Crowd Counting with Deep Negative Correlation Learning

Zenglin Shi[1], Le Zhang[2*], Yun Liu[3], Xiaofeng Cao[4],
Yangdong Ye[5], Ming-Ming Cheng[3] and Guoyan Zheng[1]
[1] *University of Bern, Switzerland*; [2] *ADSC,UIUC, Singapore*; [3] *Nankai University, China*
[4] *University of Technology Sydney, Australia*; [5] *Zhengzhou University, China*

Abstract

Deep convolutional networks (ConvNets) have achieved unprecedented performances on many computer vision tasks. However, their adaptations to crowd counting on single images are still in their infancy and suffer from severe over-fitting. Here we propose a new learning strategy to produce generalizable features by way of deep negative correlation learning (NCL). More specifically, we deeply learn a pool of decorrelated regressors with sound generalization capabilities through managing their intrinsic diversities. Our proposed method, named decorrelated ConvNet (D-ConvNet), is end-to-end-trainable and independent of the backbone fully-convolutional network architectures. Extensive experiments on very deep VGGNet as well as our customized network structure indicate the superiority of D-ConvNet when compared with several state-of-the-art methods. Our implementation will be released at https://github.com/shizenglin/Deep-NCL

1. Introduction

Crowd counting is an active research topic in computer vision due to its wide-ranging applications in video surveillance, metropolis security, human behavior analysis and resource management. Since the pioneering work for automated counting of object [1], several hand-crafted features [2, 3] and learning methods [4, 5] have consistently improved the counting performance, leading the research community to address more challenging scenarios and complex datasets [4, 5]. However, significant hurdles due to occlusions, scale variations and diverse crowd distributions make it difficult to solve this problem.

Detecting people individually [6] seems to be a straightforward solution but suffers from severe defects: detectors are prone to fail when people are in close proximity and there is no way to recover. Furthermore, high computational complexities in detection based approaches also limit their applicability for real-time applications. Counting by regression, on the other hand, learns to predict pedestrians' number through a regression function with some visual descriptors such as texture features, edge features [7] or learned representations [5, 4]. Counting by regression is perceived as state-of-the-art at present. The regression-based methods have been widely studied and reported to be computationally feasible with modern hardware, robust with parameters and accurate across various challenging scenarios.

Recently, the number of success stories of computer vision has seen an all-time rise across a wide range of topics such as image recognition [8], object detection [9], face recognition [10], image segmentation [11] and visual tracking [12]. The common idea behind these solutions is to use deep convolutional networks with many hidden layers, aiming at learning discriminative feature embedding from raw data, rather than relying on handcrafted feature extraction. Inspired by this, several ConvNets based crowd models have been proposed. [5, 13] make the fist attempt, to the best of our knowledge, to employ ConvNets for crowd counting. While tremendous progress has been achieved later by aggressively exploring deeper [14] or wider architectures [4] or heuristic engineering tricks [15, 13] with the standard "convolution + pooling" recipe, in this paper, we contribute from a different view. We are the first to provide an alternative to the commonly used learning objective with better generalization abilities by ensemble learning. Our method is end-to-end-trainable and independent of the backbone fully-convolutional network architectures. At the core of our approach is the adoption of *Negative Correlation Learning (NCL)* that has been shown, both theoretically and empirically, to work well for regressions based problems. NCL controls the *bias-variance-covariance* trade-off systematically and usually results in a regression ensemble where each base model is both "accurate and diversified". Simplicity and efficiency are central to our design, and trivially applying NCL to train multiple ConvNets is not the focus of this paper as it leads to significantly higher computational complexity. Instead, we adopt a "divide and conquer"

*This work was done during Zenglin Shi's internship at ADSC. Corresponding author: Le Zhang (zhang.le@adsc-create.edu.sg)

approach to learn a pool of regressors to regress the crowd density map on top of convolutional feature maps. Each regressor is jointly optimized with ConvNets by an amended cost function which penalizes correlations with others to make better trade-offs among the *bias-variance-covariance* in the ensemble. We call our method D-ConvNet where "D" means decorrelated. D-ConvNet based crowd counting framework has following advantages:

- D-ConvNet produces multiple counting results with a single ConvNet with no extra learning parameters. It has sound generalization capability through managing diversities among each prediction to have a better "bias-variance-covariance" trade-off.
- D-ConvNet is simple and does not rely on foreground segmentation results. It is end-to-end-trainable and does not require complicated training process like [4].
- D-ConvNet is generic and independent of the backbone fully-convolutional network architectures. Extensive experiments of very deep VGG as well as our customized network structure on several challenging crowd counting datasets such as UCF_CC_50, ShanghaiTech, and WorldExpo10 indicate the superiority of D-ConvNet when compared with several state-of-the-art methods.

2. Related Work

Regression with ConvNets has made vast inroads into crowd counting due to their commendable performances. It is widely accepted that ConvNets trained in an end-to-end manner deliver strikingly better generalization ability than shallow learning approaches with carefully engineered representations. A deep ConvNet [5] was trained alternatively with two related learning objectives, crowd density and crowd count. However, it relied heavily on a switchable learning approach and was not clear how these two objective functions can alternatively assist each other. [13] proposed to directly regress the total people number by adopting AlexNet [16] which has been later shown to be worse than methods regressing density map. This observation suggests that reasoning with rich spatial layout information from convolutional feature maps is necessary. [14] proposed a framework consisting of both deep and shallow network for crowd counting. It was reported to be more robust with perspectives and scale variations. Similarly, [4] proposed a multi-column ConvNets architecture. However, it needed careful pre-training of each base model followed by overall finetuning. "Hydra CNN" [15] was proposed to estimate object densities in different crowded scenarios in a scale-aware manner. It relied on a pyramid of image patches extracted at multiple scales and the performance on a single scale was not promising. Switching ConvNet was introduced in [17] where patches from a grid within a crowd scene were relayed to independent ConvNet regressors based on crowd count prediction quality of the ConvNet established during training.

Almost all the aforementioned approaches work well for their adopted ConvNet structures. Generating them to a much deeper network structure like [18] to further boost the discriminative ability of the learned representations for crowd counting is not straight-forward due to limited training data. Introducing large-scale datasets for crowd counting may partially alleviate the problem. However, manual labeling is costly, time-consuming and error-prone. It can also raise privacy concerns. It is often impractical as there may exist several thousands of people within a single image in dense crowd scenarios. This motivates us to study the problem of training deep ConvNets on existing crowd counting datasets with less risk of over-fitting. To address this, we draw inspirations from NCL [19, 20] and extend it to deep learning. The proposed method is readily pluggable into any ConvNets architecture and amenable to end-to-end training. With no extra learning parameter, it learns an ensemble of accurate and diversified regressors for crowd counting whose prediction errors may cancel out each other.

3. D-ConvNet

3.1. Background and Motivation

Before elaborating the proposed crowd counting method, we first briefly present the notations that we use and the background knowledge to put our D-ConvNet in a proper context. We assume that we have access to N training samples, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. The samples are M_i dimensional: $\mathbf{x}_i \in \mathcal{X} \subseteq I\!R^{M_i}, i \in \{1, \ldots, N\}, M_i = H_i \times W_i \times C_i$, where H_i, W_i and C_i denote the height, width and number of channels of i^{th} input image, respectively. Our objective is to predict the number of people in $\mathbf{x}_i$. To this end, we learn to regress a density map $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ where each $\mathbf{y}_i$ in $\mathcal{M}_i$ dimensional space. $\mathcal{M}_i = \mathcal{H}_i \times \mathcal{W}_i$. Detailed procedure of generating $\mathbf{Y}$ will be discussed in Section 3.3. In our implementation, $\frac{\mathcal{H}_i}{H_i} = \frac{\mathcal{W}_i}{W_i} = \mathcal{S}$ where $\sum \mathbf{y}_i = \mathbf{t}_i \in \mathbb{Z}$ stands for the number of people in an input image. We denote a generic data point by $\mathbf{x}$ and use $\mathbf{x}_\diamond$, with $\diamond$ denoting the placeholder for the index where ever necessary. Similarly, we use M and $\mathcal{M}$ to represent the dimensionality of a generic input data and its label, respectively. We achieve counting by learning a mapping function $G : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} \in [0, 1]^{\mathcal{M}}$.

In a typical regression ensemble, dimensionality of input data are considered to be the same, *i.e.*, $M_i = M$. In the same way, $\mathcal{M}_i = \mathcal{M}$. 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:

$$e(G) = \int (G(X,\theta) - Y)^2 p(X,Y) d(X,Y), \tag{1}$$

In practice, as data distribution $p(X,Y)$ is unknown, Eq. (1) is usually approximated by

$$e(G) = \frac{1}{N}\sum_{i=1}^{N}(G(\mathbf{x}_i,\theta) - y_i)^2, \tag{2}$$

we omit the input and parameter vectors, so where it is unambiguous, instead of $G(X,\theta)$, we write simply G. we use the shorthand expectation operator E to represent the generalization ability on testing data. *Bias-variance decomposition* theorem states that the regression error of a predictor can be decomposed into its *bias* and *variance*:

$$E\{(G-Y)^2\} = \underbrace{(E\{G\}-Y)^2}_{bias(G)^2} + \underbrace{E\{(G-E\{G\})^2\}}_{variance(G)} \tag{3}$$

It is a property of the generalization error in which bias and variance have to be balanced against each other for best performance.

Single model, however, turns out to be far from optimal in practice which has been evidenced by several studies, both theoretically [21, 20] and empirically [22, 23]. Consider the ensemble output $\tilde{G}$ by averaging individual's response G_k, i.e.,

$$\tilde{G} = \frac{1}{K}\sum_{k=1}^{K} G_k, \tag{4}$$

Here we restrict our analysis to the uniform combination case which is commonly used in practice although the decomposition presented below generalize to non-uniformly weighted ensembles as well. Posing the ensemble as a single learning unit, its bias-variance decomposition can be shown by the following equation:

$$E\{(\tilde{G}-Y)^2\} = \underbrace{(E\{\tilde{G}\}-Y)^2}_{bias(\tilde{G})^2} + \underbrace{E\{(\tilde{G}-E\{G\})^2\}}_{variance(\tilde{G})} \tag{5}$$

Consider ensemble output in Eqn. 4, it is straightforward to show:

$$\begin{aligned} E\{(\tilde{G}-Y)^2\} &= \underbrace{\frac{1}{K^2}(\sum_{k=1}^{K}(E\{G_k\}-Y))^2}_{bias(G)^2} \\ &+ \underbrace{\frac{1}{K^2}\sum_{k=1}^{K} E\{(G_k - E\{G_k\})^2\}}_{variance(G)} \\ &+ \underbrace{\frac{1}{K^2}\sum_{k=1}^{K}\sum_{j\neq k} E\{(G_k - E\{G_k\})(G_j - E\{G_j\})\}}_{covariance(G)} \end{aligned} \tag{6}$$

The so-called *bias-variance-covariance* decomposition illustrates that in addition to the internal bias and variance, the generalization error of an ensemble depends on the covariance between the individuals as well.

It is natural to show that

$$\begin{aligned} &\frac{1}{K}E\{(G_k-Y)^2\} = \textbf{bias}(G)^2 + \\ &+ [K \times \textbf{variance}(G) + \frac{1}{K}\sum_{k=1}^{K}(E\{G_k\} - E\{\tilde{G}\})^2] \end{aligned} \tag{7}$$

$$\begin{aligned} &\frac{1}{K}\sum_{k=1}^{K} E\{(G_k-\tilde{G})^2\} = \\ &- [\textbf{variance}(G) + \textbf{covariance}(G)] \\ &+ [K \times \textbf{variance}(G) + \frac{1}{K}\sum_{k=1}^{K}(E\{G_k\} - E\{\tilde{G}\})^2] \end{aligned} \tag{8}$$

then it is easy to show:

$$E\{(\tilde{G}-Y)^2\} = \frac{1}{K}\sum_{k=1}^{K} E\{(G_k-Y)^2\} - \frac{1}{K}\sum_{k=1}^{K} E\{(G_k-\tilde{G})^2\} \tag{9}$$

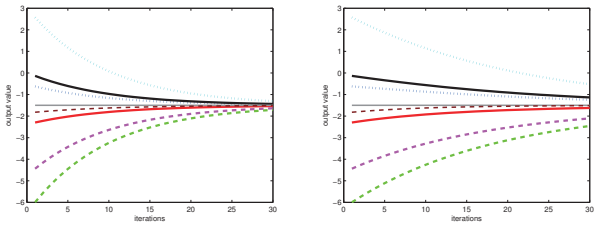
Eqn. 9 explains the effect of error correlations in an ensemble model by stating that *the quadratic error of the ensemble estimator is guaranteed to be less than or equal to the average quadratic error of the component estimators*. This is also in line with the strength-correlation theory [24], which advocates learning a set of both accurate and decorrelated models. Based on this, [25] proposed a "division of labor" approach by learning a correlation regularized ensemble by training several shallow feed forward networks with the following objective:

$$\begin{aligned} e_k &= \frac{1}{2}(G_k - \mathbf{Y})^2 + \lambda(G_k - \tilde{G})(\sum_{j\neq i}(G_j - \tilde{G})), \\ &= \frac{1}{2}(G_k - \mathbf{Y})^2 - \lambda(G_k - \tilde{G})^2, \end{aligned} \tag{10}$$

Eqn. 10 can be regarded as a smoothed version of Eq. 9 to improve the generalization ability of the ensemble models. Please note that the optimal value of λ may not necessarily be 0.5 because of the discrepancy between the training and testing data [20]. By setting $\lambda = 0$, we actually achieve conventional ensemble learning (non-boosting type) where each model is optimized independently. It is straightforward to show that the first part in Eqn 10 corresponds to bias plus an extra term $[K \times \textbf{variance}(G) + \frac{1}{K}\sum_{k=1}^{K}(E\{G_k\} - E\{\tilde{G}\})^2]$ while the second part stands for the variance, covariance and the same term $[K \times \textbf{variance}(G) + \frac{1}{K}\sum_{k=1}^{K}(E\{G_k\} - E\{\tilde{G}\})^2]$. Since the extra term appears on both sides, it cancels out when we combine them by subtracting, as done in Eqn. 10. Thus by introducing the second part in Eq. 10, we aim at achieving better "diversity" which balances the components of bias

variance and the ensemble covariance to reduce the overall MSE.

To demonstrate this, consider the scenario in Fig 1. We are using an regression ensemble consisting of 6 regressors where the ground truth is -1.5. Each curve in Fig 1 illustrates the evolution of one regressor when trained with gradient descent, i.e. $f_{in} = f_{i(n-1)} - \gamma \frac{d_E}{d_{f_{i(n-1)}}}$, where γ and E stands for the learning rate and mean-square loss function, respectively. $i \in 1, 2, \cdots 6$ is the index of individual models in the ensemble and $n \in 1, 2, \cdots 30$ stands for the index of iterations. Although both conventional ensemble learning (Fig. 1a) and NCL (Fig. 1b) may lead to correct estimations by simple model averaging, NCL results in much diversified individual models which make error cancellation being possible on testing data.



(a) Conventional ensemble learning (b) NCL

Figure 1: Demonstration of the training process of conventional learning and NCL.

3.2. Deep Negative Correlation Learning

Although the idea introduced here is theoretically general for regression ensemble, we choose crowd counting as our concrete application. We consider our mapping function as an ensemble of predictors as defined in Eqn. 4 where each base predictor is posed as:

$$G_k(\mathbf{x}_i) = G_k^{\mathcal{L}}(G_k^{\mathcal{L}-1} \cdots (G_k^1(\mathbf{x}_i)))), \quad k = 1, 2 \cdots K, i = 1, 2 \cdots N \tag{11}$$

where k and i stands for the index for individual models and data samples. More specifically, each predictor in the ensemble consists of cascades of feature extractor G_k^l, $l = 1, 2 \cdots \mathcal{L} - 1$ and regressor $G_k^{\mathcal{L}}$. Motivated by recent success of ConvNets on visual recognition tasks, each feature extractor G_k^l is embodied by a typical layer of Fully Convolutional ConvNets. By "Fully Convolutional" we mean G commutes with translation. More formally, considering a translation operator for a one-dimensional signal $(T_\kappa \mathbf{x})[i] = \mathbf{x}[i - \kappa]$, a function G that maps signals to signals is fully-convolutional with integer stride s if $G(T_{\kappa s}\mathbf{x}) = T_\kappa G(\mathbf{x})$ for any translation κ. The definition generalize to image signals in a straightforward way. Fully-convolutional realization of G is advantageous in the sense that one can provide input in an arbitrary size to the network and it will compute mapping results, which is the estimation of crowd density map.

In our implementation, lower levels of feature extractors are shared by each predictor for efficiency, i.e, $G_k^l = G^l$, $l = 1, 2 \cdots \mathcal{L} - 1$, $k = 1, 2 \cdots K$. Furthermore, building on the lessons learnt from subspace idea in ensemble learning [24], highest level of feature extractor $G_k^{\mathcal{L}-1}$ outputs a different feature subset for different regressor $G_k^{\mathcal{L}}$ to insert more diversities. Generally speaking, network specification of G_k^l is problem dependent and we show that, the proposed method, named as decorrelated ConvNet (D-ConvNet), is end-to-end-trainable and independent of the backbone fully-convolutional network architectures. Extensive experiments on very deep VGG as well as our customized network structure indicate the superiority of D-ConvNet compared with several state-of-the-art methods.

3.3. Crowd density map

It is well appreciated that the density map [1] based on annotated pedestrians' spatial location contains rich and abundant local and detailed information. Thus when learning density map using ConvNet, the learned filters become more sensitive to pedestrians of different sizes and perspective variation, benefitting to improve the counting accuracy. For a training image $\mathbf{x}_i$, the ground truth density function can be defined as a kernel density estimate based on the annotated pedestrians' points:

$$F(p) = \sum_{p \in P} \mathcal{N}(p, P, \sigma), \tag{12}$$

where p denotes a pixel, $\mathcal{N}(p, P, \sigma)$ denotes a normalized 2D Gaussian kernel evaluated at p, with an user-defined mean value P, and an isotropic covariance matrix with σ being a small value. The σ is usually set to $0.2M(p)$ when the perspective map $M(p)$ can be available [5]. However, generating such perspective maps is a laborious task and involves manually labeling several pedestrians by marking their height. Zhang et al. [4] provides an alternative to determine σ adaptively. They determine the spread parameter σ based on the size of the head for each person within the image. Due to the occlusion in many cases, it is difficult to obtain the size of the head. In this case, they assume around each head, the crowd is somewhat evenly distributed, then the average distance $\bar{d}$ between the head and its nearest k neighbors (in the image) gives a reasonable estimate of head size. As a result, the σ can be set to $\beta\bar{d}$. We found empirically $\beta = 0.3$ gives the best result.

3.4. Network Structure

Fig. 2 gives an overview of our crowd counting system. We report the performance of deep NCL for crowd counting

with two fully convolutional network configurations. The first setting, which is named as D-ConvNet-v1, employs a deep pretrained VGG16 network and make several modifications. Firstly, the stride of the fourth max-pool layer is set to 1. Secondly, the fifth pooling layer was further removed. This provides us a much larger feature map with richer information. To handle the receptive-field mismatch caused by the removal of stride in the fourth max-pool layer, we then double the receptive field of convolutional layers after the fourth max-pool layer by using the technique of holes introduced in [26]. Finally, we use a $64 \times 1 \times 1$ convolution layer as regressor $G_k^{\mathcal{L}}$ on each output feature map to get the final crowd density map. Specifically, each regressor $G_k^{\mathcal{L}}$ is sparsely connected to a small portion of feature maps from the last convolutional layer($conv5\_3$) of VGG16 network, implemented via the well-established "group convolution" strategy [16, 27].

Unlike existing work [4, 5, 13, 14, 28] which may only work on one deep or shallow network configurations, we show deep NCL introduced here can be more generalizable and thus can work well for different network structures. To demonstrate this, we train deep NCL on a relatively shallower model named as D-ConvNet-v2, which is constructed by stacking several Multi-Scale Blob as shown in Fig. 3, aiming to increase the depth and expand the width of crowd model in a single network. Multi-Scale Blob(MSB) is an Inception-like model which enhances the feature diversity by combining feature maps from different network branches. More specifically, it contains multiple filters with different kernel size (including 7×7, 5×5 and 3×3). This also makes the net more sensitive to crowd scale changing of the images.

Motivated by VGGNet [18], to make model more discriminative, we further achieve 5×5 and 7×7 convolutional layers by stacking two and three 3×3 convolutional layers, respectively. In our adopted network, the first convolution layer consists of $16 \times 5 \times 5$ filters and is followed by a 2×2 max pooling layer. After that, we stack two MSB modules as demonstrated in Fig. 3 where the first MSB modules is followed by a 2×2 max-pooling layer. The number of feature maps of each convolution layer in these two MSB modules is 24 and 32, respectively. Finally, we use the same 1×1 convolution layer on each of the feature map as regressor $G_k^{\mathcal{L}}$ to get the final crowd density map.

All the parameters in D-ConvNet can be efficiently trained and updated by back-propagating gradients of our error function given in Eq. 10 with Stochastic Gradient Descent. Taking derivatives of this objective function is straightforward and can be easily plugged into many popular ConvNet platforms, such as *Caffe* [29]. More specifically, in each iteration, we get the ensemble output $\tilde{G}$ by running one-pass feed-forward through the whole network. We then back-propagate the gradient of Eq. 10 and update parameters of all the network structure.

4. Experiments

We evaluate the proposed methods on three benchmark datasets: UCF_CC_50 dataset [30], Shanghaitech dataset [4] and WorldExpo'10 dataset [5]. We implement our crowd counting system in Caffe [29] on a single machine with a TitanX GPU. The proposed networks are trained using Stochastic Gradient Descent with a mini-batch size of 1 at a fixed constant momentum value of 0.9. Weight decay with a fixed value of 0.0005 is used as a regularizer. We use a fixed learning rate of 1e-7 in the last convolution layer of our crowd model to enlarge the gradient signal for effectively parameter updating and use a relatively smaller learning rate of 1e-9 in other layers.

4.1. Evaluation Metric

The widely used *mean absolute error* (MAE) and the *root mean squared error* (RMSE) are adopted to evaluate the performance of different methods. The MAE and RMSE are defined as follows:

$$MAE = \frac{1}{N} \cdot \sum_{i=1}^{N} |(y_i - \tilde{y}_i)|, RMSE = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - \tilde{y}_i)^2} \tag{13}$$

Here N represents the total number of images in the testing datasets, y_i and $\tilde{y}_i$ are the ground truth and the estimated value respectively for the i^{th} image.

4.2. UCF_CC_50 dataset

The challenging UCF_CC_50 dataset [30] contains 50 images that are randomly collected from the Internet. The number of head ranges from 94 to 4543 with an average of 1280 individuals per image. The total number of annotated persons within 50 images is 63974. Challenging issues such as large variations in head number among different images from a small number of training images come in the way of accurately counting for UCF_CC_50 dataset. We follow the standard evaluation protocol by splitting the dataset randomly into five parts in which each part contains ten images. Five-fold cross-validation is employed to evaluate the performance. Since the perspective maps are not provided, we generate the ground truth density map by using the Zhang's method [4] as described in Section 3.3.

We compared our method on this dataset with ten state-of-the-art methods. In [31, 1, 30], handcraft features are used to regress the density map from the input image. Several CNN-based methods in [5, 14, 4, 32, 28, 15, 17] were also considered here due to their superior performance on this dataset. Table 1 summarizes the detailed results. Firstly, it is obvious that most deep learning methods outperform hand-crafted features significantly. In [14] the authors proposed to employ a shallow network to assist the training

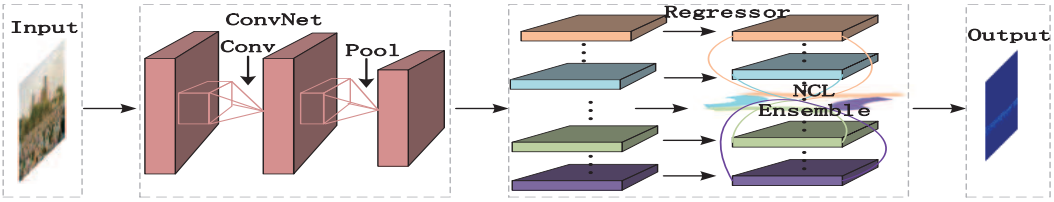


Figure 2: Details of the proposed D-ConvNest for crowd counting. We formulate a single ConvNet as ensemble learning with the same amount of parameter. D-ConvNet receives crowd images as input and processes them by stack of typical convolutional and pooling layers. Finally, a "divide and conquer" strategy is adopted to learn a pool of regressors to regress the crowd density map on top of each convolutional feature map at top layers. Each regressor is jointly optimized with the ConvNet by an amended cost function which penalizes correlations with others to make better trade-offs among the bias-variance-covariance in the ensemble. We demonstrate the feasibility of D-ConvNets on VGG and our own customized network.

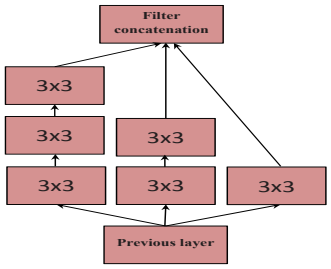


Figure 3: Demonstration of Multi-Scale Blob module used in D-ConvNet-v2.

process of deep VGG network. With the proposed deep negative learning strategy, It is also interesting to see that 1) both our deep(D-ConvNet-v1) and shallow (D-ConvNet-v2) networks works well; 2) deep networks(D-ConvNet-v1) are better than shallower networks(D-ConvNet-v2), as expected. However, shallower network(D-ConvNet-v2) still leads to competitive results and may be advantageous in resource-constrained scenarios as it is computationally cheaper; (3) finally, It is straightforward to see that D-ConvNet-v1 outperforms others on this dataset.

4.3. Shanghaitech dataset

The Shanghaitech dataset [4] is a large-scale crowd counting dataset, which contains 1198 annotated images with a total of 330,165 persons. This dataset is the largest one in the literature in terms of the number of annotated pedestrians. It consists of two parts: Part_A consisting of 482 images are randomly captured from the Internet, and Part_B including 716 images are taken from the busy streets in Shanghai. Each part is divided into training and testing subset. The crowd density varies significantly among the subsets, making it difficult to estimate the number of pedestrians.

We compare our method with four existing methods on the ShanghaiTech dataset. All the detailed results for each method are illustrated in Table 2. In the same way, we can see that all deep learning methods outperform hand-crafted features significantly. The shallow model in [4] employs a much wider structure by a multi-column design

Table 1: Comparing results of different methods on the UCF_CC_50 dataset.

Method	MAE	RMSE
Rodriguez et al.[31]	655.7	697.8
Lempitsky et al.[1]	493.4	487.1
Isrees et al.[30]	419.5	541.6
Zhang et al. [5]	467.0	498.5
CrowdNet [14]	452.5	-
Zhang et al. [4]	377.6	509.1
Zeng et al. [32]	363.7	468.4
Mark et al. [28]	338.6	424.5
Daniel et al. [15]	333.7	425.2
Sam et al. [17]	318.1	439.2
D-ConvNet-v1	**288.4**	**404.7**
D-ConvNet-v2	354.1	443.7

Table 2: Comparing performances of different methods on Shanghaitech dataset.

Method	Part_A		Part_B	
	MAE	RMSE	MAE	RMSE
LBR+RR	303.2	371.0	59.1	81.7
Zhang et al. [5]	181.8	277.7	32.0	49.8
Zhang et al. [4]	110.2	173.2	26.4	41.3
Sam et al. [17]	90.4	135.0	21.6	33.4
D-ConvNet-v1	**73.5**	**112.3**	**18.7**	**26.0**
D-ConvNet-v2	101.7	152.8	25.7	38.6

Table 3: Mean absolute errors of the WorldExpo'10 crowd counting dataset.

Method	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Average
LBP+RR	13.6	58.9	37.1	21.8	23.4	31.0
Zhang et al.[5]	9.8	14.1	14.3	22.2	3.7	12.9
Zhang et al. [4]	3.4	20.6	12.9	13.0	8.1	11.6
Sam et al. [17]	4.4	15.7	**10.0**	11.0	5.9	9.4
D-ConvNet-v1	**1.9**	**12.1**	20.7	**8.3**	**2.6**	**9.1**
D-ConvNet-v2	4.9	14.3	18.7	11.3	4.6	10.7

Table 4: Comparing transfer performances of different methods on Shanghaitech and UCF_CC_50 dataset.

Method	Part_A → Part_B		Part_B → Part_A		Part_A→ UCF_CC_50	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Zhang et al. [4]	85.2	142.3	221.4	357.8	397.7	624.1
D-ConvNet-v1	**49.1**	**99.2**	**140.4**	**226.1**	**364**	**545.8**

and performs better than the shallower CNN models in [5] in both cases. D-ConvNet-v1 performs consistently better than D-ConvNet-v2, as expected, because of employing a much deep pre-trained model. Moreover, it is interesting to see that with deep negative learning, D-ConvNet-v2 which employs a relatively shallower network structure is on a par with a much complicated and state-of-the-art switching strategy [17]. Finally, our deep structure, D-ConvNet-v1 leads to the best performance.

4.4. WorldExpo'10 dataset

The WorldExpo'10 dataset [5] is a large-scale and cross-scene crowd counting dataset. It contains 1132 annotated sequences which are captured by 108 independent cameras, all from Shanghai 2010 WorldExpo'10. This dataset consists of 3980 frames with a total of 199,923 labeled pedestrians, which are annotated at the centers of their heads. Five different regions of interest(ROI) and the perspective maps are provided for the test scenes.

We follow the standard evaluation protocol and use all the training frames to learn our model. For comparison, the quantitative results are given in Table 3. In the same way, we observe that learned representations are more robust than the handcraft features. Even without using the perspective information, our results are still comparable with another deep learning method [5] which used perspective normalization to crop 3×3 square meters patches with 0.5 overlaps on testing time. D-ConvNet-v1 outperforms all other in terms of average performance. More specifically, it achieves the best performance in four out of five scenes while method in [17] win in the remaining one cases.

Table 5: Comparing Performance of NCL regularization and conventional ensemble.

Datasets	Ensemble		D-ConvNet-v1	
	MAE	RMSE	MAE	RMSE
UCF_CC_50	380.5	527.2	**288.4**	**404.7**
Shanghaitech Part_A	91.6	127.9	**73.5**	**112.3**
Shanghaitech Part_B	21.3	30.9	**18.7**	**26.0**
WorldExpo'10	16.4	-	**9.1**	-

5. Discussions

After demonstrating the superiority of D-ConvNets by extensively comparing them with many state-of-the-art methods on multiple datasets, we now provide more discussions to shed light upon their rationale and sensitivities with some hyper-parameters. We also provide additional experiments on cross-scene evaluation to further understand the merits of the proposed method. We will focus on D-ConvNet-v1 as it consistently performs better than the other version.

5.1. NCL or Conventional Ensemble Learning?

In Table 5, we compared the performance of the proposed method with conventional ensemble learning. It is widely accepted that training deep networks like VGG remains to be challenging. In [14], a shallow network was proposed to assist the training and improve the performance of deep VGG network. When compared with results achieved on dataset UCF_CC_50 by other methods

shown in Table 1, our implementation of conventional ensemble method using a single VGG network leads to much improved results. However, it still over-fits severely compared with other state-of-the-art methods. More specifically, it was outperformed by recent methods such as multi-column structure [4], multi-scale Hydra method [15], and advanced switching strategy [17]. In contrast, the proposed D-ConvNet leads to much improved performance compared with this baseline in all cases and outperforms all aforementioned methods. As illustrated in Fig. 1, the NCL mechanism used in the proposed D-ConvNet encourages diversities in the ensemble and thus it is more likely to allow error canceling. The learning objective function in Eqn. 10 is also in line with Breiman's strength-correlation theory [24] on the VC-type bounds for generalization ability of ensemble models which advocated both accurate and decorrelated individual models. It as well appreciated that the individual model should be able to exhibit different patterns of generalization-a very simple intuitive explanation is that a million identical estimators are obviously no better than a single.

5.2. Cross Scene Evaluation

In order to test the generalization ability of the crowd counting system, here we evaluate the proposed method in a cross-scene setting where no laborious data annotation is required for counting people in new target surveillance crowd scenes unseen in the training set. More specifically, we consider the follow cross-scene scenarios: i) Part_A $\rightarrow$ Part_B, ii) Part_B $\rightarrow$ Part_A and iii) Part_A$\rightarrow$ UCF_CC_50. In each case, D-convNet is trained on the first dataset and evaluated on the second one. Results in Table 4 indicates that performances in these scenarios are worse, as expected, probably due to dataset bias. D-ConvNet performs much better in all cases compared with the state-of-the-art method of [4]. This further verifies that through managing the intrinsic diversities of each model in the ensemble, D-ConvNet leads to better generalization capability.

5.3. Effect of λ and K

Parameter λ controls the correlation between each model in the ensemble. On the one hand, setting $\lambda = 0$ is equivalent to train each regressor in an independent manner. On the other hand, employing a larger value for λ overemphasizes the effect of diversity and may lead to poor individual regressors. We empirically find that setting λ to be a relatively smaller value $\in [10^{-3}, 10^{-2}]$ usually leads to satisfactory results. Parameter K stands for the number of base regresors in the ensemble. Theoretically speaking, conventional ensemble learning such as bagging and decision tree ensemble requires larger ensemble sizes [33, 22, 23] to perform well. We empirically find that the performances of D-ConvNet works well even with a relatively smaller ensemble size (32-64). In this work, K is set to be 64 as no significant improvement is observed with a more number of regressors.

6. Conclusion

In this paper, we present a simple yet effective learning strategy for crowd counting. We pose typical counting by regression as an ensemble learning problem and learn a pool of weak regressors using convolutional feature maps. The main component of this ensemble architecture is the introduction of negative correlation learning (NCL), which aims to improve generalization capability of the ensemble models. We show the proposed method, named as Decorrelated ConvNet (D-ConvNet), has sound generalization capability through managing their intrinsic diversities. D-ConvNet is generic and independent of the backbone fully-convolutional network architectures. Extensive experiments of very deep VGG as well as our customized network structure on several challenging datasets demonstrate the superiority of D-ConvNet.

Acknowledgements: This work is supported by the Swiss National Science Foundation (Grant No. 205321_169239), the National Natural Science Foundation of China (Grant No.61772475) and the Key Science and Technology Program of Henan Province (Grant No.172102210011).

References

[1] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332.

[2] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *roceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–7.

[3] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and bayesian regression," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, 2012.

[4] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597.

[5] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 833–841.

[6] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1198–1211, 2008.

[7] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "An evaluation of crowd counting methods, features and regression

models," *Computer Vision and Image Understanding*, vol. 130, pp. 1–17, 2015.

[8] Z. Shi, Y. Ye, and Y. Wu, "Rank-based pooling for deep convolutional neural networks," *Neural Networks*, vol. 83, pp. 21–31, 2016.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[12] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302.

[13] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 1299–1302.

[14] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: a deep convolutional network for dense crowd counting," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 640–644.

[15] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 615–629.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[17] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 3, 2017, p. 6.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[19] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[20] 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.

[21] 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.

[22] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.

[23] L. Zhang and P. N. Suganthan, "Benchmarking ensemble classifiers with novel co-trained kernal ridge regression and random vector functional link ensembles [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 61–72, 2017.

[24] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] 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.

[26] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[27] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Stct: Sequentially training convolutional networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1373–1381.

[28] M. Marsden, K. McGuiness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," *arXiv preprint arXiv:1612.00220*, 2016.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *CoRR*, vol. abs/1408.5093, 2014.

[30] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554.

[31] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert, "Density-aware person detection and tracking in crowds," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 2423–2430.

[32] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," *arXiv preprint arXiv:1702.02359*, 2017.

[33] 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.