Collaborative Deep Networks for Pedestrian Detection

Hongmeng Song, Wenmin Wang, Jinzhuo Wang, Ronggang Wang
School of Electronic and Computer Engineering
Shenzhen Graduate School, Peking University
Lishui Road 2199, Nanshan District, Shenzhen, China 518055
songhongmeng4edu@pku.edu.cn, wangwm@ece.pku.edu.cn, cr7or9@163.com, rgwang@pkusz.edu.cn

***Abstract*—Conventional pedestrian detection methods construct models based on hand-crafted features or deep learning. They are powerful but limited due to finite capabilities of single classifiers. Ensemble models escape these problems by assembling multiple classifiers using some man-made criteria which synthetically utilize information from all combined models. However, these criteria lack theoretical support. Therefore, we propose a novel ensemble deep model called collaborative deep networks where multiple deep networks are meaningfully combined in a fully-connected network. For maximizing the abilities of these deep models, we incorporate a resampling process to prepare diverse datasets and pre-train them using these resampled data. Finally, a collaborative learning method is presented to train the entire model. Experimental results prove that our approach can improve the performance of single classifiers and outperform state-of-the-art methods both on Caltech dataset and ETH dataset.**

***Keywords*-Pedestrian Detection; Collaborative Deep Networks; Fully-Connected Network; Resampling Process; Collaborative Learning;**

I. Introduction

As one of the most important components of object detection, pedestrian detection has attracted a great deal of attention in recent years [1], [2]. It is still challenging because of the variations of humans and confusion caused by people's diverse poses, lighting, views, and occlusion.

Traditional methods [3], [4], [5], [6], [7], [8], [9] tend to build models based on hand-crafted features. They extract features that can capture discriminative information of pedestrians such as HOG [3], ICF [4], and ACF [8] to train SVMs [3] or boosting classifiers [7]. Abundant experiments prove that they are able to perfectly describe human body and recognize pedestrians even in complex environments. However, intrinsic limitations, like the finite ability of man-made descriptors to describe the essence of humans, still exist. Nowadays, deep models are prevailing because they are proved to be capable of conquering these limitations and largely improving the accuracy of detection [10], [11], [12], [13], [14], [15], [16]. They can not only directly learn more potential feature representations from raw data, but more importantly, provide an elastic framework inside or outside which multiple models can be incorporated.

These powerful models have made a great success in pedestrian detection but still have a lot of room to improve. Recently, many multi-classifier models [17], [18], [19], [20], [21], [22] are proposed to expand limited learning capabilities of single classifiers by embedding multiple classifiers into one model, which further improve the performance. Differed from cascade structures in general boosting algorithms that use multiple classifiers by stages [17], ensemble models emphasize parallel execution and synthetical analysis [18], [19], [20], [21], [22]. In this way, natural mistakes of one classifier can be remedied by other classifiers and more useful information can be utilized for comprehensive discrimination. Constructing ensemble models especially ensemble deep networks receives much concern mainly in sequence learning [23] and image classification [20], [21], which inspires us to focus on ensemble deep models in pedestrian detection.

Inspired by competitive working in human society, Generative Adversarial Nets (GANs) [24] design a new framework for image generation via an adversarial process where a generation model and a discriminative model are simultaneously trained like a minimax two-player game. However, cooperation normally helps humans to do more than the competition does. Therefore, as an alternative, we are devoted to designing an ensemble deep model where multiple deep models can learn from cooperating with each other.

To the best of our knowledge, we have not found such an accomplished work on ensemble deep models that build a collaborative neural network for any tasks in computer vision. Therefore, the motivation of this paper is to establish a novel ensemble model called collaborative deep networks (CDNs) where we train different deep networks to learn different features and embed them into a fully-connected network to cooperatively detect pedestrians. For training our model, a collaborative learning method is proposed which optimizes each deep model from the cooperation. We hope that our model can compensate for the weakness of single model and obtain a well-learned collaborative model with fewer errors.

In this paper, the following three contributions are made.

1) A novel ensemble model named collaborative deep networks is proposed for detecting pedestrians. In this framework, the final detection scores are produced by a fully-

This project was supported by Shenzhen Peacock Plan (20130408-183003656).

978-1-5090-6549-3/17 $31.00 © 2017 IEEE
DOI 10.1109/BigMM.2017.17

connected network which assembles the outputs of multiple deep models. In such a cooperation way, the strengths of different deep models can be jointly utilized.

2) We present a new collaborative learning method for training our model. With the help of collaborative learning, multiple deep networks can learn to cooperate better with each other to detect pedestrians and meanwhile reduce its shortages by learning from others' successful experiences.

3) A resampling process of datasets is incorporated when training multiple identical deep networks. In this way, each network can be pre-trained to recognize images with specific features, which maximizes its strengths.

The remainder of this paper is as follows: Section II describes recent related work on ensemble models. Section III elaborates the architecture and training process of our collaborative deep networks. The experimental results and evaluation of different designs of our model are shown in Section IV and V respectively. At last, in Section VI, the conclusions and future work are stated.

II. Related work

Part-based models are the first trial of ensemble methods which have achieved significant success in pedestrian detection especially when heavy deformation and occlusion happened [5], [6], [10], [14], [25], [26]. They extract local features of humans and build multiple part-oriented detectors to obtain the scores of different parts in human bodies which are then combined to train a strong classifier for pedestrian detection. For example, Deformable Part Models (DPMs) are proposed in [6] to discriminatively learn HOG features for each body part using several latent SVMs. To solve the problem on how to integrate the inaccurate scores of part detectors in DPMs, Ouyang and Wang [25] present a discriminative deep model for learning the visibility relationship among overlapping parts at multiple layers. By paying more attention to specific part information, these part-based models can extract more useful features and further combine them for more precise recognition. However, they are still confined by the finite learning capability of single-classifier models which causes some inevitable errors like easily confusing on similar samples with different labels.

Differed from part-based models, ensemble models assemble multiple independent classifiers which can successfully reduce the inevitable mistakes mainly in sequence learning [18], [19], [23] and image classification [19], [20], [21], [27]. For example, a novel design strategy for neural network macro-architecture based on self-similarity is introduced as FractalNet in [27]. In this model, multiple deep convolutional neural networks are designed with interacted subpaths of different lengths and every internal signal is transformed by a filter and nonlinearity before being seen by subsequent layers. Finally, these networks are combined in a softmax prediction layer. However, FractalNet can not assemble different kinds of deep networks, which decreases its extensibility. As for detecting pedestrians, we only find that an ensemble inference network is presented [22] which discriminate humans by the maximum, minimum, or mean of outputs of several Convolutional Neural Networks (CNNs) with different dropout designs. However, these ensemble criteria are too arbitrary which is easy to decrease the generalization ability of models. We follow this framework to design our ensemble deep networks but combine the outputs by a fully-connected network rather than some man-made rules.

GANs [24] and their variants [28] are a special type of ensemble deep models that construct an ensemble of a generation model and a discriminative model, and provide an adversarial training method. In this framework, the generation model aims at capturing data distribution, while the discriminative model is designed for estimating the probability that a sample comes from training data rather than generated images. For improving the performance of image generation, the goal of training generation model is minimizing the accuracy of the discriminative model. With such an adversarial learning method, both models can be optimized to achieve a Nash Equilibrium. Experiments prove that this kind of competitive training method generates better images than independently training generation model. As an alternative way to train ensemble deep models, we propose a collaborative learning method to train our model which is more proper for our parallel architecture. Differed from setting opposite goals of training these two models in GANs, our method establishes the same goal for all members which is minimizing detection errors of the entire model. As results, multiple pedestrian detection models can achieve better cooperative classification and simultaneously optimize themselves using the information learned in cooperation.

III. Collaborative deep networks

The Fig.1 illustrates the architecture of our collaborative deep networks. A set of region proposals are taken as input and their detection scores are output. Our method consists of three stages. Firstly, a resampling process based on K-means clustering algorithm is effectuated to prepare diverse training datasets. Then we utilize these resampled datasets and original dataset to pre-train multiple deep networks as collaborative members in our model to recognize different types of samples. Finally, a collaborative learning method is taken to train our entire model. We test these well-learned deep models by the entire training dataset and assemble their outputs to train a fully-connected network. Meanwhile, in back propagation, component models are fine-tuned by learning from cooperating with each other. More details will be elaborated in following sections.

A. Resampling training datasets

For maximizing the differences between multiple deep networks to ensure the effectiveness of collaboration, we

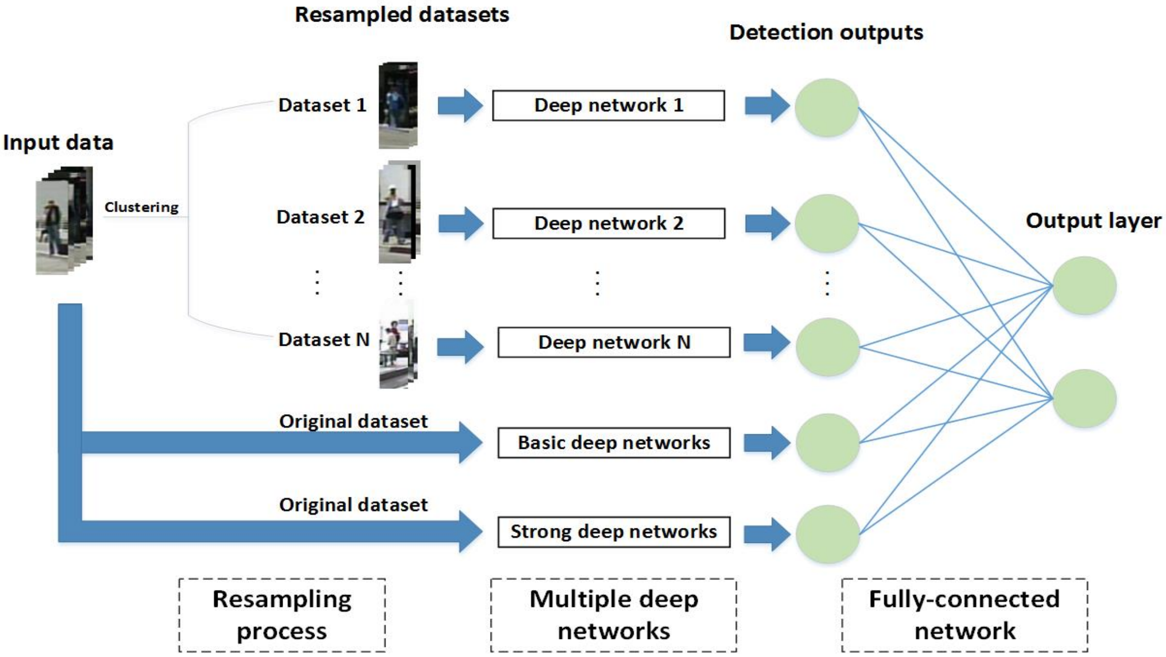

Figure 1. The framework of our collaborative deep networks. A set of region proposals are taken as input and their detection scores are output. At the first step, a resampling process based on K-means clustering algorithm is effectuated to prepare diverse training datasets. Secondly, we pre-train multiple deep networks to recognize different types of samples using these resampled datasets and original dataset. Finally, a collaborative learning method is taken to train our entire model.

need to prepare diverse training datasets at the first step. Each dataset is composed of randomly-sampled images which are similar so that these deep networks' capability of distinguishing confused samples can be enhanced. Experimental results prove that K-means algorithm is an excellent and simple way to collect similar samples randomly with the most intra-class similarities and the least inter-class ones. So we choose K-means algorithm to automatically separate a large number of data into K clusters. The detailed procedure can be described as following.

Step 1: Extract the features $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ for each sample in original training dataset $\boldsymbol{D}$.

Step 2: Select k images randomly as cluster centroids. Denote them by $\{\boldsymbol{c}_j\}_{j=1}^{k}$.

Step 3: Compute the distances between feature images and the centroids using Eq.(1).

$$d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{c}_j\|_2 \tag{1}$$

Step 4: Divide the images into corresponding clusters by seeking the minimal distances as

$$\boldsymbol{C}_j = \{i | \forall m \neq j, d_{im} > d_{ij}\} \tag{2}$$

where $1 \leqslant m \leqslant k$.

Step 5: Update $\{\boldsymbol{c}_j\}_{j=1}^{k}$ by computing the mean of images within each cluster as

$$\boldsymbol{c}_j = \frac{1}{|\boldsymbol{C}_j|} \sum_{i \in \boldsymbol{C}_j} \boldsymbol{x}_i \tag{3}$$

Step 6: Return to Step 3 and repeat it until the results of clustering are constant.

In our model, we extract a three-channel feature in [10] for clustering after the input images are converted into YUV color space. The first channel is a Y-channel image, while the three-channel images in the YUV color space are concatenated into the second channel with zero padding. In the third channel, four edge maps are concatenated where three of them are obtained from the three-channel images in the YUV color space by the Sobel edge detector and the fourth is computed by choosing the maximum magnitudes from the first three edge maps. Then considering that all samples are labeled as two classes: pedestrian and background, we separate original datasets into two clusters based on above algorithm for each channel, which eventually produces totally seven datasets for pre-training multiple deep networks. Designs of different cluster numbers and feature types for clustering will be discussed in Section V-A.

B. Pre-training multiple deep network

Due to increasing depth of deep models with fully-connected layers, they can easily be trained to fall into local optimum which decreases their capabilities of extracting useful features. To solve this problem and increase the generalization ability of the deep networks in our model, we adopt pre-training to initialize the parameters for further collaborative learning. In our collaborative deep networks, different models are pre-trained in different methods. As results, each deep network can detect pedestrians independently.

We divide the component deep networks that we choose in our model into three types. One is basic deep networks such as LeNet-5 [29], fully-connected deep neural networks, and so on. They are good but still weak to detect pedestrians which result in lots of inevitable errors. The second one includes some strong variants like Unified Deep Network (UDN) [10] which can be incorporated to enhance the performance. We pre-train both of them by original training datasets. In the last type, we construct multiple initially-identical classifiers which belong to the former two types. Then they are pre-trained by different sub-datasets that are obtained in resampling process to learn various features. All of the three deep networks are pre-trained using their own training methods. With parallel training, the computational complexity can be largely reduced.

In our experiments, we choose one UDN model for the second type of deep networks which is trained by all samples and multiple UDN models for the third type which are trained by resampled datasets as our running example to testify if our model can improve the performance of single UDN model. Other designs on different component deep networks will be discussed in Section V-A. The UDN is naturally an 8-layers CNN model containing three convolutional layers, one pooling layer following the first convolutional layer, and four full-connected layers. By designing the third convolutional layer for deformation handling and replacing the full-connected layers with a visibility reasoning network, feature extraction, deformable handling, occlusion handling, and classification are well combined in one model. More details can be referred in [10]. As results, multiple well-learned deep networks are prepared.

C. Collaborative learning method

At last, a novel collaborative learning method is proposed to train our collaborative deep networks. As an alternative to the adversarial learning process in GANs where a model learned by competing with the other model, our collaborative learning method makes multiple deep models learn from cooperating with each other to achieve the same goal. Therefore, we construct a fully-connected network which assembles the outputs of these pre-trained deep networks as input. Dropout operation is also exerted to randomly select which paths information transmits through. In this way, reliable cooperation and connection between these collaborative members can be built. For accelerating the speed of convergence, we use cross entropy to construct loss function in our collaborative learning method which is shown in Eq.4.

$$\boldsymbol{L} = -(\boldsymbol{y}\ln(\tilde{\boldsymbol{y}}) + (1-\boldsymbol{y})\ln(1-\tilde{\boldsymbol{y}})) \tag{4}$$

where $\tilde{\boldsymbol{y}}$ is the estimated detection scores.

Similar to back propagation algorithm, our collaborative learning method can be divided into two stages: collaborative process and fine-tuning process. In the first process, the original training dataset is input into all trained deep networks which produces a set of initial detection scores for each sample. Then we assemble all sets of the initial scores to obtain final scores in the last layer of the fully-connected network, which mimics the collaborative decision-making process. We denote the feature vectors of input samples as $\boldsymbol{x}$. The collaborative process can be expressed in Eq.5.

$$\begin{aligned} \tilde{\boldsymbol{z}}_{\boldsymbol{i}} &= \mathscr{F}_{\boldsymbol{i}}(\boldsymbol{x}), i = 1, 2, ..., N \\ \tilde{\boldsymbol{y}} &= \boldsymbol{\sigma}(\tilde{\boldsymbol{z}}\boldsymbol{W} + \boldsymbol{b}) \end{aligned} \tag{5}$$

where $\mathscr{F}_i$ represents all operations in the i-th deep network which produces the initial score $\tilde{\boldsymbol{z}}_i$, N denotes the number of deep networks, $\sigma(t) = (1 + \exp(-t))^{-1}$ is the sigmoid function, $\tilde{\boldsymbol{z}}$ is the ensemble of $\tilde{\boldsymbol{z}}_i$, $\boldsymbol{W}$ denotes the weights between the input nodes and output nodes, and $\boldsymbol{b}$ is bias term for the output nodes.

For the fine-tuning process, we directly use back propagation algorithm to modify the parameters both in the fully-connected network and those component deep networks. To handle error propagation between different kinds of networks, we assume that the need for each deep model to learn is the propagated error of the input layer in the fully-connected network. Then the errors are assigned into each input node and further propagate them into those deep networks for fine-tuning. Eventually, these component models can learn to cooperate with each other for achieving better pedestrian detection.

D. Implementation details

We implement our collaborative deep networks on general CPU[1]. In order to save computational resources, we directly use candidate windows as input data which are pruned from sliding windows by HOG+CSS and Linear SVM at both training and testing stages [10]. Besides, the localization process is saved by recording the coordinates of these windows in pruning process which helps us focus more attention on improving the performance of classifiers. For enhancing the capability of deep networks, the positive samples are inverted to boost training data.

We pre-train component deep networks and collaboratively train our entire model both using mini-batch stochastic gradient descent to accelerate the training speed. This method inputs a batch of samples which are randomly-sampled rather than one sample at each iteration. In addition, a batch normalization process for input data is incorporated to improve the performance. In order to find the best parameters, both training processes are reiterated for epoches. For pre-training, we fix the batch size as 71, variable learning rate as 0.025, and the number of epoches as 5. In

[1] Code is available at https://github.com/SHM1992/CDN_New

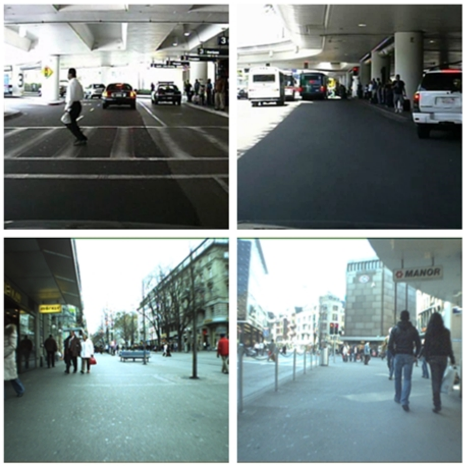

Figure 2. Example images of Caltech (up) and ETH dataset (down).

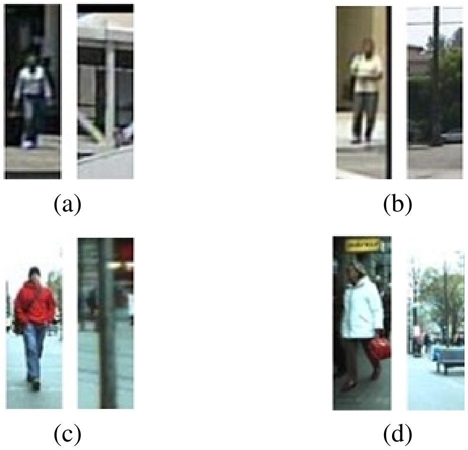

Figure 3. Example images (left) of CaltechTrain (a), CaltechTest (b), INRIA (c), and ETH (d) dataset. Each sub-figure contains a positive image (left) and a negative image (right).

collaborative learning, the counterparts remain unchanged, while the learning rate of the fully-connected network is different which is 0.1 because of its fewer layers.

IV. Experiments

A. Experimental settings

Datasets. Our collaborative deep networks is evaluated on the Caltech dataset and the ETH dataset [1] which are the most popular datasets in pedestrian detection. Both datasets contain two parts: training dataset and testing dataset which are collected by low-resolution (640*480) cameras. So the pedestrians in the images are too small to be recognized for traditional approaches. Fig. 2 shows some example images from these datasets. As we can see, some pedestrians that are far from the cameras are even difficult to be recognized by human eyes.

In the experiments on Caltech dataset, we use Caltech-Train dataset for training and Caltech-Test dataset for testing. As for ETH dataset, we follow the commonly-used methods [10] to use the INRIATrain dataset to train our model and test on ETH dataset. After the pruning process and training data augmentation, detailed data distribution of these datasets is shown in Table I. Some examples are given in Fig. 3.

Evaluation methods. In pedestrian detection, there are two basic metrics for evaluating the performance of the classifier. One is the miss rate which means the proportion of pedestrian samples that are detected as background samples in all testing images. The other is the false positive rate which shows the proportion of background samples that are detected as pedestrian samples. They can evaluate different capabilities of a classifier.

To gain a balance between both metrics, we adopt the most authoritative evaluation method provided by Dollár et al. [1] to measure the performance of our collaborative deep networks. This approach focuses only on classification process and evaluates methods directly on outputs of classifiers by the log-average miss rate. As in [1], the log-average miss rate is computed by averaging nine miss rates corresponding to nine False Positive Per Image (FPPI) rates that are evenly spaced in the log-space in the range from 10^{-2} to 10^{0}. Each miss rate and FPPI rate are calculated simultaneously after every 30 samples being detected. To avoid the influence of some extremely small pedestrians, we evaluate on ground-truth datasets where the pedestrians are larger than 49 pixels height and have 65% visible body parts in our experiments.

B. Overall experimental results

In our experiments, we utilize the UDN models as the components in our collaborative model. Therefore, the UDN is taken as the baseline model. Other compared methods are HOG [3], HOGLBP [5], DPM [6], Discriminative Deep Model [25], ICF [4], LeNet-5 [29], and ACF [8]. We denote our collaborative deep networks by CDN.

As shown in Table II, our approach improves the performance of baseline by 2.05% and 1.36% on the Caltech-Test dataset and ETH dataset respectively which outperform other compared methods. These outstanding results are attributed to the collaboration of multiple specially trained deep networks. Firstly, resampling the training data according to different features strengthens the capability of component deep networks on learning more specific features. Secondly, the pre-training process helps each deep network to avoid getting trouble in converging to local optimum which enhances the ability of feature representation. Finally, the collaborative learning method further improves the performance of any single classifier by training the fully-connected network to gather them to cooperate with each other and fine-tuning by learning from this cooperation. In this way, the intrinsic errors of each deep network can be significantly reduced.

However, it is worthy emphasizing that our framework is more general. Our purpose of proposing this novel model

Table I
DATA DISTRIBUTION OF THE EXPERIMENTAL DATASETS

Dataset	Training		Testing
	Positive	Negative	
Caltech	4396	60072	10935
ETH	2404	56966	15587

Table II
OVERALL RESULTS OF COMPARISON WITH OTHER METHODS

Methods	Log-Average Miss Rate	
	Caltech-Test	ETH
HOG	68%	64%
HOGLBP	68%	55%
DPM	63%	51%
Discriminative Deep Model	61%	47%
ICF	48%	50%
LeNet-5	46%	-
ACF	43%	50%
UDN	42.54%	45.47%
CDN	**40.49%**	**44.11%**

Table III
TRAINING TIME COMPARISON OF CDN AND UDN ON CALTECH DATASET

Methods		Total time cost
UDN		16 hours
CDN	**Resampling**	0.5 hour
	Pre-training UDNs	16 hours
	Collaborative learning	20 hours

Table IV
RESULTS OF DIFFERENT RESAMPLING METHODS ON CALTECH-TEST

Resampling Method Designs	Log-Average Miss Rate
CDN-Random sampling	42.56%
CDN	40.49%

Table V
RESULTS OF DIFFERENT FEATURES FOR CLUSTERING ON CALTECH-TEST

Clustering Feature Designs	Log-Average Miss Rate
CDN-RGB-based	41.70%
CDN-ACF-based	40.51%
CDN	40.49%

is to improve the performance of single classifier by co-operating with other classifiers. We will demonstrate it on other types in Section V-B. Therefore, it is meaningless to compare our model which uses UDNs as the component deep networks with some state-of-the-art methods.

C. Evaluation on training time

Our collaborative deep networks inevitably increases the training time of any single model even if we count the time cost of the pre-training process out due to the incorporation with fully-connected network. So, balancing the accuracy and time complexity is mainly considered in our experiments. By using K-means algorithm, we can achieve fast resampling process. For saving redundant time of pre-training multiple deep networks, we incorporate parallel computing to train them simultaneously. Moreover, in the collaborative learning process, we set a higher learning rate to accelerate the speed of convergence.

We compute time cost of training CDN on Caltech-Train dataset and compare with UDN in Table III. From this table, we can see that the time of training CDN costs totally 36.5 hours including resampling, pre-training, and collaborative learning which is 20 hours more than training UDN. If we use the well-trained deep networks directly, the total training time can be reduced to 20.5 hours which is only 4.5 hours slightly more than the UDN's. So it is tolerant to sacrifice a little time to improve the performance of single models.

V. DESIGN EVALUATION

We conduct more experiments to understand how our collaborative deep networks improves the performance of single classifier as well as to evaluate design decisions. Since the Caltech dataset is the largest among commonly used dataset, we perform these experiments on this dataset and discuss how to design the best architecture of our model in following sub-sections.

A. Which design of resampling process is the best?

In our model, a resampling process is proposed to prepare diverse training datasets to maximize the differences of multiple deep networks. It is indispensable because entirely identical deep networks can not learn from each other. We incorporate K-means algorithm to automatically divide original training data due to its convenience in collecting images with maximal similarities within a cluster and maximal differences between different clusters. Other methods like randomly selecting several datasets with same amounts of images can also help the resampling. We compare K-means algorithm and this random sampling method in Table IV. The results show that K-means is faster and more effective.

We also compare different kinds of features used in K-means clustering in Table V. Clustering datasets based on the first three channels of ACF [8] achieves nearly the same result as the three-channel features [10], which both outperform the RGB-based clustering. The results show that good features can improve the performance of CDN.

Table VI gives the results of different numbers of clusters that are resampled in K-means algorithm. The CDN-6,6,6 means that we collect 6 clusters on each channel in the three-channel features for the resampling process, while CDN-5,5,2 represents that we cluster 5 sets on the first and the second channel but only 2 sets on the third channel. It is clear in Table VI that when we resample 2 sets on each channel,

Table VI
RESULTS OF DIFFERENT CLUSTERING NUMBERS ON CALTECH-TEST

Clustering Number Designs	Log-Average Miss Rate
CDN-6,6,6	42.40%
CDN-5,5,2	42.01%
CDN	40.49%

Table VII
RESULTS OF DIFFERENT COMPONENTS ON CALTECH-TEST

Component Designs	Log-Average Miss Rate
LeNet-5	46%
UDN	42.54%
CDN-UDN+LeNet-5	42.07%
CDN-UDNs	41.08%
CDN	40.49%

Table VIII
RESULTS OF DIFFERENT ENSEMBLE METHODS ON CALTECH-TEST

Ensemble Method Designs	Log-Average Miss Rate
CDN-3-layer Neural Network	51.43%
CDN-Max	42.71%
CDN-Mean	41.88%
CDN-ANN	41.60%
CDN	40.49%

the log-average miss rate can be largely decreased. The main reason may be that the insufficient samples caused by too many clusters weaken the capabilities of deep networks.

B. Is our collaborative learning method suitable with other deep networks?

Our collaborative deep networks is a general model which can assemble different deep models and construct different pipeline architectures for collaboration. It is obvious that different choices of these deep networks can largely influence the effect of pedestrian detection. As we illustrate in Section III-B, the deep networks that can be combined in our model consist of three types. To save computational resources and speed up the pre-training process, we select LeNet-5 [29] for the first type, UDN model [10] for the second type, and 6 UDN models that are pre-trained with resampled datasets for the third type. Then, we perform experiments on three kinds of combinations which are the ensemble of UDN and LeNet-5, the ensemble of 6 UDN models, and the ensemble of UDN plus these 6 UDN models. Table VII shows the results of them on Caltech-Test dataset. As we can see, all combinations perform better than any single classifier which verifies the effectiveness of our model. Furthermore, we find that whether the ensemble of single UDN and CNN or ensemble of multiple UDNs trained by different datasets performs worse than our collaborative model. This comparison clearly demonstrates that assembling stronger classifier can boost more performance.

C. Does our collaborative learning method help?

One important contribution of our collaborative deep networks is the collaborative learning method. To initialize, a fully-connected network is built which assembles the outputs of multiple deep networks to achieve collaborative pedestrian detection. It is a novel criterion for traditional ensemble deep models using some man-made criteria. Then we implement a collaborative learning method which trains the fully-connected network in feed-forward propagation process and meanwhile finetunes component deep networks by learning from the collaboration in back propagation process. To prove that our collaborative learning method is better than other ensemble approaches, extra experiments on different ensemble deep models are performed.

Table VIII gives the comparison results. CDN-3-layer Neural Network is CDN model which adds a hidden layer in the fully-connected network. CDN-Max/CDN-Min represents the traditional ensemble deep model using maximal/minimal outputs as final detection scores. CDN-ANN is also proposed by us as a simpler version of CDN which only trains the fully-connected network without fine-tuning. It is proved that our model performs better than all compared ensemble models. The probable reasons for outperforming CDN-Max and CDN-Min are that man-made criteria are too arbitrary and extreme. Additionally, CDN-ANN is the second best model because it indeed offsets some detecting errors in deep networks by combining them in a fully-connected network. However, this model can not learn to correct these mistakes. Therefore, our collaborative learning method is necessary. The third conclusion is that deeper architecture of the fully-connected network can largely decrease the collaborative effect maybe due to the poor feature representation capability of fully-connected deep networks.

VI. CONCLUSIONS

This paper proposes a novel ensemble deep model named as collaborative deep networks for pedestrian detection. In our method, we embed multiple deep networks into a fully-connected network and design a collaborative learning method to train the entire model. To boost the capabilities of these deep networks, a resampling process is incorporated to produce diverse training datasets and component deep networks are pre-trained to gain the ability of feature representation. With our collaborative deep networks, the performance of UDN model can be improved by 2.05% on largest Caltech dataset and 1.36% on ETH dataset which both outperform relevant approaches. Extra experiments on evaluating different designs demonstrate that our model is well designed and can improve the performance of any single classifier.

In the future, we will expect larger improvement by further optimizing the architecture and learning method of our collaborative deep networks, and apply the collaborative learning method into more tasks in computer vision.

REFERENCES

[1] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *TPAMI*, vol. 34, no. 4, pp. 743–761, Apr 2012.

[2] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection," *arXiv preprint arXiv:1602.01237v1*, Feb 2016.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[4] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *BMVC*. BMVA, 2009, pp. 1–11.

[5] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *ICCV*. IEEE, 2009, pp. 32–39.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, Sep 2010.

[7] C. Cosma, R. Brehar, and S. Nedevschi, "Pedestrians detection using a cascade of lbp and hog classifiers," in *ICCP*. IEEE, 2013, pp. 69–75.

[8] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *TPAMI*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.

[9] S. Zhang, R. Benenson, and B. Shiele, "Filtered channel features for pedestrian detection," in *CVPR*. IEEE, 2015, pp. 1751–1760.

[10] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *ICCV*. IEEE, 2013, pp. 2056–2063.

[11] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *CVPR*. IEEE, 2014, pp. 899–906.

[12] W. Ke, Y. Zhang, P. Wei, Q. Ye, and J. Jiao, "Pedestrian detection via pca filters based convolutional channel features," in *ICASSP*. IEEE, 2015, pp. 1394–1398.

[13] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," in *CVPR*. IEEE, 2015, pp. 5079–5087.

[14] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *CVPR*. IEEE, 2015, pp. 437–446.

[15] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *ICCV*. IEEE, 2015, pp. 3361–3369.

[16] J. Wang, W. Wang, X. Chen, R. Wang, and W. Gao, "Deep alternative neural network: Exploring contexts as early as possible for action recognition," in *NIPS*, 2016, pp. 811–819.

[17] W. Guo, Y. Xiao, and G. Zhang, "Multi-scale pedestrian detection by use of adaboost learning algorithm," in *ICVRV*. IEEE, 2014, pp. 266–271.

[18] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *CIEL*. IEEE, 2014, pp. 1–6.

[19] W. Huang, H. Hong, K. Bian, X. Zhou, G. Song, and K. Xie, "Improving deep neural network ensembles using reconstruction error," in *IJCNN*. IEEE, 2015, pp. 1–7.

[20] D. Dai and L. V. Gool, "Ensemble projection for semi-supervised image classification," in *ICCV*. IEEE, 2013, pp. 4321–4328.

[21] C. V. Krishna Veni and T. Sobha Rani, "Ensemble based classification using small training sets: A novel approach," in *CIEL*. IEEE, 2014, pp. 1–8.

[22] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase, "Pedestrian detection based on deep convolutional neural network with ensemble inference network," in *IV*. IEEE, 2015, pp. 223–228.

[23] A. Celikyilmaz and D. Hakkani-Tur, "Investigation of ensemble models for sequence learning," in *ICASSP*. IEEE, 2015, pp. 5381–5385.

[24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, and D. Warde-Farley, "Generative adversarial nets," in *NIPS*. NIPS, 2014, pp. 2672–2680.

[25] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *CVPR*. IEEE, 2012, pp. 3258–3265.

[26] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *ICCV*. IEEE, 2015, pp. 1904–1912.

[27] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648v2*, Nov 2016.

[28] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *arXiv preprint arXiv:1606.03657v1*, Jun 2016.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of IEEE*, 1998, pp. 2278–2324.