J. Vis. Commun. Image R. 38 (2016) 530–539

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Dense crowd counting from still images with convolutional neural networks ☆

Yaocong Hu, Huan Chang, Fudong Nian, Yan Wang, Teng Li *

Anhui University, No. 111 Jiulong RD, Hefei 230061, China

ARTICLE INFO

Article history:
Received 1 July 2015
Revised 29 February 2016
Accepted 20 March 2016
Available online 29 March 2016

Keywords:
Crowd counting
Convolutional neural networks
Feature learning
Regression

ABSTRACT

For reasons of public security, modeling large crowd distributions for counting or density estimation has attracted significant research interests in recent years. Existing crowd counting algorithms rely on predefined features and regression to estimate the crowd size. However, most of them are constrained by such limitations: (1) they can handle crowds with a few tens individuals, but for crowds of hundreds or thousands, they can only be used to estimate the crowd density rather than the crowd count; (2) they usually rely on temporal sequence in crowd videos which is not applicable to still images. Addressing these problems, in this paper, we investigate the use of a deep-learning approach to estimate the number of individuals presented in a mid-level or high-level crowd visible in a single image. Firstly, a ConvNet structure is used to extract crowd features. Then two supervisory signals, i.e., crowd count and crowd density, are employed to learn crowd features and estimate the specific counting. We test our approach on a dataset containing 107 crowd images with 45,000 annotated humans inside, and each with head counts ranging from 58 to 2201. The efficacy of the proposed approach is demonstrated in extensive experiments by quantifying the counting performance through multiple evaluation criteria.

1. Introduction

Crowd counting aims at calculating the number of individuals presented in images and videos. It is an important topic with many potential practical applications, such as video surveillance (e.g., dense crowd anomaly detection, crowd management in a specific region), safety management (e.g., recording the number of people entering or leaving some regions) and web multimedia (e.g., estimating crowd size of tweet image shoot in crowd scenic spot). However, due to problems including crowd variation, occlusion, clutter and low resolution, visual analysis based crowd counting and density estimation are still very challenging tasks.

The task of crowd counting has been approached from a number of angles, but the techniques share a common framework: crowd feature extraction, followed by crowd counting using object detection or regression model. However, crowds can be various in their distributions and color patterns. And more importantly, a crowd does not have a well-defined shape as a single object does, which makes it difficult for crowd feature extraction. These difficulties cause that existing methods well-suited in pedestrian detection cannot be applicable in detecting human instances in crowd scenes. For example, we apply Deformable Parts Model (DPM) [1] in Fig. 1. Detection results show that this detection-based method is more applicable in the crowd of few tens than in the crowd of more than hundreds.

To address these challenges, some research works [2–4] indicate that the crowd in high density scenes often presents repetitive textural visual effects, namely, the crowd distribution is irregular and nonuniform in large scales, but it presents some regular visual patterns in small scales. Moreover, in derived intensity spaces such as image derivative, or edges, groups of individuals are likely to exhibit an increased level of similarity [3]. For reasons stated above, how to extract features that can well represent the information contained in the crowd is especially vital for the following procedure of this task.

In recent years, with the success of deep learning architectures for visual processing, (e.g., convolutional neural networks (ConvNets)) and availability of image databases with millions of labeled examples (e.g., ImageNet) [5], the state of the art in many different domains are advancing rapidly, including image classification [5,6], object and face detection [7,8], speech recognition [9], bioacoustics [10], etc. Unlike many previous vision approaches using hand-designed features, ConvNets can automatically learn a unique set of features optimized for a given task. Recent researches

☆ This paper has been recommended for acceptance by Dacheng Tao.

* Corresponding author.
E-mail address: liteng@ahu.edu.cn (T. Li).

http://dx.doi.org/10.1016/j.jvcir.2016.03.021
1047-3203/

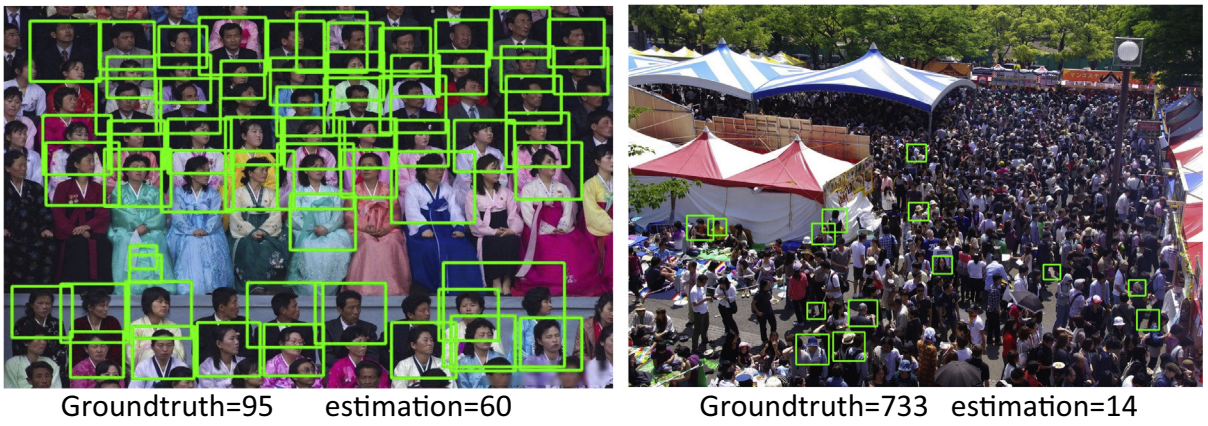


Fig. 1. Some typical examples of human detection.

Fig. 2. Two examples from UCF-CROWD dataset [3]. In both images, people far away from the scene only occupy so few pixels that even a human observer cannot distinguish them from the background.

also have shown that learned features are able to perform better than traditional hand-engineered representations (e.g., Scale Invariant Feature Transform (SIFT) [11], Histogram of Oriented Gradient (HOG) [12], Local Binary Patterns (LBP) [13]) in many domains, especially those where good features have not already been engineered [14].

Inspired by the effective and superior features learned with the deep architecture, this paper develops a simple and general discriminative learning-based framework for the problem of people counting in images. Firstly, a ConvNet structure is used to learn crowd features and then, a feature-count regressor considering two supervisory signals, i.e., crowd count and crowd density, maps the learned feature to the number of people within each local region. As a result, the total crowd estimation is the sum of that in all local regions. The proposed method evades the traditional task of learning to detect and localize individual object instances in mid-level and high-level crowd density images, which is impractical in many cases. Our sole aim is to use feature vectors learned in ConvNets to estimate people count in each local region and we expect the deviation between our estimation and groundtruth can be as small as possible.

In terms of experimental datasets, most of the previous crowd counting algorithms only have been verified on low density crowd datasets, e.g., USCD dataset [15,16] with people count of 11–46, Mall dataset [17] with count of 13–53 individuals and PETS dataset [18] containing 3–40 people per frame. To the best of our knowledge, so far, only Idrees et al. [3] provided their UCF-CROWD dataset containing between 94 and 4543 people per image, and their crowd counting algorithm achieved state-of-the-art performance on these dense crowd images. However, in some extreme dense crowd images of UCF-CROWD dataset, an individual only occupies so few pixels that even a human observer cannot distinguish it from background (as shown in Fig. 2), and such images actually are of no practical use in real world applications. To address this problem, in this paper, we provide our own AHU-CROWD dataset covering different scenarios, with head counts from 58 to 2201 per image, and all individuals visible in our dataset can be well distinguished by human observers. Moreover, in accordance with paper [3,19], images are annotated with dots, which is the natural way to count objects for humans, at least when the number of objects is large. Fig. 3 gives some examples of the counting problems and the dotted annotations we consider in this paper. The main contributions of our study can be concluded into three aspects:

- We propose a deep learning architecture to estimate the people counting in still images.
- We provide our own crowd dataset AHU-CROWD, which consists of 107 crowd images covering different scenes. All 45 K human instances are annotated with dots manually (one dot per person).

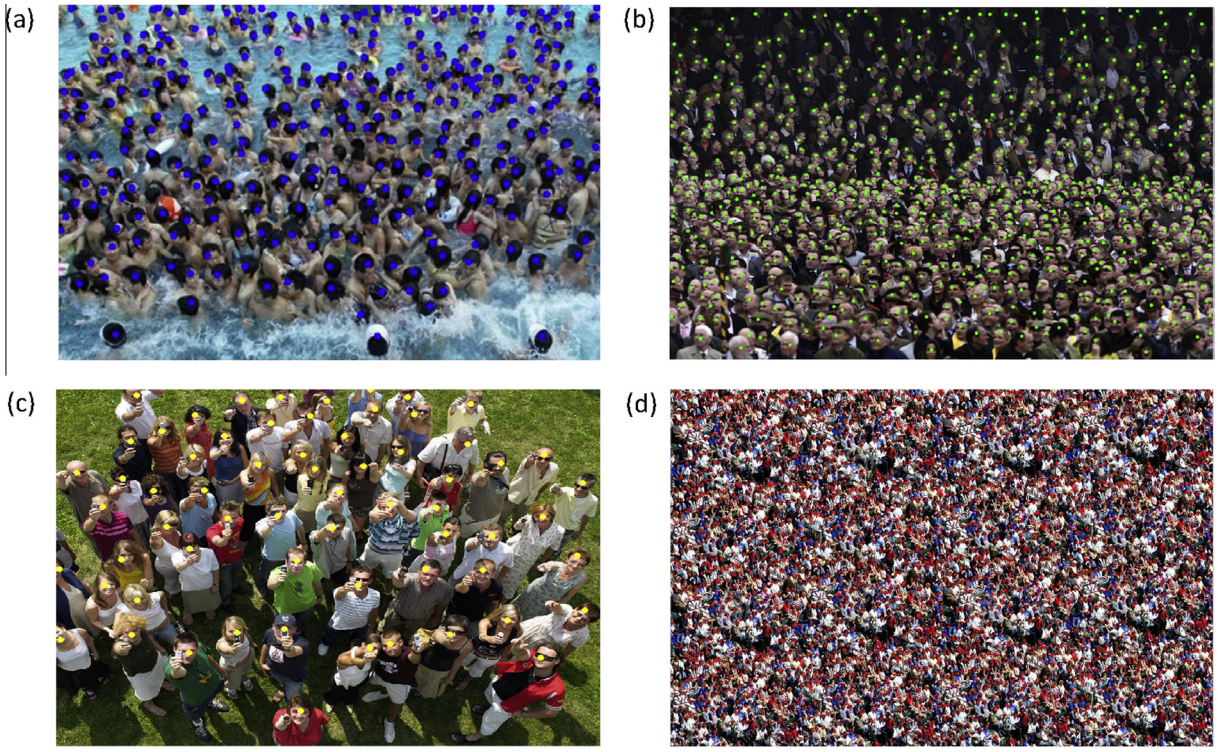


Fig. 3. Some examples from our AHU-CROWD dataset, with dotted annotations.

Table 1
Related mathematical symbols and their representation.

Symbols	Representation
$\mathbf{X}$	Training set
χ	One training sample
x	Training patch
y	Ground truth label
l	Density level label
$\mathbf{f}$	Extracted feature vector
θ_{net}	ConvNet parameters to be learned
θ_{reg}	Crowd counting regression parameters
θ_{reg}	Softmax layer parameters
$Conv$	Feature extraction function
$\mathcal{L}_{cls}$	Crowd density classification loss
$\mathcal{D}$	Crowd counting loss function

Table 2
We use CCM to abbreviate our Crowd ConvNet Model which contains three convolutional layers (conv1–3) and one crowd feature extraction layer (full). The details of each of the convolutional layers are given in three sub-rows: the first row specifies the number of convolution filters and their receptive field sizes as num × size × size; the second row indicates the convolution stride (st) and spatial padding (pad); the third row indicates the max-pooling downsampling factor.

Arch.	conv1	conv2	conv3	full
CCM	16 × 5 × 5 st 1, pad 0 ×2 max pool	32 × 3 × 3 st 1, pad 0 ×2 max pool	64 × 2 × 2 st 1, pad 0 ×2 max pool	100 Crowd -feature

- We test our architecture on our own dataset, UCSD [15] and UCF-CROWD [3] dataset and report comparisons with the state-of-the-art results.

The rest of this paper is organized as follows: we review the previous works about crowd estimation and people counting in Section 2; the proposed method and the overall framework are detailed in Section 3; experiments and the comparisons of results are summarized in Section 4; finally, we conclude this paper in Section 5.

2. Related work

In recent years, several solutions for crowd counting have been proposed. Existing crowd counting approaches roughly fall into two categories: counting by detection and counting by regression.

In counting by detection [20–23], a visual object detector is utilized and people count is estimated through detecting instances of people. e.g., Lin et al. [20] utilized Haar wavelet features combined with perspective transformation to train a human head detector. Wu and Nevatia [21] took advantages of edgelet features to learn a part detector, and it can increase the robust of occlusion in some extent. However, object detection itself is very far from being solved [24], especially for overlapping instances, and it is sensitive to illumination and perspective issues. Typically, the detection process is time-consuming since it involves exhaustive scanning of image space using a pre-trained detector with different scales. Besides, several methods assumed that objects tend to be uniform and disconnected from each other by distinct background color, which is not applicable in real crowd images.

In contrast, counting by regression [17,16,19,25,26] aims to learn a direct mapping between low-level features and people count by learning certain regression functions without segmentation or detection of individuals. Since it is more suitable for complex environments and computationally efficient, this kind of approach is favored by some recent proposed studies. e.g., Chen et al. [17] proposed a framework employing inter-dependent local features from local spatial regions as input and people count from individual regions as multi-dimensional structured output. Chan et al. [25] proposed a solution to estimate the size of inhomogeneous crowds, based on the use of Gaussian process regression and Bayesian

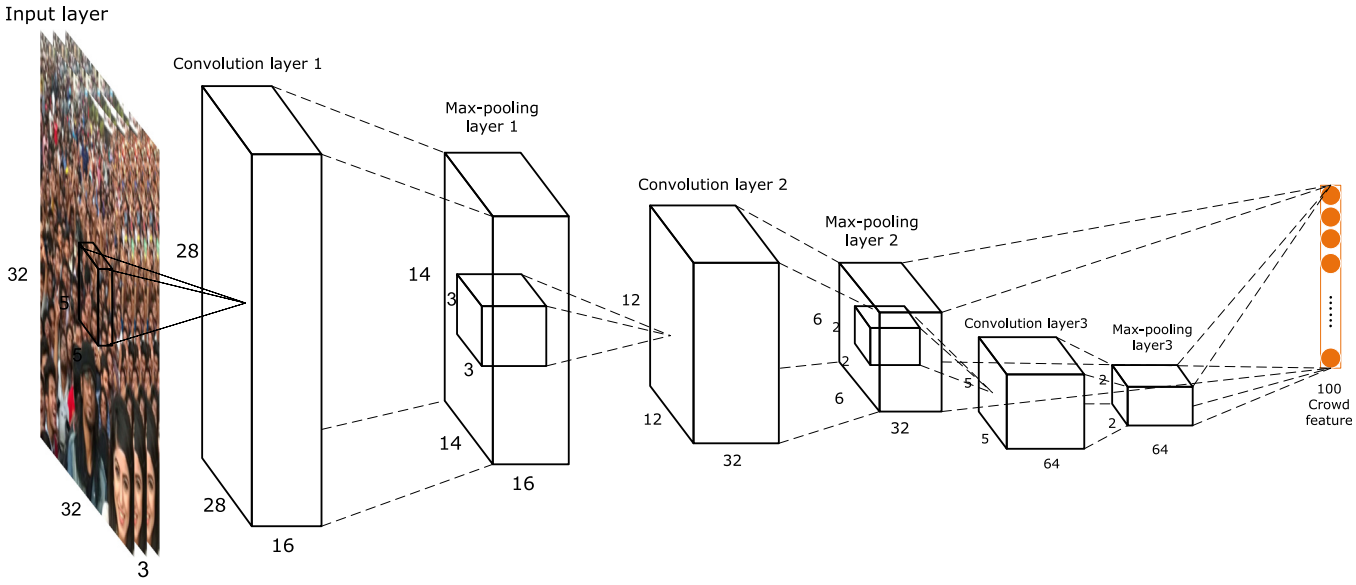


Fig. 4. The ConvNet structure for crowd feature extraction.

Fig. 5. Patches and their associated labels.

Poisson regression, which improves the adaptation to nonlinearities and generalization for small training sets. To handle the noise problem for better regression, some works have also been explored. Tao and Li proposed a probabilistic model for crowd event analysis in [27], a Cauchy Regression method to learn a robust and generality model from noisy big data in [26], and solutions to increase the robustness to noisy labels in [28,29]. These above algorithms, however, are constrained by the difficult choice of the regression function, if not properly selected, regression-based methods will not be robust to outliers and nonlinearities and are prone to over-fitting when the feature spaces are high-dimensional or when there are little training data. Moreover, most of these methods can only estimate the crowd density instead of crowd counting, due to limited learning ability of regression model.

In general, existing methods have following limitations: (1) they can handle crowds of a few tens individuals, for crowds of hundreds or thousands, they are mostly more applicable for crowd density estimation rather than crowd counting; (2) they usually rely on temporal sequence in crowd videos, which is not applicable to still images.

To overcome these limitations, we utilize a deep learning architecture to estimate people count in crowd scenes. Just around the near before, Zhang et al. [30] also proposed a deep learning based crowd density estimation algorithm, which is closely related and coincident with our perspective to this problem. The most significant distinguish with Zhang et al. [30] is that we use the density level rather than density map as density label. The more direct approach will be proven to be effective on extreme high density dataset (UCF-CROWD [3]) in Section 4, and the specific implementations and formulations are introduced in Section 3.

3. Methodology

We first provide a detailed description of our learning framework with learning settings and notations for crowd counting problem, then we introduce the proposed ConvNet architecture specialized for the crowd features learning. The related mathematical symbols and their representation are listed in Table 1.

3.1. Problem formulation

Given a set of N training images $I_1, I_2, \ldots, I_N$, we assume that each training image I_i is divided into $K = p \times q$ image patches,

and it is also assumed that K image patches $\mathbf{x} = \{x^1, x^2, x^3, \ldots, x^K\}$ are trained for each batch of iteration. We define training set $\mathbf{X} = \{\chi^1, \chi^2, \chi^3, \ldots, \chi^K\}$, $\chi^k = (x^k, y^k, l^k)$, where χ^k is the k-th training sample, x^k is its corresponding training patch, y^k is its localized labeled ground truth, and l^k represents its density level. Different density level is judged according to the total number of people in each patch. We first learn a hierarchical nonlinear feature representation of each patch.

$$\mathbf{f}^k = Conv(x^k|\theta_{net}), \tag{1}$$

where $\mathbf{f}^k$ is the extracted feature vector for the k-th patch, $Conv(\cdot|\theta_{net})$ is the feature extraction function defined by the ConvNet, and θ_{net} denotes ConvNet parameters to be learned.

Then, a regression function is selected to map the learned features to the number of people within each patch. As we observed in experiments, single crowd counting signal for learning is prone to over-fitting when training patches are not sufficient or the distribution of person label in each patch is inhomogeneous. To address these problems, we use two supervisory signals, i.e., crowd count and crowd density, to learn crowd features.

The former crowd counting signal is to encourage the regression function mapping crowd features to people count to be more accurate. Formally, given $\mathbf{f}^k$, we apply a linear transformation of the feature representation to approximate the people count at each image patch.

$$\hat{y}^k = \mathcal{F}(\mathbf{f}^k|\theta_{reg}) = \theta_{reg}^T \mathbf{f}^k. \tag{2}$$

The later crowd density signal, aims at regularizing the ConvNet and thus can effectively increase the inter-class variations. The crowd density signal classifies each crowd image patch into one of n different (e.g., $n = 10$) density levels. The parameter n needs to be set optimally according to the training data, and it will be discussed in the experiments section. Crowd density identification can be achieved by following the crowd feature layer with an n-way softmax layer, which outputs a probability distribution over the n classes. The network is trained to minimize the cross-entropy loss. Here, we call it crowd density classification loss, which is denoted as:

$$\mathcal{L}_{cls}(\mathbf{f}^k, l^k, \theta_{cls}) = -\left[\sum_{i=1}^{n} 1\{i = l^k\} \log P(i = l^k|\mathbf{f}^k, \theta_{cls})\right], \tag{3}$$

where $\mathbf{f}^k$ is the k-th crowd feature vector, l^k is its target class and θ_{cls} represents the parameters of softmax layer. $1\{\cdot\}$ is the indicator function. 1{a true statement} = 1, and 1{a false statement} = 0. $P(\cdot|\mathbf{f}^k, \theta_{cls})$ represents the predicted probability distribution.

The crowd counting problem is thus reduced to choosing a right loss function $\mathcal{D}$ and computing the optimal θ_{net}, θ_{cls} and θ_{reg} under the loss:

$$\theta_{net}^*, \theta_{cls}^*, \theta_{reg}^* = \underset{\theta_{net}, \theta_{cls}, \theta_{reg}}{\operatorname{argmin}} \left(\sum_{i=1}^{N} (\lambda \mathcal{L}_{cls} + \mathcal{D}(y, \hat{y})) + \|\theta_{net}\|_2^2 + \|\theta_{cls}\|_2^2 + \|\theta_{reg}\|_2^2\right), \tag{4}$$

where θ_{net} denotes the ConvNet parameters, θ_{cls} denotes the softmax layer parameters, and θ_{reg} represents the crowd counting regression parameters. While in the right equation, $\mathcal{L}_{cls}$ is the crowd density classification loss, $\mathcal{D}(y, \hat{y})$ is the crowd counting loss function, and λ is a hyperparameter which weights the crowd density and counting gradients.

Here, we adopt the above loss function $\mathcal{D}$ based on Maximum Excess over SubArrays (MESA) distance, which was originally proposed in Lempitsky and Zisserman [19] for object counting. We choose MESA distance as our loss distance due to its robustness to noise and turbulence, and its well-suited performance in density learning [19]. This loss involves counting accuracy over multiple sub-regions of the entire image (and not only the entire image itself). Given an image I, the MESA distance $\mathcal{D}_{MESA}$ between the total predicted counts $\hat{y}$ and ground-truth counts y is defined as the largest absolute difference between sums of $\hat{y}^k$ and y^k $(k = 1, 2, \ldots, K)$ over all box subarrays in I.

$$\mathcal{D}_{MESA}(\hat{y}, y) = \max_{box \in I} \left| \sum_{\mathbf{x} \in box} \hat{y}^k - \sum_{\mathbf{x} \in box} y^k \right|. \tag{5}$$

Our goal is to learn the parameters θ_{net} in the feature extraction function $Conv(\cdot)$, while θ_{cls} and θ_{reg} are only parameters introduced to propagate the crowd density and counting signals during training. In the testing stage, only θ_{net} is used for feature extraction, and θ_{reg} is used for crowd counting regression. The parameters are updated by stochastic gradient descent. The crowd density and counting gradients are weighted by a hyperparameter λ. The proposed learning algorithm is summarized in Algorithm 1.

Algorithm 1. The Crowd Counting Learning Algorithm.

Input: training set $\chi = \{x_i, y_i, l_i\}$, initialized parameters θ_{net}, θ_{cls}, and θ_{reg}, hyperparameter λ, learning rate $\eta(t)$, $t \leftarrow 0$, times of iteration N, batch size M.

While $t \neq N$ **do**

$t \leftarrow t + 1$ sample N training samples $\{x_i, y_i, l_i\}$ from χ

$\mathbf{f}^i = Conv(x_i, \theta_{net})$

$\nabla\theta_{cls} = \lambda \cdot \sum_{i=1}^{M} \frac{\partial \mathcal{L}_{cls}(\mathbf{f}^i, l_i, \theta_{cls})}{\partial \theta_{cls}}$

$\nabla\theta_{reg} = \sum_{i=1}^{M} \frac{\partial \mathcal{D}_{MESA}(\mathbf{f}^i, \hat{y}_i, l_i, \theta_{reg})}{\partial \theta_{reg}}$

$\nabla\mathbf{f}^i = \lambda \cdot \frac{\partial \mathcal{L}_{cls}(\mathbf{f}^i, t_i, \theta_{cls})}{\partial \theta_{cls}} + \frac{\partial \mathcal{D}_{MESA}(\mathbf{f}^i, \hat{y}_i, y_i, \theta_{reg})}{\partial \theta_{reg}}$

$\nabla\theta_{net} = \sum_{i=1}^{M} \nabla\mathbf{f}^i \cdot \frac{\partial Conv(x_i, \theta_{net})}{\partial \theta_{net}}$

update $\theta_{cls} = \theta_{cls} - \eta(t) \cdot \nabla\theta_{cls}$, $\theta_{reg} = \theta_{reg} - \eta(t) \cdot \nabla\theta_{reg}$, and $\theta_{net} = \theta_{net} - \eta(t) \cdot \nabla\theta_{net}$

End while

Output θ_{net}, θ_{reg}

3.2. The ConvNet for feature learning

As mentioned above, we learn crowd features with a variation of deep ConvNets [31]. The convolution and pooling operations in deep ConvNets are specially designed to extract visual features hierarchically, from local low-level features to global high-level ones. The deep ConvNet as listed in Table 2 takes similar structure to that in [32] and an illustration of the ConvNet structure used for crowd feature extraction is shown in Fig. 4.

The architecture takes as input a square 32×32 pixel image patches and it consists of 4 layers; The first three layers are convolutional layers and the last layer is the fully connected layer (inner product layer). The max-pooling and the rectified linear units (ReLU) non-linearity follows after the output of every convolutional layers. The first convolutional layer filters the $32 \times 32 \times 3$ input image patches with 16 kernels of size $5 \times 5 \times 3$ with a stride of 1 pixels. The second convolutional layer takes as input the output of the first convolutional layer and filters it with 32 kernels of size $3 \times 3 \times 16$. The third convolutional layer has 64 kernels of size $2 \times 2 \times 32$ connected to the outputs of the second convolutional layer. The last crowd feature layer is fully connected to both the second convolutional and the third convolutional layer, which forms a multi-scale ConvNet [32,33]. The motivation is that multi-scale inputs can provide richer representations comparing

to strict feed-forward layered architectures, by adding complementary information such as local textures and fine details lost by higher levels. In total 100-D crowd features are extracted in the end of our architecture and the regression function is as introduced in Section 3.1.

We use ReLU [34,35] for neurons in the convolutional layers and the fully-connected layer. As shown in previous researches, ReLU has better fitting abilities than the sigmoid units for large training datasets [5]. However, finding the optimal architecture of a ConvNet for a given task remains mainly empirical. More details of the proposed architecture will be detailed in the next Section 4.

In the above specified ConvNet architecture, the proposed designation of two supervisory signals is explained here. Crowd count signal is essentially regression and it ensures the capability of predicting the people count, whereas crowd density signal increases the robustness of our ConvNet model and makes sure the accuracy of counting performance when people counting is more than hundreds. On the other hand, the crowd features learned by ConvNet are independent from temporal information, so that the video sequence is not obligatory.

4. Experiments

The proposed algorithm is implemented on the basis of Caffe library [36] and some modifications are applied. The NVIDIA GTX TITAN X GPU is used. The standard Stochastic Gradient Descent (SGD) algorithm is applied to optimize ConvNet parameters with the momentum of 0.9, batch size of 100 and weight decay of 0.0004. All models are initialized with learning rate of 0.01 and this value is further reduced by hand. The training procedure terminates when the validation error does not change for ten consecutive epoches.

4.1. Data preprocessing

We searched for crowd images of different scenes from Google image search engine. In total, 107 images were selected to compose our dataset AHU-CROWD. Then, we annotated images with dots (one dot per person). Around 45,000 humans, with head count from 58 to 2201 per image, were annotated in our dataset. All crowd images were divided into patches with 32×32 pixels and each patch was associated with a groundtruth count, as illustrated in Fig. 5.

The original training set contains only about 60,000 image patches, which is undersize for training our three layers ConvNet architecture. So, here we employed data augmentation, yielding 486,576 patch samples, 8 times enlarger than original image patches. Patches were randomly perturbed in rotation and in horizontal reflection. Adding these data will yield more robust learning to potential deformation in test set and reduce overfitting in our image data.

In testing stage, we divide our test image into $p \times q$ patches. Our trained architecture can estimate the number of people within each local patches, and crowd estimation in whole image is the sum of that in all local patches.

4.2. The proposed network architecture

Parameter choice is crucial in a number of state-of-the-art methods including ConvNets architecture. So we empirically search for an optimal architecture and extract crowd features from a set of architecture with different parameters. We train the crowd density softmax classifier with a range of different parameters. The training set is used for training and the validation set is used for parameters evaluation, to obtain the following optimal architecture parameters:

- Number and size of kernels at each stage:
  -8C5-MP2-16C3-MP2-16C2-MP2
  -8C5-MP2-16C3-MP2-32C2-MP2
  -16C5-MP2-32C3-MP2-32C2-MP2
  -16C5-MP2-32C3-MP2-64C2-MP2
  -32C5-MP2-64C3-MP2-64C2-MP2,

  where each architecture contains three stages and each stage contains convolutional layer, pooling layer, and ReLU. e.g, *8C* means 8 feature map with 5×5 convolutional kernels. *MP2* means Max-pooling with 2×2 pooling kernels.
- Single or multi-scale: the single-scale architecture (SS) uses only 3rd stage features as input to generate crowd features while multi-scale architecture (MS) feeds the output of both the 2nd stage and the 3rd stage to produce crowd features.
  By observing performance of the above-mentioned network architectures, we see that MS architectures outperform SS architectures in most of time and the architecture of 16C5-MP2-32C3-MP2-32C2-MP2 achieves the best classification accuracy. For this reason, we choose this architecture to extract our crowd features.

4.3. Experiment results on AHU-CROWD

For experiment,we performed 5-fold cross validation on our dataset and to quantify our comparative experiment results on AHU-CROWD, We utilize two evaluation criteria: Absolute Deviation (AD) which was introduced in [3], and Relative Deviation (RD) derived from AD.

$$AD = \frac{1}{N} \times \sum_{i=1}^{N} |p_i - g_i| \tag{6}$$

$$RD = \frac{1}{N} \times \sum_{i=1}^{N} \frac{|p_i - g_i|}{g_i} \tag{7}$$

where N is the number of images in our dataset, p_i and g_i is the estimation value and groundtruth of the i_{th} image. As we can see in Eqs. (6) and (7), AD represents the average deviation number. On the contrary, RD pays more attention to its deviation rate. Lower AD or RD values means more accurate and better.

Such methods are also tested for comparison: Haar Wavelet [37], Deformable Parts Model (DPM) [1], Ridge Regression [17] and the method of Bag of words and Support Vector Machine (BOW–SVM) [38]. The method of Haar Wavelet [37] and DPM [1] are based on traditional head detection, and they estimate people count through detecting instances of people in the whole image. Ridge Regression [17] is a global regression method using various hand-crafted features including area, perimeter, edge and local texture features. The method of BOW–SVM [38] estimates people count within local patch, BOW model was utilized to construct visual words histogram by *K*-means quantization done on local patches, and people counting in each patches is judged by SVM regression.

The quantitative results in Table 3 show that the method of Haar Wavelet [37] and DPM [1] achieve the highest AD and RD values respectively, that is to say, the methods relies on traditional head detection cannot be applicable in our mid-level or high-level density crowd image dataset, and our proposed approach achieves lower AD and RD values comparing to the method of Ridge Regression [17] and BOW–SVM [38].

Table 3
Absolute Deviation and **Relative Deviation** of our proposed approach including **SSCCM** and **TSCCM** with different n, and comparison with Ridge Regression [17], BOW–SVM [38], Haar Wavelet [37] and DPM [1].

Method	AD	RD
Ridge Regression [17]	207.4	0.578
BOW–SVM [38]	218.8	0.604
Haar Wavelet [37]	409.0	0.912
DPM [1]	395.4	0.864
SSCCM (our proposed), $n = 1$	191.0	0.523
TSCCM (our proposed), $n = 5$	165.5	0.497
TSCCM, $n = 8$	148.2	0.418
TSCCM, $n = 10$	**137.5**	**0.365**
TSCCM, $n = 12$	141.9	0.387
TSCCM, $n = 15$	145.0	0.392

Table 4
Crowd estimations of Fig. 3 via switching the different value of parameter ***n***. The first column lists the four images of Fig. 3. The second column shows the groundtruth of each image. Here, groundtruth is abbreviated to **GT**. Crowd estimations are listed in column 3–6.

Figure	GT	$n = 1$	$n = 5$	$n = 8$	$n = 10$	$n = 12$	$n = 15$
Fig. 3(a)	312	201	236	262	**287**	273	259
Fig. 3(b)	748	491	603	699	**732**	777	769
Fig. 3(c)	64	69	**66**	69	71	73	70
Fig. 3(d)	2201	1357	1693	1786	1733	**1819**	1801

Here, Single Signal Crowd ConvNet Model (with only crowd counting signal) is abbreviated to SSCCM and Two Signal Crowd ConvNet Model is abbreviated to TSCCM. As we can see in Table 3, TSCCM outperforms SSCCM in crowd counting, while the parameter n is set to be 10, it achieves the lowest AD and RD. n represents how many different density levels we classify and it mainly lies on the crowd distributions on our training data. 10 is just the best reference in our application. To further display the effectiveness of crowd density signal, we list the crowd estimations of Fig. 3 via switching the different value of parameter n in Table 4. As we can see, TSCCM shows its superiority while crowd counting is more than hundreds.

Some typical estimations (the parameter n is set to be 10) of our proposed approach can be seen in Fig. 6. Fig. 6(a) shows the highest RD estimation of our dataset, due to its too high brightness and low contrast. On the contrary, the color balance image obtains the lowest RD estimations, as shown in Fig. 6(b). Fig. 6(c) is the image which contains the maximum individuals and obtains the RD of 0.194, and Fig. 6(d) is the image which contains the minimum individuals and obtains the RD of 0.276.

To further compare the performance of different methods on AHU-CROWD, we show the distribution of relative deviation in Fig. 7(a). The x-axis represents the range of σ (RD for single image), the y-axis shows the percentage of image distributes in the range of σ and the multi-color bars represent different methods used in our experiment. (We do not compare with the methods of Haar wavelet [37] and DPM [1], which make the worst estimations as shown in Table 3, and we only use our TSCCM ($n = 10$) for comparison.) Comparing to the methods of Ridge Regression [17] and BOW + SVM [38], we can find that higher percentage of images distribute between 0 and 0.4 and lower percentage of images distribute between 0.4 and 10 with our proposed method. In Fig. 7(b), we show the number of correct estimations ($\sigma < 0.4$) in different 6 groups, which are classified according to the density level of crowds. It shows that method of Ridge Regression [17] can only make credible estimation in the first three groups (with people count between 0 and 300) and none of the images in group one and group six (with people count from 800 to 3000) can be correctly estimated with the method of BOW–SVM [38].

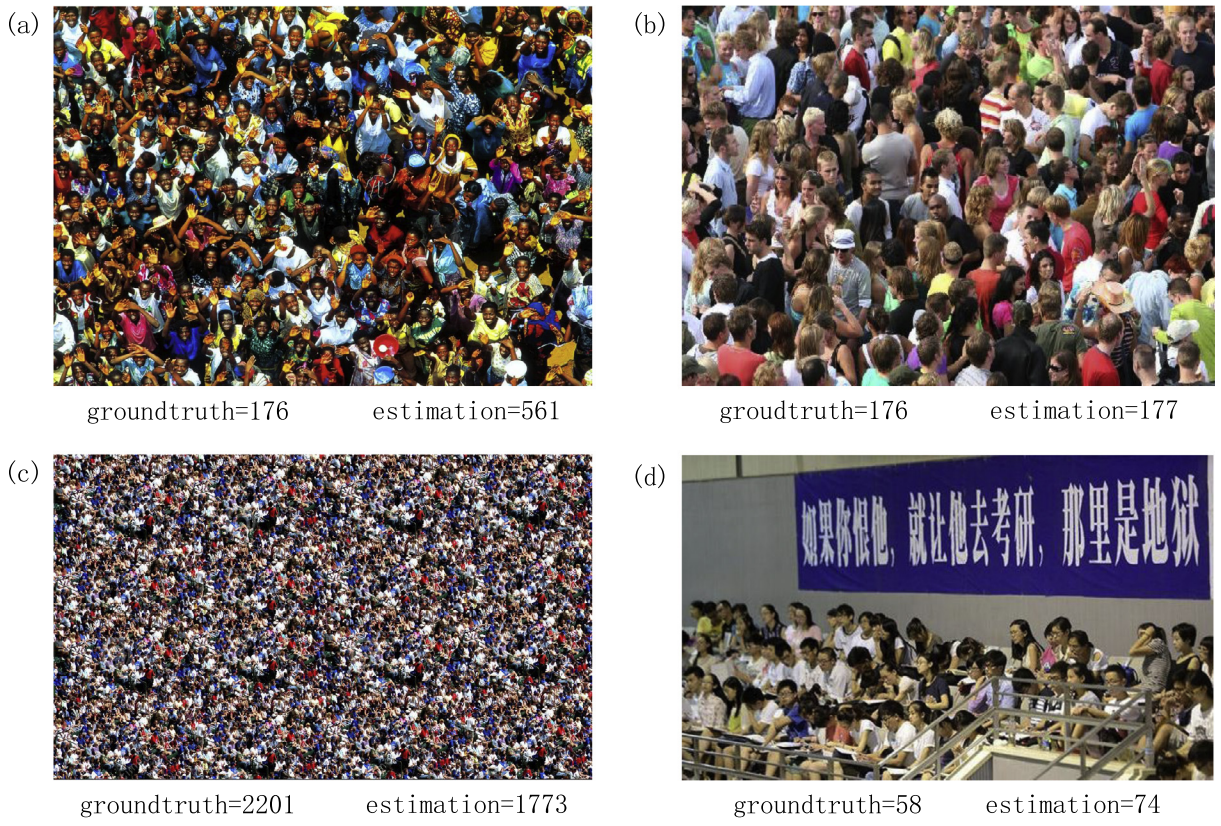
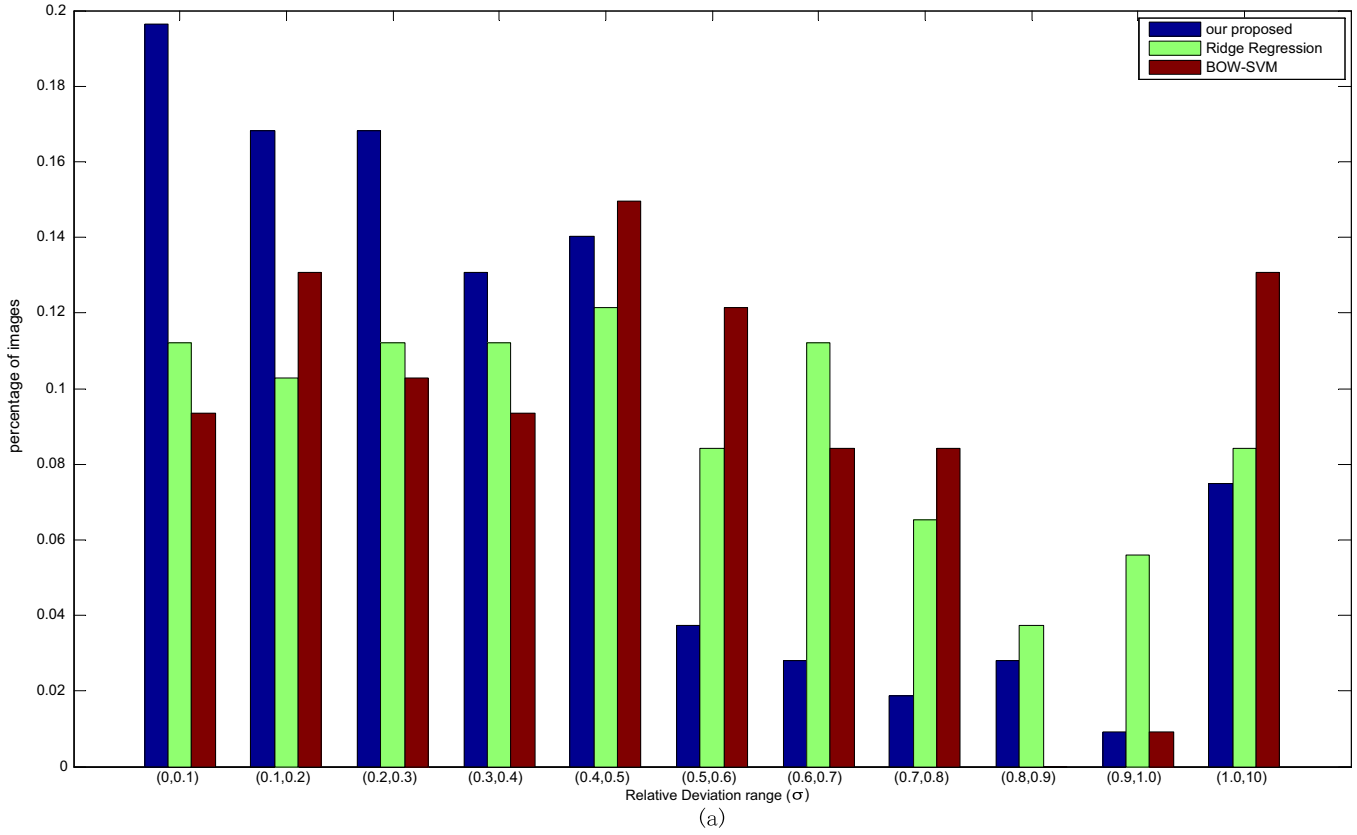
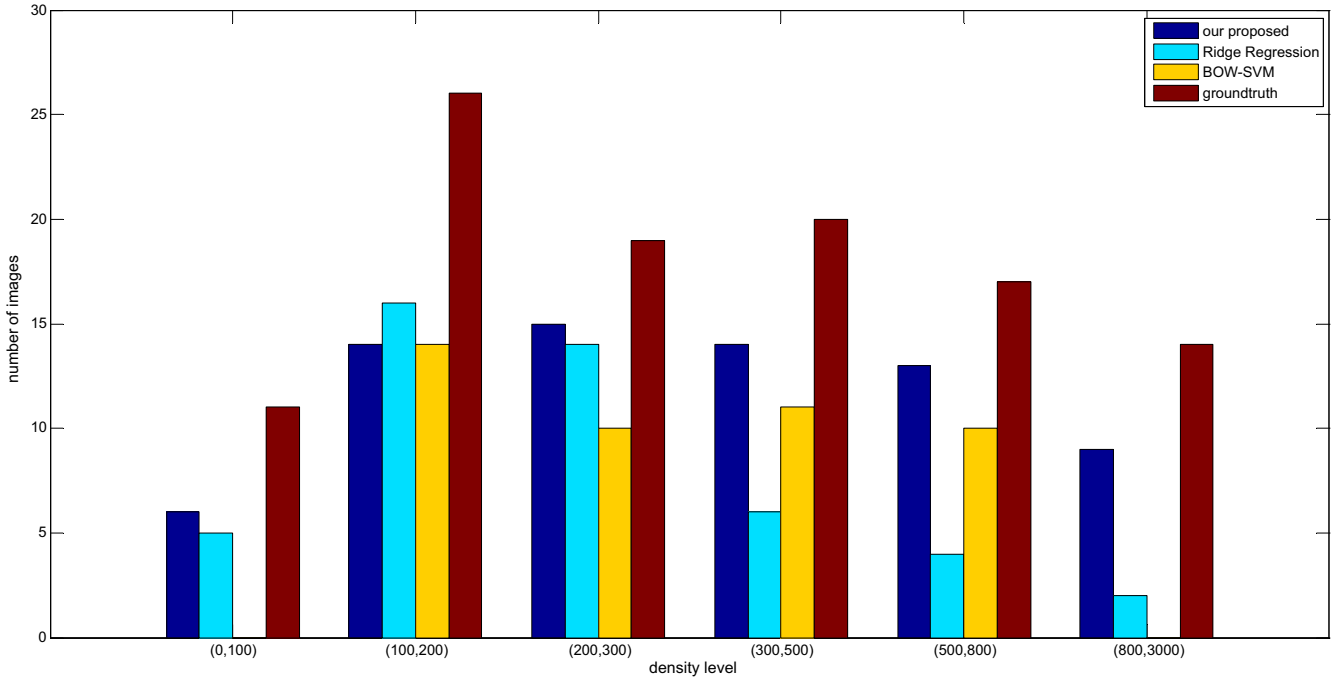


Fig. 6. The figure shows some typical results produced by our proposed method.



(a)



(b)

Fig. 7. Test results on AHU-CROWD.

Comparing to the above methods, our proposed method can be applicable in all six different density level groups.

4.4. Experiment results on UCSD

UCSD dataset contains a 2000-frame video which is chosen from one surveillance camera in the UCSD campus. The video in this dataset was recorded at 10 fps with a frame size of 158×238 and the ground truth count in each frame is from 11 to 46. We test the generalization of our method on it despite its low density.

We follow the experimental setting of [15] and employ frames 601–1400 as the training data and the remaining 1200 frames as test set. All images are divided into patches with 32×32 pixels

Table 5
Quantitative results of the proposed method and comparison with An et al. [39], Chen et al. [17], Chan et al. [15], Chen et al. [40] and Zhang et al. [30].

Method	MAE	MSE
An et al. (Kernel Ridge Regression) [39]	2.16	7.45
Chen et al. (Ridge Regression) [17]	2.25	7.82
Chan et al. (Gaussian Process Regression) [15]	2.24	7.97
Chen et al. (Cumulative Attribute Regression) [40]	2.07	6.86
Zhang et al.(CNN model) [30]	**1.60**	**3.31**
SSCCM	1.98	3.84

Table 6
Quantitative results of the proposed method and comparison with Rodriguez et al. [41], Lempitsky and Zisserman [19], Idrees et al. [3] and Zhang et al. [30].

Method	MAE	MSE
Idrees et al. [3]	468.0	590.3
Rodriguez et al. [41]	655.7	697.8
Lempitsky and Zisserman [19]	493.4	487.1
Zhang et al. [30]	467.0	498.5
SSCCM	485.9	472.3
TSCCM	**431.8**	**438.5**

and each patch is associated with a ground truth count. It is impractical to give a density level label in each patch (the ground truth count in each patch is too small). Therefore, we use Single Signal Crowd ConvNet Model (SSCCM) trained on AHU-CROWD and finetune it with the UCSD training data.

For comparison, we follow the metrics of Mean Absolute Error (MAE) and Mean Squared Error (MSE) as used in [30] to evaluate the performance of compared methods and test results can be shown in Table 5. As we can see, our ConvNet model (SSCCM) outperforms all the global regression based approaches but underperforms the method of Zhang et al. [30], largely due to the absence of crowd density signal. However, the superiority of TSCCM will be highlighted in extreme dense crowd dataset.

4.5. Experiment results on UCF-CROWD

UCF-CROWD dataset contains 50 images covering different crowd scenes. It is an extreme crowd dataset with the head count ranging from 94 to 4543. However, 50 images are undersize for training a new architecture. For this reason, we use TSCCM pretrained on AHU-CROWD and finetune the model with the UCF-CROWD training data, which is similar to the experiment on UCSD.

To compare our proposed method with the methods of Rodriguez et al. [41], Lempitsky and Zisserman [19], Idrees et al. [3] and Zhang et al. [30], we split the dataset randomly and perform 5-fold cross-validation, following by the experimental setting in [3]. MAE and MSE are utilized to quantify experiment results and the quantitative results are presented in Table 6.

It can be seen that our proposed SSCCM outperforms the method of Lempitsky and Zisserman [19] and the method of Rodriguez et al. [41]. While adding the crowd density signal, TSCCM outperforms the other four, including the state of the art methods Idrees et al. [3] and Zhang et al. [30].

5. Conclusion

In this paper, we present a deep-learning approach to estimate the number of individuals in mid-level or high-level crowd visible in a still image. The proposed deep-learning framework is used to learn a feature-count regressor, which can estimate the number of people within each local region, and the crowd estimation in whole image is therefore the sum of that in all local regions. Experiments have been performed on our own dataset AHU-CROWD, and public datasets UCSD [15] and UCF-CROWD [3]. By quantifying our experiment results, we prove that our proposed architecture outperforms both traditional methods based on head detection and learning based methods on UCSD [15], and achieves the state-of-the-art result on UCF-CROWD [3]. As a possible potential direction, to combine temporal information and multiresolution information with ConvNet can be explored in our future work, to make a better estimation and apply this deep learning architecture in video surveillance and safety management.

Acknowledgements

This work is supported by the National Natural Science Foundation (NSF) of China (Nos. 61300056 and 61572029), the Anhui Provincial Natural Science Foundation of China (No. 1408085QF118) and Science and Technology Project of Anhui Province (No. 1501b042207).

References

[1] D.M.P. Felzenszwalb, D. Ramaman, A discriminatively trained, multiscale, deformable part model, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[2] A. Marana, S. Velastin, L. Costa, R. Lotufo, Estimation of crowd density using image processing, in: IEEE Colloquium on Image Processing for Security Applications, 1997, pp. 1–9.
[3] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2547–2554.
[4] D. Ryan, S. Denman, S. Sridharan, C. Fookes, An evaluation of crowd counting methods, features and regression models, Comput. Vis. Image Und. 130 (2014) 1–17.
[5] A. Krizhevsky, S. Ilya, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[6] M. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Europe Conference on Computer Vision, 2014, pp. 818–833.
[7] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[8] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898.
[9] T. Sainath, B. Kingsbury, A. Mohamed, G.E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, B. Ramabhadran, Improvements to deep convolutional neural networks for LVCSR, in: IEEE Workshop on Automatic Speech Recognition and Understanding, 2013, pp. 315–320.
[10] E. Smirnov, North atlantic right whale call detection with convolutional neural networks, in: ICML Workshop on Machine Learning for Bioacoustics, 2013.
[11] D. Lowe, Object recognition from local scale-invariant features, 1999, pp. 1150–1157.
[12] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: International Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 886–893.
[13] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 971–987.
[14] L. Bourdev, S. Maji, J. Malik, Describing people: a poselet-based approach to attribute classification, in: IEEE International Conference on Computer Vision, 2011, pp. 1543–1550.
[15] A. Chan, Z.-S. Liang, N. Vasconcelos, Privacy preserving crowd monitoring: counting people without people models or tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–7.
[16] C.C. Loy, S. Gong, T. Xiang, From semi-supervised to transfer counting of crowds, in: IEEE International Conference on Computer Vision, 2013, pp. 2256–2263.
[17] K. Chen, C.C. Loy, S. Gong, T. Xiang, Feature mining for localised crowd counting, in: British Machine Vision Conference, 2012.
[18] J. Ferryman, A. Ellis, Pets2010: dataset and challenge, in: IEEE International Conference on Advanced Video and Signal Based Surveillance, 2010, pp. 143–150.
[19] V. Lempitsky, A. Zisserman, Learning to count objects in images, in: Advances in Neural Information Processing Systems, vol. 23, 2010, pp. 1324–1332.
[20] S.-F. Lin, J.-Y. Chen, H.-X. Chao, Estimation of number of people in crowded scenes using perspective transformation, IEEE Trans. Syst. Man Cybernet. Part A: Syst. Humans 31 (2001) 645–654.
[21] B. Wu, R. Nevatia, Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors, in: IEEE International Conference on Computer Vision, vol. 1, 2005, pp. 90–97.

[22] W. Ge, R.T. Collins, Marked point processes for crowd counting, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2913–2920.
[23] M. Li, Z. Zhang, K. Huang, T. Tan, Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection, in: International Conference on Pattern Recognition, 2008, pp. 1–4.
[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, F.-F. Li, Imagenet large scale visual recognition challenge, 2014.
[25] A.B. Chan, N. Vasconcelos, Counting people with low-level features and bayesian regression, IEEE Trans. Image Process. 21 (2012) 2160–2177.
[26] T. Liu, D. Tao, On the robustness and generalization of cauchy regression, in: 2014 4th IEEE International Conference on Information Science and Technology (ICIST), 2014, pp. 100–105.
[27] J. Li, D. Tao, A bayesian hierarchical factorization model for vector fields, IEEE Trans. Image Process. 22 (2013) 4510–4521.
[28] T. Liu, D. Tao, On the performance of manhattan nonnegative matrix factorization, IEEE Trans. Neural Networks Learn. Syst. PP (2015) 2855–2872.
[29] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, IEEE Trans. Pattern Anal. Mach. Intell. PP (2015) 447–461.
[30] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: CVPR, 2015.
[31] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998, pp. 2278–2324.
[32] P. Sermanet, Y. Lecun, Traffic sign recognition with multi-scale convolutional networks, in: International Joint Conference on Neural Networks, 2011.
[33] S.C. Pierre Sermanet, Y. LeCun, Convolutional neural networks applied to house numbers digit classification, in: International Conference on Pattern Recognition, 2012.
[34] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: International Conference on Machine Learning, 2010.
[35] A.B. Xavier Glorot, Y. Bengio, Deep sparse rectifier neural networks, in: International Conference on Artificial Intelligence and Statistics, 2011.
[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, 2014.
[37] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, T. Poggio, Pedestrian detection using wavelet templates, in: IEEE Conference on Computer Vision and Pattern Recognition, 1997.
[38] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV, 2004.
[39] S. An, W. Liu, S. Venkatesh, Face recognition using kernel ridge regression, in: IEEE Conference on Computer Vision and Pattern Recognition. CVPR '07, 2007, pp. 1–7.
[40] K. Chen, S. Gong, T. Xiang, C. Loy, Cumulative attribute space for age and crowd density estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2467–2474.
[41] M. Rodriguez, J. Sivic, I. Laptev, J.-Y. Audibert, Density-aware person detection and tracking in crowds, in: Proceedings of the International Conference on Computer Vision, 2011.