Pattern Recognition 81 (2018) 71–80

Contents lists available at ScienceDirect

Pattern Recognition

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

Entropy and orthogonality based deep discriminative feature learning for object recognition

Weiwei Shi[a], Yihong Gong[a], De Cheng[b,*], Xiaoyu Tao[a], Nanning Zheng[a]

[a] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
[b] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

ARTICLE INFO

Article history:
Received 25 July 2017
Revised 2 March 2018
Accepted 27 March 2018
Available online 28 March 2018

Keywords:
Convolutional neural network (CNN)
Discriminative feature learning
Entropy
Orthogonality
Object recognition

ABSTRACT

Inspired by the class-selectivity of the neurons in the inferior temporal (IT) area of the human visual cortex, we propose a novel discriminative feature learning method to improve the object recognition performance of convolutional neural network (CNN) without increasing the network complexity. Specifically, we apply the proposed entropy–orthogonality loss (EOL) to the penultimate layer of the CNN models in the training phase. The EOL explicitly enables the feature vectors learned by a CNN model have the following properties: (1) each dimension of the feature vectors only responds strongly to as few classes as possible, and (2) the feature vectors from different classes are as orthogonal as possible. When combined with the softmax loss, the EOL not only can enlarge the differences in the between-class feature vectors, but also can reduce the variations in the within-class feature vectors. Therefore, the discriminative ability of the learned feature vectors is highly improved. The EOL is general and independent of the CNN structure. Comprehensive experimental comparisons with both the image classification and face verification task on several benchmark datasets demonstrate that utilizing the proposed EOL during training can remarkably improve performance of CNN models compared to the corresponding baseline models trained without utilizing the EOL.

1. Introduction

To date, convolutional neural networks (CNNs) have shown excellent performances in a variety of applications, including computer vision [1–8], natural language processing [9,10], speech recognition [11,12], etc. These impressive accomplishments mainly benefit from the three factors below: (1) the rapid progress of computational hardware (e.g., GPGPUs, CPU clusters, etc.) has enabled researchers to increase the scale and complexity of their CNNs, (2) the availability of large-scale datasets (e.g., ImageNet [13]) can effectively reduce the overfitting of deep CNNs, and (3) the introduction of new training strategies, such as ReLU [14], Dropout [14], DropConnect [15], and batch normalization [16], can help produce better deep models by the back-propagation (BP) algorithm.

Recently, a common and popular method to improve performance of CNNs is to develop deeper network structures with higher complexities and then train them with large-scale datasets [17,18]. However, this strategy is unsustainable, and inevitably reaching its limit. This is because training very deep CNNs (e.g., ResNet [18]) is becoming more and more difficult to converge, and they also require GPGPU/CPU clusters and complex distributed computing platforms to implement the training process with high computational complexity. These requirements go beyond the limited budgets of many research groups and a variety of real applications.

The learned features to have good discriminative ability are very essential to object recognition [19–23]. Many discriminative feature learning methods [24–28] that are not based on deep learning have been proposed. However, constructing a highly efficient discriminative feature learning method for CNN is non-trivial. Because the BP algorithm with mini-batch is used to train CNN, a mini-batch cannot very well reflect the global distribution of the training set. Owing to the large scale of the training set, it is unrealistic to input the whole training set in each iteration. In recent years, contrastive loss [29] and triplet loss [30] are proposed to strengthen the discriminative ability of the features learned by CNN. However, both of them suffer from dramatic data expansion when composing the sample pairs or triplets from the training set. This will inevitably lead to instability and slow convergence of the training process.

* Corresponding author.
E-mail addresses: shiweiwei.math@stu.xjtu.edu.cn (W. Shi), ygong@mail.xjtu.edu.cn (Y. Gong), chengde19881214@stu.xjtu.edu.cn (D. Cheng), txy666793@stu.xjtu.edu.cn (X. Tao), nnzheng@mail.xjtu.edu.cn (N. Zheng).

https://doi.org/10.1016/j.patcog.2018.03.036
0031-3203/

We endeavor to propose a novel discriminative feature learning method for CNNs by drawing lessons from human visual cortex (HVC) object recognition mechanisms. For almost all visual tasks, the human visual system (HVS) is always superior to current machine visual systems. Hence, developing a system that simulates some properties of the HVS will be a promising research direction. Actually, existing CNNs are well known for their local connectivity and shared weight properties that originate from discoveries in visual cortex research.

Research findings in the areas of neuroscience, physiology, psychology, etc, [31–33] have shown that, object recognition in HVC is accomplished by the ventral stream, starting from the V1 area through the V2 area and V4 area, to the inferior temporal (IT) area, and then to the prefrontal cortex (PFC) area. By this hierarchy, raw input stimulus from the retina are gradually transformed into higher level representations that have better discriminative ability for speedy and accurate object recognition. Moreover, research findings have also revealed the class-selectivity of the neurons in the IT area. Specifically, the response of an IT neuron to visual stimulus is sparse with respect to classes, i.e., it only responds to very few classes. The class-selectivity implies that the feature vectors from different classes can be easily separated.

Inspired by the class-selectivity of the neurons in the IT area, we propose to improve the discriminative feature learning of CNN models by enabling the learned feature vectors to have class-selectivity. To achieve this, we propose a novel loss function, termed entropy–orthogonality loss (EOL), to modulate the neuron outputs (i.e., feature vectors) in the penultimate layer of a CNN model. The EOL explicitly enables the feature vectors learned by a CNN model to have the following properties: (1) each dimension of the feature vectors only responds strongly to as few classes as possible, and (2) the feature vectors from different classes are as orthogonal as possible. Hence our method makes an analogy between the CNN's penultimate layer neurons and the IT neurons, and the EOL measures the degree of discrimination of the learned features. The EOL and the softmax loss have the same training requirement without the need to carefully recombine the training sample pairs or triplets. Accordingly, the training of CNN models is more efficient and easier-to-implement. When combined with the softmax loss, the EOL not only can enlarge the differences in the between-class feature vectors, but also can reduce the variations in the within-class feature vectors. Therefore the discriminative ability of the learned feature vectors is highly improved, which is very essential to object recognition.

To sum up, our main contributions include:

- We propose a novel discriminative feature learning method for CNNs by enabling the learned feature vectors to have class-selectivity.
- We propose the entropy–orthogonality loss (EOL) to explicitly enforce that each dimension of the feature vectors should only respond strongly to as few classes as possible, and that the feature vectors from different classes should be as orthogonal as possible.
- We provide the optimization algorithm based on mini-batch for the proposed framework.
- Comprehensive experimental evaluations with both the image classification and the face verification tasks demonstrate the effectiveness of the proposed method.

The rest of this paper is structured as follows. Section 2 reviews the related works. Section 3.1 introduces our methodology, including the framework, the EOL and the optimization algorithm of the framework. Sections 4 and 5 provide the experimental results for the image classification and the face verification task, respectively. Section 6 concludes our work.

2. Related work

Methods to improve discriminative feature learning of CNNs mainly fall into two categories: (1) increasing CNN complexity and training data, and (2) exploiting well-designed loss functions. This section reviews the representative works that are relevant to our work for each category.

For the first category, increasing CNN complexity includes adding the depth and/or the number of feature maps at each layer. For instance, Lin et al. [34] developed the Network-In-Network (NIN) structure, where the convolution filters are replaced by a set of micro multilayer perceptrons. Data augmentation [14] is a low cost way of enlarging training samples.

In the following, we focus on listing some examples for the second category. Softmax activation function in combination with the Kullback–Leibler divergence [35,36] is used to compute the distance between the groundtruth label vector and the predicted label vector (termed softmax loss). Contrastive loss [29] is specially designed for the Siamese network, which minimizes the distance between a pair of samples from the same class and penalizes the negative pair distances for being smaller than a predefined margin. Triplet loss [30] utilizes a large number of triplets in the course of training. Each triplet consists of an anchor sample a, a positive sample p and a negative sample n, where a and p come from the same class, a and n come from two different classes. It explicitly enforces that the distance between a and n must be larger than the distance between a and p by a predefined margin.

Contrastive loss and triplet loss are well-designed for the verification task. A key difference between the datasets for image classification and those for verification is that the latter usually consist of many more classes, with much fewer training samples for each class, than those of the former. If combining the contrastive loss or triplet loss with softmax loss for classification task, compared to the number of training samples, this will result in the amount of training sample pairs or triplets dramatically increases. It inevitably leads to instability and slow convergence of the training process. These problems may be partly mitigated by carefully choosing the sample pairs or triplets. However this remarkably increases the computational complexity and will make training inconvenient. So, the contrastive loss and triplet loss are unsuitable for the more generic image classification task. While our proposed entropy–orthogonality loss (EOL) is able to be used for both verification and image classification tasks when combined with the softmax loss.

The most relevant work to our method is the deeply-supervised nets (DSN) [37]. It utilizes a classifier (e.g., softmax) at each layer of a deep network model during training to maximize both the overall classification accuracy and the layer-wise classification accuracies. We only apply the proposed EOL to the penultimate layer of a CNN during training. DSN requires a set of hyperparameters to balance the layer-wise classification accuracies and the overall accuracy, while in our method, only two trade-off hyperparameters are required for the EOL.

3. Methodology

3.1. Framework

Assume that $\mathcal{T} = \{\mathbf{X}_i, y_i\}_{i=1}^n$ is the training set, where $\mathbf{X}_i$ represents the ith training sample (i.e., input image), $y_i \in \{1, 2, \ldots, C\}$ refers to the groundtruth label of $\mathbf{X}_i$, C refers to the number of classes, and n refers to the number of training samples in $\mathcal{T}$. For the input image $\mathbf{X}_i$, we denote the output[1] of the penultimate layer

[1] Assume that the output has been reshaped into a column vector.

of a CNN by $\mathbf{x}_i$, and view $\mathbf{x}_i$ as the feature vector of $\mathbf{X}_i$ learned by the CNN.

We aim to improve discriminative feature learning of a CNN by embedding the entropy–orthogonality loss (EOL) into the penultimate layer of the CNN during training. For an L-layer CNN model, embedding the EOL into the layer $L-1$ of the CNN, the overall objective function is:

$$\min \mathcal{L}(\mathcal{W}, \mathcal{T}) = \sum_{i=1}^{n} \ell_{sm}(\mathcal{W}, \mathbf{X}_i, y_i) + \lambda \mathcal{M}(\mathbf{F}, \mathbf{c}), \tag{1}$$

where $\ell_{sm}(\mathcal{W}, \mathbf{X}_i, y_i)$ is the softmax loss for sample $\mathbf{X}_i$, $\mathcal{W}$ denotes the total layer parameters of the CNN model, $\mathcal{W} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$, $\mathbf{W}^{(l)}$ represents the filter weights of the l^{th} layer, $\mathbf{b}^{(l)}$ refers to the corresponding biases. $\mathcal{M}(\mathbf{F}, \mathbf{c})$ denotes the EOL, $\mathbf{F} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, and $\mathbf{c} = \{y_i\}_{i=1}^{n}$. Hyperparameter λ adjusts the balance between the softmax loss and the EOL.

$\mathbf{F}$ directly depends on $\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L-1}$. Hence $\mathcal{M}(\mathbf{F}, \mathbf{c})$ can directly modulate all the layer parameters from 1th to $(L-1)$th layers by BP algorithm during the training process. It is noteworthy that our proposed EOL is independent of, and able to be applied to different CNN structures. Next, we will provide the details of the proposed EOL.

3.2. Entropy–orthogonality loss (EOL)

In this subsection, an entropy and orthogonality based loss function, termed entropy–orthogonality loss (EOL), is proposed which measures the degree of discriminative ability of the learned feature vectors. For simplicity, assume that the feature vector $\mathbf{x}_i$ is a d-dimensional column vector ($\mathbf{x}_i \in \mathbb{R}^{d\times 1}$).

We call the kth ($k = 1, 2, \ldots, d$) dimension of feature vector "class-sharing" if it is nonzero on many samples belonging to many classes (we call these classes "supported classes" of this dimension). Similarly, the kth dimension of feature vector is called "class-selective" if it is nonzero on samples only belonging to a few classes. The class-selectivity of the kth dimension increases as the number of its supported classes decreases. Naturally, we can define the entropy[2] of the kth dimension to measure the degree of its class-selectivity as:

$$E(k) = -\sum_{c=1}^{C} P_{kc} \log_C(P_{kc}), \tag{2}$$

$$P_{kc} = \frac{\sum_{j\in\pi_c} |\mathbf{x}_j(k)|}{\sum_{i=1}^{n} |\mathbf{x}_i(k)|} \triangleq \frac{\sum_{j\in\pi_c} |x_{kj}|}{\sum_{i=1}^{n} |x_{ki}|}, \tag{3}$$

where, x_{ki} (i.e., $\mathbf{x}_i(k)$) refers to the kth dimension of $\mathbf{x}_i$, π_c represents the index set of the samples belonging to the cth class.

The maximum possible value for $E(k)$ is 1 when $\forall c$, $P_{kc} = \frac{1}{C}$, which means that the set of supported classes of dimension k includes all the classes and, therefore, dimension k is not class-selective at all (it is extremely "class-sharing"). Similarly, the minimum possible value of $E(k)$ is 0 when $\exists c$, $P_{kc} = 1$ and $\forall c' \neq c$, $P_{kc'} = 0$, which means that the set of supported classes of dimension k includes just one class c and, therefore, dimension k is extremely class-selective. For dimension k, the degree of its class-selectivity is determined by the value of $E(k)$ (between 0 and 1). As the value of $E(k)$ decreases, the class-selectivity of dimension k increases.

According to the discussions above, we first propose the following entropy loss $\mathcal{E}(\mathbf{F}, \mathbf{c})$:

$$\mathcal{E}(\mathbf{F}, \mathbf{c}) = \sum_{k=1}^{d} E(k), \tag{4}$$

[2] In the definition of entropy, $0 \log_C(0) = 0$.

where, $\mathbf{F} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, $\mathbf{c} = \{y_i\}_{i=1}^{n}$.

Minimizing the entropy loss is equivalent to enforcing that each dimension of the feature vectors should only respond strongly to as few classes as possible. However, the entropy loss does not consider the connection between different dimensions, which is problematic. Take 3-dimensional feature vector as an example. If we have six feature vectors from 3 different classes, $\mathbf{x}_1$ and $\mathbf{x}_2$ come from class 1, $\mathbf{x}_3$ and $\mathbf{x}_4$ come from class 2, $\mathbf{x}_5$ and $\mathbf{x}_6$ come from class 3. For the feature vector matrix $\widetilde{\mathbf{F}} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6]$, when it takes the following value of $\mathbf{A}$ and $\mathbf{B}$, respectively, $\mathcal{E}(\mathbf{A}, \widetilde{\mathbf{c}}) = \mathcal{E}(\mathbf{B}, \widetilde{\mathbf{c}})$, where $\widetilde{\mathbf{c}} = \{1, 1, 2, 2, 3, 3\}$. However, the latter one can not be classified at all, this is because $\mathbf{x}_2$, $\mathbf{x}_4$ and $\mathbf{x}_6$ have the same value. Although the situation can be partially avoided by the softmax loss, it can still cause contradiction to the softmax loss and therefore affect the discriminative ability of the learned features.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ \frac{1}{2} & 1 & \frac{1}{2} & 1 & \frac{1}{2} & 1 \end{bmatrix} \tag{5}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

To address this problem, we propose to promote orthogonality (i.e., minimize dot products) between the feature vectors of different classes. Specifically, we propose the following orthogonality loss $\mathcal{O}(\mathbf{F}, \mathbf{c})$:

$$\begin{aligned} \mathcal{O}(\mathbf{F}, \mathbf{c}) &= \sum_{i,j=1}^{n} (\mathbf{x}_i^\top \mathbf{x}_j - \phi_{ij})^2 \\ &= \|\mathbf{F}^\top \mathbf{F} - \mathbf{\Phi}\|_F^2, \end{aligned} \tag{7}$$

where,

$$\phi_{ij} = \begin{cases} 1, & \text{if } y_i = y_j, \\ 0, & \text{else}, \end{cases} \tag{8}$$

$\mathbf{\Phi} = (\phi_{ij})_{n\times n}$, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and the superscript $\top$ denotes the transpose of a matrix. Minimizing the orthogonality loss is equivalent to enforcing that (1) the feature vectors from different classes are as orthogonal as possible, (2) the L_2-norm of each feature vector is as close as possible to 1, and (3) the distance between any two feature vectors belonging to the same class is as small as possible.

Based on the above discussions and definitions, the proposed entropy–orthogonality loss (EOL) $\mathcal{M}(\mathbf{F}, \mathbf{c})$ can be obtained by integrating Eqs. (4) and (7):

$$\begin{aligned} \mathcal{M}(\mathbf{F}, \mathbf{c}) &= \alpha \mathcal{E}(\mathbf{F}, \mathbf{c}) + (1-\alpha)\mathcal{O}(\mathbf{F}, \mathbf{c}) \\ &= \alpha \sum_{k=1}^{d} E(k) + (1-\alpha)\|\mathbf{F}^\top \mathbf{F} - \mathbf{\Phi}\|_F^2, \end{aligned} \tag{9}$$

where α is the hyperparameter to adjust the balance between the two terms.

Combining Eq. (9) with Eq. (1), the overall objective function becomes:

$$\begin{aligned} \min \mathcal{L}(\mathcal{W}, \mathcal{T}) &= \sum_{i=1}^{n} \ell_{sm}(\mathcal{W}, \mathbf{X}_i, y_i) + \lambda\alpha\mathcal{E}(\mathbf{F}, \mathbf{c}) + \lambda(1-\alpha)\mathcal{O}(\mathbf{F}, \mathbf{c}) \\ &= \sum_{i=1}^{n} \ell_{sm}(\mathcal{W}, \mathbf{X}_i, y_i) + \lambda_1 \mathcal{E}(\mathbf{F}, \mathbf{c}) + \lambda_2 \mathcal{O}(\mathbf{F}, \mathbf{c}), \end{aligned} \tag{10}$$

where, $\lambda_1 = \lambda\alpha$, $\lambda_2 = \lambda(1-\alpha)$. Next, we will provide the optimization algorithm for Eq. (10).

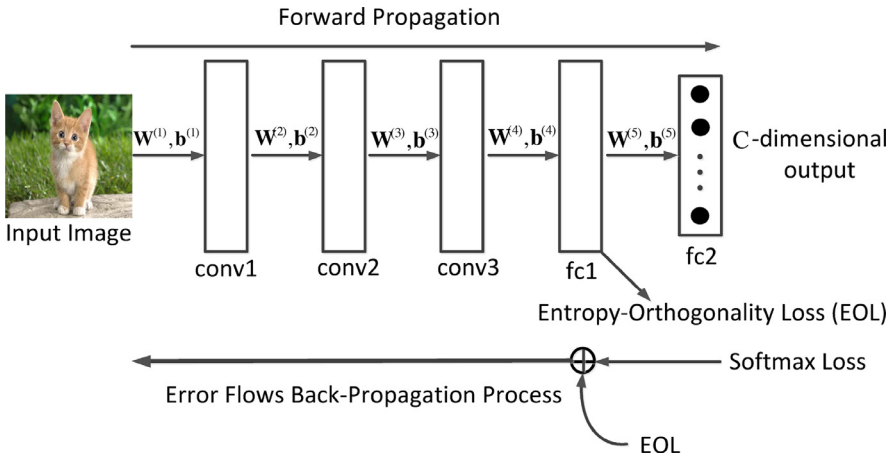

Fig. 1. The flowchart of training process in an iteration for our framework. CNN shown in this figure consists of 3 convolutional (conv) layers and 2 fully connected (fc) layers, i.e., it is a 5-layer CNN model. The last layer fc2 outputs a C-dimensional prediction vector, C is the number of classes. The penultimate layer in this model is fc1, so the entropy–orthogonality loss (EOL) is applied to layer fc1. The proposed EOL is independent of the CNN structure.

3.3. Optimization

We employ the BP algorithm with mini-batch to train the CNN model. The overall objective function is Eq. (10). Hence, we need to compute the gradients of $\mathcal{L}$ with respect to (w.r.t.) the activations of all layers, which are called the error flows of the corresponding layers. The gradient calculation of the softmax loss is straightforward. In the following, we focus on obtaining the gradients of the $\mathcal{E}(\mathbf{F},\mathbf{c})$ and $\mathcal{O}(\mathbf{F},\mathbf{c})$ w.r.t. the feature vectors $\mathbf{x}_i=[x_{1i},x_2,\ldots,x_{di}]^\top$, $(i=1,2,\ldots,n)$, respectively.

The gradient of $\mathcal{E}(\mathbf{F},\mathbf{c})$ w.r.t. $\mathbf{x}_i$ is

$$\frac{\partial \mathcal{E}(\mathbf{F},\mathbf{c})}{\partial \mathbf{x}_i}=\left[\frac{\partial E(1)}{\partial x_{1i}},\frac{\partial E(2)}{\partial x_{2i}},\ldots,\frac{\partial E(d)}{\partial x_{di}}\right]^\top, \tag{11}$$

$$\frac{\partial E(k)}{\partial x_{ki}}=-\sum_{c=1}^{C}\frac{(1+\ln(P_{kc}))}{\ln(C)}\cdot\frac{\partial P_{kc}}{\partial x_{ki}}, \tag{12}$$

$$\frac{\partial P_{kc}}{\partial x_{ki}}=\begin{cases}\dfrac{\sum_{j\notin\pi_c}|x_{kj}|}{\left(\sum_{j=1}^{n}|x_{kj}|\right)^2}\times sgn(x_{ki}), & i\in\pi_c,\\ \dfrac{-\sum_{j\in\pi_c}|x_{kj}|}{\left(\sum_{j=1}^{n}|x_{kj}|\right)^2}\times sgn(x_{ki}), & i\notin\pi_c,\end{cases} \tag{13}$$

where $sgn(\cdot)$ is sign function.

The $\mathcal{O}(\mathbf{F},\mathbf{c})$ can be written as:

$$\begin{aligned}\mathcal{O}(\mathbf{F},\mathbf{c})&=\|\mathbf{F}^\top\mathbf{F}-\mathbf{\Phi}\|_F^2=Tr((\mathbf{F}^\top\mathbf{F}-\mathbf{\Phi})^\top(\mathbf{F}^\top\mathbf{F}-\mathbf{\Phi}))\\&=Tr(\mathbf{F}^\top\mathbf{F}\mathbf{F}^\top\mathbf{F})-2Tr(\mathbf{\Phi}\mathbf{F}^\top\mathbf{F})+Tr(\mathbf{\Phi}^\top\mathbf{\Phi}),\end{aligned} \tag{14}$$

where $Tr(\cdot)$ refers to the trace of a matrix.

The gradients of $\mathcal{O}(\mathbf{F},\mathbf{c})$ w.r.t. $\mathbf{x}_i$ is

$$\frac{\partial \mathcal{O}(\mathbf{F},\mathbf{c})}{\partial \mathbf{x}_i}=4\mathbf{F}(\mathbf{F}^\top\mathbf{F}-\mathbf{\Phi})_{(:,i)}, \tag{15}$$

where the subscript $(:,i)$ represents the ith column of a matrix.

Fig. 1shows the flowchart of the training process in an iteration for our framework. Based on the above derivatives, the training algorithm for our framework is listed in Algorithm 3.1.

Algorithm 3.1 Training algorithm for our framework based on an L-layer CNN model.

Input: Training set $\mathcal{T}$, hyperparameters λ_1, λ_2, maximum number of iterations I_{max}, and counter $iter=0$.
Output: $\mathcal{W}=\{\mathbf{W}^{(l)},\mathbf{b}^{(l)}\}_{l=1}^{L}$.

1: Select a training mini-batch from $\mathcal{T}$.
2: Perform the forward propagation, for each sample, computing the activations of all layers.
3: Perform the back-propagation from layer L to $L-1$, sequentially computing the error flows of layer L and $L-1$ from softmax loss by BP algorithm.
4: Compute $\frac{\partial \mathcal{E}(\mathbf{F},\mathbf{c})}{\partial \mathbf{x}_i}$ by Eq. (11), then scale them by λ_1.
5: Compute $\frac{\partial \mathcal{O}(\mathbf{F},\mathbf{c})}{\partial \mathbf{x}_i}$ by Eq. (15), then scale them by λ_2.
6: Compute the total error flows of layer $L-1$, which is the summation of the above different items.
7: Perform the back-propagation from layer $L-1$ layer to layer 1, sequentially compute the error flows of layer $L-1$, …, 1, by BP algorithm.
8: According to the activations and error flows of all layers, compute $\frac{\partial \mathcal{L}}{\partial \mathcal{W}}$ by BP algorithm.
9: Update $\mathcal{W}$ by gradient descent algorithm.
10: $iter\leftarrow iter+1$. If $iter<I_{max}$, perform step 1.

4. Experiments with image classification task

We have conducted comprehensive experiments using the image classification and face verification tasks to demonstrate the effectiveness of the EOL.

4.1. Experimental setups

The performance evaluations are conducted using one shallow model, CNN-Quick, and three famous deep models: NIN [34], AlexNet [14] and ResNet [18], respectively. During training, we apply the EOL to the penultimate layer of the models without changing the network structures. For those hyperparameters, including dropout ratio, learning rate, weight decay and momentum, we abide by the original network settings. The hardware used in the experiments is one NVIDIA K80 GPU and one Intel Xeon E5-2650v3 CPU. The software used in the experiments is the Caffe platform [38]. All models are trained from scratch without pre-training.

The proposed EOL contains two components: (1) the entropy loss, and (2) the orthogonality loss. To study how each component affects the model performance, we conduct the four variants below for each model.

- NetXXX (Our baseline): NetXXX is trained using the softmax loss.

Table 1
Details of the CIFAR10, CIFAR100, MNIST and SVHN datasets.

Dataset	#Classes	#Samples	Size and format	Split
CIFAR10	10	60, 000	32×32 RGB	training/test:50000/10000
CIFAR100	100	60, 000	32×32 RGB	training/test:50000/10000
MNIST	10	70, 000	28×28 gray-scale	training/test:60000/10000
SVHN	10	630, 420	32×32 RGB	training/test/extra:73257/26032/531131

Table 2
Comparisons of the test error rates (%) on the CIFAR10 dataset using CNN-Quick.

Method	Param. no.	Top-1 error
CNN-Quick (Our baseline)	0.145*M*	23.47
CNN-Quick + EL	0.145*M*	17.90
CNN-Quick + OL	0.145*M*	19.88
CNN-Quick + EOL (Ours)	0.145*M*	**16.74**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

Table 3
Comparisons of the test error rates (%) on the CIFAR100 dataset using CNN-Quick.

Method	Param. no.	Top-1 error
CNN-Quick (Our baseline)	0.15*M*	55.87
CNN-Quick + EL	0.15*M*	51.05
CNN-Quick + OL	0.15*M*	53.35
CNN-Quick + EOL (Ours)	0.15*M*	**50.09**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

Table 4
Comparisons of the test error rates (%) on the SVHN dataset using CNN-Quick.

Method	Param. no.	Top-1 error
CNN-Quick (Our baseline)	0.145*M*	8.92
CNN-Quick + EL	0.145*M*	5.11
CNN-Quick + OL	0.145*M*	6.43
CNN-Quick + EOL (Ours)	0.145*M*	**4.47**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

Table 5
Comparisons of the test error rates (%) on the CIFAR10 dataset using NIN.

Method	Param. no.	Top-1 error
Without Data Augment		
Stochastic Pooling [42]	–	15.13
Maxout [43]	> 5*M*	11.68
Prob. Maxout [44]	> 5*M*	11.35
NIN [34]	0.97*M*	10.41
DSN [37]	0.97*M*	9.78
NIN (Our baseline)	0.97*M*	10.20
NIN + EL	0.97*M*	8.99
NIN + OL	0.97*M*	9.55
NIN + EOL (Ours)	0.97*M*	**8.41**
With Data Augment		
Prob. Maxout [44]	> 5*M*	9.39
Maxout [43]	> 5*M*	9.38
DropConnect [15]	–	9.32
NIN [34]	0.97*M*	8.81
DSN [37]	0.97*M*	8.22
NIN (Our baseline)	0.97*M*	8.72
NIN + EL	0.97*M*	7.15
NIN + OL	0.97*M*	7.73
NIN + EOL (Ours)	0.97*M*	**6.62**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

- NetXXX + EL: NetXXX is trained using the softmax loss and the entropy loss.
- NetXXX + OL: NetXXX is trained using the softmax loss and the orthogonality loss.
- NetXXX + EOL (Ours): NetXXX is trained using the softmax loss and the EOL.

For the hyperparameter λ_1 and λ_2, we set $\lambda_1 = 6 \times 10^{-4}$ and $\lambda_2 = 5 \times 10^{-5}$ for CNN-Quick, $\lambda_1 = 3 \times 10^{-4}$ and $\lambda_2 = 5 \times 10^{-5}$ for NIN, AlexNet and ResNet. In Section 4.8, we will show how the values of λ_1 and λ_2 have been determined, and provide the sensitivity study for λ_1 and λ_2.

4.2. Datasets

The CIFAR10 [39], CIFAR100 [39], MNIST [40], SVHN [41] and ImageNet [13] datasets are chosen to conduct performance evaluations. CIFAR10 and CIFAR100 are natural image datasets. MNIST is a dataset of hand-written digit (0–9) images. SVHN is collected from house numbers in Google Street View images. For an image of SVHN, there may be more than one digit, but the task is to classify the digit in the image center. Table 1 lists the details of the CIFAR10, CIFAR100, MNIST and SVHN datasets. These four datasets are very popular in image classification research community. This is because they contain a large amount of small images, hence they enable models to be trained in reasonable time frames on moderate configuration computers. ImageNet is chosen because it is a famous large-scale dataset with 1.28 million images and 1000 image classes, and the experiments on it can demonstrate the effectiveness of the proposed method for the large-scale classification task. The details of ImageNet will be provided in Section 4.5.

4.3. Experiments using CNN-Quick model

First, the "quick" CNN model from the official Caffe package [38] is selected as the baseline (termed CNN-Quick). It consists of 3 convolutional (conv) layers and 2 fully connected (fc) layers. We evaluated the CNN-Quick model using CIFAR10, CIFAR100 and SVHN, respectively. MNIST can not be used to evaluate the CNN-Quick model, because the input size of CNN-Quick must be 32×32, but the images in MNIST are 28×28 in size.

Tables 2–4 show the test set top-1 error rates of CIFAR10, CIFAR100 and SVHN, respectively. From these three tables, it can be seen that training CNN-Quick with the EOL is able to effectively improve performance by 6.73% on CIFAR10, by 5.78% on CIFAR100, and by 4.45% on SVHN, respectively, compared to the respective baseline. These remarkable performance improvements clearly reveal the effectiveness of the EOL.

4.4. Experiments using NIN model

Next, we apply the EOL to the well-known NIN models [34]. NIN consists of 9 conv layers without fc layer. The four datasets, including CIFAR10, CIFAR100, MNIST and SVHN, are used in the evaluation. For fairness, we complied with the same training/testing protocols and data preprocessing as in [34,37]. To be consistent with previous studies, we also used the same method as used in [34,37] to augment the CIFAR10 training set.

Tables 5–8 provide the respective comparison results of test set top-1 error rates for the four datasets. For NIN baseline, to be fair,

Table 6
Comparisons of the test error rates (%) on the CIFAR100 dataset using NIN.

Method	Param. no.	Top-1 error
Stochastic Pooling [42]	–	42.51
Maxout [43]	$>5M$	38.57
Prob. Maxout [44]	$>5M$	38.14
NIN [34]	0.98*M*	35.68
DSN [37]	0.98*M*	34.57
NIN (Our baseline)	0.98*M*	35.50
NIN + EL	0.98*M*	33.20
NIN + OL	0.98*M*	34.23
NIN + EOL (Ours)	0.98*M*	**32.54**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

Table 7
Comparisons of the test error rates (%) on the MNIST dataset using NIN.

Method	Param. no.	Top-1 error
Stochastic Pooling [42]	–	0.47
NIN [34]	0.35*M*	0.47
Maxout [43]	0.42*M*	0.47
DSN [37]	0.35*M*	0.39
NIN (Our baseline)	0.35*M*	0.47
NIN + EL	0.35*M*	0.30
NIN + OL	0.35*M*	0.36
NIN + EOL (Ours)	0.35*M*	**0.30**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

Table 8
Comparisons of the test error rates (%) on the SVHN dataset using NIN.

Method	Param. no.	Top-1 error
Stochastic Pooling [42]	–	2.80
Maxout [43]	$>5M$	2.47
Prob. Maxout [44]	$>5M$	2.39
NIN [34]	1.98*M*	2.35
DropConnect [15]	–	1.94
DSN [37]	1.98*M*	1.92
NIN (Our baseline)	1.98*M*	2.55
NIN + EL	1.98*M*	1.85
NIN + OL	1.98*M*	1.98
NIN + EOL (Ours)	1.98*M*	**1.70**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

we report the evaluation results from both our own experiments and the original paper [34]. We also include some representative methods in these four tables, including Stochastic Pooling [42], Maxout [43], Prob. Maxout [44], DropConnect [15] and DSN [37]. DSN achieved the best results on the four datasets, and it is also based on NIN with layer-wise supervisions.

The experimental results in the four tables show that our method outperforms all the compared methods. These results show:

- The proposed EOL remarkably reduces the test error rates on the four datasets, in comparison to the respective baseline.
- On CIFAR100, the reduction of error rates is most significant. NIN +EOL achieved 32.54% top-1 test error rate on it, which is 2.96% lower than our NIN baseline, and 2.03% lower than DSN.
- On MNIST and SVHN, the relative reductions of test error rates have reached 36.17% and 33.33% respectively, compared with the respective NIN baseline.

These results again reveal the effectiveness of the EOL.

4.5. Experiments using AlexNet model

The AlexNet [14] is selected as the baseline, which achieved the best performance in the ILSVRC 2012 competition of image classification. The ImageNet dataset [13] is utilized in this evaluation. It consists of variable-resolution images with 1.28 million training images, 100000 testing images and 50000 validation images. AlexNet requires a constant input size. Therefore, we warped the images to a fixed resolution of 256×256. No additional data preprocessing is used except for subtracting the image mean of the training set.

The top-1 and top-5 test error rates are listed in Table 9. AlexNet + EOL reduces the top-1 error rate by 2.65% and the top-

Table 9
Comparisons of the error rates (%) on ImageNet using AlexNet.

Method	Param. no.	Top-1 error	Top-5 error
AlexNet (Our basline)	60*M*	42.90	19.80
AlexNet + EL	60*M*	40.84	18.37
AlexNet + OL	60*M*	41.55	18.96
AlexNet + EOL (Ours)	60*M*	**40.25**	**17.90**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

Table 10
Comparisons of the test error rates (%) on the CIFAR10, CIFAR100 and SVHN datasets using ResNet.

Method	Param. no.	CIFAR10	CIFAR100	SVHN
ResNet (Our baseline)	0.27*M*	9.10	35.58	2.75
ResNet + EL	0.27*M*	7.75	32.17	2.10
ResNet + OL	0.27*M*	8.10	33.25	2.15
ResNet + EOL (Ours)	0.27*M*	**6.45**	**30.54**	**1.70**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

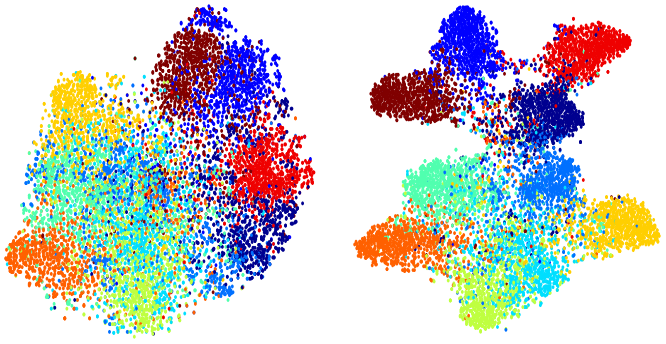

Fig. 2. Feature visualization of the CIFAR10 test set, with (a) CNN-Quick; (b) CNN-Quick+EOL. One dot denotes a image, different colors denote different classes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

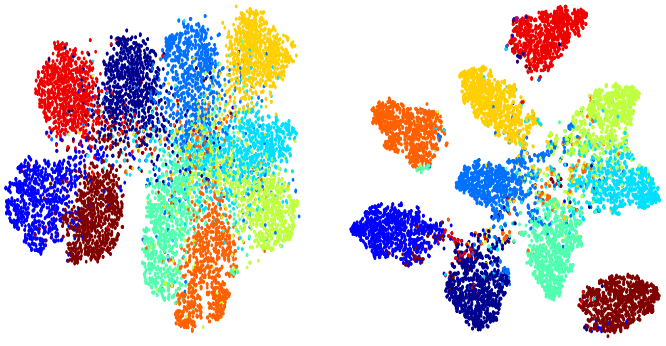

Fig. 3. Feature visualization of the CIFAR10 test set, with (a) NIN; (b) NIN + EOL.

Table 11
Comparisons of the test error rates (%) on the CIFAR10 dataset using CNN-Quick+EL by varying the λ_1 value near 6×10^{-4}. The test error rate of baseline CNN-Quick is 23.47.

λ_1	3×10^{-4}	4×10^{-4}	5×10^{-4}	6×10^{-4}	7×10^{-4}	8×10^{-4}	9×10^{-4}
Top-1 error	18.80	18.74	18.31	17.90	17.80	18.11	18.40

Table 12
Comparisons of the test error rates (%) on the CIFAR10 dataset using CNN-Quick + OL by varying the λ_2 value near 5×10^{-5}. The test error rate of baseline CNN-Quick is 23.47.

λ_2	2×10^{-5}	3×10^{-5}	4×10^{-5}	5×10^{-5}	6×10^{-5}	7×10^{-5}	8×10^{-5}
Top-1 error	20.44	20.30	20.05	19.88	19.95	20.24	20.53

Table 13
Comparisons of the verification accuracies (%) on the LFW dataset.

	Triplet loss	Softmax loss	Softmax + Triplet	Softmax + EOL
CNN-Quick	86.40	88.72	89.90	**93.20**
AlexNet	90.00	94.03	94.88	**96.45**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

Table 14
Comparisons of the verification accuracies (%) on the YTF dataset.

	Triplet loss	Softmax loss	Softmax + Triplet	Softmax + EOL
CNN-Quick	77.71	79.10	81.71	**84.33**
AlexNet	81.83	85.55	87.10	**89.28**

The bold and underlined values represent the corresponding best results under the corresponding comparison conditions.

5 error rate by 1.90% in comparison with the baseline. The relative reductions have reached 6.17% and 9.60% respectively. These results once more substantiate the effectiveness of the EOL.

4.6. Experiments using ResNet model

To further verify the effectiveness of our method, we apply the EOL to the 20-layer ResNet model [18]. The three datasets, including CIFAR10, CIFAR100 and SVHN, are used in the evaluation. We complied with the same data preprocessing as in [18].

Table 10shows the test set top-1 error rates of CIFAR10, CIFAR100 and SVHN, respectively. From Table 10, we can see that even though we apply the proposed EOL to the ResNet, it can still effectively improve performance by 2.65% on CIFAR10, by 5.04% on CIFAR100, and by 1.05% on SVHN, respectively, compared to the respective baseline. The absolute improvement on SVHN is not as large as those on the other two datasets, this is because the test error rates is almost saturated on SVHN. However, on SVHN, the relative reduction of the test error rate has reached 38.18% compared with the corresponding ResNet baseline, which is quite significant. These results further substantiate the effectiveness of the proposed EOL.

4.7. Feature visualization

We utilize t-SNE [45] to visualize the learned feature vectors extracted from the penultimate layer of the CNN-Quick and NIN models on the CIFAR-10 test set, respectively. Figs. 2 and 3 show the respective feature visualizations for the two models. It can be observed that, the EOL makes the learned feature vectors have better between-class separability and within-class compactness compared to the respective baseline. Therefore the discriminative ability of the learned feature vectors is highly improved.

4.8. Sensitivity study of λ_1 and λ_2

The hyperparameters λ_1 and λ_2 introduced by the EOL are determined on a validation set. Specifically, based on CIFAR10 dataset, we randomly select 10,000 training images as a validation set. Then, we utilize the rest training images of CIFAR10 to train CNN-Quick + EL and CNN-Quick + OL, and utilize the validation set to determine λ_1 and λ_2, respectively. After λ_1 and λ_2 are determined on the validation set ($\lambda_1 = 6 \times 10^{-4}, \lambda_2 = 5 \times 10^{-5}$), we fix λ_1 and λ_2, and use the same value of them for the CNN-Quick model on the CIFAR10, CIFAR100 and SVHN datasets. For NIN, the selection method of λ_1 and λ_2 is the same as that of CNN-Quick. We set $\lambda_1 = 3 \times 10^{-4}, \lambda_2 = 5 \times 10^{-5}$ for NIN on the CIFAR10, CIFAR100, MNIST and SVHN datasets. For AlexNet and ResNet, due to limitations in computing resources and time, we adopt the same values of λ_1 and λ_2 as those of NIN.

We perform the sensitivity study to see whether the performance varies much with the changes of the hyperparameter λ_1 and λ_2. To save the computational resources and time, we perform this study only with CNN-Quick and the CIFAR10 dataset. Specifically, we set λ_1 and λ_2 to values chosen from the predefined ranges, train CNN-Quick with these hyperparameter values on the CIFAR10 training set, and then report the top-1 error rates on the CIFAR10 test set.

To study the sensitivity of λ_1, we run CNN-Quick+EL on the CIFAR10 dataset, and change λ_1 near 6×10^{-4}, in the range of $\{3 \times 10^{-4}, 4 \times 10^{-4}, 5 \times 10^{-4}, 6 \times 10^{-4}, 7 \times 10^{-4}, 8 \times 10^{-4}, 9 \times 10^{-4}\}$. Table 11 shows the test error rates. It is seen that the test error rate does not change much, and is consistently lower than the baseline.

To study the sensitivity of λ_2, we run CNN-Quick + OL on the CIFAR10 dataset, and change λ_2 near 5×10^{-5}, in the range of $\{2 \times 10^{-5}, 3 \times 10^{-5}, 4 \times 10^{-5}, 5 \times 10^{-5}, 6 \times 10^{-5}, 7 \times 10^{-5}, 8 \times 10^{-5}\}$. Table 12 shows the test error rates. It is seen that the test error rate does not change much, and is consistently lower than the baseline.

5. Experiments with face verification task

Face verification is to distinguish whether two face images are from the same person (1:1 matching). Recently, many methods [29,30,46] utilize CNNs and the triplet loss to learn discriminative feature representations for face verification.

We conduct experimental evaluations using the CNN-Quick and AlexNet models, respectively. Both of them are adapted to the 128 × 128 input size. We employ triplet loss, softmax loss, softmax + triplet and softmax + EOL to train the two models respectively, where, softmax + triplet refers to the corresponding model is trained with the softmax loss and triplet loss, softmax + EOL refers to the corresponding model is trained with the softmax loss and EOL. The outputs of the penultimate layer in each model are

extracted to represent the input image, and utilized for face verification.

5.1. Training methodology and data preprocessing

The two models are trained on the CASIA WebFace dataset [47]. The LFW dataset [48] and the YTF dataset [49] are used to test them, respectively. The WebFace dataset consists of 493,456 face images of 10,575 persons, all of which are converted to gray-scale and normalized to 144 × 144 via landmarks. During the training process, we randomly crop 128 × 128 patches from each training image to augment the training data. The LFW dataset contains 13,233 images of 5749 persons, and all the images are also processed by the same pipeline as the training dataset and normalized to 128 × 128. The YTF dataset consists of 3425 YouTube videos of 1595 persons. The data preprocessing of the YTF dataset is the same as that of the LFW dataset.

5.2. Verification results

We follow the verification protocol described in [19,48,49]. Tables 13 and 14 show the evaluation results on LFW and YTF, respectively. It is seen that softmax + EOL is significantly better than the triplet loss, softmax loss and softmax + triplet. These results once again substantiate the effectiveness of our method.

6. Conclusion

novel discriminative feature learning method is proposed to improve object recognition performance of CNN. Specifically, we apply the proposed entropy–orthogonality loss (EOL) to the penultimate layer of the CNNs in the training phase. The EOL explicitly makes the feature vectors learned by a CNN model have the properties below: (1) each dimension of the feature vectors only responds strongly to as few classes as possible, and (2) the feature vectors from different classes are as orthogonal as possible. When combined with the softmax loss, the EOL not only can enlarge the differences of the between-class feature vectors, but also can reduce the variations of the within-class feature vectors. Therefore the discriminative ability of the learned feature vectors is highly improved. The EOL is general and independent of the CNN structure. Comprehensive experimental evaluations substantiate the effectiveness of our method.

Acknowledgments

This work is supported by National Basic Research Program of China (973 Program) under grant no. 2015CB351705, and the National Natural Science Foundation of China (NSFC) under grant no. 61332018.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[2] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, C.-L. Liu, LG-CNN: from local parts to global discrimination for fine-grained recognition, Pattern Recognit. 71 (2017) 118–131.
[3] S. Ma, S.A. Bargal, J. Zhang, L. Sigal, S. Sclaroff, Do less and achieve more: training CNNs for action recognition utilizing action images from the web, Pattern Recognit. 68 (2017) 334–345.
[4] L. Wang, L. Ge, R. Li, Y. Fang, Three-stream CNNs for action recognition, Pattern Recognit. Lett. 92 (2017) 33–40.
[5] W. Shi, Y. Gong, X. Tao, J. Wang, N. Zheng, Improving CNN performance accuracies with min-max objective, IEEE Trans. Neural Netw. Learn. Syst. (2017), doi:10.1109/TNNLS.2017.2705682.
[6] G. Cheng, P. Zhou, J. Han, Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images, IEEE Trans. Geosci. Remote Sens. 54 (12) (2016) 7405–7415.
[7] X. Yao, J. Han, G. Cheng, X. Qian, L. Guo, Semantic annotation of high-resolution satellite images via weakly supervised learning, IEEE Trans. Geosci. Remote Sens. 54 (6) (2016) 3660–3671.
[8] G. Cheng, Z. Li, X. Yao, L. Guo, Z. Wei, Remote sensing image scene classification using bag of convolutional features, IEEE Geosci. Remote Sens. Lett. 14 (10) (2017) 1735–1739.
[9] Y.-C. Wu, F. Yin, C.-L. Liu, Improving handwritten chinese text recognition using neural network language models and convolutional neural network shape models, Pattern Recognit. 65 (2017) 251–264.
[10] X. Bai, B. Shi, C. Zhang, X. Cai, L. Qi, Text/non-text image classification in the wild with convolutional neural networks, Pattern Recognit. 66 (2017) 437–446.
[11] Y. Lu, H. Wu, L. Zhou, Z. Wu, Multi-environment model adaptation based on vector Taylor series for robust speech recognition, Pattern Recognit. 43 (2010) 3093–3099.
[12] E. Trentin, S. Scherer, F. Schwenker, Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognit. Lett. 66 (2015) 4–12.
[13] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[15] L. Wan, M. Zeiler, S. Zhang, Y.L. Cun, R. Fergus, Regularization of neural networks using dropconnect, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 1058–1066.
[16] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.
[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[19] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 499–515.
[20] W. Shi, Y. Gong, J. Wang, N. Zheng, Integrating supervised Laplacian objective with CNN for object recognition, in: Proceedings of the Pacific Rim Conference on Multimedia, 2016, pp. 64–73.
[21] G. Cheng, C. Yang, X. Yao, L. Guo, J. Han, When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns, IEEE Trans. Geosci. Remote Sens. (2018), doi:10.1109/TGRS.2017.2783902.
[22] W. Shi, Y. Gong, J. Wang, Improving CNN performance with min-max objective, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 2004–2010.
[23] W. Shi, Y. Gong, X. Tao, N. Zheng, Training DCNN by combining max-margin, max-correlation objectives, and correntropy loss for multilabel image classification, IEEE Trans. Neural Netw. Learning Syst. (2017), doi:10.1109/TNNLS.2017.2705222.
[24] Z. Jiang, Z. Lin, H. Ling, F. Porikli, L. Shao, P. Turaga, Discriminative feature learning from big data for visual recognition, Pattern Recognit. 48 (10) (2015) 2961–2963.
[25] C. Li, Q. Liu, W. Dong, F. Wei, X. Zhang, L. Yang, Max-margin-based discriminative feature learning, IEEE Trans. Neural Netw. Learn. Syst. 27 (12) (2016) 2768–2775.
[26] M.-K. Zhou, X.-Y. Zhang, F. Yin, C.-L. Liu, Discriminative quadratic feature learning for handwritten Chinese character recognition, Pattern Recognit. 49 (2016) 7–18.
[27] Z. Zhang, S. Liu, X. Mei, B. Xiao, L. Zheng, Learning completed discriminative local features for texture classification, Pattern Recognit. 67 (2017) 263–275.
[28] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, Exemplar based deep discriminative and shareable feature learning for scene image classification, Pattern Recognit. 48 (10) (2015) 3004–3015.
[29] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 1988–1996.
[30] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[31] T. Serre, A. Oliva, T. Poggio, A feedforward architecture accounts for rapid categorization, Proc. Natl. Acad. Sci. 104 (15) (2007) 6424–6429.
[32] J.J. DiCarlo, D. Zoccolan, N.C. Rust, How does the brain solve visual object recognition? Neuron 73 (3) (2012) 415–434.
[33] S. Zhang, Y. Gong, J. Wang, Improving DCNN performance with sparse category-selective objective function., in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 2343–2349.
[34] M. Lin, Q. Chen, S. Yan, Network in network, arXiv:1312.4400 (2013).
[35] L. Hu, L. Dai, J. Wu, Convergent projective non-negative matrix factorization with Kullback–Leibler divergence, Pattern Recognit. Lett. 36 (2014) 15–21.
[36] M. Ponti, J. Kittler, M. Riva, T. de Campos, C. Zor, A decision cognizant Kullback–Leibler divergence, Pattern Recognit. 61 (2017) 470–478.
[37] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: Artificial Intelligence and Statistics, 2015, pp. 562–570.

[38] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, 2014, pp. 675–678.
[39] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, University of Toronto, 2009 Master's thesis.
[40] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[41] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in: Neural Information Processing Systems (NIPS) Workshop on Deep Learning and Unsupervised Feature Learning, vol. 2011, 2011, p. 5.
[42] M.D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, arXiv:1301.3557 (2013).
[43] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, Maxout networks, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 1319–1327.
[44] J.T. Springenberg, M. Riedmiller, Improving deep neural networks with probabilistic maxout units, arXiv:1312.6116 (2013).
[45] L.V.D. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (November) (2008) 2579–2605.
[46] W. Deng, J. Hu, N. Zhang, B. Chen, J. Guo, Fine-grained face verification: FGLFW database, baselines, and human-dcmn partnership, Pattern Recognit. 66 (2017) 63–73.
[47] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, arXiv:1411.7923 (2014).
[48] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report, University of Massachusetts, Amherst, 2007.
[49] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 529–534.

Weiwei Shi received the M.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2012, where he is currently pursuing the Ph.D. degree with the Institute of Artificial Intelligence and Robotics. His current research interests include image classification, image and video analysis, machine learning, and deep learning.

Yihong Gong (SM'12-F'17) received the B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from the University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively. He was an assistant professor with the School of Electrical and Electronic Engineering, Nanyang Technological University of Singapore, for four years. From 1996 to 1998, he was a Project Scientist with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. In 1999, he joined NEC Laboratories America, Princeton, NJ, USA, and established the Media Analytics Group for the labs. In 2006, he became the Site Manager to lead the entire Cupertino branch of the labs. In 2012, he joined Xi'an Jiaotong University, Xi'an, China, and became a Distinguished Professor of the National Thousand Talents Program, the Vice Director of the National Engineering Laboratory for Visual Information Processing, and the Chief Scientist of the China National Key Basic Research Project (973 Project). His current research interests include pattern recognition, machine learning, and multimedia content analysis.

De Cheng received the B.S. degree in automation control from Xi'an Jiaotong University, Xi'an, China, in 2011, and is currently a Ph.D. candidate in the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University and taking his visit in Carnegie Mellon University, Pittsburgh, PA, USA, to do some cooperation research. His research interests include pattern recognition, machine learning, and multimedia analysis.

Xiaoyu Tao received the B.S. degree in software engineering from Xi'an Jiaotong University, Xi'an, China, in 2014, where he is currently pursuing the Ph.D. degree in pattern recognition with the Institute of Artificial Intelligence and Robotics. His current research interests include image classification, object detection, and face recognition.

Nanning Zheng (SM'93-F'06) received the B.S. degree from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975; the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981; and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985. He joined Xi'an Jiaotong University in 1975, where he is currently a professor and the director of the Institute of Artificial Intelligence and Robotics. His current research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems. Dr. Zheng became a member of the Chinese Academy of Engineering in 1999. He is the Chinese Representative of the Governing Board of the International Association for Pattern Recognition. He also serves as an Executive Deputy Editor of the Chinese Science Bulletin.