-encoder sees the input images as high dimensional tensors, when training the auto-encoder patch-wisely, the auto-encoder sees the input patches as 1-D vectors. While the convolutional auto-encoder is hard to train, the conventional DAE is easy to train, as some optimization methods that have been proved to be efficient can be used properly, such as sparse constraint, whitening, etc. Hence, by training the auto-encoder patch-wisely, we can obtain better optimized weights/convolutional kernels. In the inference stage, we still use the convolutional structure, so the advantages of convolution can still be taken for better representation.

It should be noted that the layer-wise whitening technique contribute a lot to optimizing the large network. Features inside a local area $\boldsymbol{P} \in R^{K_1 \times K_2 \times C^i}$ which have C^i channels, are closely connected to each other with high correlations. This leads to redundancy in the training samples and prevents the network from obtaining better convolutional kernels. To solve this problem, the whitening layer is embedded into our model to connect two adjacent convolutional layers. In the whitening layers, ZCA whitening technique is used to preprocess the feature maps. Redundancy of the input data is removed through this kind of normalization.

The proposed model can be used to pretrain the CNN. Deep CNN has the problem of gradient vanishing, i.e., the gradients back propagated to the bottom layers are too small that the parameters in the first few convolutional layers update poorly. This problem cannot be solved by expanding the training samples, as the gradients in the first few convolutional layers are quite small in every iteration. On the contrary, our model is optimized by layer-wise training to ensure each layer is trained completely. Though the proposed model is unsupervised, it has the necessary structures that CNN also holds to learn abstract and discriminative feature representations, such as the convolution operation and the hierarchical structure. All these properties make it possible for our model to initialize CNN. This paper is supposed to be done in the future.

IV. Experiments

In this section, we design detailed hyper-parameters to verify the effectiveness of the proposed unsupervised network by conducting experiments on five benchmark datasets: the STL-10 dataset [47], the Caltech 101 dataset [48], the Land-use dataset [49], the CIFAR-10 dataset [50], and the MNIST dataset [10]. Fig. 4 shows ten examples from each image set. As an unsupervised deep network, SCDAE is compared with the well-known deep unsupervised convolutional structures: the stacked convolutional auto-encoder (SCAE) [24], the SSAE used in [21], CDBNs introduced in [28]. To evaluate the influence of the convolutional structure in deep networks, we compared our algorithm with stacked denoising auto-encoder (SDAE) [23] which does not contain a convolution structure in the network.

The performance of SCDAE is significantly impacted by the main hyper-parameters, i.e., the denoising structure, the whitening layers, the depth of the network, and the number of feature maps set in each layer. In our model, these parameters are carefully designed by conducting sufficient experiments on the validation set. The validation set is a 5% separation of the training set. For fair comparison, all the algorithms are applied only to learn the convolutional kernels and the raw images are mapped to the final representations which are classified by a linear SVM. All the experiments are conducted without data augmentation.

In the following, we first introduce five image datasets used to assess the performance of each algorithm. In this part,

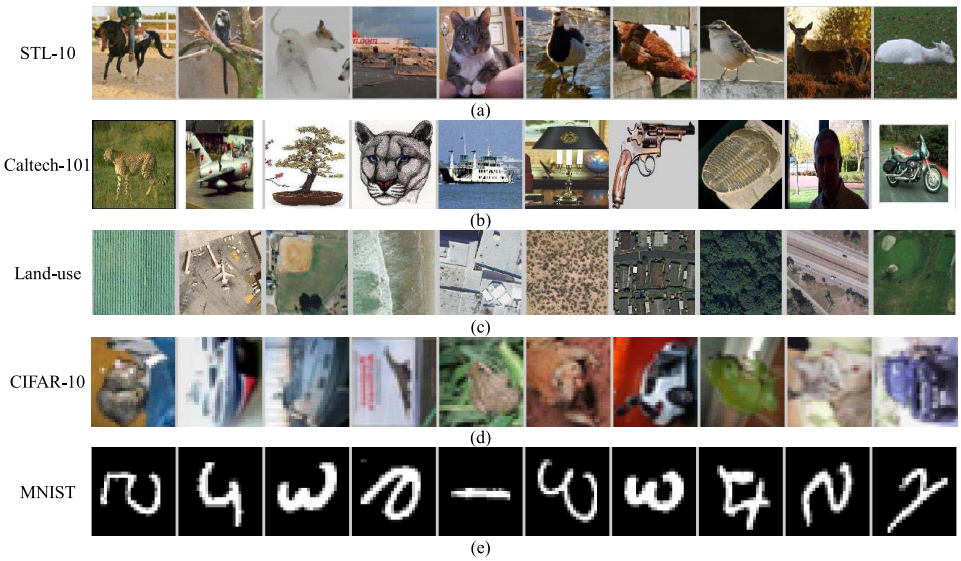


Fig. 4. Samples of the five image datasets used in our experiments. (a) STL-10. (b) Caltech-101. (c) Land-use. (d) CIFAR-10. (e) MNIST.

we also display the detailed parameter configurations of the proposed method on each dataset. Then in the next part, classification results of different models on these datasets are showed with rigorous analysis. After that, hyper-parameters and the main techniques used in our model are evaluated with certain datasets. Conclusions of the experiment results and analysis are given in the last part of this section.

A. Datasets and Basic Configurations of SCDAE

1) STL-10 Dataset: The STL-10 dataset is a natural image set for developing unsupervised feature learning, deep learning, and self-taught learning algorithms. The primary challenge is to utilize the unlabeled data to build a useful prior. STL-10 dataset contains ten classes: 1) airplane; 2) bird; 3) car; 4) cat; 5) deer; 6) dog; 7) horse; 8) monkey; 9) ship; and 10) truck with a resolution of 96×96. Each class has 500 training images and 800 testing images. An additional 100 000 unlabeled images are provided for unsupervised learning. The unlabeled images are extracted from a broader distribution of images and contain other types of examples besides the ten classes, e.g., bear and trains. Fig. 4(a) shows some examples of this dataset.

We follow the standard setting in [47] and [51]: 1) perform unsupervised feature learning on the unlabeled data; 2) perform supervised learning on the labeled data using predefined tenfolds of 100 examples from the training data; and 3) report average accuracy on the full test set. Detailed settings of our network that achieve the best results as demonstrated in Table I.

The input is a $96 \times 96 \times 3$ RGB image. The convolutional kernel size of the first layer is $9 \times 9 \times 3$. In the training stage, $9 \times 9 \times 3$ patches are extracted from the normalized and whitened image and then utilized to train the first convolutional layer. After training, each input image is convolved by the leaned filters to generate the output feature maps with a size of $87 \times 87 \times 800$. The second convolutional layer ("2nd conv" in Table I) has 4000 $2 \times 2 \times 800$ convolutional kernels which are trained by DAEs with patches extracted from the $2 \times 2 \times 800$ feature maps representing each training sample. The output of the second convolutional layer is $2 \times 2 \times 4000$ final feature maps representing each original raw image, which are then classified by the subsequent linear SVM. The average activation of units in the hidden layer of DAE (denoted as "sparse target" in Table I) in each convolutional layer is 0.01.

TABLE I
MAIN PARAMETER SETTINGS OF SCDAE ON STL-10 DATASET

Layer	1st conv	2nd conv
Kernel size	9×9×3	2×2×800
Number of maps	800	4000
Sparse target	0.01	0.01
Pool type	mean	mean
Pool size	8×8	5×5

2) Caltech-101 Dataset: The Caltech-101 dataset are collected from 101 classes of RGB images (such as animals, vehicles, and flowers) with significant shape variability. Most images are uncluttered and contain central objects that occupy most of the image. Fig. 4(b) shows some examples of this dataset. The number of images in each category varies from 31 to 800, and image sizes differ with a probable average size of 300×300. We perform training on 30 randomly selected images per class and test on the remainder. In our experiments, each sample is resized to 225×225 before input into the model for the convenience of data processing and time saving. The detailed configurations of the proposed algorithm is shown in Table II.

3) Land-Use Dataset: The Land-use dataset contains manually extracted high-resolution aerial images downloaded from the U.S. Geological Survey national map.[1] The resolution of

[1]The dataset can be downloaded from http://vision.ucmerced.edu/datasets.

TABLE II
MAIN PARAMETER SETTINGS OF SCDAE ON CALTECH-101 DATASET

Layer	1st conv	2nd conv
Kernel size	10×10×3	4×4×500
Number of maps	500	1000
Sparse target	0.05	0.1
Pool type	max	max
Pool size	24×24	2×2

TABLE III
MAIN PARAMETER SETTINGS OF SCDAE ON LAND-USE DATASET

Layer	1st conv	2nd conv
Kernel size	9×9×3	3×3×1000
Number of maps	1000	2000
Sparse target	0.5	0.05
Pool type	max	max
Pool size	31×31	2×2

TABLE IV
MAIN PARAMETER SETTINGS OF SCDAE ON CIFAR-10 DATASET

Layer	1st conv	2nd conv
Kernel size	6×6×3	2×2×800
Number of maps	800	4000
Sparse target	0.01	0.01
Pool type	mean	mean
Pool size	3×3	4×4

TABLE V
MAIN PARAMETER SETTINGS OF SCDAE ON MNIST DATASET

Layer	1st conv	2nd conv
Kernel size	11×11	2×2×1000
Number of maps	1000	1500
Sparse target	0.05	0.05
Pool type	max	max
Pool size	6×6	1×1

the images is one foot per pixel and they are cropped to 256×256 pixels. The dataset contains 21 scene categories with 100 samples per class. Some overlapping classes, such as the dense residential, medium residential, and sparse residential (which mainly differ in the density of structures), are particularly challenging to classify. Fig. 4(c) shows examples from this dataset. In the following experiments, we randomly take 80% of the images per category as the training samples and take the rest to test the performance of the algorithms. The detailed configurations of the proposed algorithm are shown in Table III.

4) CIFAR-10 Dataset: The CIFAR-10 dataset consists of 60 000 32×32 color images in ten classes (i.e., airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck) with 6000 images per class. There are 50 000 training images and 10 000 test images. The classes are completely mutually exclusive. Fig. 4(d) shows some examples of this dataset. The detailed configurations of the proposed algorithm are shown in Table IV.

5) MNIST Dataset: The MNIST database of handwritten digits has a training set of 60 000 examples, and a test set of 10 000 examples. It is a subset of a larger set available from National Institute of Standards and Technology (NIST). The digits have been size-normalized and centered in a fixed-size image. Each image is a 28×28 gray image. Some of the images are quite confusing from each other and it is challenging for the algorithms to recognize them. Fig. 4(e) demonstrates some examples of this dataset. The detailed configurations of the proposed algorithm are shown in Table V.

B. Classification Results on Various Datasets

The proposed algorithm is compared with other algorithms whose hyper-parameters have been optimized on the validation set. To evaluate the influence of the depth of the network, experiments of one-layer SCDAE are also conducted on all datasets. The classification results on each dataset are demonstrated in Table VI.

As shown, the proposed SCDAE method greatly outperforms all the other algorithms on the challenging STL-10, Caltech-101, and Land-use datasets and gets superior results to the best comparable algorithms on CIFAR-10 and MNIST datasets, indicating the effectiveness and universality of our algorithm. On Land-use dataset, there is over 10% promotion compared with the highest classification result achieved by other unsupervised deep networks. And on the challenging STL-10 dataset which is prepared for unsupervised and semi-supervised algorithms, we beat the other algorithms and achieve competitive classification performances. All these results suggest the superiority of our model on feature representation performance.

On all these datasets, our SCDAE model outperforms the SDAE and SSAE algorithms, indicating the effectiveness of convolutional structure over basic vector-form structures. Comparing the results of our model and SCAE, the importance of denoising structure and patch-wise training strategy is verified. Two-layer SCDAE model evidently promotes the classification performance based on one-layer model, proving that deep architectures can learn more discriminative features. The detailed parameter settings, techniques, and structures are evaluated in the next part.

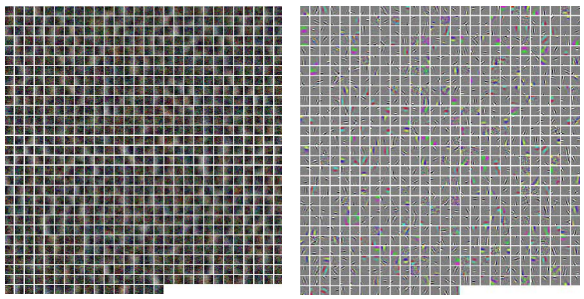
C. Analysis of SCADE's Properties

In this part, we mainly analyze the influence of structures and techniques designed in the proposed model. The key structures that contribute to the success of our network are the convolutional DAE and the whitening layer.

The importance of the whitening layers is first evaluated with experiments on CIFAR-10 dataset. The network on this dataset has two convolutional layers with a whitening layer after the input of each convolutional layer. Empirically speaking, the first whitening layer plays a key role in the training

TABLE VI
CLASSIFICATION RESULTS OF THE UNSUPERVISED DEEP NETWORKS ON ALL FIVE DATASETS

Algorithm	STL-10	Caltech-101	CIFAR-10	Land-use	MNIST
SCDAE 1 layer	56.6±0.8	71.5±1.6	75.0±1.2	75.3±1.3	99.17±0.1
SCDAE 2 layer	**60.5 ± 0.9**	**78.6± 1.2**	**80.4 ± 1.1**	**93.7 ± 1.3**	**99.38± 0.05**
SCAE	40.0±3.1	58.0±2.0	78.2 [24]	60.7±1.7	99.29 [24]
SSAE	55.5±1.2	66.2±1.2	74±0.9	83.5±1.5	96.29±0.12
CDBN	43.5±2.3	65.4±0.5 [28]	78.9 [52]	60.2±2.1	99.18 [28]
SDAE	53.5±1.5	59.5±0.3	70.1±1.0	66.4±0.8	99.06 [23]



(a) (b)

Fig. 5. Visualization of filters learned by the DAE of the first convolutional layer. (a) Filters without the first whitening layer to process the input data. (b) Filters using the first whitening layer.

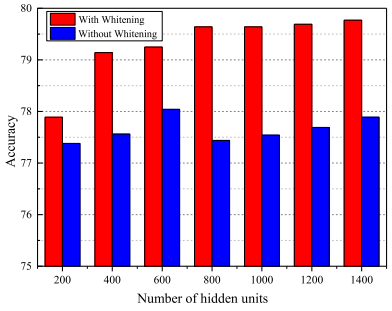


Fig. 6. Classification results on CIFAR-10 dataset by SCDAE with and without the second whitening layer. The number of the hidden neurons in the second layer is fixed at 3000.

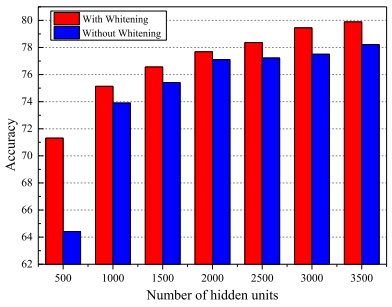


Fig. 7. Classification results on CIFAR-10 dataset by SCDAE with and without the second whitening layer. The number of the hidden neurons in the first layer is fixed at 1000.

of the DAE in the first convolutional layer, as there is usually much redundancy inside the raw images. The redundancy and correlations of the pixels can be reduced by the ZCA whitening technique. We visualize the filters learned by the DAE of the first convolutional layer with and without the first whitening layer to see its impact on the learning of convolutional kernels. Fig. 5(a) shows the visualized filters learned by the DAE without the first whitening layer on CIFAR-10 dataset, and Fig. 5(b) shows the filters with the first whitening layer.

From Fig. 5, it is revealed that the DAE with a whitening layer to process the input data learns a plenty of edges and color feature detectors, while the auto-encoder without a whitening layer learns poor features, which certifies the whitening layer's capacity of optimizing the DAEs.

We take the whitening layer as a type of optimization method and evaluate its ability of optimizing the deep network. We apply our two-layer SCDAE on CIFAR-10 dataset with and without the second whitening layer. To make a general conclusion, we conduct this experiment with different sizes of network. First, the number of hidden units in the second convolutional layer is fixed at 3000 and number of hidden units in the first convolutional layer is altered from 200 to 1400. On each size, the performance of SCDAE with and without the second whitening layer is evaluated. Classification results are shown in Fig. 6. Then, the number of hidden units in the first convolutional layer is fixed at 1000 with that in the second convolutional layer changing from 500 to 3500. Fig. 7 shows the classification results by our algorithm.

The proposed network with the second whitening layer outperforms the one without the second whitening layer in various network sizes, indicating that the whitening layer has the capacity of optimizing the large and deep neural network. It is conducive to learning discriminative and invariant feature representations from redundant input data.

We then apply the proposed SCDAE with the whitening layers on all the five datasets, and apply SCDAE without the second whitening layer on these datasets as a controlled trail. The networks are trained with relatively fine hyper-parameter settings. Performance demonstrated in Table VII further proves the importance of the whitening layers in our model.

The importance of DAE is studied following the layer-wise whitening technique. To verify the superiority of DAE over conventional auto-encoder, we apply a two-layer SCDAE network on STL-10 dataset. In the training stage, to highlight the influence of denoising structures, we replace the DAE in SCDAE with the basic sparse auto-encoder as a comparative trial. To evaluate the influence of denoising structures on

TABLE VII
RESULTS OF SCDAE WITH AND WITHOUT WHITENING LAYER APPLIED TO THE SECOND LAYER ON FIVE DATASETS

Datasets	With whitening layer	Without whitening layer
STL-10	60.5±0.9	57.6±0.7
Caltech-101	78.6±1.2	76.5±1.0
CIFAR-10	80.4±1.1	78.4±1.2
Land-use	93.7±1.3	89.3±2.2
MNIST	99.38±0.05	99.35±0.06

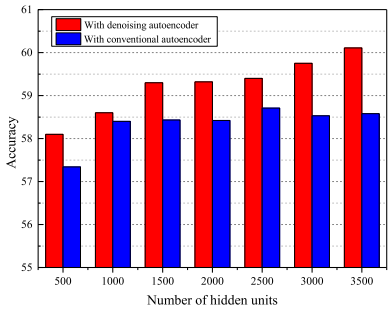


Fig. 8. Classification results on STL-10 dataset with a two-layer SCDAE stacked by DAE and the conventional auto-encoder. Number of hidden neurons in the first convolutional layer are fixed at 1000 and the number of hidden neurons in the second layer is changing.

different sizes of network, we fix the number of hidden neurons in the first convolutional layer and change the number of hidden neurons in the second layer. Classification results on STL-10 dataset are shown in Fig. 8.

As shown in Fig. 8, SCDAE with DAEs to learn the filters outperforms that with conventional auto-encoders on all the network sizes, which illustrates that the DAEs have better feature representation capacity than conventional auto-encoders and can be more helpful to boost the performance of the subsequent linear SVM classifier. To strengthen this inference, we apply the proposed network with the DAEs and with conventional auto-encoders on all the five datasets and compare the classification performance, as shown in Table VIII. Results in Table VIII show that SCDAE with denoising structures has advantages over the similar structure with the basic auto-encoder on different datasets, which indicates the proposed model has the ability of learning good features from data with various distributions.

The relations between the performance and the size of the deep network can also be inferred from the experiment results. Seen from Figs. 6 and 7, when fixing the size of the second convolutional layer, the classification results vary little as the size of the first convolutional layer changes. However, when fixing the size of the first layer in a proper value, the classification performance improves as the second convolutional layer gets larger. It indicates that the size of the second layer evidently affects the performance of the deep network, while the size of the first layer has little effect. This verdict can guide us to design an efficient network. For instance, the number of hidden units in the first convolutional layer can be set relatively small within a proper region while the size of the second layer should be as large as possible. Figs. 7 and 8 also demonstrate

TABLE VIII
CLASSIFICATION RESULTS OF A TWO-LAYER SCDAE WITH DAE AND THE BASIC SPARSE AUTO-ENCODER ON ALL THE FIVE DATASETS

Datasets	Denoising auto-encoder	Conventional auto-encoder
STL-10	60.5±0.9	58.6±0.8
Caltech-101	78.6±1.2	77.3±1.1
CIFAR-10	80.4±1.1	79.8±0.8
Land-use	93.7±1.3	93.1±1.0
MNIST	99.38±0.05	99.12±0.1

a fine property of our network: when increasing the size of the second layer, the classification results are increasing, suggesting that the proposed network has a high boundary of the capacity to learn discriminative feature representations.

D. Conclusion of Experiments

The proposed SCDAE network outperforms the conventional unsupervised deep networks on five challenging datasets, i.e., Caltech-101, STL-10, Land-use, CIFAR-10, and MNIST. The recognition performances on these datasets prove that our algorithm can learn better weights thus it can generate more representative features from the original images; Compared with SAEs, the proposed model, stacking DAEs in a convolutional way, can reserve local relevance and learn better features.

The experiments studying the impacts of the designed structures indicate that the DAE structure acts as one of the key roles in our model as it can help to learn robust and abstract feature representations compared to traditional auto-encoders. The whitening layers are indispensable to our model as there is usually a big drop in accuracy when removing this structure. By conducting experiments with various sizes of network, the relation between the complexity and the performance of the designed network is revealed: the size of deeper layers has greater influence on the final feature representation performance than the shallower layers.

V. CONCLUSION

This paper proposes the SCDAE, an unsupervised deep network inspired by recent feature learning architectures CNN and an improvement of the existing successful network SDAE. SCDAE is constructed by stacking the DAEs whose parameters are optimized through patch-wise training in a convolutional way. The deep model can learn robust and abstract hierarchical feature representations from raw visual data in an unsupervised manner. The large network is trained with layer-wise whitening technique, which proves to be an effective regularization method by the classification performance on the benchmarks. It is revealed that the proposed algorithm outperforms conventional feature learning algorithms on the challenging Land-use, Caltech-101, STL-10, CIFAR-10, and MNIST datasets, indicating that the proposed algorithm has superiority in learning robust and abstract hierarchical representations.

Although the proposed architecture is effective, there is still room for further improvements. Future work will aim to stack

more layers with better optimization methods to learn highly hierarchical features.

ACKNOWLEDGMENT

The authors would like to thank the handing Editor and the anonymous reviewers for their careful reading and helpful remarks, which have contributed in improving the quality of this paper.

REFERENCES

[1] X. Tian, L. Yang, Y. Lu, Q. Tian, and D. Tao, "Image search reranking with hierarchical topic awareness," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2177–2189, Oct. 2015, doi: 10.1109/TCYB.2014.2366740.

[2] H. Li, Y. Wei, L. Li, and C. L. P. Chen, "Hierarchical feature extraction with local neural response for image recognition," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 412–424, Apr. 2013.

[3] X. Lu, X. Li, and L. Mou, "Semi-supervised multitask learning for scene recognition," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1967–1976, Sep. 2015.

[4] L. Zhang *et al.*, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, Oct. 2015.

[5] Y. Zhang, B. Du, and L. Zhang, "A sparse representation-based binary hypothesis model for target detection in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1346–1354, Mar. 2015.

[6] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, New York, NY, USA, 2006, pp. 2169–2178.

[7] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. CVPR*, Miami, FL, USA, 2009, pp. 1794–1801.

[8] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, Zürich, Switzerland, 2014, pp. 818–833.

[9] H. Zhou, G.-B. Huang, Z. Lin, H. Wang, and Y. C. Soh, "Stacked extreme learning machines," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2013–2025, Sep. 2015.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[13] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[14] X. Lu, Y. Wang, and Y. Yuan, "Graph-regularized low-rank representation for destriping of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4009–4018, Jul. 2013.

[15] X. Lu, Y. Yuan, and P. Yan, "Image super-resolution via double sparsity regularized manifold learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 12, pp. 2022–2033, Dec. 2013.

[16] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[17] J. Weng, N. Ahuja, and T. S. Huang, "Cresceptron: A self-organizing neural network which grows adaptively," in *Proc. IJCNN*, vol. 1. Baltimore, MD, USA, 1992, pp. 576–581.

[18] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biol. Cybern.*, vol. 59, nos. 4–5, pp. 291–294, 1988.

[19] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*, document ADA164453, Defen. Tech. Inf. Center, Fort Belvoir, VA, USA, 1985.

[21] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, Helsinki, Finland, 2008, pp. 1096–1103.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[24] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. ICANN*, Espoo, Finland, 2011, pp. 52–59.

[25] M. Norouzi, M. Ranjbar, and G. Mori, "Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning," in *Proc. CVPR*, Miami, FL, USA, 2009, pp. 2735–2742.

[26] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," Dept. Comput. Sci., Colorado Univ. Boulder, Boulder, CO, USA, Tech. Rep. CU-CS-321-86, 1986.

[27] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[28] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. ICML*, Montreal, QC, Canada, 2009, pp. 609–616.

[29] H. Goh, N. Thome, M. Cord, and J.-H. Lim, "Learning deep hierarchical visual feature coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2212–2225, Dec. 2014.

[30] L. Shao, D. Wu, and X. Li, "Learning deep and wide: A spectral method for learning deep networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2303–2308, Dec. 2014.

[31] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[32] M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. CVPR*, San Francisco, CA, USA, 2010, pp. 2528–2535.

[33] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. ICCV*, Barcelona, Spain, 2011, pp. 2018–2025.

[34] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[35] K. Kavukcuoglu *et al.*, "Learning convolutional feature hierarchies for visual recognition," in *Proc. NIPS*, Vancouver, BC, Canada, 2010, pp. 1090–1098.

[36] C. Szegedy *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[37] J. Han *et al.*, "Two-stage learning to predict human eye fixations via SDAEs," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 487–498, Feb. 2016, doi: 10.1109/TCYB.2015.2404432.

[38] A. Bell and T. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.

[39] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[40] D. Tao, L. Jin, Z. Yang, and X. Li, "Rank preserving sparse learning for Kinect based scene classification," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1406–1417, Oct. 2013.

[41] D. Du, L. Zhang, H. Lu, X. Mei, and X. Li, "Discriminative hash tracking with group sparsity," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2015.2457618.

[42] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[43] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. ICML*, Banff, AB, Canada, 2004, p. 116.

[44] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. ICML*, Haifa, Israel, 2010, pp. 111–118.

[45] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *arXiv preprint arXiv:1406.4729*, 2014.

[47] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. ICAIS*, Fort Lauderdale, FL, USA, 2011, pp. 215–223.

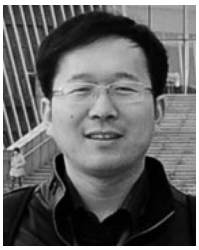
[48] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, 2007.

[49] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. ICGIS*, San Jose, CA, USA, 2010, pp. 270–279.

[50] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[51] A. Coates and A. Y. Ng, "Selecting receptive fields in deep networks," in *Proc. NIPS*, Granada, Spain, 2011, pp. 2528–2536.

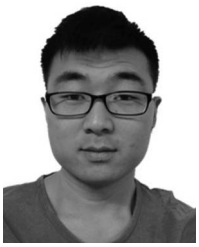
[52] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on CIFAR-10," 2010.



Bo Du (M'10–SM'15) received the B.S. degree and the Ph.D. degree in Photogrammetry and Remote Sensing from State Key Lab of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University, Wuhan, China, in 2005, and in 2010, respectively.

He is currently a professor with the School of Computer, Wuhan University, Wuhan, China. He has more than 40 research papers published in the IEEE Transactions on Geoscience and Remote Sensing (TGRS), IEEE Transactions on image processing (TIP), IEEE Journal of Selected Topics in Earth Observations and Applied Remote Sensing (JSTARS), and IEEE Geoscience and Remote Sensing Letters (GRSL), etc. His major research interests include pattern recognition, hyperspectral image processing, and signal processing.

He is currently a senior member of IEEE. He received the best reviewer awards from IEEE GRSS for his service to IEEE Journal of Selected Topics in Earth Observations and Applied Remote Sensing (JSTARS) in 2011 and ACM rising star awards for his academic progress in 2015. He was the Session Chair for the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). He also serves as a reviewer of 20 Science Citation Index (SCI) magazines including IEEE TGRS, TIP, JSTARS, and GRSL.



Wei Xiong (S'15) received the B.S. degree in School of Computer, Wuhan University, China, in 2014, where he is currently pursuing the master degree. In 2015, he was a Research Assistant in Laboratoire d'Informatique Gaspard-Monge (LIGM), Paris, France. His research interests include computer vision, machine learning, and deep learning.

Jia Wu (M'16) received the PhD degree in computer science from University of Technology Sydney (UTS), Australia. He is currently a Research Associate in the Centre of Quantum Computation and Intelligent Systems (QCIS), UTS. His research focuses on data mining and machine learning. Since 2009, Dr. Wu has published more than 20 refereed journal and conference papers (such as IEEE Trans. on KDE, and IEEE Trans. on CYB, Pattern Recognition, IJCAI, ICDM, SDM, CIKM) in these areas.

Lefei Zhang (S'11–M'14) received the B.S. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2008 and 2013, respectively. From August 2013 to July 2015, he was with the School of Computer, Wuhan University, as a Postdoctoral Researcher, and he was a Visiting Scholar with the CAD & CG Lab, Zhejiang University in 2015. He is currently a lecturer with the School of Computer, Wuhan University, and also a Hong Kong Scholar with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include pattern recognition, image processing, and remote sensing.

Dr. Zhang is a reviewer of more than twenty international journals, including the IEEE TIP, TNNLS, TMM, and TGRS.

Liangpei Zhang (M'06–SM'08) received the B.S. degree in physics from Hunan Normal University, Changsha, China, in 1982, the M.S. degree in optics from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China, in 1988, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998.

He is currently the head of the remote sensing division, state key laboratory of information engineering in surveying, mapping, and remote sensing (LIESMARS), Wuhan University. He is also a "Chang-Jiang Scholar" chair professor appointed by the ministry of education of China. He is currently a principal scientist for the China state key basic research project (2011–2016) appointed by the ministry of national science and technology of China to lead the remote sensing program in China. He has more than 450 research papers and five books. He is the holder of 15 patents. His research interests include hyperspectral remote sensing, high-resolution remote sensing, image processing, and artificial intelligence.

Dr. Zhang is the founding chair of IEEE Geoscience and Remote Sensing Society (GRSS) Wuhan Chapter. He received the best reviewer awards from IEEE GRSS for his service to IEEE Journal of Selected Topics in Earth Observations and Applied Remote Sensing (JSTARS) in 2012 and IEEE Geoscience and Remote Sensing Letters (GRSL) in 2014. He was the General Chair for the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS) and the guest editor of JSTARS. His research teams won the top three prizes of the IEEE GRSS 2014 Data Fusion Contest, and his students have been selected as the winners or finalists of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) student paper contest in recent years.

Dr. Zhang is a Fellow of the Institution of Engineering and Technology (IET), executive member (board of governor) of the China national committee of international geosphere-biosphere programme, executive member of the China society of image and graphics, etc. He was a recipient of the 2010 best paper Boeing award and the 2013 best paper ERDAS award from the American society of photogrammetry and remote sensing (ASPRS). He regularly serves as a Co-chair of the series SPIE conferences on multispectral image processing and pattern recognition, conference on Asia remote sensing, and many other conferences. He edits several conference proceedings, issues, and geoinformatics symposiums. He also serves as an associate editor of the International Journal of Ambient Computing and Intelligence, International Journal of Image and Graphics, International Journal of Digital Multimedia Broadcasting, Journal of Geo-spatial Information Science, and Journal of Remote Sensing, and the guest editor of Journal of applied remote sensing and Journal of sensors.

Dr. Zhang is currently serving as an associate editor of the IEEE Transactions on Geoscience and Remote Sensing.

Dacheng Tao (F'15) received the B.Eng. degree from the University of Science and Technology of China, Hefei, China, the M.Phil. degree from the Chinese University of Hong Kong, Hong Kong, and the Ph.D. degree from the University of London, London, U.K.

He is a professor of Computer Science with the Centre for Quantum Computation & Intelligent Systems, and the Faculty of Engineering and Information Technology in the University of Technology, Sydney. He mainly applies statistics and mathematics to data analytics problems and his research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and 100+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM07, the best student paper award in IEEE ICDM13, and the 2014 ICDM 10 Year Highest-Impact Paper Award. He received the 2015 Australian Museum Scopus-Eureka Prize. He is a Fellow of the IEEE, OSA, IAPR and SPIE.