arXiv:2212.07070v1 [cs.CV] 14 Dec 2022

Deep Negative Correlation Classification

Le Zhang, Qibin Hou, Yun Liu, Jia-Wang Bian, Xun Xu, Joey Tianyi Zhou and Ce Zhu, *Fellow, IEEE*

***Abstract*—Ensemble learning serves as a straightforward way to improve the performance of almost any machine learning algorithm. Existing deep ensemble methods usually naïvely train many different models and then aggregate their predictions. This is not optimal in our view from two aspects: i) Naïvely training multiple models adds much more computational burden, especially in the deep learning era; ii) Purely optimizing each base model without considering their interactions limits the diversity of ensemble and performance gains. We tackle these issues by proposing deep negative correlation classification (DNCC), in which the accuracy and diversity trade-off is systematically controlled by decomposing the loss function seamlessly into individual accuracy and the "correlation" between individual models and the ensemble. DNCC yields a deep classification ensemble where the individual estimator is both accurate and "negatively correlated". Thanks to the optimized diversities, DNCC works well even when utilizing a shared network backbone, which significantly improves its efficiency when compared with most existing ensemble systems. Extensive experiments on multiple benchmark datasets and network structures demonstrate the superiority of the proposed method.**

***Index Terms*—Ensemble Learning, Diversity, Negative Correlation Learning, Deep Learning**

I. Introduction

ENSEMBLE learning typically fuses multiple models to get better performance than its individual models. It has been used in multiple research fields such as machine learning [1]–[4], computer vision [5]–[7], and so on. Dietterich [8] explained the success of ensemble learning from the statistical, computational, and representational views. In addition, bias-variance decomposition [6], [9] and strength-correlation [2] also shed light upon the rationale of ensemble learning theoretically.

Apart from the accuracy of the individual estimator, it is widely convinced that much of the success of ensemble learning is attributed to the degree of disagreement, or "diversity", within the system. A simple yet intuitive explanation is that millions of identical estimators are obviously no better than any individual amongst them. Under this umbrella, a frenzy of efforts has been devoted to encouraging better accuracy and diversity trade-offs. A typical solution is to fully optimize the individual estimator while injecting randomness into the ensemble, *e.g.*, randomly manipulating the training data to provide each learner with a different subset of patterns or features [2], [3].

Le Zhang and Ce Zhu are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China.

Qibin Hou is with the School of Computer Science, Nankai University.

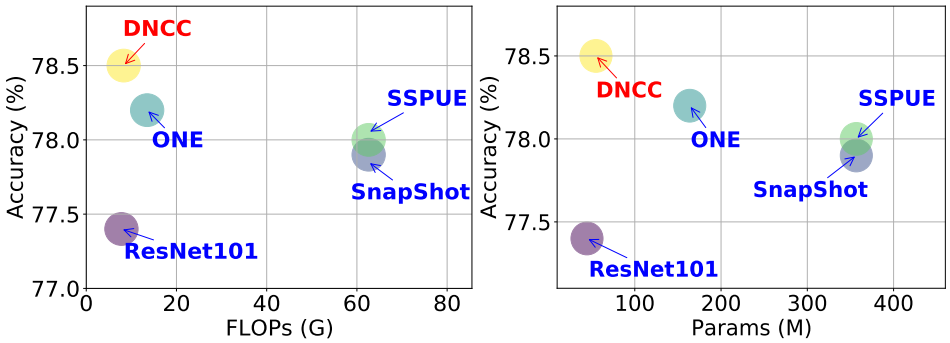
Jia-Wang Bian is with Department of Engineering Science, University of Oxford.

Yun Liu, Xun Xu and Joey Tianyi Zhou are with the Agency for Science, Technology and Research, Singapore.

However, the exact roles of accuracy and diversity in ensemble systems have not been well addressed although some theoretical analyses exist for the regression ensemble. Motivated by *bias-variance* decomposition, [10] showed that the mean squared error (MSE) for ensemble can be further decomposed into the *bias-variance-covariance* three-way trade-off. In this way, the optimal ensemble is said to be both "accurate" and "diversified". Based on the bias-variance-covariance, [11] first proposed the well-known negative-correlation learning from the evolutionary computation point of view by explicitly managing the accuracy and diversity of the individual estimator. This has motivated several works such as [12], [13], and more recent ones in the deep learning era [6], [14], [15].

As for ensemble classification, where the individual estimators output discrete class labels, the "diversity" is not yet well understood and remains an open research issue. Although [16] have made the first attempt by utilizing the one-hot coding on the category labels and training multiple models with the MSE under the negative correlation learning framework, this method is far from optimal. On one hand, the MSE is prone to outliers and is less robust than commonly used classification loss functions, *e.g.*, Softmax Cross-Entropy loss in the deep learning era. On the other hand, this method has a high computational complexity owing to the computation of the pseudo-inverse of a large matrix. Therefore, existing ensemble classification methods mainly utilize heuristic strategies to enhance diversities implicitly. Examples include using different diversity measures [17], randomly sampling data/feature subset [2], [3], utilizing different hyper-parameters [18], randomly dropping network activations/connections [19], [20] and aggregating different network outputs along the optimization path [21]. Bian *et al.* provide some theoretical insights on the diversity measurement in [22]. However, their work is still lacking because it only focus on binary classification in shallow learning scenarios.

From the optimization point of view, a better way for ensemble classification is to jointly optimize the accuracy as well as the diversity explicitly and this has been barely studied for deep classification ensembles. In this work, we tap into this gap and propose deep negative correlation classification (DNCC) that is backbone-independent and end-to-end trainable for optimizing the long-standing accuracy-diversity trade-offs for classification ensemble. The main idea of DNCC, as illustrated in Fig. 2, is a new loss function for ensemble deep classification, inspired by the "negative correlation learning" [11] which is commonly used in ensemble regression. More specifically, we seamlessly decompose the Softmax Cross-Entropy loss of ensemble deep networks into the individual loss of each network and their Bregman information, a quantity originally motivated by *the rate-distortion theory* and used to measure the correlation amongst



(a)Accuracy-FLOPs trade-off. (b) Accuracy-Params trade-off.

Fig. 1. Performance and network complexity trade-offs of different methods. We choose ResNet101 as the baseline network and test them with the input size of 224×224. Obviously, DNCC obtains better trade-off between the performance and complexity. Please note that some methods utilize multiple networks only in either the training (*i.e.*, ONE) or the inference stages (*i.e.*, Snapshot). For fair comparisons, here we report the maximum number of parameters/FLOPs in both stages.

the ensemble here, and then derive a simple and efficient method for deep ensemble classification. The loss function is readily pluggable into any network architecture and amenable to training via backpropagation. Finally, we show that our DNCC outperforms challenging baselines on multiple benchmark datasets and network structures including CNNs [23], [24], Transformers [25], and MLPs [26].

- We provide the definition of diversity in the deep classification ensembles by decomposing the commonly used soft-max crossentropy loss seamlessly into individual accuracy and the "correlations".
- Based on our framework, we show it is easy to optimize the accuracies and the the diversities of the base learners in an end-to-end manner. In this way, the proposed method naturally yields both " accurate" and "diversified" deep ensembles.
- We demonstrate the effectiveness of our approach on different datasets when taking different network backbones. We show the proposed method is able to show consistent improvement over existing ensemble methods with significantly less FLOPs.

The rest of this paper is organized as follows. The related work is summarized in Section II. And then, the definition of the diversity and the "accuracy-diversity" decomposition is presented in Section III. Finally, the empirical results are presented in Section IV, followed by the conclusion in Section V.

II. Related Work

A. Conventional Ensemble Classification

Representative conventional ensemble methods include bagging and boosting. They have been well studied in recent years and applied widely in different applications. [1] works by training multiple classifiers, which are formed by making bootstrap replicates of the learning set, using these as new learning sets, and then aggregating individual results. Due to the independence amongst the ensemble, each base model could be trained parallelly. As a special case of bagging, random forest [2] utilizes multiple decision trees as the base classifier and demonstrates its superiority in a wide range of applications. Boosting [27] works in a curriculum learning manner by first solving easy samples and progressively giving more focus to samples that are difficult to classify. Bian *et al.* [28] formulate ensemble pruning problem as an objection maximization problem based on information entropy. For more details on conventional ensemble classification, please refer to [29].

B. Ensemble Deep Classification

Although deep learning based methods have proven to surpass their shallow counterparts in various tasks, researchers have successfully shown that their performance could be further enhanced by ensemble learning. [19] introduced a dropout strategy to prevent the co-adaptation of feature learners, in which the key idea is to randomly drop units (along with their connections) from a network during training. It can be seen as an extreme case of bagging and each parameter of the network is very strongly regularized by sharing it with the corresponding parameter in all the other models [19]. The adaptive version of dropout is proposed in [30] where a binary belief network is overlaid on a network and is used to regularize its hidden units by selectively setting activities to zero. Motivated by dropout, [20] introduced DropConnect to regularize large fully-connected layers within neural networks. It sets a randomly selected subset of weights within the network to zero and thus each unit essentially receives input from a random subset of units in the previous layer. DropConnect could be regarded as a larger ensemble of deep neural networks than dropout [20]. In [31], multi-column structures are proposed where each column is actually a convolutional neural network (CNN) with different parameters, and outputs of all columns are averaged. The proposed method improves state-of-the-art performance on several benchmark datasets. Moreover, [32] proposed an ensemble of randomized deep networks by the way of entropy minimization strategy [33] and achieved improved results in visual tracking.

Stochastic multiple choice learning (sMCL) is proposed in [34] to train diverse deep ensembles, which follows a "winner-take-gradient" training strategy. Experimental results demonstrate the broad applicability and efficacy of sMCL for training diverse deep ensembles. An ensemble of deep CNN is

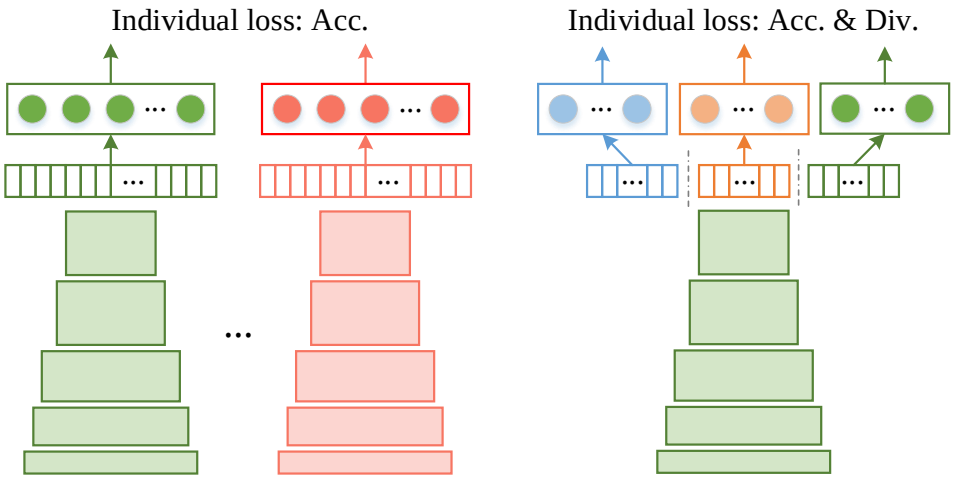


Fig. 2. Comparison between conventional ensemble methods and the proposed DNCC. Left: Conventional ensemble naïvely trains many different models and optimizes individual accuracies independently. Right: DNCC jointly trains an efficient deep ensemble (with negligible extra parameters) where the individual estimator is both accurate and diversified.

introduced in [5], where the individual results are aggregated by the KemenyYoung method [35]. Deep neural decision forest [36], [37] unifies random forest with the representation learning functionality from deep convolutional networks in an end-to-end manner. In [38], [39], the deep forest is proposed to generate a deep forest ensemble with a cascade structure that enables deep forest to do representation learning, and the number of cascade levels can be adaptively determined such that the model complexity can be automatically set. BatchEnsemble is established in [40] where each weight matrix is defined as the Hadamard product of a shared weight among all ensemble members. Besides, [41] showed how deep learning methods could be applied in the context of crowd-sourcing and unsupervised ensemble learning. Ensemble-based Decorrelation Method is introduced in [42] to regularize deep neural networks and avoid overfitting. Snapshot ensemble [21] trains a single neural network converging to several local minima along its optimization path and saves the model parameters. In [43], the authors introduced a simple and scalable predictive uncertainty estimation using Deep Ensembles. In the same way, other methods [44]–[46] approximate Bayesian inference for neural networks with Bayesian model averaging. A naive on-the-fly-ensemble is introduced in [47]. It works by training a single multi-branch network while simultaneously establishing a strong teacher on the fly by aggregating all the results to enhance the learning of the target network in a knowledge distilling strategy. Random subspace strategy is used in [48] for video classification. Chen *et al.* [49] design a new loss function to rectify the bias toward the majority classes for class-imbalance deep learning.

Different from existing methods which usually implicitly encourage diversity in the ensemble system, we explicitly decompose the ensemble Softmax Cross-Entropy loss into individual classification loss and the pairwise correlations between individual predictions and the ensemble outputs. This is beneficial in the sense that both accuracy and diversity are fully optimized by back-propagation. To summarize, we make the following contributions:

III. Methodology

Before elaborating on the proposed ensemble deep classification method, we first briefly present the notations and background knowledge. We assume that we have access to N training samples, *i.e.*, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Our objective is to predict their category labels, *i.e.*, $Y = \{y_1, \ldots, y_N\}$. We denote a generic data point/feature tensor by $\mathbf{x}$ and use $\mathbf{x}_\diamond$, with $\diamond$ denoting the place-holder for the index wherever necessary. y and $y_\diamond$ are similarly defined. Suppose we have $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y} = \{1, \cdots, K\}$, in which K is the number of classes.

We achieve our goal by learning a mapping function $G : \mathcal{X} \rightarrow \mathcal{Y}$. Then the learning problem is to use the set X to learn a mapping function G, parameterized by θ, to approximate their label Y as accurate as possible:

$$L(G) = -\frac{1}{N}\sum_{i=1}^{N} log(\frac{e^{f_{y_i}}}{\sum_{j=1}^{K} e^{f_j}}), \tag{1}$$

where f_j denotes the jth element ($j \in \{1, \cdots, K\}$) of the vector of class scores f. For simplicity, we will use L to represent $L(G)$ whenever the dependence of the loss function with the parameters can be easily inferred from the context. We consider the mapping function G to be an ensemble of deep networks, composed of M base classifier $\mathcal{G} = \{\mathcal{G}^m\}_{m=1}^{M}$, where the classifiers $\mathcal{G}^m : \mathcal{X} \rightarrow \mathcal{Y}, m \in \{1, \ldots, M\}$, called *based deep networks*, are combined using averaging:

$$\overline{L} = -\frac{1}{N}\sum_{i=1}^{N} log(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_{y_i}^m}}{\sum_{j=1}^{K} e^{f_j^m}}), \tag{2}$$

where f_j^m denotes the jth element ($j \in \{1, \cdots, K\}$) of the vector of the mth classifiers' class scores. Now we provide the definition of the Bregman divergence as follows:

Definition 1. (Bregman Divergence [50]). Let $\mathbf{g} : \mathcal{S} \rightarrow \mathbb{R}, \mathcal{S} = dom(\mathbf{g})$ be a strictly convex function defined on a convex set $\mathcal{S} \subseteq \mathbb{R}^d$ such that $\mathbf{g}$ is differentiable on $ri(\mathcal{S})$, assumed to be

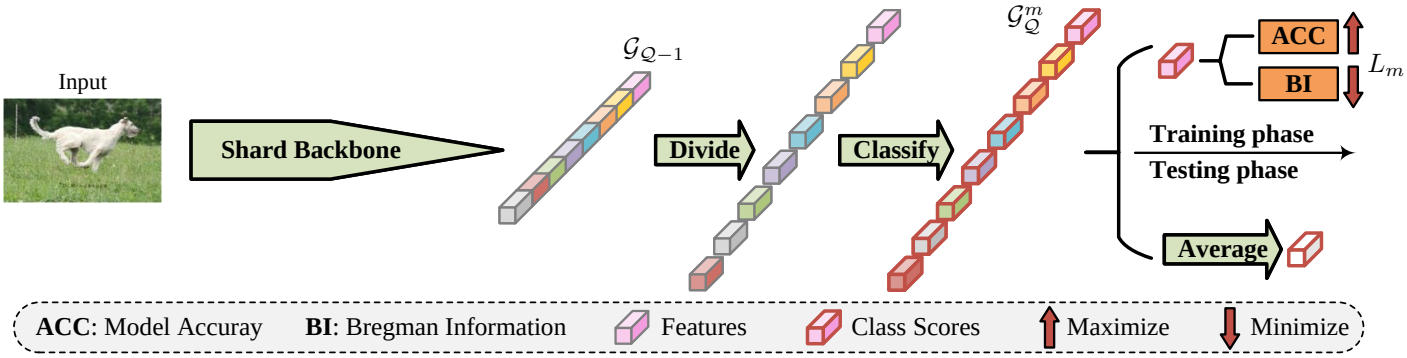


Fig. 3. DNCC consists of multiple networks with a shared backbone. In the penultimate layer, we divide the features into multiple non-overlapping subsets and train multiple classifiers accordingly. In the training phase, each classifier is forced to be both accurate and "negative correlated". In the testing phase, we simply use the average aggregation.

nonempty. The *Bregman divergence* $d_{\mathcal{G}} : \mathcal{S} \times ri(\mathcal{S}) \rightarrow [0, \infty)$ is defined as:

$$d_{\mathbf{g}}(t+\delta, t) = \mathbf{g}(t+\delta) - \mathbf{g}(t) - < \delta, \bigtriangledown \mathbf{g}(t) >, \tag{3}$$

where $\bigtriangledown \mathbf{g}(t)$ and $ri(\mathcal{S})$ represent the gradient vector of $\mathbf{g}$ evaluated at t and the relative interior of $\mathcal{S}$, respectively. $dom(\mathbf{g})$ represents the effective domain of $\mathbf{g}$, *i.e.*, set of all t such that $|\mathbf{g}(t)| < +\infty$ is denoted by $dom(\mathbf{g})$.

Definition 2. (Bregman Information [51]). Let T be a random variable that takes values in $\mathcal{T} = \{t_i\}_{i=1}^{N}$ following a probability measure ν. Let $\mu = E_{\nu}[T] = \sum_{i=1}^{N} \nu_i t_i \in ri(\mathcal{S})$ and $d_{\mathbf{g}} : \mathcal{S} \times ri(\mathcal{S}) \rightarrow [0, \infty)$ be the Bregman divergence.Then the *Bregman information* of T in terms of $d_{\mathbf{g}}$ is defined as:

$$I_{\mathbf{g}} = E_{\nu}[d_{\mathbf{g}}(T, \mu)] = \sum_{i=1}^{N} \nu_i d_{\mathbf{g}}(t_i, \mu). \tag{4}$$

With Definition 1 and 2, we have the following results:

Lemma 1. Given any convex function $\mathbf{g}$, for any random variable T, we have:

$$E[\mathbf{g}(T)] - \mathbf{g}(E[T]) = I_{\mathbf{g}}(T). \tag{5}$$

Proof.

$$\begin{aligned} & E[\mathbf{g}(T)] - \mathbf{g}(E[T]] \\ & = E[\mathbf{g}(T)] - \mathbf{g}(E[T]) - E[< T - E[T], \bigtriangledown \mathbf{g}(E[T]) >] \\ & = E[\mathbf{g}(t) - \mathbf{g}(E[T]) - < T - E[T], \bigtriangledown \mathbf{g}(E[T]) >] \\ & = E[d_{\mathbf{g}}(T, E[T])] \\ & = I_{\mathbf{g}}(T) \end{aligned} \tag{6}$$

With the above results, it is easy to obtain the following results by considering the convexity of the *-log* function.

Proposition 1. For an ensemble of network, its Softmax Cross-entropy Loss $\overline{L}$, as defined in Eq. (2), can be decomposed into the average loss of its base network and the Bregman Information:

$$\overline{L} = \frac{1}{M} \sum_{m=1}^{M} L_m - I_{-log}. \tag{7}$$

Proposition 1 explains the effect of error correlations in an ensemble system by stating that the Softmax Cross-entropy Loss of the ensemble network is guaranteed to be less than or equal to the average loss of the base networks. Existing ensemble classification methods mainly work by training multiple models independently. This may not be optimal because, as demonstrated in Proposition 1, the ensemble loss consists of both the individual loss and the non-negative Bregman information of the inputs. Based on this, we propose to learn a correlation-regularized ensemble system with the following objective:

$$\begin{aligned} L_m = & -\frac{1}{N} \sum_{i=1}^{N} [log(\frac{e^{f_{y_i}^m}}{\sum_{j=1}^{K} e^{f_j^m}}) \\ & - \lambda * d_{-log}(\frac{e^{f_{y_i}^m}}{\sum_{j=1}^{K} e^{f_j^m}}, \frac{1}{M} \sum_{m=1}^{M} \frac{e^{f_{y_i}^m}}{\sum_{j=1}^{K} e^{f_j^m}})], \end{aligned} \tag{8}$$

where d_{-log} could be obtained by setting $\mathbf{g}$ as $-log$ in Definition 1. Eq. (8) can be regarded as a smoothed version of Eq. (7) to improve the generalization ability of the ensemble models. The parameter λ here controls the ensemble's accuracy and diversity and note that its optimal value may not necessarily be 1 because of the discrepancy between the training and testing data [6], [9], [15].

More specifically, we consider the function $\mathcal{G}$ as an ensemble of networks as defined in Eq. (2) where each base network is posed as:

$$\begin{aligned} \mathcal{G}^m(\mathbf{x}_i) = \mathcal{G}_{\mathcal{Q}}^m(G_{\mathcal{Q}-1}^m \cdots (\mathcal{G}_1^m(\mathbf{x}_i))), \\ m = 1, 2 \cdots M, \quad i = 1, 2 \cdots N, \end{aligned} \tag{9}$$

where m, i, and $\mathcal{Q}$ stand for the index for individual networks, the index for data samples and the depth of the network, respectively. More specifically, each predictor in the ensemble consists of cascades of feature extractors $\mathcal{G}_q^m$, $q = 1, 2 \cdots \mathcal{Q}-1$ and classifier $\mathcal{G}_{\mathcal{Q}}^m$. As the diversity in the system are enhanced by regularizing the Bregman Information, we are able to use a shared network backbone for high efficiency. Formally, the lower levels of feature extractors are shared by each predictor, *i.e.*, $\mathcal{G}_q^m = \mathcal{G}_q$, $q = 1, 2 \cdots, \mathcal{Q}-1$, $m = 1, 2 \cdots, M$. Based on that, we further divide the outputs of the highest level feature extractor $\mathcal{G}_{\mathcal{Q}-1}$ to different subsets, each of which is used as input for different classifier $\mathcal{G}_{\mathcal{Q}}^m$. This has been shown to

TABLE I
COMPARISON OF DNCC WITH OTHER BASELINE METHODS ON THE CIFAR DATASET.

Method	CIFAR10	CIFAR100
Res50 [23]	94.80	77.39
Res50-Snapshot [21]	94.78	78.47
Res50-ONE [47]	94.89	78.56
Res50-SSPUE [43]	94.82	78.39
Res50-DNCC	**95.05**	**79.05**
Res101 [23]	94.98	77.78
Res101-Snapshot [21]	95.30	78.52
Res101-ONE [47]	95.41	78.61
Res101-SSPUE [43]	95.28	78.55
Res101-DNCC	**95.53**	**78.82**

be effective in generating an ensemble system without extra computational overhead than a standard single network [6]. Apparently, this strategy is significantly more efficient than previous efforts in naively fusing multiple different deep networks [52], [53]. An overview of the proposed method can be found in Fig. 3 and illustrations on the accuracy and complexity trade-off are presented in Fig. 1.

A. Discussion

Advocating both accuracy and diversities amongst individual models has been shown to be effective in the ensemble learning community. [2] derived the upper bound for ensemble's generation error by the way of both the "strength" and the "correlation" for base models. In addition, the Probably Approximately Correct (PAC) learning theory [54] shows that a good learner should be both accurate and with low hypothesis space complexity. In the proposed method, the complexity has been reduced in two ways. First, by using a shared network backbone, the proposed method is efficient and the complexity has been reduced significantly [6]. Second, the existing method [55] has shown that enhancing the ensemble diversity could also lead to a small hypothesis space complexity. The proposed loss function, as shown in Eq. (8), penalizes the pairwise correlation amongst the ensemble by reducing the Bregman distance between the individual networks and the ensemble results, and is thus beneficial in reducing the hypothesis space complexity.

IV. EXPERIMENTS

To demonstrate the feasibility of DNCC, we evaluate it on several benchmark datasets, including CIFAR10 [56], CIFAR100 [56], and ImageNet [57]. For CIFAR10 and CIFAR100, we employ the well-established residual networks (ResNet) [23] and use ResNet50 and ResNet100. We split the features of the last global average pooling into 8 non-overlapping subsets with equal dimensionality and train 8 classifiers accordingly. For CIFAR10, the batch size is set to 128 and we train the network for 150 epochs. The initial learning rate is 0.1 which is decreased by a factor of 0.1 for every 50 epochs. For CIFAR100, we train the network for 200 epochs and we decrease the learning rate by a factor of 0.1 at epochs 60, 120, and 160 with an initial learning rate of 0.1. As for the ImageNet dataset, apart from ResNet50 and ResNet101, we also evaluate DenseNet121 [24]. Moreover, we also add the recently proposed Transformer and MLP networks in the comparisons to further understand the merits of the proposed methods. For Transformer and MLP architectures, we choose the Swin-Transformer [25] and the Vision Permutator [26]. For all the networks trained on the ImageNet dataset, we firstly expand the features in the penultimate layer to $8\times$ of its original dimension, then split the resulting features into 8 non-overlapping subsets. Finally, we train 8 classifiers accordingly. For each network, we follow the original training protocol. For CNNs, we adaptively control the value of λ by setting it to be $\frac{epoch}{\#epoch} \times 10^{-2}$, in which $epoch$ and $\#epoch$ stand for the number of current epoch and the number of the maximum epoch, respectively. For Swin-Transformers and Vision Permutator, λ is simply set to be $5e-4$. We run all experiments with PyTorch [58].

A. Main Results

We compared our proposed method with the baseline network (*i.e.*Res50, Res101) and several state-of-the-art ensemble learning strategies. Firstly, we consider the Snapshot Ensemble [21]. It trains a single neural network and saves the model parameters when converging to several local minima along the optimization path. We also consider the On-the-Fly Native Ensemble (ONE) [47]. ONE trains multi-branch networks and distills the knowledge from the ensemble results to each branch on the fly. The other method we compare is the SSPUE (Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles) which is a combination of ensembles and adversarial training [43].

CIFAR10. The CIFAR10 dataset [56] consists of natural color images sized at 32×32 pixels. It has 50,000 training images and 10,000 testing images from 10 classes. We use a standard data augmentation scheme [59], in which the images are zero-padded with 4 pixels on each side and then randomly cropped to generate 32×32 images. Besides, we also horizontally flip the inputs with a probability of 0.5. We use ResNet50 and ResNet101 [23]. The experimental results are summarized in Tab. I. From the experimental results, we can see that with the same network backbone, the proposed DNCC method yields the best results for both ResNet50 and ResNet101 and surpasses state-of-the-art ensemble learning strategies such as Snapshot Ensemble [21], ONE [47] and SSPUE [43].

CIFAR100. The CIFAR100 dataset [56] has the same statistics as CIFAR10 except that the images are sampled from 100 classes. We use the same data augmentation technique as done for CIFAR10. In the same way, we use ResNet50 and ResNet101 [23]. Results for Snapshot Ensemble, ONE, and SSPUE with the same architecture are also provided. Results are presented in Tab. I. Conclusions from the CIFAR10 dataset are also applicable here.

ImageNet. The ILSVRC 2012 classification dataset [57] consists of 1000 images classes, with a total of 1.2 million training

TABLE II
COMPARISON OF DNCC WITH OTHER BASELINE METHODS ON THE IMAGENET DATASET.

Network	Accuracy	Params	FLOPs
Res50 [23]	76.1	**25.6**M	**4.1G**
Res50-Snapshot	76.4	204.8M	32.8G
Res50-ONE	76.6	144.7	9.8G
Res50-SSPUE	76.3	204.8M	32.8G
Res50-DNCC	**76.8**	36.0M	4.5G
Res101 [23]	77.4	**44.6**M	**7.8G**
Res101-Snapshot	77.9	356.8M	62.6G
Res101-ONE	78.2	163.7M	13.5G
Res101-SSPUE	78.0	356.8M	62.6G
Res101-DNCC	**78.5**	55.0M	8.3G
Dense121 [24]	74.4	**8.0M**	**2.9G**
Dense121-Snapshot	74.7	63.8M	23.0G
Dense121-ONE	75.0	26.8M	4.4G
Dense121-SSPUE	74.8	63.8M	23.0G
Dense121-DNCC	**75.2**	23.6M	**2.9G**

images and 50,000 validation images. We adopt the same data augmentation scheme as in [23] and apply a 224×224 center crop to images for testing. For this dataset, we use various architectures including CNN, Transformer, and MLP. As for CNN,we use ResNet50, ResNet101, and DenseNet121. Results for Snapshot Ensemble, ONE, and SSPUE with the same architecture for each network are also provided Results are summarized in Tab. II. In order to further understand the merits of DNCC, we also report the number of parameters and FLOPs (with input size of 224×224) of each method. Please note that some methods, such as ONE, use multiple auxiliary heads in the training phase and prune them in the inference phase. In addition, other methods, such as the Snapshot Ensemble, save multiple models and use all of them in the inference phase. For more fair comparisons, we report the maximum number of parameters and FLOPs for each method in both training and testing stages. It is also straightforward to see that DNCC yield better trade-offs between the performances and network complexities.

In Tab. III, we also evaluate DNCC on more advanced non-CNN architectures. We choose the Swin-Transformer [25] and the Vision Permutator (ViP) [26] as the representative work of the transformer and MLP respectively. As the Swin-Transformer and the Vision Permutator typically need more training epochs (*i.e.*, 300) to converge, we did not compare other ensemble methods because they typically need more FLOPs/Parameters and thus significantly slow down the training process. We can show that the proposed DNCC improves different baselines.

B. Trade-off between Accuracy and Diversity

In order to further understand the merits of DNCC, we shed light upon the trade-off between accuracy and diversity in this section. In [60], the authors showed that enhancing diversity could in principle lead to a small hypothesis space complexity which is essential in improving the generalization ability of the learning system. Here we compare the accuracy

TABLE III
EFFECT OF DNCC ON POPULAR TRANSFORMER AND MLP-LIKE MODELS ON THE IMAGENET DATASET.

Network	Accuracy	Params	FLOPs
Swin-T [25]	81.2	**28.3**M	**4.5G**
Swin-T-DNCC	**81.5**	38.4M	**4.5G**
ViP-Small/7 [26]	81.5	**25.1M**	**6.9G**
ViP-Small/7-DNCC	**81.9**	29.4M	**6.9G**

and diversities of DNCC with the conventional ensemble. In this study, we compare Res101-DNCC and Res101-Ensemble, which is trained by setting $\lambda = 0$, and train them on the CIFAR10 dataset. We split the training data into two non-overlapping subsets with a ratio of 4:1 and use them to train and validate the methods, respectively. Following the previous section, the ensemble size is set to 8. Motivated by [3], we compare the pairwise accuracy and diversity of both methods. For the ith and j^{th} classifier in the ensemble, the accuracy is measured by the mean accuracy of both classifiers, and the diversity is defined as:

$$Diversity(i,j) = \frac{1}{K}\sum_{k=1}^{K}(1 - \frac{W_k^i * W_k^j}{||W_k^i|| * ||W_k^j||}), \qquad (10)$$

where W_k^i is the weights of the ith classifier for the kth class. Eq. (10) is essentially the average angle between the corresponding decision hyper-planes for the two classifiers [60].

In this example, as there are 8 individual classifiers in the ensemble, there exist $C_8^2 = 28$ pairs and we visualize the diversity and accuracy for DNCC and the conventional ensemble in Fig. 4. More specifically, in Fig. 4(a) and Fig. 4(b), we visualize the diversity and accuracy trade-off of both methods. The x-axis and y-axis stand for the accuracy and diversity improvement (positive value in the y-axis indicates that DNCC has better pairwise accuracy and diversity over the conventional ensemble, respectively) for each pair, respectively.

For the diversity part, DNCC shows clear superiority over the conventional ensemble baseline, as expected. Fig. 4(a) compares the accuracy improvement of DNCC over the conventional ensemble. The x-axis stands for the index of each pair. The y-axis is the accuracy difference between DNCC and the conventional ensemble. In this case, a positive value indicates that DNCC achieves better pairwise mean accuracy than the conventional ensemble. It is interesting that in most cases, the pairwise mean accuracy of DNCC is better than the conventional ensemble. Two reasons could lead to this improvement: i) The penalty term introduced in Eq. (8) could work as a strong regularizer for the individual models in the ensemble and thus reduce its over-fitting. ii) Advocating the diversity of DNCC in our setting could also lead to more diversified feature representation in the bottom network backbone, which essentially prevents their output units from co-adapting. Similarly, Fig. 4(b) shows the diversity difference in the same manner. Obviously, DNCC demonstrates a clear advantage in the diversity part over the conventional ensemble. Finally, we observe that the DNCC and the conventional ensemble achieve

94.26% and 93.97% accuracy, respectively. However, for the individual classifier in both methods, the accuracy is mostly within $60\%-90\%$, which again demonstrates the effectiveness of ensemble methods in improving the final performance.

C. Trade-off between Efficiency and Accuracy

Previous ensemble methods [52], [53] fuse outputs from M different models in the inference phase, and hence requiring M times of computational overhead than a standard single model. In contrast, the proposed method is able to have the same inference time in our design [6]. The speed of a single model using different backbones can be found in [61].

Although we mainly advocate an efficient solution of "train 1 and get N for free", in practice, one could simply further enhance the classification accuracy by having the base network branch at an earlier stage. To demonstrated this, we have done extra experiments on the CIFAR100 dataset with the ResNet50 backbone. In particular, we grew different network branches after the Res2X, Res3X, Res4X blocks. We achieved the accuracy of 80.04%, 79.70%, and 79.05%, respectively. In addition, the accuracy of 80.26% was obtained in a naïve setting in which 8 independent ResNet50 were used. Those results are also provided in Tab. IV. The results verify that inserting more diversities in the ensemble, by having different randomly initialized branches at the early stage of the network, could lead to better ensemble performances.

TABLE IV
ACCURACY-EFFICIENCY TRADE-OFF OF DNCC ON CIFAR100 WITH A RESNET50 BACKBONE.

Split Position	Shared Params	Accuracy
Input	0%	**80.26**%
Res2X	6%	80.04%
Res3X	36%	79.70%
Res4X	91%	79.05%

D. Ablation Studies

In this section, we provide some ablation studies to further understand the merits of the proposed DNCC. In particular, we study the effect of the ensemble size N and the regularization parameter λ in the following sections.

Effect of the Ensemble Size: The existing ensemble learning theory [2] shows that it will bring no harm in terms of the final accuracy to increase the ensemble size. However, this improvement does not come with no cost: increasing the ensemble size will dramatically increase the computational complexity as well. Therefore, in practice, one may need to control the ensemble size to achieve a better trade-off between performance and computational resources.

In order to investigate the role of the ensemble size in DNCC, we conduct a set of experiments on the CIFAR100 dataset. We use a tight version of ResNet18 [23] which we call *tight-ResNet18* for this experiment. More specifically, for each residual block of ResNet18, we reduce the number of output channels by a factor of 8 to save the training time. We train M different networks with the same training protocol with different weight initialization. All the networks are trained for 300 epochs with an initial learning rate of 0.1 and a batch size of 128. We decrease the learning rate by a factor of 0.1 at epochs 60, 120, and 160, respectively. λ in Eq. (8) is set to $1e$-4. Fig. 5(a) displays the performance of *tight-ResNet18* ensemble as the effective ensemble size, M, is varied. It can be seen that both methods yield better performance than the single model in most cases and most importantly, DNCC performs better than the conventional ensemble method when we have $N > 4$, which demonstrates the effectiveness of the proposed DNCC.

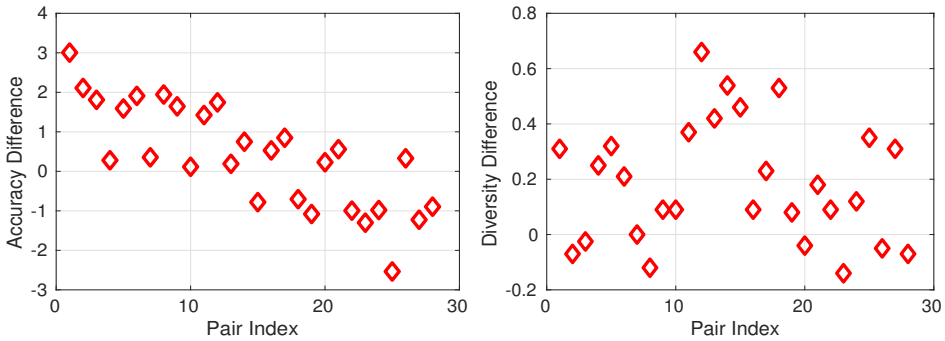
TABLE V
COMPARISON OF DNCC WITH THE CONVENTIONAL ENSEMBLE BY SETTING $\lambda = 0$.

Method	CIFAR10	CIFAR100	ImageNet
Res50 [23]	94.80	77.39	76.14
Res50-Ensemble	94.85	78.42	76.51
Res50-DNCC	**95.05**	**79.05**	**76.82**
Res101 [23]	94.98	77.78	77.37
Res101-Ensemble	95.36	78.48	78.12
Res101-DNCC	**95.53**	**78.82**	**78.45**

Effective of λ: The parameter λ controls the correlation among base models in the ensemble system. On the one hand, setting $\lambda = 0$ is equivalent to training each classifier in an independent manner. On the other hand, a larger value of λ could yield a less-correlated ensemble with high diversities. However, we also observe that a larger value of λ could also have a negative effect on the accuracy of individual models, which could lead to worse final accuracy. This is because a larger value of λ will affect the network optimization towards maximizing accuracy by weakening the effect of the first term in Eq. (8). In order to study the effect of λ, we conduct ablation experiments with different values of λ using the previous settings. More specifically, we train 8 *tight-ResNet18* models with the parameter λ in $\{0, 1e\text{-}8, 1e\text{-}6, 1e\text{-}4, 1e\text{-}2, 1\}$ and report the results in Fig. 5(b). As can be observed, setting λ to a small value of $1e$-4 could yield the best result. In addition, we also report the results of different backbones on different datasets when setting $\lambda = 0$ in Tab. V. Results clearly show that the proposed DNCC outperforms the conventional ensemble by setting $\lambda = 0$. Hence, it is beneficial to manage the accuracy-diversity amongst the ensemble.

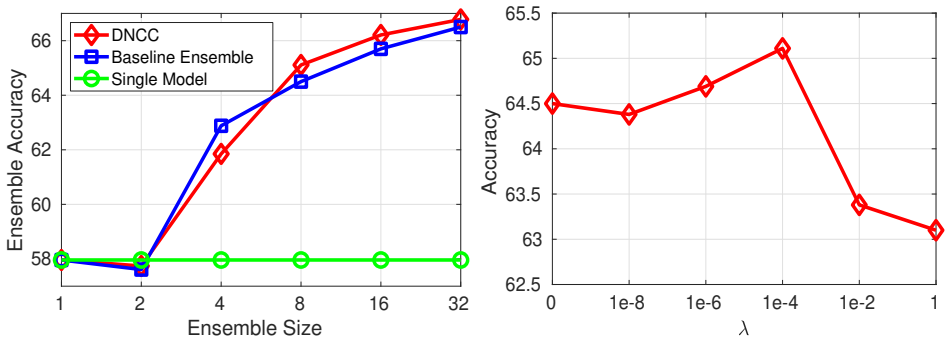
V. CONCLUSION

In this paper, we have presented the deep negative correlation classification (DNCC) algorithm to learn efficient and structure-independent deep network ensembles by involving Bregman information. Our analysis presents a new view of ensemble Softmax Cross-Entropy loss by decomposing it into individual accuracy and diversity between individual prediction and the ensemble output. Apart from the high efficiency, our proposed method is also advantageous when compared with



(a) Accuracy improvement. (b) Diversity improvement.

Fig. 4. Diversity and accuracy comparison between DNCC and the conventional ensemble on the CIFAR10 dataset. The accuracy is calculated by the mean accuracy of each pair and the diversity is defined in Eq. (10). For figures (a) and (b), a positive value in the y-axis indicates that DNCC has better pairwise accuracy and diversity over the conventional ensemble, respectively. The experimental results indicate that, for each pair of classifiers, DNCC achieves competitive accuracy yet much better diversities over the conventional ensemble.



(a) Effect of the ensemble size. (b) Effect of the parameter λ.

Fig. 5. Effect of the ensemble size N and the parameter λ on the CIFAR100 dataset. The figure (a) shows that both DNCC and the conventional ensemble perform better when a larger ensemble size is utilized. Furthermore, with a larger ensemble size (≥ 4), DNCC outperforms the conventional ensemble. The figure (b) also indicates that setting λ to a small value of $1e$-4 could yield improved results.

existing ensemble methods by jointly optimizing the accuracy and diversities through back-propagation. Extensive experiments have shown the feasibility of the proposed method with multiple network structures on multiple benchmark datasets. Our work would be valuable in developing new accurate yet efficient deep ensemble learning algorithms. One limitation of our work is, for more complicated classification problems such as ImageNet, it may need an extra, but still affordable, amount of parameters.

REFERENCES

[1] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[2] ——, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.

[4] G. Valentini and T. G. Dietterich, "Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods," *Journal of Machine Learning Research*, vol. 5, no. Jul, pp. 725–775, 2004.

[5] L. Zhang, Z. Shi, J. T. Zhou, M.-M. Cheng, Y. Liu, J.-W. Bian, Z. Zeng, and C. Shen, "Ordered or orderless: A revisit for video based person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[6] L. Zhang, Z. Shi, M.-M. Cheng, Y. Liu, J.-W. Bian, J. T. Zhou, G. Zheng, and Z. Zeng, "Nonlinear regression via deep negative correlation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[7] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[8] T. G. Dietterich, "Ensemble methods in machine learning," in *International Workshop on Multiple Classifier Systems*. Springer, 2000, pp. 1–15.

[9] G. Brown, J. L. Wyatt, and P. Tiňo, "Managing diversity in regression ensembles," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1621–1650, 2005.

[10] N. Ueda and R. Nakano, "Generalization error of ensemble estimators," in *International Conference on Neural Networks*, vol. 1. IEEE, 1996, pp. 90–95.

[11] Y. Liu, X. Yao, and T. Higuchi, "Evolutionary ensembles with negative correlation learning," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 380–387, 2000.

[12] C. Perales-González, F. Fernández-Navarro, M. Carbonero-Ruz, and J. Pérez-Rodríguez, "Global negative correlation learning: A unified framework for global optimization of ensemble models," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[13] H. Chen, B. Jiang, and X. Yao, "Semisupervised negative correlation learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5366–5379, 2018.

[14] Z. Shi, L. Zhang, Y. Liu, X. Cao, Y. Ye, M.-M. Cheng, and G. Zheng, "Crowd counting with deep negative correlation learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5382–5390.

[15] W. Zhao, B. Zheng, Q. Lin, and H. Lu, "Enhancing diversity of defocus blur detectors via cross-ensemble network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8905–8913.

[16] M. Alhamdoosh and D. Wang, "Fast decorrelated neural network ensembles with random weights," *Information Sciences*, vol. 264, pp. 104–117, 2014.

[17] E. K. Tang, P. N. Suganthan, and X. Yao, "An analysis of diversity measures," *Machine Learning*, vol. 65, no. 1, pp. 247–271, 2006.

[18] L. Zhang and P. N. Suganthan, "Benchmarking ensemble classifiers with

novel co-trained kernel ridge regression and random vector functional link ensembles [research frontier],” *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 61–72, 2017.

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

[20] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, “Regularization of neural networks using dropconnect,” in *International Conference on Machine Learning*, 2013, pp. 1058–1066.

[21] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, “Snapshot ensembles: Train 1, get M for free,” in *International Conference on Learning Representations*, 2017.

[22] Y. Bian and H. Chen, “When does diversity help generalization in classification ensembles?” *IEEE Transactions on Cybernetics*, 2021.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical vision transformer using shifted windows,” in *International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[26] Q. Hou, Z. Jiang, L. Yuan, M.-M. Cheng, S. Yan, and J. Feng, “Vision permutator: A permutable MLP-like architecture for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[27] Y. Freund, R. E. Schapire *et al.*, “Experiments with a new boosting algorithm,” in *International Conference on Machine Learning*, vol. 96. Citeseer, 1996, pp. 148–156.

[28] Y. Bian, Y. Wang, Y. Yao, and H. Chen, “Ensemble pruning based on objection maximization with a general distributed framework,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3766–3774, 2019.

[29] Y. Ren, L. Zhang, and P. N. Suganthan, “Ensemble classification and regression-recent developments, applications and future directions,” *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 41–53, 2016.

[30] J. Ba and B. Frey, “Adaptive dropout for training deep neural networks,” in *Advances in Neural Information Processing Systems*, 2013, pp. 3084–3092.

[31] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, “Multi-column deep neural network for traffic sign classification,” *Neural Networks*, vol. 32, pp. 333–338, 2012.

[32] L. Zhang and P. N. Suganthan, “Visual tracking with convolutional random vector functional link network,” *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3243–3253, 2016.

[33] Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in *Advances in Neural Information Processing Systems*, 2005, pp. 529–536.

[34] S. Lee, S. P. S. Prakash, M. Cogswell, V. Ranjan, D. Crandall, and D. Batra, “Stochastic multiple choice learning for training diverse deep ensembles,” in *Advances in Neural Information Processing Systems*, 2016, pp. 2119–2127.

[35] J. Levin and B. Nalebuff, “An introduction to vote-counting schemes,” *Journal of Economic Perspectives*, vol. 9, no. 1, pp. 3–26, 1995.

[36] P. Kontschieder, M. Fiterau, A. Criminisi, and S. Rota Bulo, “Deep neural decision forests,” in *IEEE International Conference on Computer Vision*, 2015, pp. 1467–1475.

[37] S. Rota Bulo and P. Kontschieder, “Neural decision forests for semantic image labelling,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 81–88.

[38] Z. Zhou and J. Feng, “Deep forest: Towards an alternative to deep neural networks,” *arXiv preprint arXiv:1702.08835*, 2017.

[39] M. Pang, K.-M. Ting, P. Zhao, and Z.-H. Zhou, “Improving deep forest by confidence screening,” in *IEEE International Conference on Data Mining*. IEEE, 2018, pp. 1194–1199.

[40] Y. Wen, D. Tran, and J. Ba, “BatchEnsemble: Efficient ensemble of deep neural networks via rank-1 perturbation,” in *Advances in Neural Information Processing Systems Workshop*, 2019.

[41] U. Shaham, X. Cheng, O. Dror, A. Jaffe, B. Nadler, J. Chang, and Y. Kluger, “A deep learning approach to unsupervised ensemble learning,” in *International Conference on Machine Learning*, 2016, pp. 30–39.

[42] S. Gu, Y. Hou, L. Zhang, and Y. Zhang, “Regularizing deep neural networks with an ensemble-based decorrelation method.” in *International Joint Conference on Artificial Intelligence*, 2018, pp. 2177–2183.

[43] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[44] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *International Conference on Machine Learning*, 2016, pp. 1050–1059.

[45] H. Ritter, A. Botev, and D. Barber, “A scalable laplace approximation for neural networks,” in *International Conference on Learning Representations*, vol. 6, 2018.

[46] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” *arXiv preprint arXiv:1505.05424*, 2015.

[47] X. Zhu, S. Gong *et al.*, “Knowledge distillation by on-the-fly native ensemble,” in *Advances in Neural Information Processing Systems*, 2018, pp. 7517–7527.

[48] J. Zheng, X. Cao, B. Zhang, X. Zhen, and X. Su, “Deep ensemble machine for video classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 553–565, 2018.

[49] Z. Chen, J. Duan, L. Kang, and G. Qiu, “Class-imbalanced deep learning via a class-balanced ensemble,” *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[50] L. M. Bregman, “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming,” *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217, 1967.

[51] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, “Clustering with bregman divergences,” *Journal of Machine Learning Research*, vol. 6, no. Oct, pp. 1705–1749, 2005.

[52] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[53] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[54] L. G. Valiant, “A theory of the learnable,” *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[55] Y. Yu, Y.-F. Li, and Z.-H. Zhou, “Diversity regularized machine,” in *International Joint Conference on Artificial Intelligence*, 2011.

[56] A. Krizhevsky, G. Hinton *et al.*, *Learning multiple layers of features from tiny images*. Citeseer, 2009.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[59] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[60] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, “Diversity techniques improve the performance of the best imbalance learning ensembles,” *Information Sciences*, vol. 325, pp. 98–117, 2015.

[61] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, 2019, pp. 6105–6114.