Pattern Recognition 81 (2018) 294–308

Contents lists available at ScienceDirect

Pattern Recognition

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

Deep variance network: An iterative, improved CNN framework for unbalanced training datasets

Shuai Li[a], Wenfeng Song[a], Hong Qin[b], Aimin Hao[a,*]

[a] State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science, Beihang University, China
[b] Stony Brook University (SUNY Stony Brook), USA

ARTICLE INFO

Article history:
Received 15 August 2017
Revised 18 March 2018
Accepted 27 March 2018
Available online 4 April 2018

MSC:
00-01
99-00

Keywords:
Deep variance network
Unbalanced training datasets
Convolutional neural network
Homogeneity
Heterogeneity

ABSTRACT

Convolutional neural network (CNN) has demonstrated its superior ability to achieve amazing accuracy in computer vision field. Nevertheless, for practical domain-specific image recognition tasks, it still remains difficult to obtain massive high-quality labeled datasets due to the strong requirements for extensive, tedious manual processing. Inspired by the well-known observation that human brain can accurately recognize objects without relying on massive congeneric examples, we propose a novel deep variance network (DVN) to further enhance the generalization ability of CNN in this paper, which could still produce higher recognition accuracy even with unbalanced training datasets than original CNN. The key idea of our DVN is built upon the intrinsic exploitation of inter-class homogeneity and intra-class heterogeneity. Towards such goal, we make the first attempt to incorporate a hierarchical Bayesian model into the powerful CNN framework, which can transfer the joint feature distribution from certain object's complete training dataset to other object's incomplete training dataset in an iterative way. In each training cycle, the CNN-resulted features are clustered into discrimination-related subspaces to guide the learning and adaptive adjustment of homogeneity and heterogeneity over unbalanced training datasets. In practice, we furnish several state-of-the-art deep networks with our proposed DVN, and conduct extensive experiments and comprehensive evaluations over CIFAR-10, MNIST, and SVHN benchmarks. The experiments have shown that, most of the furnished deep networks can benefit from our DVN, wherein they gain at most 6.9% accuracy improvement over CIFAR-10 benchmark, 52.83% error reduction over MNIST benchmark, and an improvement of 6.2% over SVHN datasets.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The accurate and rapid classification of wild images plays a critical role in computer vision related applications. In practice, deep learning methods typically require thousands of well-labeled training samples to learn new concepts. Specifically, different categories often involve unbalanced samples, which means the sample number of each class is not equal. The instinct of unbalance problem is the unequal difficulties to obtain the labeled images of each class [4,5,30,56,60]. For example, the samples of automobiles and planes can be more easily obtained from the internet than those of ships and rockets, and the samples of cats and dogs can be more easily obtained than those of any other rare animals. Furthermore, even if the numbers of the samples are equal, the class-specific accuracy is affected by the diversity of the class samples. Thus, the balance of training dataset is vital for classification tasks.

Deep learning methods can achieve satisfactory performance with sufficient well labeled data sets [50,54], but still face the unbalanced challenges. It should be noted that the performance gains achieved from existing deep networks can be attributed to the increase of network depth and large-scale well-labeled training dataset. For example, some works [16,50,54,55] employ the blocks mode to increase the depth of deep convolutional architecture, and rank tops in recognition and detection tasks for Imagenet [46] challenges. However, deep models increase the dependence of the well-labeled training dataset at the same time [38,39]. Unfortunately, the annotation of the dataset is expensive and time-consuming. Imagenet dataset needs tens of thousands of people to manually annotate 1500-class objects over fifteen million images in a crowdsourcing way, which costs about five years. It is hard to guarantee the balanced number of each class. However, the balance of sub-datasets is not essential for human's complex neurons system, while only a few examples are sufficient for human to understand a new category and further make meaningful reasoning about other similar instances. Therefore, the learning method is to

[*] Corresponding author.
E-mail address: ham@buaa.edu.cn (A. Hao).

https://doi.org/10.1016/j.patcog.2018.03.035
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

be improved in the sense of extracting more information on the unbalanced dataset. Thus, we aim to improve the learning ability of CNNs over the unbalanced dataset.

To solve the unbalance problem in deep learning field, previous works mainly focus on the re-sampling over numbers and distributions in-between categories [4,5,8,30,31,51,56] and the cost-sensitive loss [18]. The traditional methods can benefit to the "shallow" learning methods [13], however, it is not the most effective way to deal with unbalanced data in the context of deep learning. Moreover, such works commonly have inherent limitations [18]. For instance, over-sampling [36] can easily introduce undesirable noise, which gives rise to overfitting risks. Such limitations are also negative to deep learning methods [18]. These methods do not consider the essential problems during the unbalanced learning process, which can be summarized as three main challenges below. The first one is how to properly leverage practical unbalanced dataset with complex variances in scales, qualities, and acquisition difficulties. With this method, overfitting problem can be avoided as much as possible. The second one is how to adaptively enrich the information of scale-limited training dataset. With this method, unbalanced dataset is trained sufficiently. The third challenge is how to effectively accommodate the inexhaustible feature changes involved in the same-class instances. With this method, the generalization ability can be guaranteed.

Furthermore, the unbalance problem defined by traditional classifiers is limited to the number of each class. According to the previous works [2], the balanced dataset should satisfy two principles: (1) each of the involved sub-dataset has equal number of training samples; (2) the sub-datasets corresponding to each class should play an equal role in the training phase. This number-based definition makes the previous works focus on expanding number of samples, which can not expand the information for minority classes. When trained in an information-insufficient way, the CNNs easily produce an overfitting problem. Besides, the two principles are more or less ignored in the training process, which depends on the difficulty degrees of each class, and it is the instinct factor for training datasets of CNNs. Therefore, we extend the number-based unbalanced dataset to the accuracy-based unbalanced dataset for CNNs. On that basis, current CNNs-related methods typically suffer from unbalanced training datasets (such as MNIST, CIFAR-10, CIFAR-100, and SVHN datasets). In addition, the classification accuracy on the number-based unbalanced datasets, such as, CIFAR-10, CIFAR-100 (we re-organize them as number-based unbalanced dataset on purpose) can also be improved when adopting proper unbalanced learning schemes in the training phase. Therefore, the accuracy-based unbalance principles are more general than number-based ones (unbalance refers to accuracy-based unbalance in this paper unless otherwise specified). However, unbalanced learning will inevitably give rise to great challenges in CNNs, as it becomes very hard to learn from the samples of the minority classes [4,5,8,30,31,51].

Bayesian based techniques, which are able to learn the distribution based on the training datasets, show potential over the small-scale dataset [49]. And the techniques can avoid the overfitting in large-scale datasets. For instance, Bayesian Program Learning (BPL) [26] is capable of learning a large class of visual concepts from a single example and generalizes in ways that are mostly indistinguishable from people. One-shot learning [48] developed a deep generative model to combine the representational power of deep learning with the inferential power of Bayesian reasoning. In fact, such methods are hard to represent high diversity in visual features. Motivated by both of the learning methods, we propose to learn the features via deep learning and further resort to the Bayesian network to augment the distribution, so as to solve the unbalance problem during CNNs training. Our method is based on the observation that the inter-class objects may have similar sample distribution, while the intra-class objects may have variations.

To tackle the aforementioned challenges, we propose to integrate the advantages of probability graph method built on the inter-class and intra-class distributions based on Bayesian hierarchical model to embed into CNNs. It is firstly trained on the CNNs to extract features, the features are clustered into groups via hypergraph. The groups are further clustered into three subspaces based on the degree of data overlapping among different classes. Then based on the inter-class and intra-class distributions, new virtual samples are generated to augment the training datasets to solve the unbalance problem. Specifically, the salient contributions of this paper can be summarized as follows:

(1) We pioneer a generic *deep variance network* (DVN) by integrating three subspaces as the prior in the Bayesian network to iteratively back propagate into the powerful CNN framework, which can greatly improve the CNN performance for unbalanced training datasets.
(2) We propose a *hierarchical Bayesian model* for the transfer learning of intra-class heterogeneity and inter-class homogeneity over CNN-produced feature space, which can intrinsically transfer the joint feature distribution from certain complete training dataset to other incomplete datasets, and expedite the training convergence of CNN.
(3) We propose a *virtual example generation* method based on Gaussian kernel density estimation, which conduces to complete the unbalanced training dataset via passing the information from feature level to image level in a top-down way.
(4) We verify our DVN by using it to *furnish several state-of-the-art CNN networks*, and conduct extensive experiments and comprehensive evaluations over CIFAR-10, CIFAR-100, MNIST and SVHN benchmarks, which demonstrate its superiorities in effectiveness and universality.

The rest of the paper is organized as follows. Section 2 firstly analyzes the unbalance problem in detail, and then briefly reviews the deep neural networks from three different perspectives. Sections 3, 4 and 5 detail our DVN framework. Section 6 evaluates our DVN over four datasets under accuracy-based and number-based unbalanced settings. Section 7 discusses the theoretic boundary of the proposed DVN. Finally, Section 8 makes conclusions and conducts discussions for future work.

2. Related work

This section discusses problems caused by the utility of unbalanced training datasets first and then briefly reviews deep neural networks from three different perspectives.

2.1. Unbalanced datasets

To solve the unbalanced(number-based) learning problem, there are mainly two types of methods. The first one is to balance the sample number of different-class training images according to the covering the ratios of each class' samples (resampling) [32,64]. Among the resampling methods, the most famous one is the synthetic minority over-sampling technique (SMOTE) [7]. It generates synthetic samples, which are interpolations of intra-class neighboring samples, to augment the minority class. And its underlying assumption is that, the augmented interpolations deviate from a locally linear feature space instead of the original class distribution. Recently, Tzelepis et al. [56] proposed re-sampling via a multi-dimensional Gaussian distribution and reformulated a maximum margin classifier. Such methods are efficient for isotropic classes. Most works in this field commonly use classifiers to adjust the hyperplane to discriminate the classes. For example, Li et al.

[30] proposed a deep generative model, and Brun et al. [5] used dynamic classifier selection to improve the unbalanced datasets. Some other works focused on augmenting the dataset. For example, Son et al. [51] proposed to cluster the dataset via exemplars from heterogeneous data, which further showed the analysis of data distribution could improve the unbalance dataset learning. Li et al. [32] proposed a class-specific feature group method. And Zhu et al. [64] studied the synthetic minority over-sampling technique for multi-class imbalance problems. However, most of the existing over-sampling methods are demonstrated to generate wrong synthetic samples for minority incomplete classes in some cases, and thus, making the unbalanced learning tasks much harder.

The second one focuses on the class-wise weights' adjustment in training phase according to the accuracy of each class' learning results (cost-sensitive) [15]. Wang et al. [57] proposed class-weighted loss to enhance the minority class accuracy. The diversity of each class' samples could also affect the performance of the cost-sensitive based methods. Considering F-measure is a more reasonable performance metric compared to accuracy, some recent works proposed to systematically optimize F-measures for the classification and feature selection in unbalance datasets. For example, Dembczynski et al. [11] optimized the F-measure in multi-label classification tasks, Liu et al. [34] employed F-measures for cost-sensitive feature selection, and Puthiya Parambath et al. [42] optimized F-measures via cost-sensitive classification. Furthermore, Brun et al. [6] suggested that the problem was not solely caused by class imbalances, but was also related to the difficulty of data overlapping among the classes.

In deep learning techniques, some works proposed to extend the previous two types of works to facilitate CNNs. Peng et al. [40] proposed feature fusion from several unbalanced datasets. Ando and Hensman [3] extended the synthetic samples, but was limited to the number-based unbalanced dataset. Masko and Hensman [36] demonstrated the over-sampling method could improve the accuracy of the unbalanced datasets, but it was limited to simply replicating images of the minority class.

Recently, to make CNN available on general devices, many works try to compress the CNN models. For example, Wang et al. [58] leveraged K-means clustering to compress the convolutional layers of CNNs in discrete cosine transform space. Rastegari et al. [45] proposed Xnor-net by approximating the convolutions with primarily binary operations, and it could save 32 times memory. Luo et al. [35] focused on the filter level pruning, which did not change the original network structure but could be perfectly supported by any off-the-shelf deep learning frameworks.

In summary, the two methods are initially designed for the unbalance problem in traditional classifiers without CNNs and some limited works try to extend the methods to improve CNNs'. However, the works fail to balance the CNN classes, and they are not efficient for the large-scale dataset and can not totally learn the semantic features from CNNs. To the best of our knowledge, we are the first to embed the bayesian network into CNNs (DVN) to learn the semantic features over unbalanced datasets.

2.2. Improved convolution neural networks via depth increasing

The CNNs mostly focus on improving the accuracy of the unbalance classes, but choose to ignore the unbalanced distribution and weights. Most of the existing deep neural network methods improve the feature capture capability for many visual recognition tasks by increasing the depth of CNNs. Krizhevsky et al. [24] proposed a network with 60 million parameters, which significantly improved the accuracy with respect to the original neural network. Since then, more complex and deeper networks have been proposed to further increase the performance. For example, Szegedy et al. [54] introduced the inception deep convolutional architecture in GoogLeNet/Inception. Besides, Simonyan Zisserman [50] investigated a very deep network (VGG) to improve the performance on large scale datasets. Lin et al. [33] proposed a "Network In Network" (NIN) structure to enhance model discriminability within the receptive field. And Srivastava et al. [52] introduced a highway network architecture to overcome the gradient-vanishing problem, which could be trained with hundreds of layers. In practice, this network is hard to guarantee the convergence for different datasets. Meanwhile, He et al. [16] presented a residual learning framework to ease the training of deeper networks, which achieved higher accuracy in object recognition, detection, and segmentation. Recently, Yu et al. [61,62] and Hong et al. [17], respectively, proposed multi-layer deep neural network and multi-modal features to add the data diversity via multiple tasks combinations. However, most of the proposed depth-increasing CNNs are limited to the well-labeled and balanced dataset. For the ideal dataset, the deep models perform better than the "shallow" models. Thus, it is significant to propose a framework to learn sufficient information from the unbalanced datasets.

2.3. Improved convolution neural networks via information expanding

Recently, some works also focus on the information expanding of scale-limited training datasets. For example, Jaderberg et al. [20] and Rasmus et al. [44] augment the information learned by CNNs on the fixed dataset to capture more features. At the same time, some works propose to improve feature information extraction. For example, von Noord and Postma [39], Nogueira et al. [38] and Li et al. [31], respectively, improved the multi-class classification performance via improving the feature information of the same classes. Jiao et al. [22] proposed a lightweight nonlinear CNN. Chen et al. [8] proposed to encode localized and second-order feature into the network. Bohne et al. [4] learned local metrics from pairwise similarity data. Afridi et al. [1] demonstrated that, it is possible to automatically rank pre-trained CNNs for a given task. In addition, some works [9,10] focus on focus on the leverage of image structure features. Furthermore, the GANs are proposed to produce interpretable and diverse candidates. The GANs and their variants cover most of the data generation tasks, including semi-supervised learning, domain adaption, image inverting, etc. [12,63]. The ability of GAN is strong under the condition of sufficient datasets (labeled or unlabeled). However, in the unbalance learning field, the GANs face some of the main challenges: The require of large scale of training dataset and the limited quality of images. Meanwhile, one-shot learning [48] is proposed to learn the human ability to recognize a new concept and then generate compelling alternative variations of the concept, which is limited for representing the elementary simple features.

2.4. Bayesian networks augmented convolution neural networks

In contrast to the methods to improve the CNN performance, Bayesian network considers the distribution of the classes instead of enlarging the model and the datasets. Bayesian network, deep Boltzman machine [47], etc. show strong ability in automatically learning discriminative features. And the ability in multiple-layer learning allows capturing sophisticated domain-specific features. For example, Salakhutdinov and Hinton [47] proposed Boltzmann machines, of which, data-dependent expectations were estimated using a variational approximation, and data independent expectations were approximated using persistent Markov chains. Li and Perona [14] proposed a hierarchical Bayesian model to represent codewords in an unsupervised way. Brenden et al. [26] proposed a computational model to explain the observed examples under a

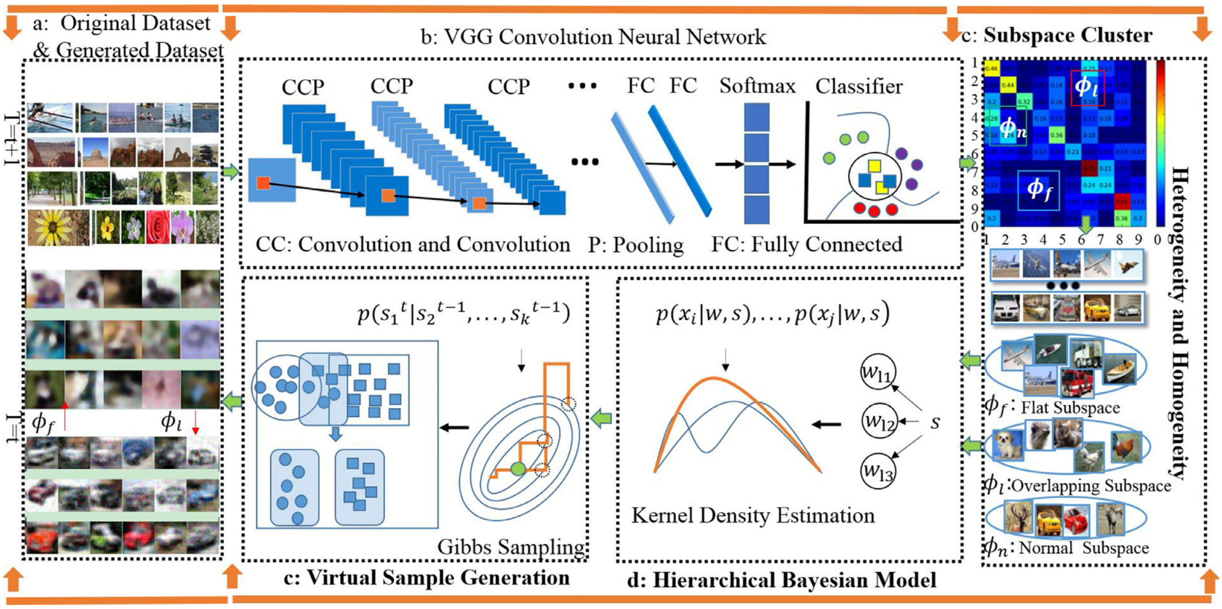

a: Original Dataset & Generated Dataset
I=I+1
I=1
ϕ_f
ϕ_l
b: VGG Convolution Neural Network
CCP
CCP
CCP
FC FC
Softmax
Classifier
CC: Convolution and Convolution
P: Pooling
FC: Fully Connected
c: Subspace Cluster
ϕ_l
ϕ_n
ϕ_f
Heterogeneity and Homogeneity
ϕ_f: Flat Subspace
ϕ_l: Overlapping Subspace
ϕ_n: Normal Subspace
$p(s_1{}^t|s_2{}^{t-1},...,s_k{}^{t-1})$
Gibbs Sampling
c: Virtual Sample Generation
$p(x_i|w,s),...,p(x_j|w,s)$
w_{11}
w_{12}
w_{13}
s
Kernel Density Estimation
d: Hierarchical Bayesian Model

Fig. 1. The pipeline of our DVN framework.

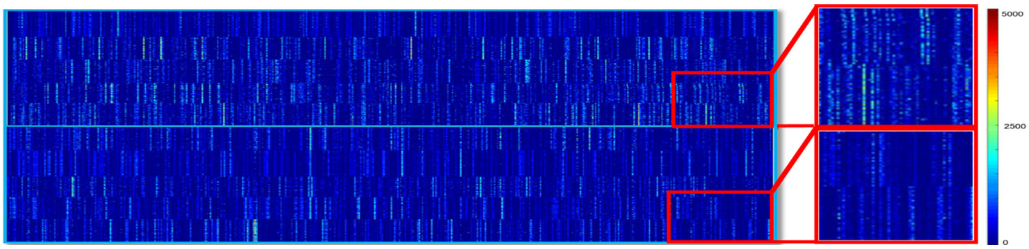


Fig. 2. Feature visualization of original Lenet network (top row) and DVN (bottom row). Each row of the feature matrix is a 500-dimensional feature vector. Each column involves 10-classes image set, and each image set has 1000 image samples, which are used to show their homogeneous and heterogeneous properties.

Bayesian criterion, which could mimic the human's learning abilities in recognizing handwritten characters. However, this method only has limited generalization ability, which is hard to be employed to improve other existing deep networks.

To enhance advantages and avoid disadvantages of the aforementioned works, this paper will design a deep variance network framework by integrating the deep neural network architecture and the Bayesian network, which is expected to further improve the performance of most state-of-the-art networks.

3. Method overview

As shown in Fig. 1, an iterative hierarchical Bayesian model is embedded in a CNN framework to transfer the feature distribution among quality-varying training sub-datasets, which respectively corresponds to different object classes. We briefly overview the key technical elements of our framework as follows.

Heterogeneity and homogeneity analysis. We define the heterogeneity and homogeneity by extracting the intra-class diversity and inter-class similarity. We first extract features based on traditional CNN. And then, we construct a hyper-graph to automatically cluster the extracted CNN features into three subspaces according to their discriminations. Upon that, we further calculate and compare the feature distributions in/across the classes grouped in the certain subspaces to obtain their intra-class heterogeneity and inter-class homogeneity (Please refer to Section 4 for details).

Hierarchical Bayesian model. Based on our proposed heterogeneity and homogeneity, we further propose a hierarchical Bayesian model, wherein the feature distributions over well-labeled complete (balance) training dataset can serve as conditions of incomplete training dataset (unbalance). Therefore, the hierarchical Bayesian model incrementally generate and refine multi-level features top down for the incomplete training dataset in an iterative way, so that it can expedite the performance improvement of CNN(Please refer to Section 5.1 for details).

Virtual sample generation. Throughout our framework, we take into account the hierarchical Bayesian priors, which span from pixel to class, image, subspaces gradually. And then, based on the obtained three subspaces and the priors of class distributions, we generate virtual images via kernel density estimation over the iteratively updated class distributions, so that we can obtain an relatively-balanced accuracy for incomplete classes (please refer to Sections 5.2 and 5.3 for details).

4. Heterogeneity and homogeneity analysis on clustered CNN features

4.1. Hyper-graph based CNN feature subspace construction

Given i training images belonging to the class $\mathbf{C}$, we denote them as $\mathbf{X_C} = (x_1, \ldots, x_i)$. Let R denote the relationship between different classes. In essence, the challenge for recognition is the fuzziness (in some sense it reflects that certain classes have close relations) among different classes. Therefore, we model R with a hyper-graph to amplify the subtle differences among different classes. The hyper-graph enables the representation of non-linear relationships among the whole classes rather than the pairwise similarities, wherein hierarchical priors and clustered subspace properties can be encoded in the latent variables $\mathbf{L}$ [41]. Given the hyper-graph $H = (V, E)$, a set of vertices V represent the object classes, edge $e \in E$ represents the pairwise relationship, and the degree $d(v)$ of the vertex v is computed by summing all the weights of the adjacent edges.

The hyper-graph could then be employed to gather some classes into a group according to their feature distances. Let $(S; S_C)$ be a partition of the vertices V in H, with $S \cup S_C = V$. We gather V into groups based on the graphcut method, and the discriminative ability of each group is encoded in $d(v)$. The higher the value is, the lower discriminative ability v has. Our goal is to adaptively obtain three subspaces. The "flat" subspace, denoted by ϕ_f, consists of groups that the involved classes have long distance from one another. In other words, the involved features have the best distinguishing capability. The "overlapping" subspace, denoted by ϕ_l, consists of the groups that the involved classes are most indistinguishable. The "normal" subspace ϕ_n, consists of the groups that the involved classes are moderately discriminative. In our recurrent framework, we would add a regularizer to the back propagation for ϕ_f to slow down its gradient decrease, and respectively model the features of ϕ_l and ϕ_n with Bayesian networks in their corresponding back propagation procedures. Benefiting from such schemes, in the training phase, the loss of ϕ_l is expected to be larger than that without DVN, while the accuracy is higher. When we conduct back propagation for ϕ_f, ϕ_l and ϕ_n, relatively stable and balanced gradients will be adaptively adopted to decrease their losses. The hyper-graph based clustering involves two steps. The first is to cluster all the classes into K groups (K is computed based on the distances between the classes, and we empirically set $K = 70\%$ of the total class number). The second is to cluster the K groups into three subspaces according to each group purity (the number of categories in this group). In our implementation, ϕ_f consists of the groups with 10% object classes; ϕ_l consists of the groups with 20%; and the others belong to ϕ_n.

4.2. Definition of inner-class heterogeneity and inter-class homogeneity

Based on the hyper-graph relation R, we can obtain three subspaces. In order to generate virtual samples to enhance the differences of the classes in each subspace, we first define intra-class heterogeneity H_e and inter-class homogeneity H_o. Inter-class homogeneity describes the general feature distribution involved in two or more object classes, which will be employed to balance the incomplete labeled dataset. Inner-class heterogeneity describes the feature distribution variances within the same object class. Most previous works focus more on the similar properties of the same class, however, we find that the heterogeneity in the same class is more valuable for the balance of training data in CNN-based object recognition. Let $\mathbf{\Sigma}_\xi$ denote the variances of the same-class feature distribution, in previous works [53], it is linearly formulated as $x = \tilde{x} + \xi,\ \xi \sim N(0, \mathbf{\Sigma}_\xi)$, which is the most basic assumption to represent the distribution of the low-level visual properties, such as color and noise. However, it is hard to represent the object's variances in scale, rotation, position, and deformation. We resort to modeling the variances by transforming the homogeneity and heterogeneity from feature level to image level. We employ the latent parameter $\mathbf{s} = (s_1, \ldots, s_k)$ to describe such condition distributions, wherein s_i represents the variable related to feature distribution of the i_{th} group. The unbalanced dataset ϕ_l is adaptively refined on the condition of the well-labeled dataset ϕ_n and ϕ_f. To embed the constraint as a new layer in CNN, it can be defined as

$$\min \lambda \mathrm{KL}(p(x|\mathbf{s}), p(\tilde{x}|\mathbf{s})) + \log \sum_j exp(z_j) - z_y. \tag{1}$$

Here, λ is a regularization parameter, which balances the weight of the term $\mathrm{KL}(p(x|\mathbf{s}), p(\tilde{x}|\mathbf{s}))$ in the loss function. z_j represents the j_{th} element of softmax layer in CNN. $\mathrm{KL}(\cdot)$ is the Kullback–Leibler divergence, and $\mathbf{s}$ is the aforementioned latent parameter depicting distribution. The inner-class heterogeneity and inter-class homogeneity can be formulated via nonlinear feature mapping as

$$H_e(x) = cov(x - \tilde{x}, x - \tilde{x}), H_o(x) = \frac{1}{n}\sum_i (f(x_i) - f(\tilde{x_j})). \tag{2}$$

Here, H_o is a k-dimension vector, which describes the intrinsic change rule in different classes. In real world, an object's appearance may be influenced by several factors, such as color, shape, scale, etc. And the objects belonging to different classes may be subject to similar change principles. H_e is a matrix, which describes the variance of the CNN features resulted from certain-class instances.

$$p(H_e) = \sum_k p(H_e|s_k),\ \ p(H_o) = \sum_k p(H_o|s_k). \tag{3}$$

5. Hierarchical Bayesian model for training data balance

5.1. Hierarchical Bayesian model construction

Most previous methods for unbalanced training data processing tend to transfer information among the same-level features [20], however, at image level it is hard to eliminate the inner-class heterogeneity only with the inner-class homogeneity information. Thus, we design a hierarchical Bayesian model to learn different-class homogeneity and same-class heterogeneity. Our model supports cross-level information transfer, spanning from pixel, low-level feature, image, object class, to subspace in a top-down way.

Our Bayesian model is a dynamic hybrid model with hierarchical-structure feature learning ability, which can leverage cross-level feature interaction as prior relationships to compute the conditional probability distribution of level-wise features. Thus, given the known feature distribution of certain layers in our hierarchical model, we can obtain other layer's conditional probability distribution function. Under the mean field assumption, by factorizing the same-level posterior knowledge into independent partitions, the model can be defined as

$$\begin{gathered} p(x_i) \propto p(x_i|\mathbf{w}),\ \ p(x|\mathbf{w}) = \prod_i p(x_i|w_i), \\ p(w_l) \propto p(w_l|f(x)),\ \ p(\mathbf{w}|f(x)) = \prod_l p(w_l|f(x_l)), \\ p(f(x_i)) \propto p(f(x_i)|\mathbf{s}),\ \ p(f(x)|\mathbf{s}) = \prod_i p(f(x_i)|s_i). \end{gathered} \tag{4}$$

Here, $\mathbf{w}$ is the neuron parameter of CNN, the subscript l denotes layer index of the Bayesian network. Based on Eq. (4), the variance regularity can be transferred from the feature level to image level. Given unbalanced training datasets, based on the CNN features and the cluster subspaces via hyper-graph (obtained in Section 3), we transfer H_e from the well-labeled complete training dataset to the incomplete one. Given $x_l \in \phi_l$ and $y_f \in \phi_f$, the distribution can be computed via

$$p(x, y) = \prod_{l,f} p(x_l|H_e, H_o)p(H_e, H_o|y_f)p(y_f). \tag{5}$$

The challenge to solve Eq. (5) is that, the distribution of H_e can not be obtained directly. Notice the hierarchical structure in Eq. (4), we address this problem by introducing latent variable $\mathbf{s}$ as

$$p(x) = \sum_k p(s_k)p(x|s_k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_k|\mu_k, \mathbf{\Sigma_k}). \tag{6}$$

Here ,π_k represents the distribution of the latent variable **s**, which is related to H_e, H_o. With the relation between two images x, y, Eq. (5) can be rewritten as

$$p(x,y) = \sum_k p(x|s_k)p(s_k|H_e,H_o)p(H_e,H_o|y)p(y). \tag{7}$$

Besides, given x, we can get the distribution of the latent variable **s** according to its distribution via

$$\begin{aligned} p(s_k|x) &= \frac{p(s_k)p(x|s_k)}{\sum_k p(s_k)p(x|s_k)} \\ &= \frac{\pi_k \mathcal{N}(x|\mu_{\mathbf{k}}, \mathbf{\Sigma_k})}{\sum_k \pi_k \mathcal{N}(x|\mu_{\mathbf{k}}, \mathbf{\Sigma_k})}. \end{aligned} \tag{8}$$

In fact, π_k serves as the weight coefficient, which can significantly affect the changing strength when generating virtual samples.

5.2. Bayesian prior guided virtual sample generation

Based on the constructed hierarchical Bayesian model, according to Eq. (6), we can generate more virtual image samples for the unbalanced training dataset. Here, the key challenge is how to compute the distribution of **s**. In practice, we can analyze and extract the correlations between **s** and the features from the convolution and pooling layers of CNN. The relations between x and **s** can be formulated as

$$p(\mathbf{s}|x,\mu,\mathbf{\Sigma}) \propto p(x|\mathbf{s},\mu)p(\mathbf{s}|\mathbf{\Sigma}) \propto p(x|\mathbf{s},\mu,\mathbf{\Sigma}), \tag{9}$$

where μ, $\mathbf{\Sigma}$ are the parameters learnt from the ground truth. The probability density function (PDF) of **s** can be derived from Eq. (9). If certain elements of **s** have been learnt, the unbalanced dataset can be augmented by Eq. (5), which biases to complement the features missed in the original incomplete training dataset. Thus, for original incomplete training dataset, we can remedy the limitations due to the lack of certain-class variations or changes. Therefore, to compute the distribution of **s**, we should first determine $p(x|\mathbf{s})$ by integrating over variables **w** and $\boldsymbol{\theta}$ (see Eq. (10)), and then estimate **s** via Gibbs sampling in EM iterations.

$$p(x|\mathbf{s}) = \int p(\mathbf{w}|s)(\prod_{i=1}^{n}\sum_{\theta_i} p(\theta_i|\mathbf{w})p(x_i|\theta_i))d\mathbf{w}. \tag{10}$$

In Eq. (10), since the parameters **w** and θ are dependent on each other, $p(x|\mathbf{s})$ can not be analytically computed. To this end, we resort to variational approximation, which can maximize the log-likelihood of the data and minimize the Kullback–Leibler divergence (between the approximation and the posterior truth). And we use the distribution of $q(\mathbf{s})$ to approximate the true distribution of $p(x|\mathbf{s})$. We can optimize Eq. (10) by maximizing the lower bound of the likelihood function. For a labeled image, its variational lower bound of the marginal likelihood is bound via Jensen's inequality in the following way:

$$\log p(x|\mathbf{s}) \geq J(x), \tag{11}$$

$$\begin{aligned} J(x) = &\int \sum_w q(\mathbf{w},\theta)\log p(\mathbf{w},\theta,x|\mathbf{s})d\mathbf{s} \\ &- \int \sum_s q(\mathbf{w},\theta)\ln q(\mathbf{w},\theta)d\mathbf{s}, \end{aligned} \tag{12}$$

$$J(x) = \mathrm{L}(\mathbf{w},x,\mathbf{s}) = E_q[\log p(\mathbf{w},\theta,x|\mathbf{s})] - E_q[\log q(\mathbf{w},\theta)], \tag{13}$$

where $q(\mathbf{w}, \mathbf{s}|\theta)$ is an arbitrary variational distribution. Considering Eqs. (11)–(13) we have

$$\log p(x|\mathbf{s}) = \mathrm{L}(\mathbf{w},x,\theta) + \mathrm{KL}(q(\mathbf{w},\theta|\mathbf{s})||p(\mathbf{w},\theta|x,\mathbf{s})). \tag{14}$$

Our goal is to maximize $\log p(x|\mathbf{s})$. By maximizing the lower bound $\mathrm{L}(\mathbf{w}, x, \theta)$ with respect to s it is the same as minimizing the KL distance between the estimated posterior probability and the true probability.

5.3. Back-propogation of DVN

In fact, based on the examine the loss function in Eq. (1), we have two basic constraints. One is the CNN loss (softmax loss), which enforces constraints on the differences between the predicted label and ground truth. The other is about the distribution estimation of x. By minimizing the loss function, we can estimate the distribution of x, and further get the enhanced CNN model from the complete training dataset. In practice, we separately optimize the two components of the loss function. That is, when decreasing the KL divergence, the CNN related function is fixed, and vice versa. The iterative algorithm alternates between the following two steps until convergence. The standard gradient decent flow is represented by following Eqs. (15) and (16).

$$\frac{\partial L}{\partial \mathbf{w}^l} = x^{l-1}(\delta^l)^T, \tag{15}$$

where δ is the loss from the higher layers l. **w** encodes the parameters of CNN nodes. x^l is the feature map of layer l. And the gradient decent flow is computed as follows:

$$\Delta W^l = -\eta \frac{\partial L}{\partial \mathbf{w}^l}. \tag{16}$$

Here η denotes the learning rate. In E-step, for each class of images, the variational parameters **s**, θ are initialized or updated according to Eq. (10). The CNN weight **w** is updated via back propagation, and **s** is updated via Gibbs sampling. The virtual image is generated by Gaussian mixture model based kernel density estimation. The detailed algorithm is showed in Algorithm 1. The backward propagation process of the KL loss is computed by following Eqs. (17) and (18):

$$\mu = \mu^{t-1} + \rho\left(H_o + \frac{\partial L}{\partial p(x)}\frac{\partial p(x)}{\partial \mu}\right), \tag{17}$$

where ρ is the learning rate for μ, which is set to 1.0 in our experiment.

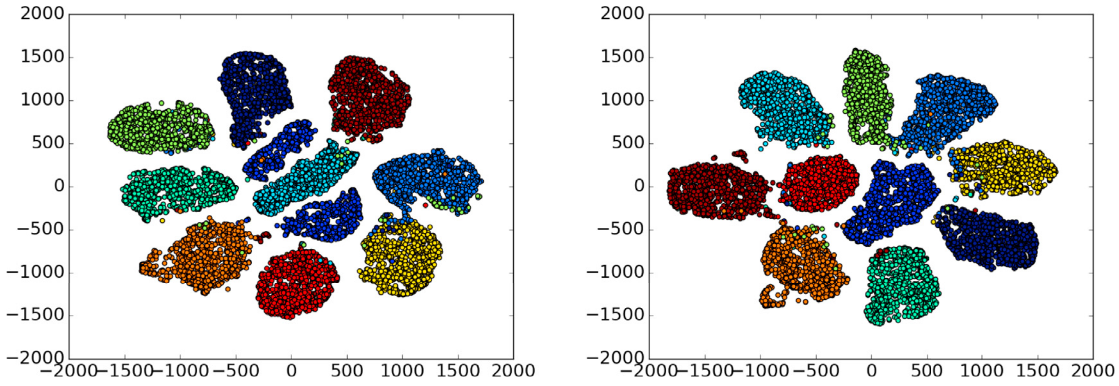
$$\sigma = \sigma^{t-1} + \beta\left(H_e + \frac{\partial L}{\partial p(x)}\frac{\partial p(x)}{\partial \sigma}\right). \tag{18}$$

Here, β is set to 1.0 as the learning rate of σ. In M-step, we update the subspaces, maximize the resulted lower bound of the log-likelihood function with respect to model parameters θ and **w**. We conduct this task by finding the maximum likelihood estimation with expected statistics computed in E-step.

6. Experimental results and evaluations

6.1. Experimental settings

We have implemented our DVN framework using Caffe [21] on 2 GPU clusters with 32 NVIDIA Tesla k20ms and 8 K80s. Given the features extracted from original CNN, we cluster them into three subspaces via hyper-graph cut. A hierarchical Bayesian model is trained to transfer the latent variables H_e and H_o. Here H_o and H_e serve as prior knowledge of ϕ_l. On that basis, virtual images are generated for the "flat" subspace and "overlapping" subspace ϕ_l. After that, we feed the virtual images generated by current iteration to CNN, and the training process will be stopped when the loss function keeps stable. In our experiments, the epoch times may range from 1 to 5, which is related to the training dataset and the practically-employed CNN structure. For each object class, we



(a) Original Caffenet

(b) DVN furnished Caffenet

Fig. 3. The classification results' comparison over MNIST datasets respectively based on the features resulted from original Lenet and DVN furnished Caffenet.

empirically generate 10% of its original training images according to ϕ_l.

We study the unbalanced image recognition task both on the number-based and the accuracy-based unbalanced datasets. For the first experiment, the classification of originally number-based balanced dataset, such as, CIFAR-10 dataset. The CIFAR-100 [23] is also studied to experiment with controlled unbalance levels. Further, to verify our DVN framework on accuracy based unbalanced datasets, we apply it to equip three types of state-of-the-art CNN networks, and conduct extensive experiments over five public benchmarks: MNIST [29], CIFAR-10, CIFAR-100, Imagenet and the Street View House Numbers (SVHN) Dataset [37]. The first type is the depth-unlimited networks, including Highway network [52], dense network [19] and deep residual network [16]. The second type is the task-oriented network, which can achieve remarkable accuracy on certain datasets in spite of its shallower-depth structure, including the Lenet [28] over MNIST dataset, a 7-layer network over MNIST dataset (denoted as Caffenet), and a 3-layer network over CIFAR-10 dataset (denoted as Simple network). The third type is a classic network with intermediate-depth structure, including VGG [50], network in network [33] and the Google inception network [54]. All the involved networks are trained in two ways: with DVN and without DVN. As for evaluation, we reimplement these methods according to the experimental settings documented in their original papers, and compare their testing error rates (only considering the top-one(five) accuracy rate) resulted from the cases with and without DVN.

6.2. Feature analysis

As shown in Fig. 3, the features extracted by DVN show consistent characteristics for the same class images, and have clear separations across different classes. The homogeneous features are extracted from the MNIST testing dataset, wherein each class has about 1000 images to be tested. The employed network is Lenet, whose network structure keeps the same with original paper, and the features we used to represent the image are from the last fully connected layer. In order to show the discriminative ability of heterogeneous features, we illustrate the feature distributions with the t-Distributed Stochastic Neighbor Embedding (t-SNE) method proposed in [27]. Fig. 3(a) shows the original Caffenet features. Fig. 3(b) shows the features resulted from DVN furnished Caffenet, wherein it has a relatively larger margin for each class and the points in the same classes are close to each other.

Thus, we can find that DVN can make the features extracted from CNN more discriminative. Furthermore, to analyze the resulted medium-level features, taking the intermediate convolution layers for example, we compare the feature maps resulted from DVN and original CNN. As shown in Fig. 4, we randomly select some samples from the ten classes of CIFAR-10, and show the feature maps learned from the second convolutional layer for feature activations. The second row shows feature maps extracted from DVN furnished GoogLenet, while the first row shows feature maps extracted from original GoogLenet. The DVN framework can activate a larger perception scope for high-level recognition tasks, so that the high level features focus more on the object related region than the original GoogLenet. The activations reserve more info about object-related features in the feature maps, while dropping the background regions in the images.

6.3. Comparisons with the state-of-the-art

We at first compare our DVN with two number-based unbalanced methods: cost sensitive [57] and oversampling [36]. Further, we compare our method with two high performance methods for improving the performance of CNNs, generative adversarial networks and center loss [59]. We re-organize the CIFAR-10 dataset into 11 levels of unbalanced datasets (unb1,... ,unb11) as described in [36] (Table 1 in supplement material). To control the unbalanced level, the subsets contain 65% of the images in the original CIFAR-10 dataset. Here, higher percentages are not possible, as certain amounts of data need to be removed to achieve the unbalanced distributions; and extremely small percentages are excluded, since such subsets would not train the network sufficiently, which goes beyond the research. The distributions are selected to be as mutually exclusive from each other as possible. All the settings are designed with linearly unbalanced representation, exponentially unbalanced representation, and major/minor/singular under/over representation according to the method [36]. With the reduced CIFAR-10, we conduct experiments with the oversampling method and DVN framework based on the Alexnet. Fig. 5(a) and (b) documents the class-wise accuracy on unb7 dataset. The accuracies of 70% classes are improved ranging from 1% to 7%, and the total accuracy is improved from 84.70% to 87.4%. The minor class is the airplane, which has an improvement of 7% than oversampling method. The ROC curves in Fig. 7 show improvements in AUC for most of the classes in "unb1" compared with the oversampling scheme [36] (The remaining results are provided in Fig.2 of the supplement material). The histogram in Fig. 6 shows 5% improvement over the equal-number distribution in "unb1", and the other unbalanced distributions have improvements from 1.8% to 4.3%, respectively. The results shown in Fig. 6 illustrate that, our method performs better than the oversampling method under all the distributions. In fact, oversampling based methods can

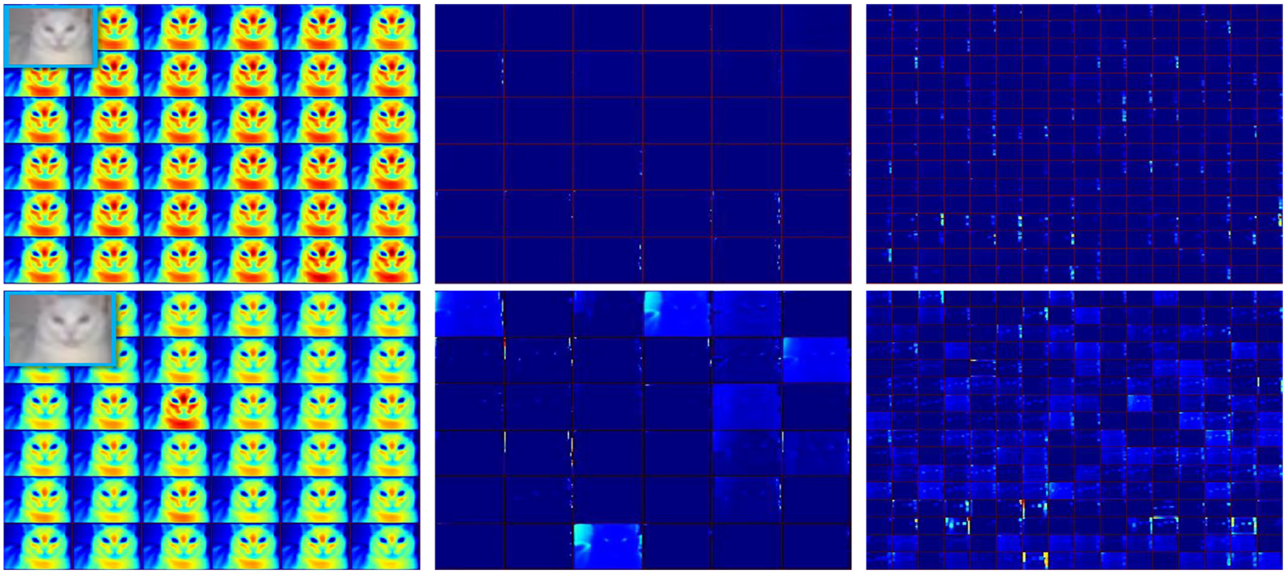
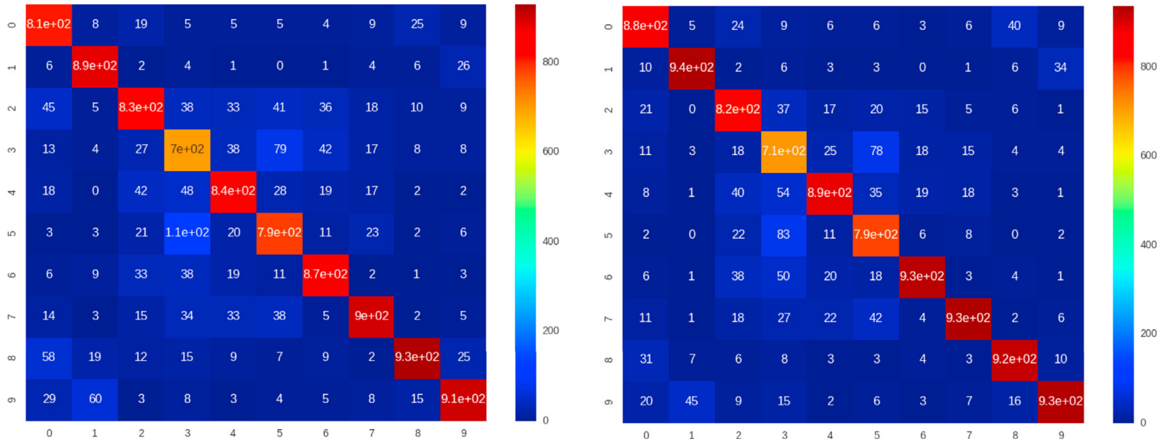


Fig. 4. Comparison of the feature maps learned from the ‘conv17x7_s2’, ‘conv23x3’ and ‘inception_3bpool’ layers, (convolutional layer of original GoogLenet and DVN furnished GoogLenet). The DVN framework can activate a larger perception scope for high-level recognition tasks, so that the high level features could focus more on the object related region. (More are provided in Fig. 11 of the supplement material).



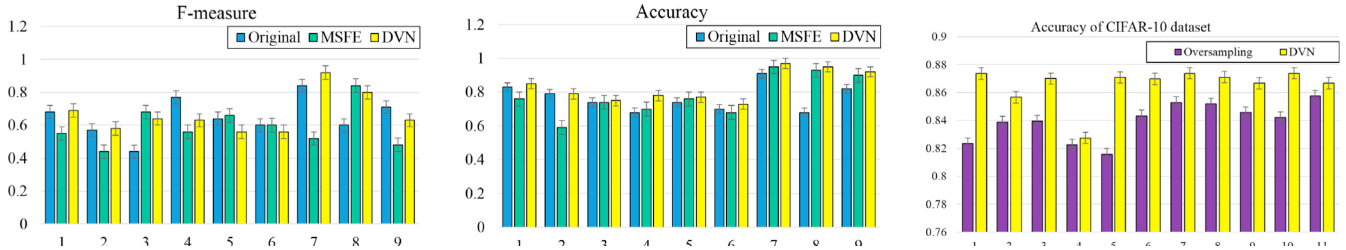
	0	1	2	3	4	5	6	7	8	9
0	8.1e+02	8	19	5	5	5	4	9	25	9
1	6	8.9e+02	2	4	1	0	1	4	6	26
2	45	5	8.3e+02	38	33	41	36	18	10	9
3	13	4	27	7e+02	38	79	42	17	8	8
4	18	0	42	48	8.4e+02	28	19	17	2	2
5	3	3	21	1.1e+02	20	7.9e+02	11	23	2	6
6	6	9	33	38	19	11	8.7e+02	2	1	3
7	14	3	15	34	33	38	5	9e+02	2	5
8	58	19	12	15	9	7	9	2	9.3e+02	25
9	29	60	3	8	3	4	5	8	15	9.1e+02

(a) Original Alexnet

	0	1	2	3	4	5	6	7	8	9
0	8.8e+02	5	24	9	6	6	3	6	40	9
1	10	9.4e+02	2	6	3	3	0	1	6	34
2	21	0	8.2e+02	37	17	20	15	5	6	1
3	11	3	18	7.1e+02	25	78	18	15	4	4
4	8	1	40	54	8.9e+02	35	19	18	3	1
5	2	0	22	83	11	7.9e+02	6	8	0	2
6	6	1	38	50	20	18	9.3e+02	3	4	1
7	11	1	18	27	22	42	4	9.3e+02	2	6
8	31	7	6	8	3	3	4	3	9.2e+02	10
9	20	45	9	15	2	6	3	7	16	9.3e+02

(b) DVN furnished Alexnet

Fig. 5. Confusion matrix of original and DVN furnished Alexnet over CIFAR-10 dataset.



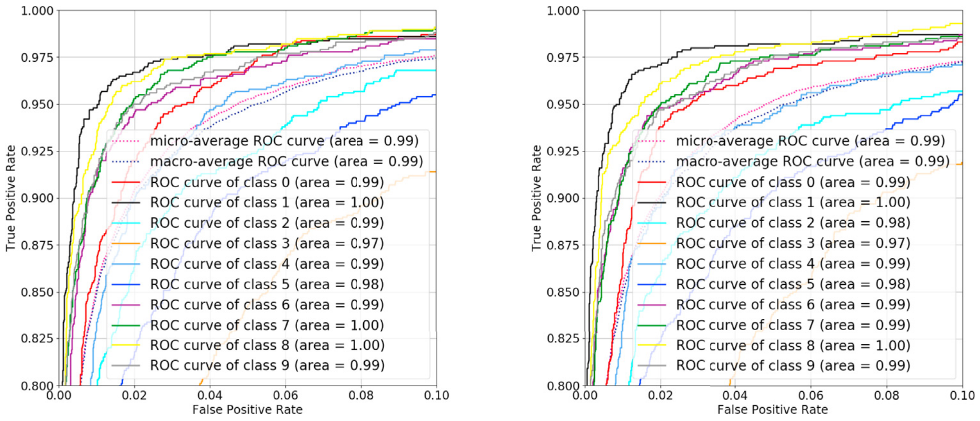
(a) F-measure with AlexNet

(b) Accuracy with AlexNet

(c) Unbalance Levels

Fig. 6. Accuracy and F-measure comparisons of original CNN, MSFE loss and DVN furnished Alexnet on CIFAR-100. The unbalance levels on CIFAR-10.

not expand the original distribution of training samples for all the classes via simple number measurement. Our DVN will learn the inter-class homogeneity from the major classes so as to extend the distribution of the minor ones. Then we compare our DVN with the cost-sensitive method MSFE [57]. We conduct the experiment over CIFAR-100 by making the numbers of the classes unbalanced based on the original settings [57] (imb1, imb2, ... , imb9). CIFAR-100 contains 60,000 images belonging to 100 classes (600 images/class), which are further divided into 20 super-classes. The numbers of training and testing image samples for each class are, respectively, 500 and 100. To evaluate our algorithm on various scales of datasets, three different-size datasets are extracted. The first one is relatively large, a mixture of two superclass samples (household furniture and electrical devices). The other two small



(a) DVN Furnished Alexnet

(b) Oversampling Furnished Alexnet

Fig. 7. ROC curves of DVN furnished and original Alexnet over CIFAR-10 dataset in a representative unbalance level "unb1" (more are provided in Fig. 2 of the supplement material).

Table 1
Unbalance levels of MNIST for GAN and center loss. The original MNIST dataset is split into 10 small datasets by random equally. '10' represents all the samples.

Class	0	1	2	3	4	5	6	7	8	9
UNB1	1	1	1	1	1	1	1	1	1	10
UNB2	10	10	10	10	10	10	10	10	10	1
UNB3	10	10	10	10	10	1	1	1	1	1
UNB4	5	5	5	5	5	5	10	10	10	10

ones have approximate sizes, each of which is the combination of two-class samples, which are randomly selected from the superclass trees [57]. The results in Fig. 6(c) show that, our DVN furnished network has similar stability with MSFE while improving the accuracy for the number-based unbalanced dataset. The experiments on the CIFAR-10 and CIFAR-100 datasets show that, the accuracy-based methods can be benefit to the number-based unbalanced datasets.

In addition, we further explore the DVN with DCGAN [43]. In order to test if DVN could benefit the classification task, and to compare with the GAN single augmented CNNs, we conduct two experiments on MNIST and CIFAR-10 datasets. The first is conducted on MNIST-UNB4 with different levels in Table 1. The results are shown in Table 2. The proposed DVN furnished Alexnet performs better than DCGAN from 'UNB1' to 'UNB4'. The small number of samples in unbalanced datasets also makes DCGAN fail to generate reasonable images. We also conduct experiments with DCGAN on CIFAR-10 dataset to test the effect on images with various backgrounds. The results show that, the performance of hierarchical Bayesian is much better than DCGAN's, although DCGAN improves the Caffenet slightly by 1.2%. (*More results are listed in Fig. 17 in the supplement material*).

The center loss shows good performance when visualizing the MNIST dataset. In order to compare with it, we conduct the experiments on MNIST-UNB4 dataset and MNIST original dataset. The results show that the performance of DVN furnished center loss is better. Fig. 8 shows that, at 'UNB3' level, the DVN recognizes all the 10 classes instead of 9 as the center loss do(*Others are listed in supplement material: Figs. 14–16*).

6.4. Performance evaluations

In this section, we evaluate our method on accuracy-based unbalanced datasets. Furthermore, in Fig. 9, the original CIFAR-10, SVHN, and MNIST datasets are demonstrated as accuracy-based unbalanced ones with corresponding networks. After some (ranging from 40,000 to 50,000 for different networks) times of iterations, the loss is stable, but the diagonal elements of the confusion matrix are still not equal. Therefore, we classify the three datasets as accuracy-based unbalanced datasets. For this unbalanced datasets, we use our DVN to improve the performance.

Performance comparison. We have conducted extensive experiments on both toy and real world dataset, the classes ranges from digit handwritings to animals with complex backgrounds. The experiments can be classified into two types.

Firstly, we conduct experiments on 8 networks over 5 datasets without unbalance level control, as shown in Tables 3 and 4.

Secondly, we conduct experiments on imagenet100 dataset, which are randomly selected in the 1000 classes. The dataset is split with train and test dataset as 8:2. Most of the images have more complex background and largely-varying objects. The results are shown in Table 4.

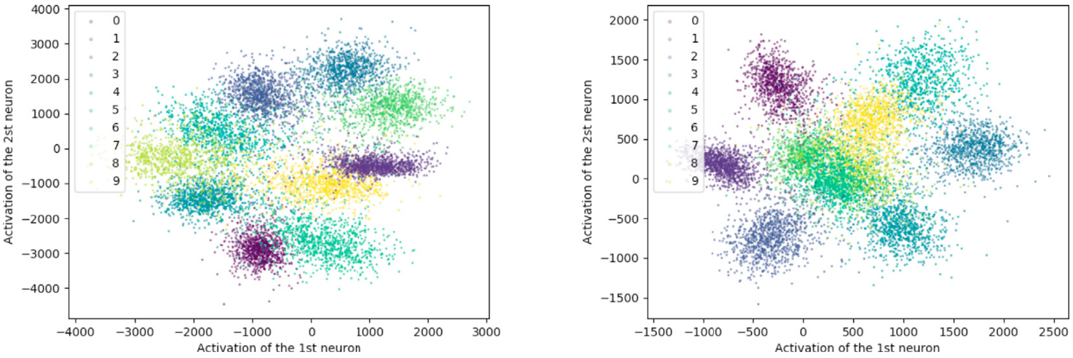
In the first experiment, as documented in Table 3, most of the involved networks' performances can benefit from our DVN. The bold text is the highest of all the tested networks. On CIFAR-10 benchmark, DVN-furnished VGG achieves nearly state-of-the-art performance (95.7% accuracy), where each class has a more balanced accuracy than original VGG network. And other networks can also gain performance improvement ranging from 2.3% to 6.9%. On MNIST benchmark, the DVN-furnished Caffenet network achieves state-of-the-art performance (only with 0.25% error rate), whose error rate has been reduced by 52.83%. For other networks, the largest decrease of error rate reaches to 50%.

Particularly, in our defined three subspaces, the generated samples are different, the samples in the "overlapping" subspace have higher resolution than the samples in ϕ_f subspace. The obscure ones are used for adding harder cases for the top K_c (K_c is computed based on the distances between the classes, and we empirically set $K_c = 30\%$ in the following experiments) easily-recognized classes, which are used as a regularizer to slow down the convergence speed to avoid overfitting. The generated images demonstrates that, Bayesian network tends to generate more varying samples for the "overlapping" space ϕ_l.

With the generated images, the accuracy of the ϕ_l subspace is improved, while the accuracies of the top 3 classes in "flat" subspace ϕ_f are decreased slightly. As indicated in Fig. 11(b), the loss values of the original network for CIFAR-10 and MNIST datasets are lower than the loss values with DVN, but the accuracy is also lower than DVN. Thus, for the unbalanced cases, the original Alexnet may easily cause overfitting. Furthermore, we find that DVN performs even better when being combined with the deeper network. And

Table 2
Comparsions of GAN and DVN on MNIST-UNB4 datasets with different unbalance levels.

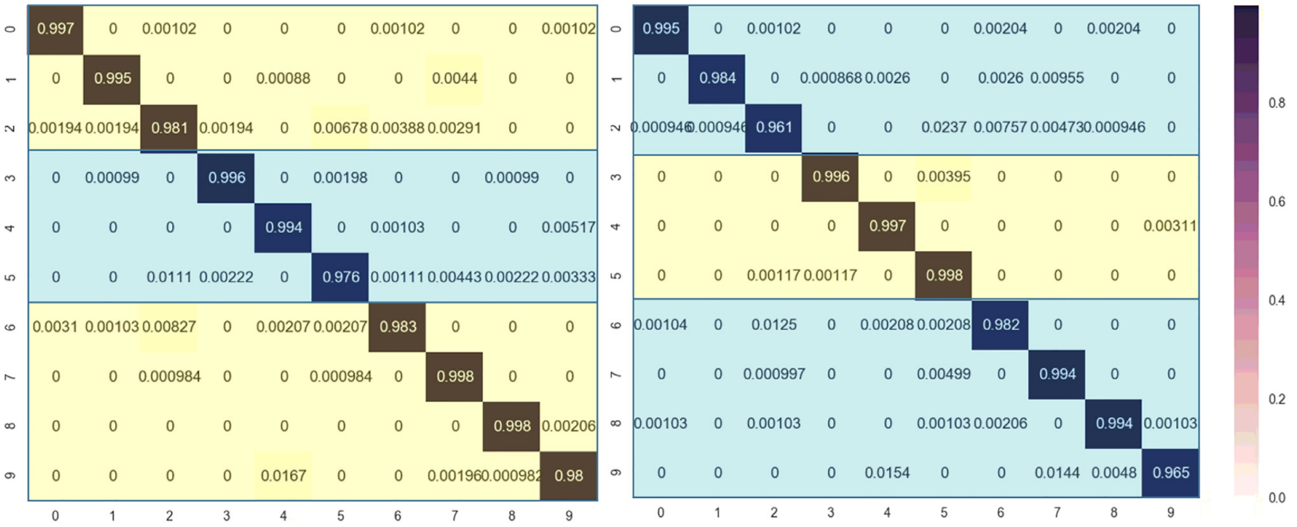
ID	ORI		GAN		DVN		GAN+DVN	
Dataset	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
UNB1	*0.978188*	0.0590697	0.984398	0.0749743	**0.980398**	0.05468816	0.980099	0.0973185
UNB2	*0.983937*	0.0293378	0.992838	0.0371489	**0.992838**	0.0314566	0.889201	1.63811
UNB3	*0.988625*	0.0550345	0.985359	0.0717597	**0.985359**	0.0550024	0.984879	0.0826827
UNB4	*0.98975*	0.0287776	0.993938	0.0363117	**0.994625**	0.0314365	0.985719	0.0698933



(a) DVN Furnished Center Loss

(b) Original Center Loss

Fig. 8. Features' distributions in “ip1” layer of Lenet++ over MNIST dataset (center loss weight is set as 0.01).



	0	1	2	3	4	5	6	7	8	9
0	0.997	0	0.00102	0	0	0	0.00102	0	0	0.00102
1	0	0.995	0	0	0.00088	0	0	0.0044	0	0
2	0.00194	0.00194	0.981	0.00194	0	0.00678	0.00388	0.00291	0	0
3	0	0.00099	0	0.996	0	0.00198	0	0	0.00099	0
4	0	0	0	0	0.994	0	0.00103	0	0	0.00517
5	0	0	0.0111	0.00222	0	0.976	0.00111	0.00443	0.00222	0.00333
6	0.0031	0.00103	0.00827	0	0.00207	0.00207	0.983	0	0	0
7	0	0	0.000984	0	0	0.000984	0	0.998	0	0
8	0	0	0	0	0	0	0	0	0.998	0.00206
9	0	0	0	0	0.0167	0	0	0.00196	0.000982	0.98

	0	1	2	3	4	5	6	7	8	9
0	0.995	0	0.00102	0	0	0	0.00204	0	0.00204	0
1	0	0.984	0	0.000868	0.0026	0	0.0026	0.00955	0	0
2	0.000946	0.000946	0.961	0	0	0.0237	0.00757	0.00473	0.000946	0
3	0	0	0	0.996	0	0.00395	0	0	0	0
4	0	0	0	0	0.997	0	0	0	0	0.00311
5	0	0	0.00117	0.00117	0	0.998	0	0	0	0
6	0.00104	0	0.0125	0	0.00208	0.00208	0.982	0	0	0
7	0	0	0.000997	0	0	0.00499	0	0.994	0	0
8	0.00103	0	0.00103	0	0	0.00103	0.00206	0	0.994	0.00103
9	0	0	0	0	0.0154	0	0	0.0144	0.0048	0.965

Fig. 9. Demonstrations of the accuracy-based unbalanced problem dealing with MNIST dataset.

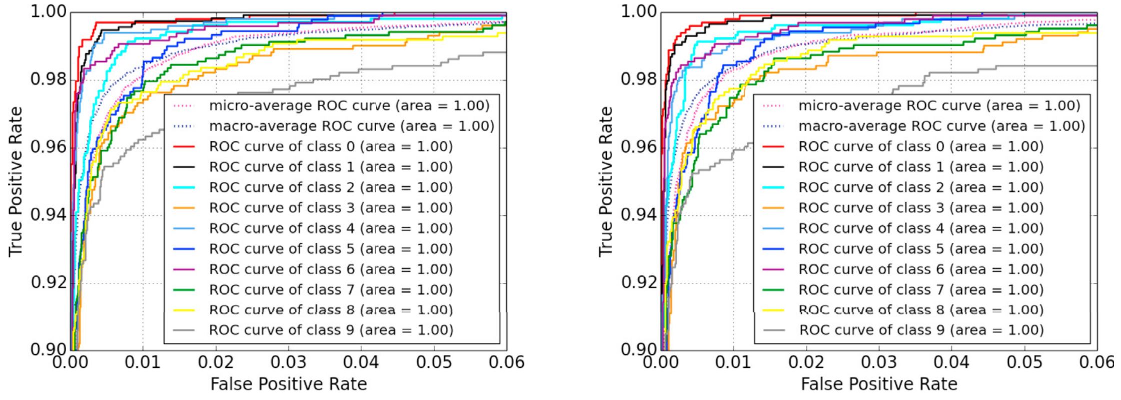
the result confirms the belief that deep networks commonly perform better than shallow ones.

As for SVHN [37] dataset, we use their training-testing split data published on the website. In order to test our DVN's ability in handling raw images and small scale training datasets, we train the networks without any preprocessing and data augmentation operations on the Alexnet [25], Simple network [21] and GoogLenet [54], VGGNet [50], etc. In order to compare DVN with the simple data augmented technologies (including cropping, horizontal reflection and scaling), we respectively test such cases using the VGG network [50] and Network in network (NIN) [33], using Global Contrast Normalization (GCN) and Zero-Phase Component Analysis (ZCA) whitening to normalize the dataset. In the testing process, the DVN resulted accuracy improvement of each class is more balanceable than that benefitting from the simple data augmentation for original network. To verify all the classes are learned and classified in balance, we compare the multiple-classes ROC curves over the three datasets, and show the curves in Fig. 10 (*the ROC curves of the SVHN and CIFAR-10 are provided in Figs. 9 and 10 of the supplement material*). The results state that, most of the network furnished with DVN will be better than the corresponding original ones. Besides, the DVN furnished VGG network has the similar improvement over the CIFAR-10 dataset. The performance of the DVN on SVHN dataset is shown in Table 3. The results show two advantages of DVN: the first one is to improve the performance of the existed network including Simple Network, Caffenet, GoogLenet and Alexnet. The second is to improve the class-wise accuracy in a balanced way, including VGG network and NIN.

The second experiment is conducted on Imagenet100 datasets with more deeper network including residual networks [16], Densenet [19]. As Table 4 illuminated, The improvement ranges from 0.76% to 10.39%. The best performance is from the DVN-furnished VGGNet with 500,000 iterations. DVN fails to improve lenet and Simplenet due to the large scale dataset with too sim-

Table 3
Performance comparisons of DVN-furnished networks over CIFAR-10, MNIST and SVHN datasets. VGGNet has 19 layers.

ID	Dataset	Network	Performance improvement	Original performance	DVN-furnished performance
1	CIFAR-10	VGGNet(19) [50]	3.2	92.5	95.7
2	CIFAR-10	GoogLenet [54]	2.3	87.1	89.4
3	CIFAR-10	Simple network [21]	6.9	80.5	87.4
4	CIFAR-10	Network in network (NIN) [33]	2.1	92.4	94.5
5	CIFAR-10	Caffenet [21]	0.9	88.5	89.4
6	CIFAR-10	Alexnet [25]	2.0	83.4	85.4
7	CIFAR-10	Highway [52]	1.9	60.0	61.9
8	CIFAR-10	Lenet [28]	−0.5	65.9	65.4
9	MNIST	VGGNet(19) [50]	0.5	98.9	99.4
10	MNIST	GoogLenet [54]	0.3	99.2	99.5
11	MNIST	Simple network [21]	1.1	97.4	98.5
12	MNIST	Network in network (NIN) [33]	0.9	98.9	99.3
13	MNIST	Caffenet [21]	0.3	99.5	99.8
14	MNIST	Alexnet [25]	0.3	99.5	99.8
15	MNIST	Highway [52]	0.2	99.5	99.7
16	MNIST	Lenet [28]	0.1	99.1	99.2
17	SVHN	VGGNet (19) [50]	0.1	96.8	96.9
18	SVHN	GoogLenet [54]	0.9	94.6	95.5
19	SVHN	Simple network [21]	1.2	93.1	94.3
20	SVHN	Network in network (NIN) [33]	0.1	96.7	96.8
21	SVHN	Caffenet [21]	–	–	–
22	SVHN	Alexnet [25]	2.1	94.1	96.2
23	SVHN	Highway [52]	0.8	76.1	76.9
24	SVHN	Lenet [28]	−0.6	87.4	86.8



(a) DVN furnished Alexnet.

(b) Original Alexnet.

Fig. 10. ROC curves of DVN and original networks on MNIST dataset (More results are shown in Fig. 8 in supplement material.).

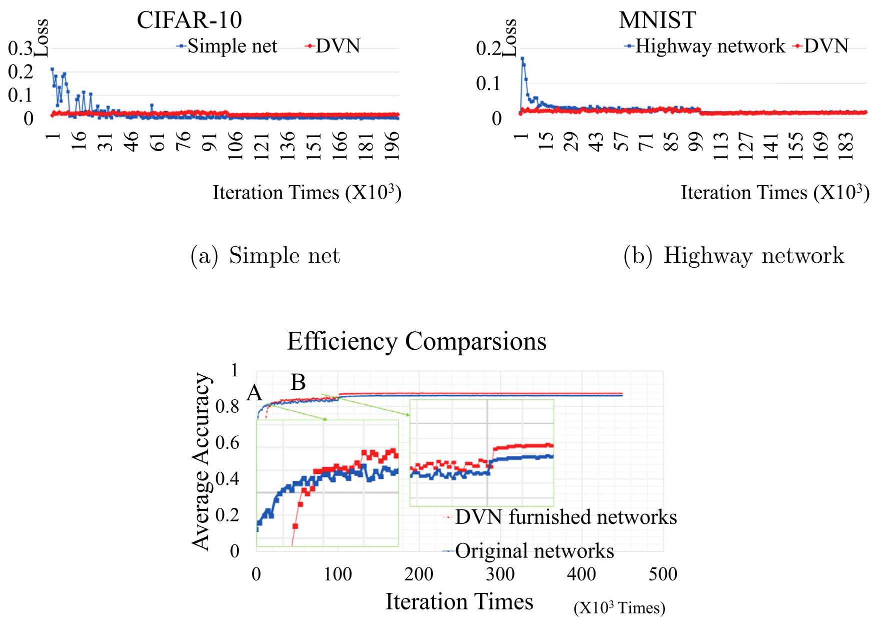
Table 4
Deep CNNs on Imagenet100 dataset.'@*k*' means top *k* ranking. Residualnet has 50 layers, while Densenet has 121 layers. VGGNet has 19 layers.

Imagenet100	Original (@1)	Original (@5)	DVN (@1)	DVN (@5)
Residualnet(50) [16]	75.46	94.21	**79.24**	**96.73**
GoogLenet [54]	72.13	92.36	**72.89**	**94.52**
NIN [33]	67.68	89.24	**69.39**	**90.21**
VGGNet(19) [50]	89.27	98.29	**89.99**	**98.49**
CaffeNet [21]	65.99	86.47	**66.64**	**86.77**
SimpleNet [21]	**41.76**	**68.17**	33.06	59.59
Highway [52]	42.64	–	**48.07**	–
Lenet [28]	**19.21**	**22.83**	14.89	19.37
Densenet(121) [19]	75.28	93.21	**78.89**	**95.62**
Alexnet [25]	54.97	79.61	**65.36**	**85.20**

ple and shallow networks. The results show that on more complex dataset with a large number of classes, the DVN performs better on deeper networks including residual net and VGGNet.

Comparison of training cost and convergence. For the involved networks, we compare their 450,000-iteration time costs with that of our DVN. In fact, our method can benefit the unbalanced datasets both in efficiency and performance in the early training procedure, while the efficiency and performance need to be traded off in the late training procedure. We have compared the Alexnet with Caffenet on CIFAR-10 dataset. The efficiency and accuracy curves are shown in Fig. 11(d), where the red curve records the single time cost of DVN framework, and the blue curve records the single time cost of original network. For the original settings in CIFAR-10 dataset, the cost of DVN framework is a trade-off choice between accuracy and iteration times. To reach the same accuracy of original networks, the efficiency is not decreased. Before point A, which is the first turning point of the accuracy curve, the efficiency is lower than original networks. During the processing between point A and point B, the efficiency is close to original networks, but after point B, the efficiency is higher than original ones. After point B, both original and DVN-furnished networks keep stable.

To evaluate the convergence of our DVN, we compare the training process of Simple network and Highway network by measuring the gradient decrease of the loss functions. Simple network is trained with the “basic learning rate” of 0.1 and the “momentum” of 0.9, while Highway network (10 layers) is trained on MNIST, whose learning policy is the same as that in [52]. As shown in Fig. 11 benefitting from the additional term added in our loss func-



(a) Simple net

(b) Highway network

(c) Average Efficiency

Fig. 11. Comparisons of the convergence of DVN over original networks. (Zoom in to see more clearly.)

Algorithm 1 DVN training process.

```
Input: Original Training Dataset: $x$
Output: Virtual images with Bayesian network
Initialize $x \sim \sum_s N(\mu, \Sigma)$
while KL term not converged do
    for $i \in [1, 2, \ldots, t]$ do
        Sample $\{x_i\}_{i=1}^m \sim \mathbb{P}$ a mini batch from real data
        Randomly generate virtual samples:
        $s_1^i \sim p(S_1 = s_1 | S_2 = s_2^{i-1}, S_3 = s_3^{i-1}, \ldots, S_K = s_K^{i-1})$
        $s_2^i \sim p(S_2 = s_2 | S_1 = s_1^{i-1}, S_3 = s_3^{i-1}, \ldots, S_K = s_K^{i-1})$
        ...
        $s_K^i \sim p(S_K = s_K | S_1 = s_1^{i-1}, S_2 = s_2^{i-1}, \ldots, S_{K-1} = s_{K-1}^{i-1})$
        $f(x) \leftarrow$ CNN forward and back propagations
        $g_W \leftarrow \Delta_W E$
        $w \leftarrow w + \eta$ SGDProp$(w, g_w)$
    end for
    Cluster $[\phi_f, \phi_l, \phi_n] \leftarrow hyper - graph(f(x))$
    Kernel density estimation of $p(x \in \phi_{p=\{f,n,l\}})$
    $p(x \in \phi_{p=\{f,n,l\}}) \leftarrow \frac{1}{s}\sum_{j=1}^{s} K_h(x - x_j) \leftarrow \frac{1}{sh}\sum_{j=1}^{s} K(\frac{x-x_j}{h})$ (h
is set by grid search by each class)
    $f(x|s) \leftarrow \mathcal{N}(\mu(H_o, H_e), \Sigma(H_o, H_e))$
    $H_o \leftarrow \sum f(x_\phi) / \sum ||\phi||$
    $H_e \leftarrow cov(x_\phi, x_\phi)$
end while
```

tion, DVN converges faster than original networks, of which, our loss function converges faster than the original loss function at the starting and middle stages of the training process (DVN is initialized from 1000 iterations original Networks). At the last stage, our loss function's performance is stable, whose curve remains relatively higher than that of Simple network but lower than that of Highway network. It indicates that, DVN could provide an effective scheme to avoid the overfitting problem in shallow networks.

7. Discussion

We try to find the proper theoretic boundary of our DVN via the analysis in the aspects of features' distribution and loss domination. *(1) Features' distribution in training dataset.* To make each minority class learn balanced homogeneity from majority class, we suggest an empirical equation to calculate the theoretical least training sample number for each class. The equation is under the assumption that, the majority class' samples can cover most of the characteristics involved in the corresponding object instances, and whose feature distribution in some sense has similarity with the minority class.

$$N_{min} = \sigma * \max_i (N_i)/C. \tag{19}$$

Here, N_i is the number of the i_{th} class' samples. C is the total number of all the classes involved in the training data set. In MNIST-UNB4 datasets, $N_{\max} = 6000$ is considered sufficiently enough to cover most of the characteristics [16,19]. Here, $C = 10$, based on Eq. (19), $N_{\min} = 600$. For "UNB1", "UNB2" and "UNB3" datasets, the sample numbers of the minority classes are right close to the boundary ($N_{\min} = 600$), our DVN's effect on accuracy ranges from −0.003 to 0.001. For "UNB4", the sample number of the minority classes are 3000, which are larger than $N_{\min}$, thus, the total accuracy gain of our DVN is 0.01. For the well-balanced levels, the DVN-furnished Alexnet improves the accuracy by 0.005. *(2)Domination of loss.* In the training process, the majority class samples are most likely to dominate the loss, which can be observed from the trend of the class-specific loss (accuracy confusion matrix). To obtain relatively-balanced training loss, our DVN facilitates to enforce the minority class having the same or relatively higher voting weight for the total loss. If the classification accuracy of certain class is significantly lower than others', it is hard to be improved by the DVN-furnished network. Therefore, with the training dataset satisfying our empirically theoretical boundary of unbalance level, our DVN can achieve better accuracy gain than that achieved on the datasets out of the theoretical unbalance boundary.(More validation experiments are conducted in section 'Discussion' in supplement material).

8. Conclusion and further work

In this paper, we have presented an iterative, improved CNN framework to handle unbalanced training datasets. It provides a universal and powerful scheme to further enhance the performance of the sate-of-the-art deep neural networks. Meanwhile, many of the involved technical elements, including hypergraph based CNN feature subspace construction, inner-class heterogeneity, and inter-class homogeneity definition, multi-level features/distributions involved hierarchical Bayesian model, and Bayesian prior guided iterative generation of virtual training samples, collectively contribute to many other pattern recognition related applications in computer vision and other related fields. Moreover, different types of experiments have demonstrated our framework's advantages in terms of accuracy, robustness, convergence, efficiency, and versatility.

However, at present our framework still has some limitations. According to our experiments, both GoogLenet and deep residual networks, trained over the willfully-tailored subset of the well-balanced Imagenet dataset, perform worse than those trained over complete Imagenet dataset. It indicates that, the synthetic images generated by our DVN may have some semantic conflicts with the real-captured images. Therefore, in the near future, we will also endeavor our upcoming efforts to exploit and encode more semantically meaningful priors in our framework.

Acknowledgments

This research is supported in part by National Natural Science Foundation of China (Nos. 61672077 and 61532002), Applied Basic Research Program of Qingdao (No. 161013xx) and National Science Foundation of USA (Nos. IIS-0949467, IIS-1047715, IIS-1715985, IIS-61672149, and IIS-1049448). The capital health research and development of special 2016-1-4011.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patcog.2018.03.035.

References

[1] M.J. Afridi, A. Ross, E.M. Shapiro, On automated source selection for transfer learning in convolutional neural networks, Pattern Recognit. 73 (Supplement C) (2018) 65–75, doi:10.1016/j.patcog.2017.07.019.

[2] A. Ali, S.M. Shamsuddin, A.L. Ralescu, Classification with class imbalance problem: a review, Int. J. Adv. Soft Comput. Appl. 7 (3) (2015) 176–204.

[3] S. Ando, C.Y. Huang, Deep over-sampling framework for classifying imbalanced data. arXiv preprint arXiv:1704.07515. [cs.CV], 2017.

[4] J. Bohne, Y. Ying, S. Gentric, M. Pontil, Learning local metrics from pairwise similarity data, Pattern Recognit. 75 (Supplement C) (2018) 315–326, doi:10.1016/j.patcog.2017.04.002.

[5] A.L. Brun, A.S. Britto, L.S. Oliveira, F. Enembreck, R. Sabourin, A framework for dynamic classifier selection oriented by the classification problem difficulty, Pattern Recognit. 76 (Supplement C) (2018) 175–190, doi:10.1016/j.patcog.2017.10.038.

[6] A.L. Brun, A.S. Britto, et al., A framework for dynamic classifier selection oriented by the classification problem difficulty, Pattern Recognit. 76 (Supplement C) (2018) 175–190, doi:10.1016/j.patcog.2017.10.038.

[7] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.

[8] B. Chen, J. Li, G. Wei, B. Ma, A novel localized and second order feature coding network for image recognition, Pattern Recognit. 76 (2018) 339–348, doi:10.1016/j.patcog.2017.10.039.

[9] C. Chen, S. Li, H. Qin, A. Hao, Structure-sensitive saliency detection via multilevel rank analysis in intrinsic feature space, IEEE Trans. Image Process. 24 (8) (2015) 2303–2316.

[10] C. Chen, S. Li, Y. Wang, H. Qin, A. Hao, Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion, IEEE Trans. Image Process. 26 (7) (2017) 3156–3170.

[11] K. Dembczynski, A. Jachnik, W. Kotłowski, W. Waegeman, E. Hüllermeier, Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization, in: Proceedings of the Thirtieth International Conference on International Conference on Machine Learning, 28, JMLR.org, 2013.

[12] A. Dosovitskiy, T. Brox, Generating images with perceptual similarity metrics based on deep networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 658–666.

[13] D. Erhan, Y. Bengio, et al., Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11 (2010) 625–660.

[14] F.-F. Li, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: Proceedings of the Computer Vision and Pattern Recognition, IEEE, 2005, pp. 524–531.

[15] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: adaptive synthetic sampling approach for imbalanced learning, in: Proceedings of the IEEE International Joint Conference on Neural Networks, 2008, pp. 1322–1328.

[16] 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.

[17] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670, doi:10.1109/TIP.2015.2487860.

[18] C. Huang, Y. Li, C. Change Loy, X. Tang, Learning deep representation for imbalanced classification, in: Proceedings of the Computer Vision and Pattern Recognition, IEEE, 2016, pp. 5375–5384.

[19] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[20] M. Jaderberg, K. Simonyan, Zisserman, Andrew, Spatial transformer networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 2017–2025.

[21] Y. Jia, E. Shelhamer, J. Donahue, et al., Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2017, pp. 675–678.

[22] L. Jiao, S. Zhang, L. Li, F. Liu, W. Ma, A modified convolutional neural network for face sketch synthesis, Pattern Recognit. 76 (2018) 125–136, doi:10.1016/j.patcog.2017.10.025.

[23] A. Krizhevsky, Learning multiple layers of features from tiny images, Department of Computer Science, University of Toronto, 2009.

[24] 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.

[25] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[26] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338.

[27] V.D.M. Laurens, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2605) (2008) 2579–2605.

[28] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[29] LeCun, Y., Cortes, C., Burges, C. J., 2012. The MNIST database of handwritten digits, 1998. Available electronically at http://yann.lecun.com/exdb/mnist.

[30] C. Li, J. Zhu, B. Zhang, Max-margin deep generative models for (semi-) supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. (2017) 1, doi:10.1109/TPAMI.2017.2766142.

[31] S. Li, M. Shao, Y. Fu, Person re-identification by cross-view multi-level dictionary learning, IEEE Trans. Pattern Anal. Mach. Intell. (2017) 1, doi:10.1109/TPAMI.2017.2764893.

[32] X. Li, D. Zhu, M. Dong, Multinomial classification with class-conditional overlapping sparse feature groups, Pattern Recognit. Lett. 101 (Supplement C) (2018) 37–43, doi:10.1016/j.patrec.2017.11.002.

[33] M. Lin, Q. Chen, S. Yan, Network in network. arXiv preprint arXiv:1312.4400. [cs.CV], 2013.

[34] M. Liu, C. Xu, Y. Luo, C. Xu, Y. Wen, D. Tao, Cost-sensitive feature selection by optimizing f-measures, IEEE Trans. Image Process. 27 (3) (2018) 1323–1335, doi:10.1109/TIP.2017.2781298.

[35] J.-H. Luo, J. Wu, W. Lin, Thinet: a filter level pruning method for deep neural network compression. arXiv preprint arXiv:1707.06342. [cs.CV], 2017.

[36] D. Masko, P. Hensman, The impact of imbalanced training data for convolutional neural networks, Degree Project in Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.

[37] Y. Netzer, T. Wang, et al., Reading digits in natural images with unsupervised feature learning, in: Proceedings of the NIPS Workshop on Deep Learning Unsupervised Feature Learning, 2010, p. .

[38] K. Nogueira, O.A.B. Penatti, J.A Dos Santos, Towards better exploiting convolutional neural networks for remote sensing scene classification, Pattern Recognit. 61 (2017) 539–556, doi:10.1016/j.patcog.2016.07.001.

[39] N. van Noord, E. Postma, Learning scale-variant and scale-invariant features for deep image classification, Pattern Recognit. 61 (2017) 583–592, doi:10.1016/j.patcog.2016.06.005.

[40] L. Peng, H. Zhang, Y. Chen, B. Yang, Imbalanced traffic identification using an imbalanced data gravitation-based classification model, Comput. Commun. 102 (3) (2017) 177–189.

[41] P. Purkait, T. Chin, et al., Clustering with hypergraphs: the case for large hyperedges, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 672–687.

[42] S. Puthiya Parambath, N. Usunier, Y. Grandvalet, Optimizing F-measures by cost-sensitive classification, in: Proceedings of the Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 2123–2131.

[43] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. [cs.CV], 2015.

[44] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, T. Raiko, Semi-supervised learning with ladder networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 3532–3540.

[45] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: imagenet classification using binary convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 525–542.

[46] O. Russakovsky, J. Deng, H. Su, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[47] R. Salakhutdinov, G.E. Hinton, Deep Boltzmann machines, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2009, pp. 448–455.

[48] A. Santoro, S. Bartunov, et al., One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065. [cs.CV], 2016.

[49] X. Shang, Z. Zhu, B. Leimkuhler, A.J. Storkey, Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 37–45.

[50] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[51] Son, Y., Lee, S., Park, S., Lee, J.,. Learning representative exemplars using one-class Gaussian process regression. Pattern Recognit. 74, 185–197. doi:10.1016/j.patcog.2017.09.002.

[52] R.K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 2368–2376.

[53] J. Susskind, R. Memisevic, G. Hinton, M. Pollefeys, Modeling the joint density of two images under a variety of transformations, in: Proceedings of the Computer Vision and Pattern Recognition, IEEE, 2011, pp. 2793–2800.

[54] C. Szegedy, W. Liu, Y. Jia, et al., Going deeper with convolutions, in: Proceedings of the Computer Vision and Pattern Recognition, IEEE, 2015, pp. 1–9.

[55] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the Computer Vision and Pattern Recognition, IEEE, 2015, pp. 2818–2826.

[56] C. Tzelepis, V. Mezaris, I. Patras, Linear maximum margin classifier for learning from uncertain data, IEEE Trans. Pattern Anal. Mach. Intell. (2017) 1, doi:10.1109/TPAMI.2017.2772235.

[57] Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., Kennedy, P. J., Training deep neural networks on imbalanced data sets. In: Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 4368–4374.

[58] Y. Wang, C. Xu, S. You, D. Tao, C. Xu, Cnnpack: Packing convolutional neural networks in the frequency domain, in: Proceedings of the Advances in Neural Information Processing Systems 29, Curran Associates, Inc., 2016, pp. 253–261.

[59] 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, Springer, 2016, pp. 499–515.

[60] Y. Yang, J. Jiang, Bi-weighted ensemble via HMM-based approaches for temporal data clustering, Pattern Recognit. 76 (2018) 391–403, doi:10.1016/j.patcog.2017.11.022.

[61] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. PP (99) (2017) 1–11.

[62] J. Yu, B. Zhang, Z. Kuang, D. Lin, J. Fan, Iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning, IEEE Trans. Inf. Forensics Secur. 12 (5) (2017) 1005–1016.

[63] Zheng, Z., Zheng, L., Yang, Y., 2017. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. arXiv preprint arXiv:1701.07717.

[64] T. Zhu, Y. Lin, Y. Liu, Synthetic minority oversampling technique for multi-class imbalance problems, Pattern Recognit. 72 (Supplement C) (2017) 327–340, doi:10.1016/j.patcog.2017.07.024.

Shuai Li received the Ph.D. degree in computer science from Beihang University. He is currently an associate professor at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include computer vision, image processing, computer graphics, physics-based modeling and simulation, and virtual surgery simulation.

Wenfeng Song is currently a Ph.D. candidate at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. Her research interests include computer vision, machine learning and image processing.

Hong Qin received the B.S. and M.S. degrees in computer science from Peking University. He received the Ph.D. degree in computer science from the University of Toronto. He is a professor of computer science in the Department of Computer Science, Stony Brook University. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer-aided geometric design, visualization, and scientific computing. He is a senior member of the IEEE.

Aimin Hao is a professor in Computer Science School and the Associate Director of State Key Laboratory of Virtual Reality Technology and Systems at Beihang University. He received his B.S., M.S., and Ph.D. in Computer Science at Beihang University. His research interests are on virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.