Bilinear Deep Learning for Image Classification

Sheng-hua Zhong
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon
999077 Hong Kong, P. R. China
csshzhong@comp.polyu.edu.hk

Yan Liu
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon
999077 Hong Kong, P. R. China
csyliu@comp.polyu.edu.hk

Yang Liu
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon
999077 Hong Kong, P. R. China
csygliu@comp.polyu.edu.hk

ABSTRACT

Image classification is a well-known classical problem in multimedia content analysis. This paper proposes a novel deep learning model called bilinear deep belief network (BDBN) for image classification. Unlike previous image classification models, BDBN aims to provide human-like judgment by referencing the architecture of the human visual system and the procedure of intelligent perception. Therefore, the multi-layer structure of the cortex and the propagation of information in the visual areas of the brain are realized faithfully. Unlike most existing deep models, BDBN utilizes a bilinear discriminant strategy to simulate the "initial guess" in human object recognition, and at the same time to avoid falling into a bad local optimum. To preserve the natural tensor structure of the image data, a novel deep architecture with greedy layer-wise reconstruction and global fine-tuning is proposed. To adapt real-world image classification tasks, we develop BDBN under a semi-supervised learning framework, which makes the deep model work well when labeled images are insufficient. Comparative experiments on three standard datasets show that the proposed algorithm outperforms both representative classification models and existing deep learning techniques. More interestingly, our demonstrations show that the proposed BDBN works consistently with the visual perception of humans.

Categories and Subject Descriptors

I.2.0 [**Artificial Intelligence**]: General –*cognitive simulation*;
I.2.6 [**Artificial Intelligence**]: Learning –*connection and neural nets*

General Terms: Algorithm

Keywords: Deep learning, bilinear discriminant projection, image classification.

1. INTRODUCTION

Image classification, a classical problem in multimedia content analysis, aims to understand the semantic meaning of visual information and determine the category of the images according to some predefined criteria [1]. Existing image classification methods can be roughly divided into two broad families of approaches: parametric and nonparametric classifiers. Parametric classifiers, also known as learning-based classifiers, require an intensive training phase of the classifier parameters (e.g., the parameters of SVM [2], Boosting [3], fragments and object parts [4], decision trees [5], web graphs [6], hierarchical classification models [7], etc.). To date, the leading image classifiers are parametric classifiers, particularly SVM-based methods. Nonparametric classifiers make their classification decisions directly on the data, and require no training of parameters [8]. Recently, in the literature on multimedia, many papers focused on the specific applications; for instance, landmark image classification [9], sports genre & view type classification [10], age images classification [11] and affective images classification [12] [13]. In addition, camera metadata are utilized for classification [14].

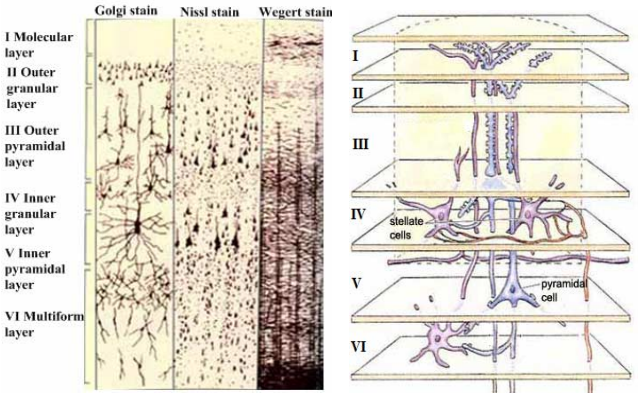
Despite more than fifteen years of extensive research, image classification for real-world applications remains a well-known challenge in the field of multimedia. But humans, even children, do not have difficulty with classifying images. Before the age of 25 months, children have already developed the ability to recognize novel three-dimensional objects [15]. Motivated by this fact, researchers in the fields of cognitive science and neuroscience have conducted pioneering work on modeling the human brain using computational architectures. Among these computational architectures, deep architecture composed of multiple layers of parameterized nonlinear modules is a representative paradigm that has achieved notable success in modeling the human visual system.

In this paper, we focus on designing a proper deep architecture and corresponding learning algorithms for the tasks of image classification. The latest research results and findings from neuroscience have indicated that the deep model is consistent with the physical structure, evolution of intelligence, and propagation of information in the human visual cortex. Thus, it shows great potential to provide human-like judgment using a human-like system in tasks of multimedia content analysis. The following sections contain a detailed discussion from three aspects:

1) Deep architecture is identical to the multi-layer physical structure of the human visual cortex. The neocortex, which is associated with many cognitive abilities, has a complex multi-layer hierarchy [16]. The laminar structure and a multi-layer illustration of the neocortex are shown in Figure 1. The neocortex can be roughly divided into six functionally distinct layers from Molecular layer I to Multiform layer VI. Layer IV in the primary visual cortex (V1) is further divided into four layers, labeled 4A, 4B, 4Cα, and 4Cβ. Therefore, dozens of cortical layers are involved in generating even the simplest vision [17].

*Area Chair: Nicu Sebe.

MM'11, November 28–December. 1, 2011, Scottsdale, Arizona, USA.



(a)Laminar structure of the neocortex

(b) Multi-layer illustration of the neocortex

Figure 1. Multi-layer structure of the cerebral cortex.

2) The development of intelligence follows with the multi-layer structure. From an evolutionary viewpoint, the phylogenetically most recent part of the brain is the neocortex. In humans and other primates, starting from catarhinians, the multi-layers structure began to appear in the neocortex [18]. Therefore, a deep architecture actually represents the result of human intelligence evolution. It thus provides a possible way to achieve the ultimate target of multimedia content analysis, which is to enable the computer to understand images semantically as humans do.

3) The manner in which data is delivered in a deep architecture is a good simulation of the information propagation in the visual cortex. There are several reasons for believing that our visual systems contain multi-layer generative models in which top-down connections can be used to generate low-level features of images from high-level representations, and bottom-up connections can be used to infer the high-level representations that would have generated an observed set of low-level features [19]. Single cell recordings and the reciprocal connectivity between cortical areas [20] both suggest a hierarchy of progressively more complex features in which each layer can influence the layers below it.

From considerations in the field of neuroscience, deep model is chosen in this paper for the task of image classification. To better adapt the image data and the image classification application, we propose a novel deep model called bilinear deep belief network (BDBN) with a new deep architecture and a new deep learning algorithm.

The deep architecture of BDBN is designed by referencing the human visual system and the human procedure of perception. In the primary visual cortex, all the way through the optic tract to a nerve position is a direct correspondence from an angular position in the field of view of the eye, just like a matrix. Therefore, the input layer and all hidden layers in BDBN are constructed by a set of second-order planes, which are also consistent with the natural tensor structure of images. All of these planes are fully connected with the adjacent ones until the output layer, which is a vector to indicate the label of the images.

Based on this new deep architecture, we propose a novel deep learning algorithm with three stages: bilinear discriminant initialization, greedy layer-wise reconstruction, and global fine-turning. The rationale for three-stage learning comes from the phenomenon of two peaks of activation in the visual cortex areas. With regard to object recognition, the early peak is related to the activation of an "initial guess" based on the discriminative knowledge that has been acquired, while the late peak reflects the post-recognition activation of conceptual knowledge related to the recognized object [21]. In most existing deep models, "post activation" is modeled by the fine-tuning stage, but the "initial guess" process is neglected. In our model, two peaks of activation and the propagation of information in the visual cortex are faithfully realized.

We model the peak activation of the "initial guess" by preserving the disciminant information of the labeled data to the greatest extent. Most existing deep models initialize the parameter space in a random manner and gradually approximate a locally optimal solution by learning. Unfortunately, a bad initial parameter space may lead to a poor local optimum and thus seriously affect the following learning procedure. To address this problem, we utilize a bilinear discriminant strategy to construct a second-order plane from the lower layer. The symmetrically weighted connections between these two adjacent layers are used as the initial parameter space for further learning. Moreover, the discriminant-based "initial guess" brings an additional advantage to the meaningful architecture. Currently, the number of neurons in each layer is fixed and pre-defined intuitively. In our model, the size of the deep architecture is determined based on the optimum dimension for retaining the discriminant information.

Last but not least, we develop our deep model under a semi-supervised learning framework because of the insufficiency of the labeled images in real-world applications. However, when relying on the efforts of experienced human annotators, labeled instances are often difficult, expensive, or time consuming to obtain [22]. By contrast, with the growing availability of a large number of images from photo-sharing sites such as Flickr, abundant unlabeled data are available [23].

The remainder of this paper is organized as follows. Related work on deep learning is reviewed in Section 2. A novel deep architecture and a new deep learning algorithm are introduced in Section 3. Section 4 shows the performance of the proposed techniques in real image classification tasks and Section 5 concludes this paper.

2. RELATED WORK ON DEEP LEARNING

Different from shallow learning models, deep learning is about learning multiple levels of representation and abstraction that helps to make sense of data. Besides evidence from neuroscience, some theoretical analyses from machine learning also provide support for the argument that deep models are more compact and expressive than shallow models in representing most learning functions, especially highly variable ones. For example, to model the d-dimensional parity function, Gaussian SVM uses $O(d2^d)$ parameters while deep learning only needs $O(d^2)$ parameters with $O(log_2 d)$ hidden layers [24]. The effectiveness of a deep model makes it promising for use in solving hard learning problems, for example, in semantically identifying the class of images from low-level visual features.

The performance of deep learning has been notable, especially after the introduction of the deep belief networks (DBN) model. The learning procedure of DBN can be divided into two stages: abstracting information layer by layer and fine-tuning the whole deep network to the ultimate learning target [25]. Figure 2 shows a DBN with one input layer H^1, three hidden layers H^2, H^3, H^4, while x is the unfolding vector of input data, and y is

the learning target. In the first stage, DBN pairs each feed-forward layer with a feed-back layer that attempts to reconstruct the input of the layer from the output. In Figure 2, the layer-wise reconstruction happens between H^1 and H^2, H^2 and H^3, H^3 and H^4, which is implemented by a family of Restricted Boltzmann Machines (RBMs) [26]. After a greedy unsupervised learning of each pair of layers, the lower-level features are progressively combined into more compact high-level representations. The whole deep network is then refined using a contrastive version of the "wake-sleep" algorithm via a global gradient-based optimization strategy.

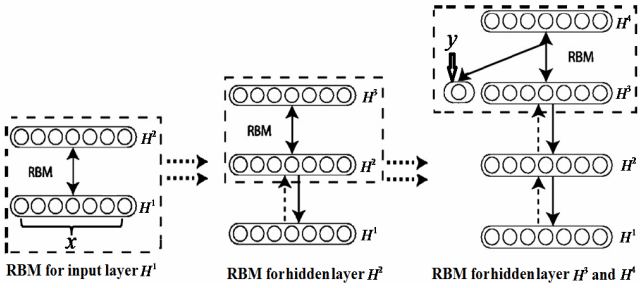


Figure 2. Structure of the deep belief network (DBN).

Owing to this two-stage fast greedy learning, deep learning has also exhibited a notable performance under a situation of insufficient training data [27]. Weston et al. simply leveraged shallow algorithms to deep architecture and already achieved a competitive performance in semi-supervised learning tasks [28]. DBN-rNCA is a semi-supervised learning algorithm that combines DBN architecture and neighborhood component analysis (NCA) techniques for dimensionality reduction [27]. Experimental validations have demonstrated that DBN-rNCA obviously improves the performance of handwritten digit recognition by using abundant unlabeled data. Zhou et al. proposed a new semi-supervised classifier called discriminative deep belief network (DDBN) [29], which integrated the abstraction ability of DBN for unlabeled data and the discriminative ability of the backpropagation strategy for labeled data. Moreover, empirical validations in various real-world applications have shown that DBN performs impressively in analyses of visual data, such as in image classification [27], image annotation [30], and image retrieval [31].

In recent years, deep convolutional architectures have been attracting an increasing amount of attention because of their ability to preserve the space structure and resistance to small variations in the images [32][33]. As early as in 1989, LeCun et al. proposed a convolutional network that used a feature detection layer followed by a feature pooling layer as the basic module, and that was trained to minimize the overall loss for classification [34]. While the convolutional nets are deep, i.e., including a series of multiple detection/pooling modules, they do not seem to suffer from the convergence problems that plague deep fully-connected neural nets [35]. Similar with DBN, deep convolutional network (DCNN) has no distinct feature extractor and classifier. All of the layers in DCNN are trained from data in an integrated fashion. Currently, DCNN has been successfully used to extract spatial features [36] and spatial-temporal features [33] in different applications, such as image classification [6] [37] and human action recognition [38].

3. BILINEAR DEEP LEARNING MODEL

In this section, we propose a novel learning framework based on bilinear deep belief network (BDBN). Our bilinear deep belief network, which is aimed at the task of image classification, is demonstrated in Section 3.1. The bilinear discriminant initialization stage is discussed in Section 3.2. Section 3.3 contains details of the greedy layer-wise reconstruction. The global fine-tuning process of the whole deep network is described in Section 3.4. We provide the procedure of BDBN in Section 3.5.

3.1 Bilinear Deep Belief Network

Let X be a set of data samples as shown below:

$$X = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_k, ..., \mathbf{X}_K] \tag{1}$$

where $\mathbf{X}_k$ is a sample datum in the image space $\mathbb{R}^{I \times J}$ and K is the number of sample data. Let Y be a set of labels corresponding to X, which can be seen as:

$$Y = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k, ..., \mathbf{y}_K] \tag{2}$$

And $\mathbf{y}_k$ is the label vector of $\mathbf{X}_k$ in $\mathbb{R}^C$, where C is the number of classes.

$$y_k^c = \begin{cases} 1 & \text{if } \mathbf{X}_k \in c\text{th class} \\ 0 & \text{if } \mathbf{X}_k \notin c\text{th class} \end{cases} \tag{3}$$

Based on the given training set, the aim in image classification is to learn a mapping function from the image set X to the label set Y, and then classify the new coming data points according to the learned mapping function.

To address the problem of image classification, we propose a novel bilinear deep learning technique BDBN. Figure 3 shows the architecture of BDBN. A fully interconnected directed belief network includes input layer H^1, hidden layer H^2,…, H^N, and one label layer La at the top. The input layer H^1 has $I \times J$ units, and this size is equal to the dimension of the input features. In our model, we use the pixel values of sample datum $\mathbf{X}_k$ as the original input features. In the top, the label layer has C units, which is equal to the number of classes. The search of the mapping function from X to Y is transformed to the problem of finding the optimum parameter space θ^* for the deep architecture.

The learning procedure of our proposed BDBN is listed below:

1. The strategy of bilinear discriminant projection is utilized to construct a projection to map the original data into a discriminant bilinear subspace.
2. The initial symmetrically weighted connections are constructed between adjacent layers according to the "initial guess" based on the discriminant information. The size of the deep architecture is determined automatically based on the optimum dimension to retain the discriminant information.
3. After the architecture of the next layer is determined, the parameter space is refined by the greedy layer-wise information reconstruction using RBMs as building blocks.
4. Repeat the first to third stages until the parameter space θ in all N layers is constructed.
5. In the "post activation" stage, the whole deep model is fine-tuned to minimize the classification error based on backpropagation.

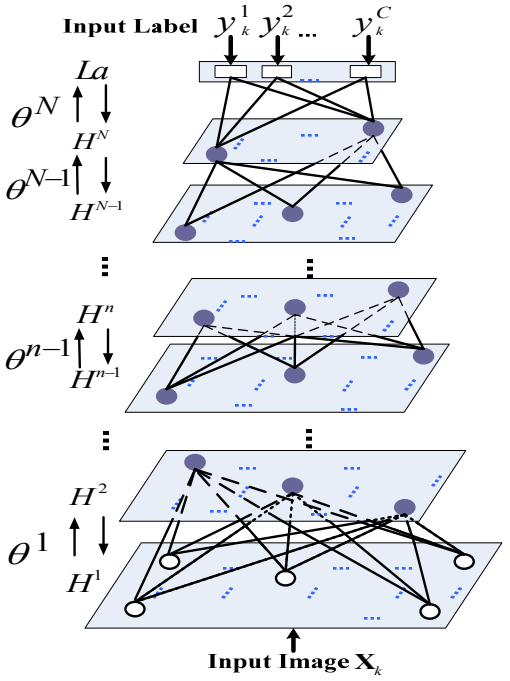


Figure 3. Architecture of the bilinear deep belief network.

3.2 Bilinear Discriminant Initialization

In this subsection, we introduce the bilinear discriminant projection (BDP), which is used to extract the discriminant information from the original image datasets.

Given the labeled training data points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_L \in \mathbb{R}^{I \times J}$, without unfolding the input data to vectors, BDP aims to find two projection matrices $\mathbf{U} \in \mathbb{R}^{I \times P}$ and $\mathbf{V} \in \mathbb{R}^{J \times Q}$ such that by $\mathbf{TX}_s = \mathbf{U}^T \mathbf{X}_s \mathbf{V}$ (s = 1, ..., L), just as depicted in Figure 4, the latent representation $\mathbf{TX}_1, \mathbf{TX}_2, \ldots, \mathbf{TX}_L \in \mathbb{R}^{P \times Q}$ can be obtained.

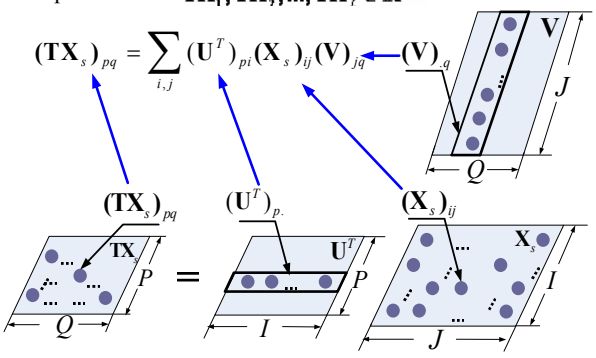


Figure 4. Latent representation with projection matrices $\mathbf{U}$ and $\mathbf{V}$.

In order to preserve the discriminant information in the learning procedure, the objective function of BDP could be represented as follows:

$$\arg\max_{\mathbf{U},\mathbf{V}} J(\mathbf{U},\mathbf{V}) = \sum_{s,t=1}^{K} \| \mathbf{U}^T (\mathbf{X}_s - \mathbf{X}_t) \mathbf{V} \|^2 (\alpha \mathbf{B}_{st} - (1-\alpha) \mathbf{W}_{st}) \quad (4)$$
$$s.t.\ \mathbf{U}^T \mathbf{U} = \mathbf{I}_P,\ \mathbf{V}^T \mathbf{V} = \mathbf{I}_Q$$

where $\alpha \in [0,1]$ is the parameter used to balance the between-class weights $\mathbf{B}_{st}$ and the within class weights $\mathbf{W}_{st}$, which are defined as follows [39][40]:

$$\mathbf{B}_{st} = \begin{cases} \frac{1}{n_d} - \frac{1}{n_c}, & \text{if } \mathbf{y}_s^c = \mathbf{y}_t^c = 1, \\ \frac{1}{n_d}, & \text{else,} \end{cases}, \quad \mathbf{W}_{st} = \begin{cases} \frac{1}{n_c}, & \text{if } \mathbf{y}_s^c = \mathbf{y}_t^c = 1, \\ 0, & \text{else,} \end{cases} \quad (5)$$

where $\mathbf{y}_s^c$ denotes the class label of datum point $\mathbf{X}_s$, n_d is the number of data points in all classes and n_c is the number of data points in class c, where $c \in \{1, \ldots, C\}$.

By simultaneously maximizing the distances between data points from different classes and minimizing the distances between data points from the same class, the discriminant information is preserved to the greatest extent in the projected feature space. Optimizing $J(\mathbf{U},\mathbf{V})$ is a non-convex optimization problem with respect to the projection matrices $\mathbf{U}$ and $\mathbf{V}$. However, solving $\mathbf{U}$ (or $\mathbf{V}$) with fixed $\mathbf{V}$ (or $\mathbf{U}$) is a convex optimization problem. Let $\mathbf{E}_{st} = \alpha \mathbf{B}_{st} - (1-\alpha) \mathbf{W}_{st}$, with the fixed $\mathbf{V}$. The optimal $\mathbf{U}$ is composed of the first P eigenvectors of the following eigendecomposition problem:

$$\mathbf{D}_\mathbf{V} \mathbf{u} = \lambda \mathbf{u} \quad (6)$$

where $\mathbf{D}_\mathbf{V} = \sum_{st} \mathbf{E}_{st} (\mathbf{X}_s - \mathbf{X}_t) \mathbf{V} \mathbf{V}^T (\mathbf{X}_s - \mathbf{X}_t)^T$. Similarly, with the fixed $\mathbf{U}$, the optimal $\mathbf{V}$ is composed of the first Q eigenvectors of the following eigendecomposition problem:

$$\mathbf{D}_\mathbf{U} \mathbf{v} = \lambda \mathbf{v} \quad (7)$$

where $\mathbf{D}_\mathbf{U} = \sum_{st} \mathbf{E}_{st} (\mathbf{X}_s - \mathbf{X}_t)^T \mathbf{U} \mathbf{U}^T (\mathbf{X}_s - \mathbf{X}_t)$.

Therefore, we can alternately optimize $\mathbf{U}$ (with a fixed $\mathbf{V}$) and $\mathbf{V}$ (with a fixed $\mathbf{U}$). The above steps monotonically increase $J(\mathbf{U},\mathbf{V})$ and since the function is upper bounded, it will converge to a critical point with transformation matrices $\mathbf{U}$, $\mathbf{V}$.

The sizes of P and Q are determined by the number of positive eigenvalues in $\mathbf{D}_\mathbf{V}$ and $\mathbf{D}_\mathbf{U}$, respectively, since adding the eigenvectors corresponding to the nonpositive eigenvalues will not increase $J(\mathbf{U},\mathbf{V})$ in Equation (4). As a result, the original dimension $I \times J$ is automatically reduced into $P \times Q$.

3.3 Greedy Layer-Wise Reconstruction

The sample data set $\mathbf{X}$ is inputted to the deep architecture as the input layer H^1 to construct an RBM with the first hidden layer H^2.

The energy of the state $(\mathbf{h}^1, \mathbf{h}^2)$ in the first RBM is:

$$\begin{aligned} & E(\mathbf{h}^1, \mathbf{h}^2; \theta^1) \\ &= -(\mathbf{h}^1 \mathbf{A}^1 \mathbf{h}^2 + \mathbf{b}^1 \mathbf{h}^1 + \mathbf{c}^1 \mathbf{h}^2) \\ &= -\sum_{i=1,j=1}^{i \le I, j \le J} \sum_{p=1,q=1}^{p \le P^2, q \le Q^2} h_{ij}^1 A_{ij,pq}^1 h_{pq}^2 - \sum_{i=1,j=1}^{i \le I, j \le J} b_{ij}^1 h_{ij}^1 - \sum_{p=1,q=1}^{p \le P^2, q \le Q^2} c_{pq}^1 h_{pq}^2 \end{aligned} \quad (8)$$

where $\theta^1 = (\mathbf{A}^1, \mathbf{b}^1, \mathbf{c}^1)$ are the model parameters between the input layer H^1 and first hidden layer H^2. $A_{ij,pq}^1$ is the symmetric

interaction term between the input unit (i,j) in H^1 and the hidden unit (p,q) in H^2. b_{ij}^1 is the $(i,j)^{th}$ bias of layer H^1 and c_{pq}^1 is the $(p,q)^{th}$ bias of layer H^2. $I \times J$ is the number of units in H^1, while $P^2 \times Q^2$ is the number of units in H^2. Therefore, the first RBM has the following joint distribution:

$$P\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)=\frac{1}{Z}e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)}=\frac{e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)}}{\sum_{\mathbf{h}^1}\sum_{\mathbf{h}^2}e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)}} \tag{9}$$

where Z is the normalization constant. The probability of the model assigned to $\mathbf{h}^1$ in H^1 is:

$$P\left(\mathbf{h}^1\right)=\frac{1}{Z}\sum_{\mathbf{h}^2}e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)}=\frac{\sum_{\mathbf{h}^2}e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)}}{\sum_{\mathbf{h}^1}\sum_{\mathbf{h}^2}e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)}} \tag{10}$$

And the log-likelihood of $P\left(\mathbf{h}^1\right)$ is:

$$\log P\left(\mathbf{h}^1\right)=\log\sum_{\mathbf{h}^2}e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)}-\log\sum_{\mathbf{h}^1}\sum_{\mathbf{h}^2}e^{-E\left(\mathbf{h}^1,\mathbf{h}^2;\theta^1\right)} \tag{11}$$

Gibbs sampling from an RBM proceeds by sampling $\mathbf{h}^2$ given $\mathbf{h}^1$, then sampling $\mathbf{h}^1$ given $\mathbf{h}^2$, and so on. The conditional distributions over input state $\mathbf{h}^1$ in layer H^1 and hidden state $\mathbf{h}^2$ in layer H^2 are given by the logistic functions Equation (12) and Equation (13), where $\sigma(x)=\frac{1}{1+\exp(-x)}$.

$$p\left(\mathbf{h}^2\mid\mathbf{h}^1\right)=\prod_{p,q}p\left(h_{pq}^2\mid\mathbf{h}^1\right),p\left(h_{pq}^2=1\mid\mathbf{h}^1\right)=\sigma\Big(\sum_{i=1,j=1}^{i\le I,j\le J}h_{ij}^1A_{ij,pq}^1+c_{pq}\Big) \tag{12}$$

$$p\left(\mathbf{h}^1\mid\mathbf{h}^2\right)=\prod_{i,j}p\left(h_{ij}^1\mid\mathbf{h}^2\right),p\left(\mathrm{h}_{ij}^1=1\mid\mathbf{h}^2\right)=\sigma\Big(\sum_{p=1,q=1}^{p\le P^2,q\le Q^2}A_{ij,pq}^1h_{pq}^2+b_{ij}\Big) \tag{13}$$

Denote $\mathbf{h}^2(t)$ for the t^{th} of $\mathbf{h}^2$ sample from the chain, starting at $t=0$ with $\mathbf{h}^1(\mathbf{0})$, which is the input observation for the RBM, and $(\mathbf{h}^2(t),\mathbf{h}^1(t))$ for $t\longrightarrow\infty$ is a sample from the Markov chain. Therefore, we can calculate the derivative of Equation (11) with respect to the parameter $\theta^1=\left(\mathbf{A}^1,\mathbf{b}^1,\mathbf{c}^1\right)$ below:

$$\begin{aligned}\frac{\partial\log p(\mathbf{h}^1(0))}{\partial\theta^1}&=-\sum_{\mathbf{h}^2(0)}p(\mathbf{h}^2(0)\mid\mathbf{h}^1(0))\frac{\partial E(\mathbf{h}^2(0),\mathbf{h}^1(0))}{\partial\theta^1}+\\&\sum_{\mathbf{h}^2(t)}\sum_{\mathbf{h}^1(t)}p(\mathbf{h}^2(t),\mathbf{h}^1(t))\frac{\partial E(\mathbf{h}^2(t),\mathbf{h}^1(t))}{\partial\theta^1}\end{aligned} \tag{14}$$

The idea of the Contrastive Divergence [41] algorithm using the difference between two Kullback-Liebler divergences is to take t small (typically $t=1$) to run the chain for only one step. When $t=1$, the derivative to the model parameter $\mathbf{A}^1$ can be obtained by Equation (15),

$$\begin{aligned}&\frac{\partial\log P(\mathbf{h}^1(0))}{\partial\mathbf{A}^1}\\&=-\sum_{\mathbf{h}^2(0)}P(\mathbf{h}^2(0)\mid\mathbf{h}^1(0))\frac{\partial E(\mathbf{h}^2(0),\mathbf{h}^1(0))}{\partial\mathbf{A}^1}+\sum_{\mathbf{h}^2(1)}\sum_{\mathbf{h}^1(1)}P(\mathbf{h}^2(1),\mathbf{h}^1(1))\frac{\partial E(\mathbf{h}^2(1),\mathbf{h}^1(1))}{\partial\mathbf{A}^1}\\&=<\mathbf{h}^1(0)\mathbf{h}^2(0)>_{data}-<\mathbf{h}^1(1)\mathbf{h}^2(1)>_{recon}\end{aligned} \tag{15}$$

where $\langle\cdot\rangle_{data}$ denotes an expectation with respect to the data distribution and $\langle\cdot\rangle_{recon}$ denotes the "reconstruction" distribution of data after one step. This leads to a simple learning rule for performing the stochastic steepest ascent in the log probability of the training data in Equation (16) and Equation (17).

$$A_{ij,pq}^1=\vartheta A_{ij,pq}^1+\Delta A_{ij,pq}^1 \tag{16}$$

$$\Delta A_{ij,pq}^1=\varepsilon_{\mathbf{A}}(<h_{ij}^1(0)h_{pq}^2(0)>_{data}-<h_{ij}^1(1)h_{pq}^2(1)>_{recon}) \tag{17}$$

Other parameters in the θ^1 update function can be calculated in a similar manner.

$$b_{ij}^1=\vartheta b_{ij}^1+\Delta b_{ij}^1=\vartheta b_{ij}^1+\varepsilon_{\mathbf{b}}(h_{ij}^1(0)-h_{ij}^1(1)) \tag{18}$$

$$c_{pq}^1=\vartheta c_{pq}^1+\Delta c_{pq}^1=\vartheta c_{pq}^1+\varepsilon_{\mathbf{c}}(h_{pq}^2(0)-h_{pq}^2(1)) \tag{19}$$

where ϑ is the momentum and $\varepsilon_{\mathbf{A}}$, $\varepsilon_{\mathbf{b}}$, $\varepsilon_{\mathbf{c}}$ are the learning rate of model parameters $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$.

As far as we know, all existing deep learning models determine the structure, such as the sizes of the hidden layers, based on intuition. In our proposed model, we intend to provide a more meaningful architecture by integrating the determinative information from labeled data. To integrate discriminative information obtained from bilinear discriminant projection for classification, we have two procedures: determining the sizes of hidden layers and calculating the discriminative initial symmetrically weighted connections.

As described before, we find a bilinear projection that can automatically reduce the original dimension $I \times J$ to $P \times Q$ through the transformation matrices $\mathbf{U}^1$ and $\mathbf{V}^1$**.** As a result, the number of neurons in layer H^2 is determined by the row and column size of the transformation matrices $\mathbf{U}^1$ and $\mathbf{V}^1$.

$$P^2=row(\mathbf{U}^1),\ Q^2=column(\mathbf{V}^1) \tag{20}$$

Furthermore, in existing deep learning models, the weights of the symmetrical connections $\mathbf{A}$ are initialized to small random values chosen from a zero-mean Gaussian with a standard deviation of about 0.01. Differently from them, we set the discriminative transformation parameters obtained from the bilinear discriminant projection as the initial weights of the symmetrical connections by Equation (21).

$$A_{ij,pq}^1(0)=(\mathbf{U}_{ip}^1)^T\mathbf{V}_{jq}^1 \tag{21}$$

The above discussion is the greedy layer-wise abstraction for the first layer H^1 with its next adjacent layer H^2. Similar operations can be performed on the higher layer pairs.

3.4 Global Fine-Tuning

Above, we use the greedy layer-by-layer algorithm to learn a deep model with the help of discriminant information obtained from bilinear discriminant projection. In this section, we use backpropagation through the whole deep model to fine-tune the parameters $\theta = \mathbf{[A,b,c]}$ for optimal reconstruction.

In the greedy layer-by-layer information abstraction stage, a global search has been performed for a sensible and good region in the whole parameter space. Therefore, before proceeding to the process of fine-tuning, we have already constructed a good data concept extraction model. In our model, backpropagation is utilized to adjust the entire deep network to find good local optimum parameters $\theta^* = [\mathbf{A}^*, \mathbf{b}^*, \mathbf{c}^*]$ to effectively classify the data. In this stage, the learning algorithm is used to minimize the classification error $[-\sum_l \mathbf{y}_l \log \hat{\mathbf{y}}_l]$, where $\mathbf{y}_l$ and $\hat{\mathbf{y}}_l$ are the correct label and the output label value of labeled sample datum $\mathbf{X}_l$ in X^L.

3.5 Bilinear Deep Learning Algorithm

In this section, the detailed procedure of the BDBN is described in Algorithm 1.

Algorithm 1: Bilinear Deep Belief Network

Input: Training data set X, Labeled samples X^L in X

Corresponding labels set Y

Number of layers N, Number of epochs E

Number of labeled data L, Parameter α

Between-class weights $\mathbf{B}_{st}$, Within class weights $\mathbf{W}_{st}$

Initial bias parameters $\mathbf{b}$ and $\mathbf{c}$

Momentum ϑ and learning rate $\varepsilon_{\mathbf{A}}$, $\varepsilon_{\mathbf{b}}$, $\varepsilon_{\mathbf{c}}$

Output: Optimal parameter space $\theta^* = [\mathbf{A}^*, \mathbf{b}^*, \mathbf{c}^*]$

for n = 1,…, N **do**
  for e = 1,…, E **do**
    if n = 1
      $T^n = X^L$
    else
      for l = 1,…, L **do**
        $\mathbf{T}_l^n = \sigma(\mathbf{T}_l^{n-1} A^{n-1} + c^{n-1})$
      end for
    end if
    while not convergent **do**
      $\mathbf{D}_{\mathbf{V}} = \sum_{st} \mathbf{E}_{st} (\mathbf{T}_s^n - \mathbf{T}_t^n) \mathbf{V} \mathbf{V}^T (\mathbf{T}_s^n - \mathbf{T}_t^n)^T$
      $\mathbf{D}_{\mathbf{U}} = \sum_{st} \mathbf{E}_{st} (\mathbf{T}_s^n - \mathbf{T}_t^n)^T \mathbf{U} \mathbf{U}^T (\mathbf{T}_s^n - \mathbf{T}_t^n)$
      Fix $\mathbf{V}$, compute $\mathbf{U}$ by solving $\mathbf{D}_{\mathbf{V}} \mathbf{u} = \boldsymbol{\lambda} \mathbf{u}$
      Fix $\mathbf{U}$, compute $\mathbf{V}$ by solving $\mathbf{D}_{\mathbf{U}} \mathbf{v} = \boldsymbol{\lambda} \mathbf{v}$
    end while
    Determine the size of the next layer

$$P^{n+1} = row(\mathbf{U}^n), \; Q^{n+1} = column(\mathbf{V}^n)$$

Compute initial weights of the connections

$$A_{ij,pq}^n(0) = (\mathbf{U}_{ip}^n)^T \mathbf{V}_{jq}^n$$

Calculate the state of the next layer

$$p\left(h_{pq}^{n+1} = 1 \mid \mathbf{h}^n\right) = \sigma\Big(\sum_{i=1,j=1}^{i \le P^n, j \le Q^n} h_{ij}^n A_{ij,pq}^n + c_{pq}^n\Big)$$

$$p\left(\mathrm{h}_{ij}^n = 1 \mid \mathbf{h}^{n+1}\right) = \sigma\Big(\sum_{p=1,q=1}^{p \le P^{n+1}, q \le Q^{n+1}} A_{ij,pq}^n h_{pq}^{n+1} + b_{ij}^n\Big)$$

Update the weights and biases

$$A_{ij,pq}^n = \vartheta A_{ij,pq}^n + \varepsilon_{\mathbf{A}} (< h_{ij}^n(0) h_{pq}^{n+1}(0) >_{data} - < h_{ij}^n(1) h_{pq}^{n+1}(1) >_{recon})$$

$$b_{ij}^1 = \vartheta b_{ij}^1 + \varepsilon_{\mathbf{b}} (h_{ij}^1(0) - h_{ij}^1(1))$$

$$c_{pq}^1 = \vartheta c_{pq}^1 + \varepsilon_{\mathbf{c}} (h_{pq}^2(0) - h_{pq}^2(1))$$

  end for
end for

Calculate optimal parameter space $\theta^* = \arg\min_{\theta} [-\sum_l \mathbf{y}_l \log \hat{\mathbf{y}}_l]$

4. EXPERIMENTS AND RESULTS

In this section, three standard datasets with different kinds of visual data are used to demonstrate the performance of the proposed BDBN. The first dataset is the Caltech101, a standard dataset for image classification, which includes images of 100 different objects plus a background category [42]. In this paper, we use images from the first five categories. The second dataset is the Urban and Natural Scene. This dataset is composed of 2,688 color images with eight categories [43]. The third dataset is the CMU pose, illumination, and expression (PIE) dataset [44].

For simplicity, we set the balance weight α as 0.5 in our experiments. For parameters such as the learning rate and the momentum in the deep learning model, we simply follow the general setting of previous work on deep learning [45], although a more careful choice may lead to better performance. For example, in greedy layer wise learning, the number of epochs is fixed at 30 and the learning rate η is equal to 0.1.The initial momentum ϑ is 0.5. After five epochs, the momentum is set to 0.9. In the fine-tuning stage, the method of conjugate gradients is utilized and three line searches are performed in each epoch until convergence.

We compare the performance of BDBN with other representative classifiers, including k-nearest neighbor (KNN), support vector machines (SVM) [46], transductive SVM (TSVM) [47], neural network (NN) [48], EmbedNN [28], Semi-DBN [45], DBN-rNCA [27], DDBN [29], and DCNN [37]. KNN, a typical nonlinear classifier, is always used as the baseline for comparisons of performance. In this paper, we set k equal to 3. SVM and NN are two powerful methods of classification. EmbedNN is the semi-supervised version of NN with deep architecture. Semi-DBN, DBN-rNCA, and DDBN are the semi-supervised versions of DBN. As a new deep learning model, DCNN demonstrated great classification ability due to its ability to preserve visual locality and space structure.

4.1 Experiments on Caltech101

In this experiment, we work on the frequently used subset of the Caltech101 [29], which includes 2,935 images from the first five categories: 435 images of ‘‘Faces,’’ 435 images of ‘‘Faces_easy,’’ 798 images of ‘‘Motorbikes,’’ 467 images of ‘‘Back_google,’’ and 800 images of ‘‘Airplanes.’’ As shown in Figure 5, the images in the same category vary greatly.

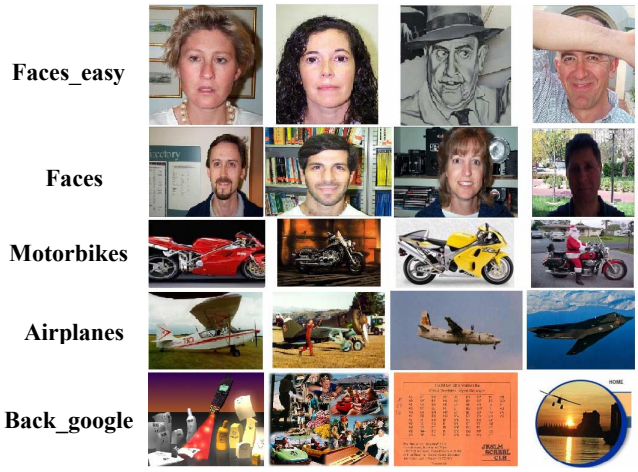


Figure 5. Sample images from the dataset Caltech101.

First, we compare the classification accuracy of different methods with a various number of labeled data. Because the number of images in each category in Caltech101 is different, 50 images are randomly selected for each category to form the test set and the rest to form the training set. As the previous setting in [29], the number of labeled data is equal to 5, 25, 50, and 75 per category, respectively. We perform 10 random splits and report the average results over the 10 trials. As shown in Table 1, the performance of BDBN on this dataset is stable and impressive.

Table 1. Classification accuracy rate (%) on the test data with different number of labeled data per category on Caltech101.

Num./Cat.	5	25	50	75
KNN	44.60	58.20	63.20	64.60
SVM	49.80	66.20	67.40	68.20
TSVM	50.00	70.20	70.50	72.80
NN	53.20	64.00	66.80	70.60
EmbedNN	51.20	55.50	58.60	64.00
Semi-DBN	55.40	65.80	67.60	69.60
DBN-rNCA	55.80	64.20	65.40	69.80
DDBN	58.30	71.40	72.00	74.20
DCNN	58.20	70.80	73.40	75.20
BDBN	**61.80**	**71.60**	**75.60**	**78.80**

Then, we compare the convergence of the proposed BDBN with two other deep learning models: Semi-DBN and DDBN, all of which have a fine-tuning stage. Figure 6 shows that BDBN converges much more quickly than Semi-DBN and DDBN. Although they are all deep learning models, BDBN requires an average of 106 iterations in comparison to 290 iterations for Semi-DBN and 161 iterations for DDBN. The improvement comes from the strategy of bilinear discriminant projection. This strategy helps BDBN to achieve a better “initial guess” when constructing the symmetrically weighted connections between layers.

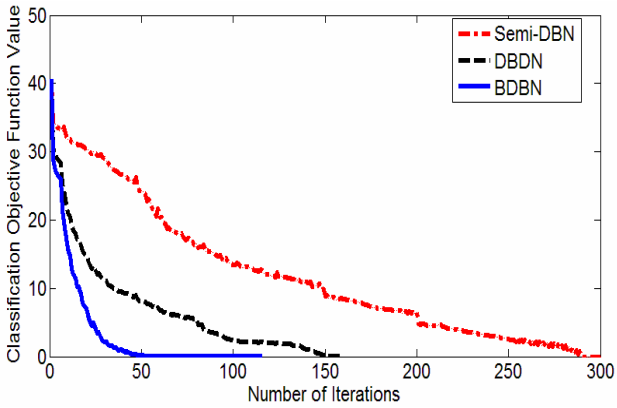


Figure 6. Convergence curve of Semi-DBN, DDBN and BDBN on Caltech 101.

In the third experiment, we visualize the parameter space between the input layer and the first hidden layer of BDBN. Some samples are shown in Figure 7. Obviously, BDBN abstracts the shape information from the training data and delivers it to the deeper layer for the determination of image classes.

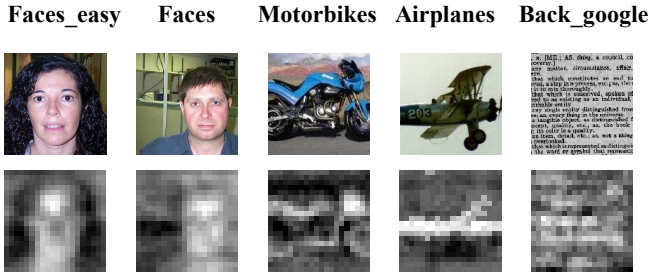


Figure 7. Some samples of parameter space visualization between the input layer and the first hidden layer of BDBN.

4.2 Experiments on Urban and Natural Scene

In this section, we demonstrate the performance of BDBN on the Urban and Natural Scene dataset [43]. This dataset is composed of 2,688 color images with eight categories, namely “coast & beach,” “highway,” “open country,” “tall building,” “forest,” “street,” “mountain,” and “city center.” In the preprocessing stage, images are downsampled to 32×32 as the input of BDBN. In our experiment, 50 images are randomly selected from each category to form the test set and the rest of the images are used for training. Sample images of each category are shown in Figure 8.

All existing deep learning models determine the structure, such as the sizes of the hidden layers, based on researchers’ intuition. In our model, the number of the neurons in each layer can be determined automatically based on bilinear discriminant strategy. Table 2 demonstrates this advantage by comparing the real running time and classification accuracy of BDBN with other five neural networks. The number of labeled data is equal to 5, 25, 50 and 75 per category, respectively. We perform 10 random splits and report the average results over the 10 trials. For BDBN, the number of neurons in layer H^1 is the size of the input image, i.e. 32×32.

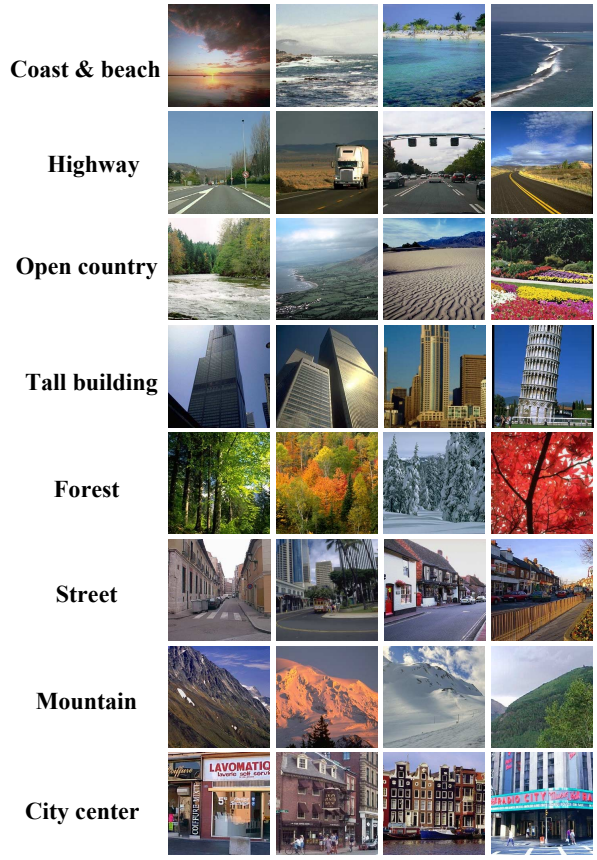


Figure 8. Sample images from the Urban & Natural Scene.

The number of neurons in H^2, H^3, H^4 is 24×24, 21×21, and 19×20, respectively. The classical setting of neurons numbers in H^2, H^3, H^4 are 500, 500, and 2000, according to previous publications. The results with different sizes of the deep architecture are provided for the models under comparison. In the table, "_d" is used to represent the compared models with the same size of BDBN, and "_c" is utilized to represent the compared models with the classical sizes. Clearly, BDBN has lower time complexity and better classification accuracy.

In Figure 9, we discuss the limitation of image classification based on visual similarity. Figure 9 (a) is a representative image of "Street", and Figure 9 (b) is a representative image of "Highway". Figure 9 (c) is classified to be "Highway" by BDBN and all other classifiers in this experiment, although the ground-truth of this image is "Street". Only according to visual similarity, Figure 9 (b) and Figure 9 (c) should be grouped together. However, human can give the correct judgment of Figure 9 (c) by referencing the buildings and cars along the street, which is a kind of contextual cueing acquired from past experiences of regularities. We list it as the future work of integrating contextual cueing in the deep modeling.

(a) (b) (c)

Figure 9. Limitation of image classification via visual similarity. (a) A representative image of "Street". (b) A representative image of "Highway". (c) The misclassified image. The ground-truth category of it is "Street" and the misclassified category is "Highway".

4.3 Experiments on CMU PIE

In this part, we demonstrate the performance of BDBN on image dataset of the CMU PIE dataset [44]. The CMU PIE face dataset contains 68 subjects with a total of 41,368 face images. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. As with the general setting of experiments to build the sub dataset [50], we use all the images under different illuminations and expressions with five near frontal poses (C05, C07, C09, C27, C29). In this way, about 170 images with the resolution of 32×32 are obtained for each individual. The preprocessing is applied following the general setting of experiment [50].

In the above experiments, the convolutional deep learning model demonstrates a better performance than other existing deep models. Therefore, in the experiment for dataset PIE, we compare the robustness of our deep model BDBN with that of the convolutional deep model DCNN. For the dataset, 120 images are randomly selected for each person to form the training set and the rest to form the test set. We perform 10 random splits and report the average results over the 10 trials.

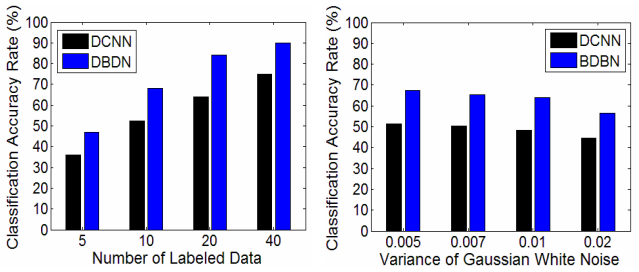
First, we compare the influence from different number of labeled data with the same extent of noise. Of the 120 images for each person, different numbers of images are randomly selected and labeled while the others remain unlabeled. The number of labeled

Table 2. Comparisons of run-time (s) and classification accuracy (%) with different labeled numbers and different deep architectures.

Num. / Cat.	5		25		50		75	
	Run-time(s)	Acc.(%)	Run-time(s)	Acc.(%)	Run-time(s)	Acc.(%)	Run-time(s)	Acc.(%)
NN_d	**378**	22.25	1340	30.50	2693	31.50	5796	32.75
NN_c	438	22.50	3602	27.25	6791	30.25	9948	32.50
EmbedNN_d	435	26.75	1373	32.50	2722	35.00	5913	37.50
EmbedNN_c	523	27.50	3702	32.75	6831	36.50	10219	38.25
Semi-DBN_d	769	29.50	1275	33.50	2402	37.25	5945	40.25
Semi-DBN_c	1394	30.50	3467	34.25	7792	37.70	22887	39.50
DBN-rNCA_d	712	29.25	1156	35.25	2209	36.50	5197	41.25
DBN-rNCA_c	1134	30.75	3223	35.25	6565	37.00	18452	42.50
DDBN_d	658	31.25	1051	37.00	2126	41.25	5142	49.20
DDBN_c	1045	32.00	2987	38.25	5292	42.50	16737	51.00
BDBN	392	**35.25**	**963**	**42.50**	**2056**	**50.75**	**5101**	**55.25**

data per subject is equal to 5, 10, 20 and 40. The Gaussian white noise with a mean of 0 and a variance 0.003 is added to the intensity image. According to the average classification results shown in Figure 10 (a), it is obvious that the classification accuracy increases with the number of labeled data. In addition, BDBN exhibits better performance than DCNN under all conditions.

Second, we compare the influence from different extents of noise with same number of labeled data. Here, we fix the number of labeled data per subject to be 10. The variance of Gaussian white noise changes from 0.005 to 0.02. From Figure 10 (b), although the classification accuracy decreases along with the increase of noise in both BDBN and DCNN, our technique performs better. Thus, we are able to conclude that, although DCNN is famously invariant to variations or noises [32][33], our proposed BDBN is more robust than DCNN.



(a) (b)

Figure 10. (a) Classification Accuracy rate (%) with different number of labeled data (b) Classification Accuracy rate (%) with different extents of noise.

Why does BDBN always performs better than DCNN for noisy images? Figure 11 is intended to provide some interpretation from the data reconstruction. The images with Gaussian white noise with a mean of 0 and variance of 0.005 are inputted to BDBN, as shown in the first row. The results of the reconstruction in every layer are shown from the second to the fourth row. It is apparent that, after three layer-wise information reconstructions, the noises have been removed. In addition, the reconstructed images are more similar to the original images shown in the fifth row.

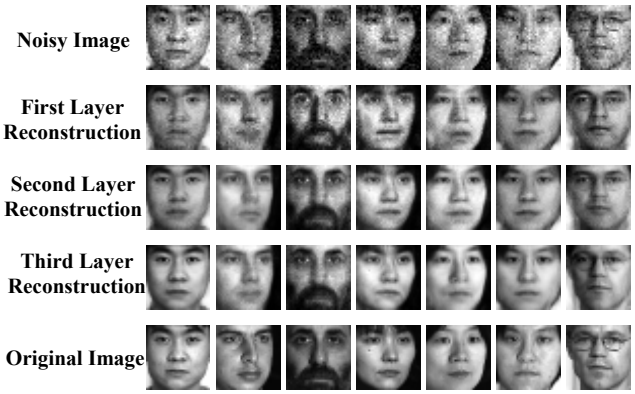


Figure 11. The reconstruction of BDBN in every layer. The first row shows the noisy images. The reconstruction results of every layer are shown from the second to the fourth row. The original images are shown in the fifth row.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel learning model, BDBN for a classical multimedia content analysis task, image classification. BDBN has several attractive characters. First, the novel deep architecture of BDBN simulates the multi-layer physical structure of the visual cortex and enables the preservation of the natural tensor structure of the input image in the information propagation. Second, the three-stage learning of BDBN faithfully realizes the procedure of object recognition by human beings, especially for the "initial guess" part, which has never been modeled in deep learning. Third, the bilinear discriminant initialization of BDBN not only prevents the propagation of information from falling into a bad local optimum but also provides a more meaningful setting for deep architecture. Fourth, the semi-supervised learning ability of BDBN causes the proposed deep techniques to work well with an insufficient number of labeled data. Experiments on three real-world image classification tasks not only show the distinguishing classification ability of BDBN but also clearly demonstrate our intention of providing a human-like image analysis by referencing the human visual system and perception procedure. Future work will be explored from two aspects. The first possible extension is providing more semantic understanding of the images by integrating contextual cueing in deep modeling. The second direction is utilizing deep learning for multimedia content analysis in a large scale dataset with noisy tags.

6. ACKNOWLEDGMENTS

This research was supported by HK PolyU 5245/09E.

7. REFERENCES

[1] F. Moosmann, E. Nowak and F. Jurie, "Randomized Clustering Forests for Image Classification", In *PAMI*, 2008.

[2] A. Kumar , C. Sminchisescu, "Support kernel machines for object recognition", In *ICCV*, 2007.

[3] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "Weak hypotheses and boosting for generic object detection and recognition", In *ECCV*, 2004.

[4] J. Yang, K.Yu, Y. Gong, T. Huang, "Linear spatial pyramid matching using sparse coding for image classification", In *CVPR*, 2009.

[5] A. Bosch, A. Zisserman, X. Munoz, "Image classification using random forests and ferns", In *ICCV*, 2007.

[6] D. Mahajan, and M. Slaney, "Image classification using the web graph", In *ACMMM*, 2010.

[7] M.H. Tsai, S.F. Tsai, T.S. Huang, "Hierarchical image feature extraction and classification", In *ACMMM*, 2010.

[8] O. Boiman, E. Shechtman, M. Irani, "In defense of nearest-neighbor based image classification", In *CVPR*, 2008.

[9] X. Xian, C.S. Xu, J.Q. Wang, "Landmark image classification using 3D point clouds", In *ACMMM*, 2010.

[10] L.F. Li, N Zhang, L.Y. Duan, Q.M. Huang, J. Du, L. Guan, "Automatic sports genre categorization and view-type classification over large-scale dataset", In *ACMMM*, 2009.

[11] W.T. Chu, W.L. Liu, J. Y. Yu, "Age classification for pose variant and occluded faces", In *ACMMM*, 2010.

[12] J. Machajdik and A. Hanbury, "Affective image classification using features inspired by psychology and art theory," In *ACMMM*, 2010.

[13] R. Valenti, A. Jaimes, N. Sebe, "Sonify your face: facial expressions for sound generation", In *ACMMM*, 2010.

[14] Z. Li, H.Z. Luo, J.P. Fan, “Incorporating camera metadata for attended region detection and consumer photo classification”, In *ACMMM*, 2009.

[15] G. Wallis, H. Bülthoff, “Learning to recognize objects”, In *Trends. Cogn. Sci,* 1999.

[16] T. Lee, D. Mumford, “Hierarchical Bayesian inference in the visual cortex”, In *JOSAA*, 2003.

[17] G. Leuba, R. Kraftsik, “Changes in volume, surface estimate, 3-dimensional shape and total number of neurons of the human primary visual-cortex from midgestation until old-age”, In *Inat. Embryol.*, 1994.

[18] R. A. Barton, “Neocortex size and behavioural ecology in primates”, In *Royal Society of London*, 1996.

[19] G. E. Hinton, “Learning Multiple Layers of Representation”, In *Trends. Cogn. Sci*, 2007.

[20] D. J. Felleman, D. C. Van Essen, “Distributed hierarchical processing in the primate cerebral cortex”, In *Cereb. Cortex.*, 1991.

[21] R. VanRullen, S. J. Thorpe, "The time course of visual processing: from early perception to decision-making," In *JOCN,* 2001.

[22] X. Zhu, “Semi-supervised learning literature survey,” Technical report 1530, Univ. of Wisconsin-Madison, 2006.

[23] R. Gross, L. Sweeney, F. D. la Torre, S. Baker, “Semi-supervised learning of multi-factor models for face de-identification,” In *CVPR*, 2008.

[24] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, “An empirical evaluation of deep architectures on problems with many factors of variation”, In *ICML*, 2007.

[25] G. E. Hinton, S. Osindero, Y. Teh, “A fast learning algorithm for deep belief nets”, In *Neural Comput.*, 2006.

[26] P. Smolensky, “Information processing in dynamical systems: foundations of harmony theory”, In *Parallel Distributed Processing: Explorations in The Microstructure of Cognition*, vol. 1: Foundations, MIT Press, pp. 194-281, 1986.

[27] R.R. Salakhutdinov, G.E. Hinton, “Learning a nonlinear embedding by preserving class neighbourhood structure”, In *AISTATS*, 2007.

[28] J. Weston, F. Ratle, R. Collobert, “Deep learning via semi-supervised embedding”, In *ICML*, 2008.

[29] S.S. Zhou, Q.C. Chen, and X.L. Wang. "Discriminate Deep Belief Networks for Image Classification", In *ICIP*, 2010.

[30] Z. Wang, D. Xia, E.Y. Chang, “A deep-learning model-based and data-driven hybrid architecture for image annotation”, In *VLS-MCMR, ACM*, 2010.

[31] E. Hörster, and R. Lienhart, “Deep networks for image retrieval on large-scale databases”, In *ACMMM*, 2008.

[32] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”, In *ICML*, 2009.

[33] G. Taylor, R. Fergus, Y.L. Cun and C. Bregler, “Convolutional learning of spatio-temporal features,” In *ECCV*, 2010.

[34] Y.L. Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, “Backpropagation applied to handwritten zip code recognition,” In *Neural Comput.*, 1989.

[35] Y. Bengio, and Y.L. Cun, “Scaling Learning Algorithms towards AI,” In *Large-Scale Kernel Machines*, 2007.

[36] R. Memisevic, G.E. Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines,” In *Neural Comput.*, 2010.

[37] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y.L. Cun, “What is the best multi-stage architecture for object recognition?”, In *ICCV*, 2009.

[38] S. Ji, W. Xu, M. Yang, K. Yu, “3D convolutional neural networks for human action recognition,” In *ICML*, 2010.

[39] S. Yan, D. Xu, B. Zhang, H.J. Zhang, Q. Yang and S. Lin, “Graph embedding and extension: a general framework for dimensionality reduction”, In *PAMI*, 2007.

[40] M. Sugiyama, “Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis”, In *JMLR*, 2007.

[41] G.E. Hinton, “Training products of experts by minimizing contrastive divergence”, In *Neural Comput.*, 2002.

[42] F.F. Li, R. Fergus, P. Pernoa, “Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories”, In *CVPR* , 2004.

[43] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope, ” In *IJCV*, 2001.

[44] T. Sim, S. Baker, “The CMU pose, illumination, and expression database”, In *PAMI*, 2003.

[45] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, “Greedy layer-wise training of deep networks”, In *NIPS*, 2006.

[46] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers”, In *COLT*, 1992.

[47] R. Collobert, F. Sinz, J. Weston, L. Bottou, “Large scale transductive SVMs, In *JMLR*”, 2006.

[48] T.M. Mitchell, “Machine Learning”, 1997.

[49] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner. “Gradient-based learning applied to document recognition,” In *Proceedings of the IEEE,* pp. 2278-2324, 1998.

[50] X.F. He, D. Cai, and P. Niyogi, “Tensor subspace analysis”, In *NIPS*, 2005.