Improving the Learning of Multi-column Convolutional Neural Network for Crowd Counting

Zhi-Qi Cheng[1,2*], Jun-Xiu Li[1,3*], Qi Dai[3], Xiao Wu[1†], Jun-Yan He[1], Alexander G. Hauptmann[2]
[1]Southwest Jiaotong University, [2]Carnegie Mellon University, [3]Microsoft Research
{zhiqic,alex}@cs.cmu.edu,{lijunxiu@my,wuxiaohk@home}.swjtu.edu.cn,qid@microsoft.com,junyanhe1989@gmail.com

ABSTRACT

Tremendous variation in the scale of people/head size is a critical problem for crowd counting. To improve the scale invariance of feature representation, recent works extensively employ Convolutional Neural Networks with multi-column structures to handle different scales and resolutions. However, due to the substantial redundant parameters in columns, existing multi-column networks invariably exhibit almost the same scale features in different columns, which severely affects counting accuracy and leads to overfitting. In this paper, we attack this problem by proposing a novel Multi-column Mutual Learning (McML) strategy. It has two main innovations: 1) A statistical network is incorporated into the multi-column framework to estimate the mutual information between columns, which can approximately indicate the scale correlation between features from different columns. By minimizing the mutual information, each column is guided to learn features with different image scales. 2) We devise a mutual learning scheme that can alternately optimize each column while keeping the other columns fixed on each mini-batch training data. With such asynchronous parameter update process, each column is inclined to learn different feature representation from others, which can efficiently reduce the parameter redundancy and improve generalization ability. More remarkably, McML can be applied to all existing multi-column networks and is end-to-end trainable. Extensive experiments on four challenging benchmarks show that McML can significantly improve the original multi-column networks and outperform the other state-of-the-art approaches.

KEYWORDS

Crowd Counting; Multi-column Network; Mutual Learning Strategy

ACM Reference Format:
Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, Jun-Yan He, Alexander G. Hauptmann. 2019. Improving the Learning of Multi-column Convolutional Neural Network for Crowd Counting. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3343031.3350898

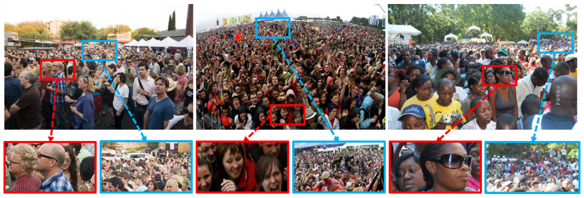

Figure 1: Examples of ShanghaiTech Part A dataset [69]. Crowd counting is a challenging task with the significant variation in the people/head size due to the perspective effect.

1 INTRODUCTION

With the growth of wide applications, such as safety monitoring, disaster management, and public space design, crowd counting has been extensively studied in the past decade. As shown in Figure 1, a significant challenge of crowd counting lies in the extreme variation in the scale of people/head size. To improve the scale invariance of feature learning, Multi-column Convolutional Neural Networks are extensively studied [3, 12, 21, 32, 44, 50, 69]. As illustrated in Figure 2, the motivation of multi-column networks is intuitive. Each column is devised with different receptive fields (e.g., different filter sizes) so that the features learned by different columns are expected to focus on different scales and resolutions. By assembling features from all columns, multi-column networks are easily adaptive to the large variations of the scale due to the generalization ability across scales and resolutions.

Although multi-column architecture is naturally employed for addressing the issue of various scale change, previous works [12, 21, 30, 44, 62] have pointed out that different columns always generate features with almost the same scale, which indicates that existing multi-column architectures cannot effectively improve the scale invariance of feature learning. To further verify this observation, we have extensively analyzed three state-of-the-art networks, i.e., MCNN [69], CSRNet [30] and ic-CNN [44]. It is worth noting that CSRNet is a single column network, which has four different configurations (i.e., different dilation rates). We remould CSRNet to treat each configuration as a column, and design a four-column network as an alternative. The Maximal Information Coefficient (MIC)[1] and the Structural SIMilarity (SSIM)[2] are computed based on the results of different columns. MIC measures the strength of association between the outputs (i.e., crowd counts) and SSIM measures the similarity between density maps. As shown in Table 1, different columns (Col.↔Col.) always output almost the same counts (i.e., high MIC) and the similar estimated density maps (i.e., high SSIM). In contrast, a large gap between the ensemble of all columns and the ground truth (Col.↔GT.) still exists. This comparison shows

*Equal contribution. This work was done when Zhi-Qi Cheng and Jun-Xiu Li visited at Microsoft Research. †Xiao Wu is the corresponding author.

MM '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350898

[1] https://en.wikipedia.org/wiki/Maximal_information_coefficient
[2] https://en.wikipedia.org/wiki/Structural_similarity

Table 1: The result analysis of three multi-column networks. The values in the table are the average of all columns. Col.↔Col. is the result between different columns. Col.↔GT is the result between the ensemble of all columns and the ground truth.

	Col.↔ Col.		Col.↔ GT	
Method	**MIC**	**SSIM**	**MIC**	**SSIM**
ShanghaiTech Part A [69]				
MCNN [69]	0.94	0.71	0.52	0.55
CSRNet [30]	0.93	0.84	0.74	0.71
ic-CNN [44]	0.92	0.72	0.70	0.68
UCF_CC_50 [24]				
MCNN [69]	0.81	0.53	0.70	0.36
CSRNet [30]	0.87	0.72	0.71	0.48
ic-CNN [44]	0.93	0.70	0.57	0.52

that there are substantial redundant parameters among columns, which makes multi-column architecture fails to learn the features across different scales. On the other hand, it indicates that existing multi-column networks tend to overfit the data and can not learn the essence of the ground truth.

Inspired by previous works [30, 44, 62], we reveal that the problem of existing multi-column networks lies in the difficulty of learning features with different scales. Generally speaking, there are two main problems: 1) There is no supervision to guide multiple columns to learn features at different scales. The current learning objective is only to minimize the errors of crowd count. Although we have designed different columns to have different receptive fields, they are still gradually forced to generate features with almost the same scale along with the network optimization. 2) There are huge redundant parameters among columns. Because of parallel column architectures, multi-column networks naturally brought in redundant parameters. As the analysis of [1], with the increase of parameters, a more substantial amount of training data is also required. It implies that existing multi-column networks are typically harder to train and easier to overfit.

In this paper, we propose a novel Multi-column Mutual Learning (McML) strategy to improve the learning of multi-column networks. As illustrated in Figure 3, our McML addresses the above two issues from two aspects. 1) A statistical network is proposed to measure the mutual information between different columns. The mutual information can approximately measure the scale correlation between features from different columns. By additionally minimizing the mutual information in the loss, different column structures are forced to learn feature representations with different scales. 2) Instead of the conventional optimization that updates the parameters of multiple columns simultaneously, we devise a mutual learning scheme that can alternately optimize each column while keeping the other columns fixed on each mini-batch training data. With such asynchronous learning steps, each column is inclined to learn different feature representation from others, which can efficiently reduce the parameter redundancy and improve the generalization ability. The proposed McML can be applied to all existing multi-column networks and is end-to-end trainable. We conduct extensive experiments on four datasets to verify the effectiveness of our method.

The main contribution of this work is the proposal of Multi-column Mutual Learning (McML) strategy to improve the learning of multi-column networks. The solution also provides the elegant views of how to explicitly supervise multi-column architectures to learn features with different scales and how to reduce the enormous redundant parameters and avoid overfitting, which are problems not yet fully understood in the literature.

2 RELATED WORK

2.1 Detection-based Methods

These models use visual object detectors to locate people in images. Given the individual localization of each people, crowd counting becomes trivial. There are two directions in this line, i.e., detection on 1) whole pedestrians [4, 16, 58, 70] and 2) parts of pedestrians [17, 25, 31, 61]. Typically, local features [16, 31] are first extracted and then are exploited to train various detectors (e.g., SVM [31] and AdaBoost [59]). Although these works achieve satisfactory results for the low-density scenario, they are unable to generalize for high-density images since it is impossible to train a detector for extremely crowded scenes.

2.2 Regression-based Methods

Different from detection-based models, regression-based methods directly estimate crowd count using image features. It has two steps: 1) extract powerful image features, 2) use various regression models to estimate the crowd count. Specifically, image features include edge features [9, 11, 36, 45, 47] and texture features [10, 11, 24, 43]. Regression methods cover Bayesian [9], Ridge [11], Forest [43] and Markov Random Field [24, 41]. Since these works always use handcrafted low-level features, they still cannot obtain satisfactory performance.

2.3 CNN-based Methods

Due to substantial variations in the scale of people/head size, most recent studies extensively use Convolutional Neural Networks (CNN) with multi-column structures for crowd counting. Specifically, a dual-column network is proposed by [3] to merge shallow and deep layers to estimate crowd counts. Inspired by this work, a great three-column network named MCNN is proposed by [69], which employs different filters on separate columns to obtain the various scale features. Noted that there are a lot of works to continually improve MCNN [26, 55, 56, 60]. Sam et al. [50] introduce a switching structure, which uses a classifier to assign input image patches to best column structures. Recently, Liu et al. [32] propose a multi-column network to simultaneously estimate crowd density by detection and regression models. Ranjan et al. [44] employ a two-column structure to iterative train their model with different resolution images.

In addition to multi-column networks, there are a lot of methods to improve scale invariance of feature learning by 1) studying on the fusion of multi-scale features [35, 57, 62, 63], 2) studying on multi-blob based scale aggregation networks [7, 64], 3) designing scale-invariant convolutional or pooling layers [21, 30, 33, 56, 62], and 4) studying on automated scale adaptive networks [48, 49, 66]. On the other hand, a lot of studies devote to using perspective maps [52], geometric constraints [34, 68], and region-of-interest [33] to further improve the counting accuracy.

These state-of-the-art methods aim to improve the scale invariance of feature learning. Inspired by recent studies [30, 44, 62], we reveal that existing multi-column networks cannot effectively learn different scale features as Sec. 1. To solve this problem, we propose

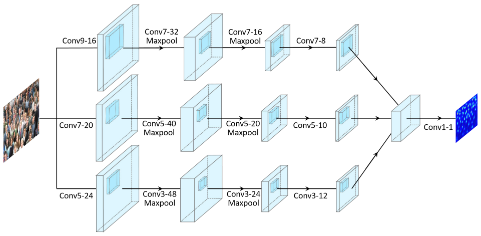

Figure 2: The architecture of MCNN [69]. It is a classical Multi-column Convolutional Neural Network. It employs different size of filters on three columns to obtain different scale features.

a novel Multi-column Mutual Learning (McML) strategy, which can be applied to all existing CNN-based multi-column networks and is end-to-end trainable. It is noted that the previous work ic-CNN [44] also proposes an iterative learning strategy to improve the learning of multi-column networks. Different from our McML, since ic-CNN is designed for a specific neural architecture, it can not be generalized to all multi-column networks. Additionally, we have tested our McML on the same network of ic-CNN. Experimental results show that McML can still significantly improve the performance of the original ic-CNN.

3 MULTI-COLUMN MUTUAL LEARNING

In this section, we present the proposed Multi-column Mutual Learning (McML) strategy. The problem formulation is first introduced in Sec. 3.1. Then the overview of our McML is described in Sec. 3.2. More details of McML are illustrated in Sec. 3.3 to 3.5.

3.1 Problem Formulation

Recent studies define crowd counting task is as a density regression problem [7, 29, 69]. Given N training images $\mathbf{X} = \{x_1, \cdots, x_N\}$ as the training set, each image x_i is annotated with a total of c_i center points of pedestrians' heads $\mathbf{P}_i^{gt} = \{P_1, P_2, \cdots, P_{c_i}\}$. Typically, the ground truth density map y_i of image x_i is generated as,

$$\forall p \in x_i, y_i = \sum_{P \in \mathbf{P}_i^{gt}} \mathcal{N}^{gt}(p; \mu = P, \sigma^2), \tag{1}$$

where p is a pixel and $\mathcal{N}^{gt}$ is a Gaussian kernel with standard deviation σ. The number of people c_i in image x_i is equal to the sum of density of all pixels as $\sum_{p \in x_i} y_i(p) = c_i$. With these training data, crowd counting models aim to learn a regression model G with parameters θ to minimize the difference between estimated density map $G_\theta(x_i)$ and ground truth density map Y_i. Specifically, Euclidean distance, i.e., L_2 loss is employed to get an approximate solution,

$$L_2 = \frac{1}{2N} \sum_{i=1}^{N} (G_\theta(x_i) - y_i)^2, \tag{2}$$

where as the size of input images are different, the value of Eqn. 2 is further normalized by the number of pixels in each image.

It is noted that, as shown in Figure 1, enormous variation in the scale of people/head size is a critical problem for crowd counting. Many studies [5, 13, 19, 22, 46, 67] have proved that only using an individual regression model is theoretically far from the global optimal. To improve the scale invariance of feature learning, Convolutional Neural Networks with multi-column structures are extensively studied by recent works [26, 55, 56, 60, 69]. Figure 2 illustrates a typical multi-column network named MCNN [69]. The intentions of multi-column networks are natural, where each column structure is devised with different receptive fields (e.g., different filter sizes) so that the features learned by individual column is expected to focus on a particular scale of people/head size. With the ensemble of features from all columns, multi-column networks are easily adaptive to handle the large scale variations.

Although the motivation for multi-column structures is straightforward, previous works [30, 44, 62] have pointed out that existing multi-column networks cannot improve the scale invariance of features learning. As analyzed in Sec. 1, we are convinced that there are abundant redundant parameters between columns, which causes multi-column structures to fail to learn the features across different scales and invariably get almost the same estimated crowd counts and density maps. After thoroughly surveying previous works [30, 44, 46, 62, 67] and analyzing our experimental results in Table 1, we further reveal that the main problem of existing multi-column networks lies in the learning process. Generally speaking, current learning strategy has two main weaknesses. 1) It only optimizes the objective of crowd counting, while completely ignores the intention of using multi-column structures to learn different scale features. 2) It instantly optimizes multi-column structures at the same time, which can result in the enormous redundant parameters among columns and overfitting on the limited training data. To address these problems, our work aims to propose a general learning strategy named Multi-column Mutual learning (McML) to improve the learning of multi-column networks.

3.2 Overview of McML

In this section, we present an overview of Multi-column Mutual Learning (McML) strategy. For the sake of simplicity, we introduce the case of two columns as an example. As shown in Figure 3, our McML has two main innovations.

- McML has integrated a statistical network into multi-column structures to automatically estimate the mutual information between columns. The essential of the statistical network is a classifier network. Specifically, the inputs are features from different columns, and the output is the mutual information between columns. We use mutual information to approximately indicate the scale correlation between features from different columns. By minimizing the mutual information between columns, McML can guide each column to focus on different image scale information.
- McML is a mutual learning scheme. Different from updating the parameters of multiple columns simultaneously, McML alternately optimizes each column in turn until the network converged. In the learning of each column, the mutual information between columns is first estimated as prior knowledge to guide the parameter update. With the help of the mutual information between columns, McML can alternately make each column to be guided by other columns to learn different image scales/resolutions. It is proved that this mutual learning scheme can significantly reduce the volume of redundant parameters and avoid overfitting.

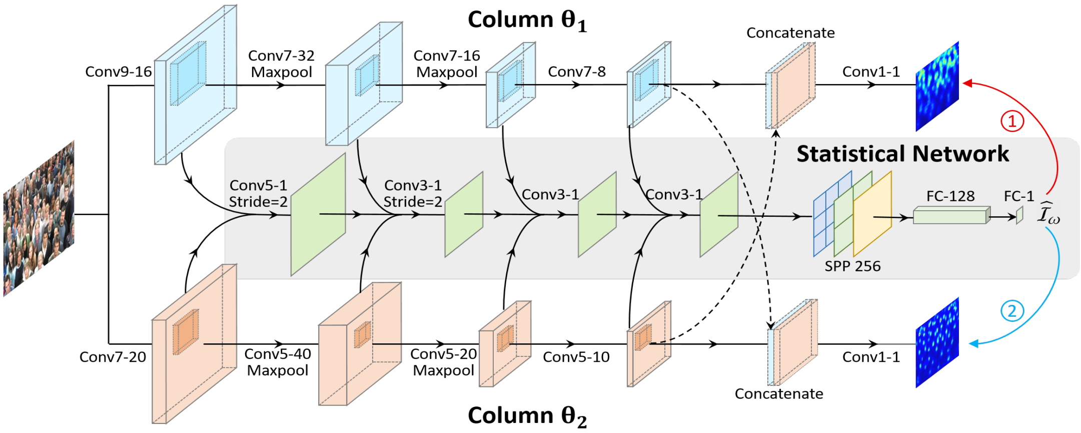

Figure 3: Overview of our Multi-column Mutual Learning (McML) strategy. It is equivalent to adding a statistical network to estimate the mutual information $\widehat{\mathcal{I}}_\omega$ between columns. By minimizing the mutual information, it can guide multi-columns to learn different scale information. Additionally, McML is a mutual learning scheme as arrows ① and ②, where each column is alternately optimized while keeping the other columns fixed on each mini-batch training data. Specifically, this is an example of two columns (Θ_1 and Θ_2) in MCNN [69]. ConvX-Y implies a convolution layer has Y filters with X×X kernel size. The stride of all convolutional layers is 1, except for the special reminder. MaxPool is the max pooling layer with a stride of 2. [Best viewed in color].

Mathematically, two columns with parameters θ_1 and θ_2 are alternately trained as,

$$L_{\theta_1} = \min_{\theta_1} L_2(\mathrm{Conv}(F_{\theta_1 \circ \theta_2}(X)), Y) + \alpha \widehat{\mathcal{I}}_\omega(C_{\theta_1}; C_{\theta_2}), \qquad (3)$$

$$L_{\theta_2} = \min_{\theta_2} L_2(\mathrm{Conv}(F_{\theta_1 \circ \theta_2}(X)), Y) + \alpha \widehat{\mathcal{I}}_\omega(C_{\theta_1}; C_{\theta_2}), \qquad (4)$$

where each column is trained by two losses. L_2 loss (Eqn. 2) is used to minimize counting errors, and $\widehat{\mathcal{I}}_\omega$ (Eqn. 7) is employed to minimize the mutual information between columns. α is the weight to trade off two losses. The value of mutual information $\widehat{\mathcal{I}}_\omega$ is computed by the statistical network with parameters ω. Here we have slightly abused symbols. C_{θ_1} and C_{θ_2} are features from different convolutional layers of two columns, which are used to estimate the mutual information. $F_{\theta_1 \circ \theta_2}(X)$ means the ensemble (i.e., concatenation) of features at the last convolutional layers for two both columns. Conv is a 1×1 convolutional layer that is used to predict density maps for crowd counting.

Typically, our proposed McML is also a mutual learning scheme. Two columns are alternately optimized until convergence. In the learning of each column, the mutual information $\widehat{\mathcal{I}}_\omega$ is first estimated as prior knowledge to guide the parameter update. Once the optimization of one column (e.g., θ_1) is finished, we will update the mutual information $\widehat{\mathcal{I}}_\omega$ again and alternately to update the other column (e.g., θ_2). Additionally, it is noted that Eqns. 3 and 4 show the situation of most multi-column networks (e.g., CrowdNet [3], AM-CNN [68], and MCNN [69]), where the features of multi-columns are concatenated to estimate density maps. However, a few multi-column networks (e.g., ic-CNN [44]) predict density maps in all columns. In these cases, $F_{\theta_1 \circ \theta_2}$ of Eqns. 3 and 4 should be replaced with F_{θ_1} and F_{θ_2} respectively. Where F_{θ_1} and F_{θ_2} are features from the last convolutional layers at two columns.

Specifically, we will introduce the mutual information estimation (i.e., computation of $\widehat{\mathcal{I}}_\omega$) in Sec. 3.3, the mutual learning scheme in Sec. 3.4 and neural architectures of statistical networks in Sec. 3.5.

3.3 Mutual Information Estimation

In this section, we first briefly introduce the definition of mutual information. Then we present the statistical network in details.

Mutual information is a fundamental quantity for measuring the correlation between variables. We treat column structures as different variables. Inspired by the success of previous works [28, 38], we use mutual information to indicate the degree of parameter redundancy between columns. Moreover, mutual information can also approximately measure the scale correlation between features from different columns. Instead of estimating the mutual information with parameters of columns, similar to [6, 15, 20], we chooses to compute the mutual information using the features of multi-columns since our objective is to learn different scale features. Typically, the mutual information between features C_{θ_1} and C_{θ_2} is defined as,

$$\mathcal{I}(C_{\theta_1}; C_{\theta_2}) := H(C_{\theta_1}) - H(C_{\theta_1} \mid C_{\theta_2}), \qquad (5)$$

where H is the Shannon entropy. $H(C_{\theta_1} \mid C_{\theta_2})$ measures the uncertainty in C_{θ_1} given C_{θ_2}. Previous works [6, 28, 38, 42] widely use Kullback-Leibler (KL) divergence to compute the mutual information,

$$\mathcal{I}(C_{\theta_1}; C_{\theta_2}) = D_{KL}(\mathbb{P}_{C_{\theta_1} C_{\theta_2}} \mid\mid \mathbb{P}_{C_{\theta_1}} \otimes \mathbb{P}_{C_{\theta_2}}), \qquad (6)$$

where $\mathbb{P}_{C_{\theta_1} C_{\theta_2}}$ is the joint distribution of two features. $\mathbb{P}_{C_{\theta_1}}$ and $\mathbb{P}_{C_{\theta_2}}$ are the marginal distributions. $\otimes$ means the production. Since the joint distribution $\mathbb{P}_{C_{\theta_1} C_{\theta_2}}$ and the product of marginal distributions $\mathbb{P}_{C_{\theta_1}} \otimes \mathbb{P}_{C_{\theta_2}}$ are unknown in our case, the mutual information of two columns is challenging to compute [40].

Fortunately, inspired by the previous work named MINE [2], we propose a statistical network to estimate the mutual information. The essence of the statistical network is a classifier. It can be used to distinguish the samples between the joint distribution and the product of marginal distributions. Instead of computing Eqn. 6, the statistical network chooses to use Donsker-Varadhan representation [18] i.e., $\mathcal{I}(C_{\theta_1}; C_{\theta_2}) \geq \widehat{\mathcal{I}}_\omega(C_{\theta_1}; C_{\theta_2})$, to get a lower-bound for

Algorithm 1 Mutual Information Estimation

```
Input: Randomly sampled $b$ images.
Draw features from two columns as the joint distribution,
$(C_{\theta_1}^{(1)}, C_{\theta_2}^{(1)}), \ldots, (C_{\theta_1}^{(b)}, C_{\theta_2}^{(b)}) \sim \mathbb{P}_{C_{\theta_1} C_{\theta_2}}$;
Randomly disrupt $C_{\theta_2}$ as the product of marginal distribution,
$(C_{\theta_1}^{(1)}, \widehat{C_{\theta_2}^{(1)}}), \ldots, (C_{\theta_1}^{(b)}, \widehat{C_{\theta_2}^{(b)}}) \sim \mathbb{P}_{C_{\theta_1}} \otimes \mathbb{P}_{C_{\theta_2}}$;
Evaluate mutual information $\widehat{\mathcal{I}}_\omega$ Das Eqn. 7;
Use moving average to get the gradient,
$\widehat{G}(\omega) \leftarrow \widetilde{\nabla}_\omega \widehat{\mathcal{I}}_\omega$;
Update the statistical network parameters,
$\omega \leftarrow \omega + \widehat{G}(\omega)$;
```

the mutual information estimation,

$$\widehat{\mathcal{I}}_\omega \leftarrow \frac{1}{b} \sum_{i=1}^{b} T_\omega(C_{\theta_1}^{(i)}, C_{\theta_2}^{(i)}) - \log(\frac{1}{b} \sum_{i=1}^{b} e^{T_\omega(C_{\theta_1}^{(i)}, \widehat{C_{\theta_2}^{(i)}})}), \tag{7}$$

where T_ω is the statistical network with parameters ω. To compute the lower-bound $\widehat{\mathcal{I}}_\omega$, we randomly select b training images. With the forward pass of the network, we directly get b pairs of features from two column structures as the joint distribution $(C_{\theta_1}, C_{\theta_2}) \sim \mathbb{P}_{C_{\theta_1} C_{\theta_2}}$. At the same time, we randomly disrupt the order of C_{θ_2} in $(C_{\theta_1}, C_{\theta_2}) \sim \mathbb{P}_{C_{\theta_1} C_{\theta_2}}$ to get b pairs of features as the product of the marginal distribution $(C_{\theta_1}, \widehat{C_{\theta_2}}) \sim \mathbb{P}_{C_{\theta_1}} \otimes \mathbb{P}_{C_{\theta_2}}$. Then we input these features to the statistical network T_ω. By calculating the b outputs of the statistical network as Eqn. 7, we can get a lower-bound for the mutual information estimation. Here we use moving average to get the gradient of Eqn. 7. By maximizing this lower-bound, we can approximately obtain the real mutual information. More details of the mutual information estimation are provided in Alg. 1.

Without loss of generality, the statistical network T_ω can be designed as any classifier networks according to the different multi-column networks. We have tested McML on three multi-column networks (i.e., MCNN [69], CSRNet [30] and ic-CNN [44]). The statistical networks for these baselines are described in Sec. 3.5.

3.4 Mutual Learning Scheme

Our proposed McML is a mutual learning scheme. For the sake of simplicity, we present the case of two columns as an example. As shown in Alg. 2, we alternately optimize two columns in each mini-batch until convergence. In each learning iteration, we randomly sample b training images. Before optimizing column θ_1, the mutual information is first estimated as prior knowledge to guide the parameter update. With forward of the network, the features of two column structures are sampled to update the statistical network T_ω and estimate the mutual information $\widehat{\mathcal{I}}_\omega$ as Alg. 1. With the guidance of the mutual information, our McML can supervise column θ_1 to learn as much as possible different scale features from column θ_2. It is noted that we have fixed parameters of other columns (i.e., θ_2) and statistical network (T_ω), and only update column θ_1. Since the size of input images are different, we have to update column structure on each image. After back-propagation of a total of b images, the column θ_2 will be optimized in similar steps.

Algorithm 2 Mutual Learning Scheme

```
Input: Training set $X$, Ground truth $Y$.
$\theta_1$, $\theta_2$ and $\omega \leftarrow$ initialize network parameters;
repeat
    Randomly sampled $b$ images from $X$;
    Estimate mutual information $\widehat{\mathcal{I}}_\omega$ and update statistical network $T_\omega$ as Alg. 1;
    Update column $\theta_1$ as Eqn. 3 on each image,
    $\theta_1 \leftarrow \theta_1 + \frac{\partial L_{\theta_1}}{\partial \theta_1}$;
    Estimate mutual information $\widehat{\mathcal{I}}_\omega$ and update statistical network $T_\omega$ as Alg. 1;
    Update column $\theta_2$ as Eqn. 4 on each image,
    $\theta_2 \leftarrow \theta_2 + \frac{\partial L_{\theta_2}}{\partial \theta_2}$;
until Convergence
```

It is noted that our McML can be naturally extended to multi-columns architectures. For the case of $K > 2$, the loss function of a column θ_k is computed as,

$$L_{\theta_k} = L_2(\text{Conv}(F_{\theta_1 \circ \theta_2, .., \theta_K}(X)), Y) + \frac{\alpha}{K-1} \sum_{l=1, k \neq l}^{K} \widehat{\mathcal{I}}_\omega(C_{\theta_l}; C_{\theta_k}). \tag{8}$$

Similar to Eqns. 3 and 4, where $F_{\theta_1 \circ \theta_2, .., \theta_K}$ means the ensemble (i.e., concanation) of features from the last convolutional layers at multi-columns. C_{θ_*} is the features from different convolutional layers at each column. α is a weight to trade off two losses. At this point, we only need to add more steps to estimate mutual information of multi-columns. Once the mutual information is obtained, multi-column structures are still alternately optimized until convergence.

3.5 Network Architectures

We employ McML to improve three state-of-the-art networks, including MCNN [69], CSRNet [30], and ic-CNN [44]. Table 2 shows the neural architecture of statistical networks. To better understand the details, Figure 3 gives a real example of two columns in MCNN [69]. With sharing the parameters, no matter how many columns are adopted, each multi-column network only needs one single statistical network. The inputs of statistical networks are the features from different layers. We use convolutional layers with one output channel to reduce the feature dimension. Since training images have different size and inspired by the previous work [14], one spatial pyramid pooling (SPP) layer is applied to reshape the features from the last convolutional layer into a fixed dimension.

Table 2: The structure of statistical network. The convolutional, spatial pyramid pooling, and fully connected layers are denoted as "Conv (kernel size)-(number of channels)-(stride)", "SPP (size of outputs)", and "FC (size of outputs)".

MCNN [69]	**CSRNet** [30]	**ic-CNN** [44]
Conv 5-1-2	Conv 3-1-1	Conv 3-1-1
Conv 3-1-2	Conv 3-1-1	Conv 3-1-1
Conv 3-1-1	Conv 3-1-1	Conv 3-1-1
Conv 3-1-1	Conv 3-1-1	Conv 3-1-1
SPP 256	Conv 3-1-1	Conv 3-1-1
FC 128	Conv 3-1-1	SPP 256
FC 1	SPP 256	FC 128
	FC 128	FC 1
	FC 1	

Table 3: Performance of ablation studies. Comparison of Org. (Original Baseline), MLS (Mutual Learning Scheme), MIE (Mutual Information Estimation), and McML (Multi-column Mutual Learning) on four crowd counting datasets.

	ShanghaiTech A [69]		ShanghaiTech B [69]		UCF_CC_50 [24]		UCSD [65]		WorldExpo'10 [8]
Method	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
MCNN [69]	110.2	173.2	26.4	41.3	377.6	509.1	1.07	1.35	11.6
MCNN+MLS	105.2	160.3	22.2	34.2	332.8	425.3	1.04	1.35	10.8
MCNN+MIE	106.7	160.5	25.4	35.6	338.6	447.4	1.12	1.47	11.0
MCNN+McML	101.5	157.7	19.8	33.9	311.0	402.4	1.03	1.24	10.2
CSRNet [30]	68.2	115.0	10.6	16.0	266.1	397.5	1.16	1.47	8.6
CSRNet+MLS	64.2	109.3	9.9	12.3	254.2	376.3	**1.00**	1.31	8.4
CSRNet+MIE	65.6	111.0	9.3	12.8	264.9	387.1	1.06	1.40	8.3
CSRNet+McML	**59.1**	**104.3**	**8.1**	**10.6**	246.1	367.7	1.01	1.27	**8.0**
ic-CNN [44]	68.5	116.2	10.7	16.0	260.9	365.5	1.14	1.43	10.3
ic-CNN+MLS	67.4	112.8	10.3	14.6	248.4	364.3	1.02	1.28	9.7
ic-CNN+MIE	66.3	111.8	11.3	15.1	255.3	368.2	1.06	1.34	9.8
ic-CNN+McML	63.8	110.5	10.1	13.9	**242.9**	**357.0**	**1.00**	**1.20**	8.5

Finally, two fully connected layers are employed as a classifier. Similar to [2], Leaky-ReLU [37] is used as the activation function for all convolutional layers, and no activation function for other layers.

Specifically, MCNN adopts 3 column structures. Each column contains 4 convolutional layers. Intuitively, the statistical network of MCNN uses 4 convolutional layers to embed the features as Figure 3. CSRNet is a single column network. The first 10 convolutional layers are from pre-trained VGG-16 [54]. The last 6 dilated convolutional layers are utilized to estimate the crowd counts. The original version has 4 configurations for 6 dilated convolutional layers (i.e., different dilation rates). Here we treat 4 configurations as 4 different columns. Similarly, as shown in Table 2, the statistical network of CSRNet utilizes 6 convolutional layers to embed the features for 6 dilated convolutional layers in each column. ic-CNN contains two columns (i.e., Low Resolution (LR) and High Resolution (HR) columns). LR contains 11 convolutional layers and 2 max-pooling layers, and HR has 10 convolutional layers with 2 max-pooling layers and 2 deconvolutional layers. As Table 2 shows, the statistical network of ic-CNN uses 5 convolutional layers to embed features from corresponding 5 convolutional layers after the second max pooling layer at both columns.

4 EXPERIMENT

4.1 Experiment Settings

Datasets. To evaluate the effectiveness of our McML, we conduct experiments on four crowd counting datasets, i.e., ShanghaiTech [69], UCF_CC_50 [24], UCSD [8], and WorldExpo'10 [65]. Specifically, ShanghaiTech dataset consists of two parts: Part_A and Part_B. Part_A is collected from the internet and usually has very high crowd density. Part_B is from busy streets and has a relatively sparse crowd density. UCF_CC_50 is mainly collected from Flickr and contains images of extremely dense crowds. UCSD and WorldExpo'10 are both collected from actual surveillance cameras and have low resolution and sparse crowd density. More details of datasets split are illustrated in supplementary material.

Learning Settings. We use our McML to improve MCNN, CSRNet, and ic-CNN. For MCNN, the network is initiated by a Gaussian distribution with a mean of 0 and a standard deviation of 0.01. Adam optimizer [27] with a learning rate of 1e-5 is used to train three columns. For CSRNet, the first 10 convolutional layers are fine-tuned from the pre-trained VGG-16 [54]. The other layers are initiated in the same way as MCNN. We use Stochastic gradient descent (SGD) with a fixed learning rate of 1e-6 to finetune four columns. For ic-CNN, input features from Low-resolution column to High-resolution column are neglected. The SGD with the learning rate of 1e-4 is used to train two columns. The learning settings of the statistical network for all baselines are the same. The number of samples b is 75. Moving average is used to evaluate gradient bias. Adam optimizer with a learning rate of $1e-4$ is used to optimize the statistical network. More details of ground truth generation and data augmentation are illustrated in supplement materials.

Evaluation Details. Following previous works [30, 44, 69], we use mean absolute error (MAE) and mean square error (MSE) to evaluate the performance:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Z_i - Z_i^{gt}\right|, \quad MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Z_i - Z_i^{gt}\right)^2}, \tag{9}$$

where Z_i is the estimated crowd count and Z_i^{gt} is the ground truth count of the i-th image. N is the number of test images. The MAE indicates the accuracy of the estimation, while the MSE indicates the robustness.

4.2 Ablation Studies

We have conduct extensive ablation studies on our McML.

MIE vs. MLS. We separately investigate the roles of our proposed two improvements, i.e., Mutual Learning Scheme (MLS) and Mutual Information Estimation (MIE). Experimental results are shown in Table 3. Org. is the original baseline, MLS means that we ignore the mutual information estimation (i.e., $\widehat{\mathcal{I}}_{\omega}$) in Eqns. 3 and 4, and MIE indicates that we optimize all columns at the same time (i.e., do not alternately optimize each column). Generally speaking, MLS achieves better performance than all original baselines. After integrated MIE, there is a noticeable improvement. It fully demonstrates the effectiveness of our method.

Statistical Network. We intend to compare different statistical networks. We have modified the proposed statistical network as follows: 1) Only-1-Conv means only keep the last convolutional layer. 2) Last-3-Conv denotes to preserve the last three convolutional layers. 3) First-3-Conv indicates to retain the first three convolutional layers. 4) FC-3 (64) means to add one fully connected (FC) layer with 64 outputs between the original two FC layers. 5) FC-1 (64)

Table 4: Comparison with state-of-the-art methods on ShanghaiTech [69], UCF_CC_50 [24] and UCSD [65] datasets.

			ShanghaiTech A [69]		ShanghaiTech B [69]		UCF_CC_50 [24]		UCSD [65]	
Method	**Venue & Year**		**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
Idrees et al. [24]	CVPR	2013	-	-	-	-	419.5	541.6	-	-
Zhang et al. [65]	CVPR	2015	181.8	277.7	32.0	49.8	467.0	498.5	1.60	3.31
CCNN [39]	ECCV	2016	-	-	-	-	-	-	1.51	-
Hydra-2s [39]	ECCV	2016	-	-	-	-	333.7	425.3	-	-
C-MTL [55]	AVSS	2017	101.3	152.4	20.0	31.1	322.8	397.9	-	-
SwitchCNN [50]	CVPR	2017	90.4	135.0	21.6	33.4	318.1	439.2	1.62	2.10
CP-CNN [56]	ICCV	2017	73.6	106.4	20.1	30.1	295.8	**320.9**	-	-
Huang at al. [23]	TIP	2018	-	-	20.2	35.6	409.5	563.7	**1.00**	1.40
SaCNN [66]	WACV	2018	86.8	139.2	16.2	25.8	314.9	424.8	-	-
ACSCP [51]	CVPR	2018	75.7	**102.7**	17.2	27.4	291.0	404.6	-	-
IG-CNN [49]	CVPR	2018	72.5	118.2	13.6	21.1	291.4	349.4	-	-
Deep-NCL [53]	CVPR	2018	73.5	112.3	18.7	26.0	288.4	404.7	-	-
MCNN [69]	CVPR	2016	110.2	173.2	26.4	41.3	377.6	509.1	1.07	1.35
CSRNet [30]	CVPR	2018	68.2	115.0	10.6	16.0	266.1	397.5	1.16	1.47
ic-CNN [44]	ECCV	2018	68.5	116.2	10.7	16.0	260.9	365.5	1.14	1.43
MCNN+McML	-	-	101.5	157.7	19.8	33.9	311.0	402.4	1.03	1.24
CSRNet+McML	-	-	**59.1**	104.3	**8.1**	**10.6**	246.1	367.7	1.01	1.27
ic-CNN+McML	-	-	63.8	110.5	10.1	13.9	**242.9**	357.0	**1.00**	**1.20**

indicates to reduce the outputs of the first FC layer into 64. 6) FC-1 (256) states to increase the outputs of the first FC layer into 256. Comparison results are illustrated in Table 6. In general, different statistical networks have no significant difference in performance. Even using only one convolutional layer, our proposed training strategy still obviously improve the original baseline. These results fully demonstrate the robustness of our method.

The number of samples b. We study the effect of the number of samples b. As shown in Figure 4, we obverse that with the number of b increases, the performance first increases and then decreases. Typically, when b is too small, because of the estimated mutual information has a severe bias, our method intuitively gets poor performance. In contrast, when b is too large, although the mutual information has been accurately estimated, the performance of our model is still severely affected since the iterations of the mutual learning scheme are inadequate. Based on that we use a binary search to find the best value of b. After extensive cross-validation, b is set to 75 for all baselines.

The weight of α. We have verified the impact of the weight of α. To get a more accurate setting, we perform a grid search with the step of 0.1. The best values of α for different datasets are illustated in Table 5. Since ShanghaiTech Part A and UCF_CC_50 have more substantial scale changes, they have a larger α than other datasets. We assume that the weight of α positively correlates to the degree of scale changes.

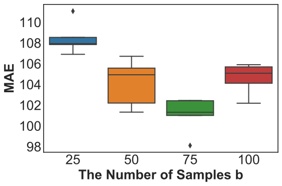

Figure 4: Effects of samples b.

Datasets	α
ShanghaiTech A	0.3
ShanghaiTech B	0.2
UCF_CC_50	0.4
UCSD	0.1
WorldExpo'10	0.2

Table 5: The values of α.

Table 6: Ablation studies of statistical networks on ShanghaiTech Part A dataset [69].

	MCNN [69]		CSRNet [30]		ic-CNN [44]	
Structures	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
Only-1-Conv	104.2	160.8	61.7	106.9	66.2	114.1
Last-3-Conv	103.5	160.1	61.1	106.8	65.5	113.3
First-3-Conv	103.2	159.7	60.7	106.1	64.8	113.2
FC-3 (64)	101.6	157.8	59.3	**104.3**	63.9	**110.5**
FC-1 (64)	102.0	158.2	59.8	105.1	64.5	111.3
FC-1 (256)	102.2	158.4	59.7	104.8	63.9	111.0
Ours (Table 2)	**101.5**	**157.7**	**59.1**	**104.3**	**63.8**	**110.5**

4.3 Comparisons with State-of-the-art

We demonstrate the efficiency of our McML on four challenging crowd counting datasets. Tables 4 and 7 show the comparison with the other state-of-the-art methods. We observe that McML can significantly improve three baselines (i.e., MCNN, CSRNet, and ic-CNN) on all datasets. Notably, after using McML, the optimized CSRNet and ic-CNN also obviously outperform the other state-of-the-art approaches. It fully demonstrates that our method can not only be applied to any multi-column network but also works on both dense and sparse crowd scenes. Additionally, although ic-CNN also propose an alternate training process, our McML can still achieve better results than the original ic-CNN. It means that our McML is more effective than ic-CNN.

For ShanghaiTech dataset, McML significantly boosts MCNN, CSRNet, and ic-CNN with relative MAE improvements of 7.9%, 13.3% and 6.9% on Part A, and 25.0%, 23.6% and 5.6% on Part B, respectively. Similarly, for UCF_CC_50 dataset, McML provides the relative MAE improvements of 17.6%, 7.5%, and 6.9% for three baselines. These results clearly state McML can not only handle dense-crowd scenes but also work for small datasets. On the other hand, experimental results of UCSD dataset show McML can improve the accuracy (i.e., lower MAE) and gain the robustness (i.e., lower MSE). This result states the effectiveness of McML on the sparse-crowd scene. Additionally, on WorldExpo'10 dataset, although our proposed McML does not utilize perspective maps, they still achieve

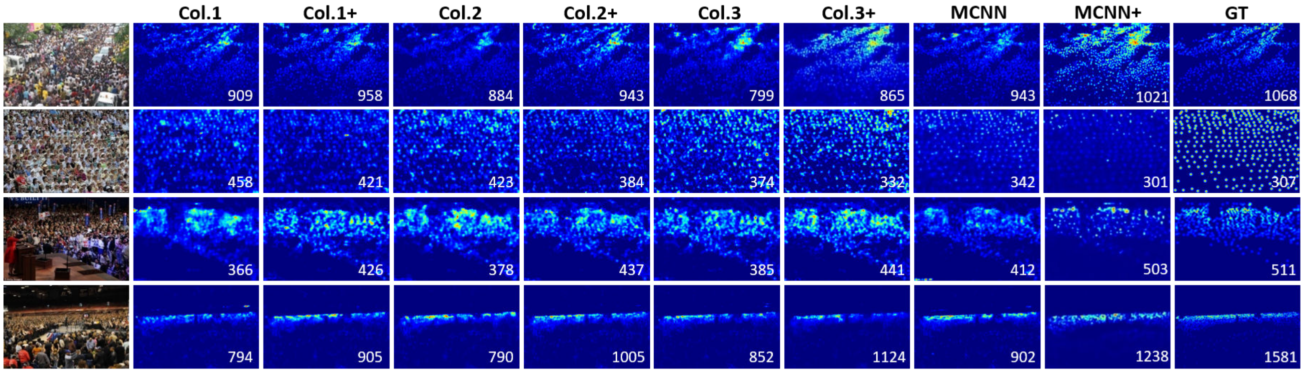

Figure 5: Comparison of estimated density maps between MCNN [69] and McML. '+' indicates employing McML on the original MCNN.

Table 7: Comparison with state-of-the-art methods on World-Expo'10 [8] dataset. Only MAE is computed for each scene and then averaged to evaluate the overall performance.

Method	S1	S2	S3	S4	S5	Avg.
Zhang et al. [65]	9.8	14.1	14.3	22.2	3.7	12.9
Huang et al. [23]	4.1	21.7	11.9	11.0	3.5	10.5
Switch-CNN [50]	4.4	15.7	10.0	11.0	5.9	9.4
SaCNN [66]	**2.6**	13.5	10.6	12.5	**3.3**	8.5
CP-CNN [56]	2.9	14.7	10.5	10.4	5.8	8.9
MCNN [69]	3.4	20.6	12.9	13.0	8.1	11.6
CSRNet [30]	2.9	11.5	8.6	16.6	3.4	8.6
ic-CNN [44]	17.0	12.3	9.2	8.1	4.7	10.3
MCNN+McML	3.4	15.2	14.6	12.7	5.2	10.2
CSRNet+McML	2.8	**11.2**	9.0	13.5	3.5	**8.0**
ic-CNN+McML	10.7	**11.2**	**8.2**	**8.0**	4.5	8.5

better results than other state-of-the-art methods that use perspective maps.

4.4 Why does McML Work

We attempt to give more insights to show why our McML works. The statistical analysis is illustrated in Table 8. Compared with the results without McML (in Table 1), we observe that McML can significantly reduce Maximal Information Coefficient (MIC) and Structural SIMilarity (SSIM) between columns. It denotes that our method can indeed reduce the redundant parameters of columns and avoid overfitting. On the other hand, McML can efficiently improve MIC and SSIM between the ensemble of all columns and the ground truth. It means that our method can guide multi-column structures to learn different scale features and improve the accuracy of crowd counting.

To further verify that our McML can indeed guide multi-column networks to learn different scales, we showcase the generated density maps from different columns of MCNN in Figure 5. These four examples typically contain different crowd densities, occlusions, and scale changes. We observe that estimated density maps of McML have more different salient areas than the original MCNN. It means that our method can indeed guide multi-column structures to focus on different scale information (i.e., different people/head sizes). It is noted that the ground truth itself is generated with center points of pedestrians' heads, which inherently contains inaccurate information. Thus the result of our method is still unable to produce the same density map to the ground truth.

Table 8: The result analysis of our proposed McML. The values in the table are the average of all columns. Col.↔Col. is the result between different columns. Col.↔GT is the result between the ensemble of all columns and the ground truth.

	Col.↔ Col.		Col.↔ GT	
Method	MIC	SSIM	MIC	SSIM
ShanghaiTech Part A [69]				
MCNN+McML	0.74	0.61	0.68	0.70
CSRNet+McML	0.77	0.70	0.82	0.82
ic-CNN+McML	0.76	0.55	0.80	0.76
UCF_CC_50 [24]				
MCNN+McML	0.69	0.48	0.79	0.47
CSRNet+McML	0.73	0.60	0.75	0.61
ic-CNN+McML	0.75	0.60	0.72	0.64

5 CONCLUSION

In this paper, we propose a novel learning strategy called Multi-column Mutual learning (McML) for crowd counting, which can improve the scale invariance of feature learning and reduce parameter redundancy to avoid overfitting. It could be applied to all existing CNN-based multi-column networks and is end-to-end trainable. Experiments on four challenging datasets fully demonstrate that it can significantly improve all baselines and outperforms the other state-of-the-art methods. In summary, this work provides the elegant views of effectively using multi-column architectures to improve the scale invariance. In future work, we will study how to handle different image scales and resolutions in the ground truth generation.

6 ACKNOWLEDGEMENTS

This research was supported in part through the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00340, National Natural Science Foundation of China (Grant No: 61772436), Foundation for Department of Transportation of Henan Province, China (2019J-2-2), Sichuan Science and Technology Innovation Seedling Fund (2017RZ0015), China Scholarship Council (Grant No. 201707000083) and Cultivation Program for the Excellent Doctoral Dissertation of Southwest Jiaotong University (Grant No. D-YB 201707).

REFERENCES

[1] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *Advances in neural information processing systems*. 2654–2662.

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. 2018. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062* (2018).

[3] Lokesh Boominathan, Srinivas S. S. Kruthiventi, and R. Venkatesh Babu. 2016. CrowdNet: A Deep Convolutional Network for Dense Crowd Counting. In *Proceedings of ACM International Conference on Multimedia*. 640–644.

[4] Gabriel J Brostow and Roberto Cipolla. 2006. Unsupervised bayesian detection of independent motion in crowds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Vol. 1. 594–601.

[5] Gavin Brown, Jeremy L Wyatt, and Peter Tiňo. 2005. Managing diversity in regression ensembles. *Journal of Machine Learning Research* 6, Sep (2005), 1621–1650.

[6] Atul J Butte and Isaac S Kohane. 1999. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Biocomputing*. 418–429.

[7] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. 2018. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *Proceedings of European Conference on Computer Vision*. 757–773.

[8] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. 2008. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–7.

[9] Antoni B Chan and Nuno Vasconcelos. 2012. Counting people with low-level features and Bayesian regression. *IEEE Transactions on Image Processing* 21, 4 (2012), 2160–2177.

[10] Ke Chen, Shaogang Gong, Tao Xiang, and Chen Change Loy. 2013. Cumulative attribute space for age and crowd density estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2467–2474.

[11] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. 2012. Feature Mining for Localised Crowd Counting. In *Proceedings of British Machine Vision Conference*. 1–11.

[12] Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, and Alexander Hauptmann. 2019. Learning Spatial Awareness to Improve Crowd Counting. In *Proceedings of IEEE International Conference on Computer Vision*.

[13] Zhi-Qi Cheng, Xiao Wu, Siyu Huang, Jun-Xiu Li, Alexander G. Hauptmann, and Qiang Peng. 2018. Learning to Transfer: Generalizable Attribute Learning with Multitask Neural Model Search. In *Proceedings of the 26th ACM International Conference on Multimedia*.

[14] Zhi-Qi Cheng, Xiao Wu, Yang Liu, and Xian-Sheng Hua. 2017. Video2shop: Exact matching clothes in videos to online shopping images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4048–4056.

[15] Zhi-Qi Cheng, Hao Zhang, Xiao Wu, and Chong-Wah Ngo. 2017. On the selection of anchors and targets for video hyperlinking. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. 287–293.

[16] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Vol. 1. 886–893.

[17] Piotr Dollár, Boris Babenko, Serge Belongie, Pietro Perona, and Zhuowen Tu. 2008. Multiple component learning for object detection. In *Proceedings of European Conference on Computer Vision*. 211–224.

[18] Monroe D Donsker and SR Srinivasa Varadhan. 1983. Asymptotic evaluation of certain Markov process expectations for large time. IV. *Communications on Pure and Applied Mathematics* 36, 2 (1983), 183–212.

[19] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research* 15, 1 (2014), 3133–3181.

[20] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).

[21] Siyu Huang, Xi Li, Zhiqi Cheng, Zhongfei Zhang, and Alexander G. Hauptmann. 2018. Stacked Pooling: Improving Crowd Counting by Boosting Scale Invariance. *CoRR* abs/1808.07456 (2018).

[22] Siyu Huang, Xi Li, Zhi-Qi Cheng, Zhongfei Zhang, and Alexander Hauptmann. 2018. GNAS: A Greedy Neural Architecture Search Method for Multi-Attribute Learning. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2049–2057.

[23] Siyu Huang, Xi Li, Zhongfei Zhang, Fei Wu, Shenghua Gao, Rongrong Ji, and Junwei Han. 2018. Body Structure Aware Deep Crowd Counting. *IEEE Trans. Image Processing* 27, 3 (2018), 1049–1059.

[24] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. 2013. Multi-source Multi-scale Counting in Extremely Dense Crowd Images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2547–2554.

[25] Haroon Idrees, Khurram Soomro, and Mubarak Shah. 2015. Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning. *IEEE transactions on pattern analysis and machine intelligence* 37, 10 (2015), 1986–1998.

[26] Di Kang and Antoni B. Chan. 2018. Crowd Counting by Adaptively Fusing Predictions from an Image Pyramid. In *Proceedings of British Machine Vision Conference*. 89.

[27] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[28] Nojun Kwak and Chong-Ho Choi. 2002. Input feature selection by mutual information based on Parzen window. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 12 (2002), 1667–1671.

[29] Victor S. Lempitsky and Andrew Zisserman. 2010. Learning To Count Objects in Images. In *Proceedings of Conference on Neural Information Processing Systems*. 1324–1332.

[30] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1091–1100.

[31] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. 2001. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 31, 6 (2001), 645–654.

[32] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann. 2018. DecideNet: Counting Varying Density Crowds Through Attention Guided Detection and Density Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 5197–5206.

[33] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. 2018. ADCrowdNet: An Attention-injective Deformable Convolutional Network for Crowd Understanding. *CoRR* abs/1811.11968 (2018).

[34] Weizhe Liu, Krzysztof Lis, Mathieu Salzmann, and Pascal Fua. 2018. Geometric and Physical Constraints for Head Plane Crowd Density Estimation in Videos. *CoRR* abs/1803.08805 (2018).

[35] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. 2018. Context-Aware Crowd Counting. *CoRR* abs/1811.10452 (2018).

[36] Zheng Ma and Antoni B. Chan. 2013. Crossing the Line: Crowd Counting by Integer Programming with Local Features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2539–2546.

[37] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

[38] Frederik Maes, Andre Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. 1997. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging* 16, 2 (1997), 187–198.

[39] Daniel Oñoro-Rubio and Roberto Javier López-Sastre. 2016. Towards Perspective-Free Object Counting with Deep Learning. In *Proceedings of European Conference on Computer Vision*. 615–629.

[40] Liam Paninski. 2003. Estimation of entropy and mutual information. *Neural computation* 15, 6 (2003), 1191–1253.

[41] Nikos Paragios and Visvanathan Ramesh. 2001. A MRF-based approach for real-time subway monitoring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Vol. 1. I–I.

[42] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 8 (2005), 1226–1238.

[43] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. 2015. COUNT Forest: CO-Voting Uncertain Number of Targets Using Random Forest for Crowd Density Estimation. In *Proceedings of International Conference on Computer Vision*. 3253–3261.

[44] Viresh Ranjan, Hieu Le, and Minh Hoai. 2018. Iterative Crowd Counting. In *Proceedings of European Conference on Computer Vision*. 278–293.

[45] Carlo S Regazzoni and Alessandra Tesei. 1996. Distributed data fusion for real-time crowding estimation. *Signal Processing* 53, 1 (1996), 47–63.

[46] Ye Ren, Le Zhang, and Ponnuthurai N Suganthan. 2016. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational intelligence magazine* 11, 1 (2016), 41–53.

[47] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. 2009. Crowd counting using multiple local features. In *Digital Image Computing: Techniques and Applications*. 81–88.

[48] Deepak Babu Sam and R. Venkatesh Babu. 2018. Top-Down Feedback for Crowd Counting Convolutional Neural Network. In *Proceedings of Conference on Artificial Intelligence*. 7323–7330.

[49] Deepak Babu Sam, Neeraj N. Sajjan, R. Venkatesh Babu, and Mukundhan Srinivasan. 2018. Divide and Grow: Capturing Huge Diversity in Crowd Images With Incrementally Growing CNN. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 3618–3626.

[50] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. 2017. Switching Convolutional Neural Network for Crowd Counting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 4031–4039.

[51] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. 2018. Crowd Counting via Adversarial Cross-Scale Consistency Pursuit. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 5245–5254.

[52] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. 2018. Perspective-Aware CNN For Crowd Counting. *CoRR* abs/1807.01989 (2018).

[53] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. 2018. Crowd Counting With Deep Negative Correlation Learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 5382–5390.

[54] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[55] Vishwanath A. Sindagi and Vishal M. Patel. 2017. CNN-Based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *Proceedings of International Conference on Advanced Video and Signal Based Surveillance*. 1–6.

[56] Vishwanath A. Sindagi and Vishal M. Patel. 2017. Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs. In *Proceedings of International Conference on Computer Vision*. 1879–1888.

[57] Yukun Tian, Yimei Lei, Junping Zhang, and James Z. Wang. 2018. PaDNet: Pan-Density Crowd Counting. *CoRR* abs/1811.02805 (2018).

[58] Paul Viola and Michael Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Vol. 1. I–I.

[59] Paul Viola, Michael J Jones, and Daniel Snow. 2005. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision* 63, 2 (2005), 153–161.

[60] Elad Walach and Lior Wolf. 2016. Learning to Count with CNN Boosting. In *Proceedings of European Conference on Computer Vision*. 660–676.

[61] Meng Wang and Xiaogang Wang. 2011. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3401–3408.

[62] Ze Wang, Zehao Xiao, Kai Xie, Qiang Qiu, Xiantong Zhen, and Xianbin Cao. 2018. In Defense of Single-column Networks for Crowd Counting. In *Proceedings of British Machine Vision Conference*. 78.

[63] Xingjiao Wu, Yingbin Zheng, Hao Ye, Wenxin Hu, Jing Yang, and Liang He. 2018. Adaptive Scenario Discovery for Crowd Counting. *CoRR* abs/1812.02393 (2018).

[64] Lingke Zeng, Xiangmin Xu, Bolun Cai, Suo Qiu, and Tong Zhang. 2017. Multi-scale convolutional neural networks for crowd counting. In *Proceedings of International Conference on Image Processing*. 465–469.

[65] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 833–841.

[66] Lu Zhang, Miaojing Shi, and Qiaobo Chen. 2018. Crowd Counting via Scale-Adaptive Convolutional Neural Network. In *Proceedings of Winter Conference on Applications of Computer Vision*. 1113–1121.

[67] Le Zhang and Ponnuthurai Nagaratnam Suganthan. 2017. Benchmarking ensemble classifiers with novel co-trained kernel ridge regression and random vector functional link ensembles [research frontier]. *IEEE Computational Intelligence Magazine* 12, 4 (2017), 61–72.

[68] Youmei Zhang, Chunluan Zhou, Faliang Chang, and Alex C. Kot. 2018. Attention to Head Locations for Crowd Counting. *CoRR* abs/1806.10287 (2018).

[69] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 589–597.

[70] Tao Zhao, Ram Nevatia, and Bo Wu. 2008. Segmentation and tracking of multiple humans in crowded environments. *IEEE transactions on pattern analysis and machine intelligence* 30, 7 (2008), 1198–1211.