Neurocomputing 166 (2015) 151–163

Contents lists available at ScienceDirect

Neurocomputing

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

Crowd counting in public video surveillance by label distribution learning

Zhaoxiang Zhang [a,*], Mo Wang [a], Xin Geng [b]

[a] State Key Laboratory, Beihang University, Beijing, China
[b] School of Computer Science and Engineering, Southeast University, Nanjing, China

ARTICLE INFO

Article history:
Received 16 October 2014
Received in revised form
30 January 2015
Accepted 19 March 2015
Communicated by Ran He
Available online 20 April 2015

Keywords:
Crowd counting
Information reuse
Label distributions
Visual surveillance

ABSTRACT

The increase of population causes the raise of security threat in crowed environment, which makes crowd counting becoming more and more important. For common complexity scenes, existing crowd counting approaches are mainly based on regression models which learn a mapping between low-level features and class labels. One of the major challenges for generating a good regression function is the insufficient and imbalanced training data. Observationally, the problem of crowd counting has the characteristic that crowd images with adjacent class labels contain similar features, which can be utilized to reduce the effect of insufficiency and imbalance by the strategy of information reuse. Consequently, this paper introduces a label distribution learning (LDL) strategy into crowd counting, where crowd images are labelled with label distributions instead of the conventional single labels. In this way, one crowd instance can contribute to not only the learning of its real class label, but also the learning of neighboring class labels. Hence the training data are increased significantly, and the classes with insufficient training data are supplied with more training data, which belongs to their adjacent classes. Experimental results prove the effectiveness and robustness of the LDL method for crowd counting. We have also shown the outstanding performance of the approach in different dataset.

1. Introduction

With the increase of population, threats in crowded environment are rising, including fighting, rioting, and violent protest. The most common indicator of these behaviors is the crowd size, and its evaluation known as crowd counting or crowd density estimation attracts more attentions.

Generally, when given a video captured by a static camera in crowd situations, crowd counting approaches can estimate the number of people or the level of crowd density. There are many potential real-world applications in crowd counting [1,2], e.g. surveillance in public for safety and security by detecting abnormally large crowd, resource management in retail sectors for optimizing floor plan or product display by quantifying the number of people entering and existing at different times of the day, and urban planning for developing long-term crowd management strategies or designing evacuation routes in public spaces by statistically analysing the flow rate of people around an area. In other fields, crowd counting methods are also applicative. For instance, animals pass through a particular boundary, blood cells flow through a blood vessel under a microscope, and the rate of car traffic.

Existing methods for crowd counting could be roughly divided into the following three categories: pedestrian detection based approaches, trajectory clustering based approaches, and the feature-based regression approaches [1]. The pedestrian detection based approaches [3–6] estimate the number of people by detecting the whole instances of people in a crowd image, using a trained detector to scan the image with different scales. The trajectory clustering based approaches [7,8] count the number of people by analyzing the feature trajectory extracted from each crowd frame. The approaches based on pedestrian detection and trajectory clustering either rely on explicit object segmentation or feature point tracking, which requires sufficient computational expense and high frame rate video. Thus, they are not suitable to crowd scenes with cluttered background and frequent occlusion. In contrast, the feature-based regression approaches [9–13,2] aim to learn a direct mapping between multi-dimensional features and class labels, only depending on the low-level features extracted from crowd frames with ordinary frame rate. Based on the above analysis, the feature-based regression approaches are more appropriate to be adopted in real applications.

However, the feature-based regression approaches also have inherent disadvantages. As we know, the performance of an appropriate regression function always depends on the population of training data. Existing benchmarking datasets for crowd counting such as Mall and UCSD are insufficient and imbalanced, as

* Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2015.03.083
0925-2312/

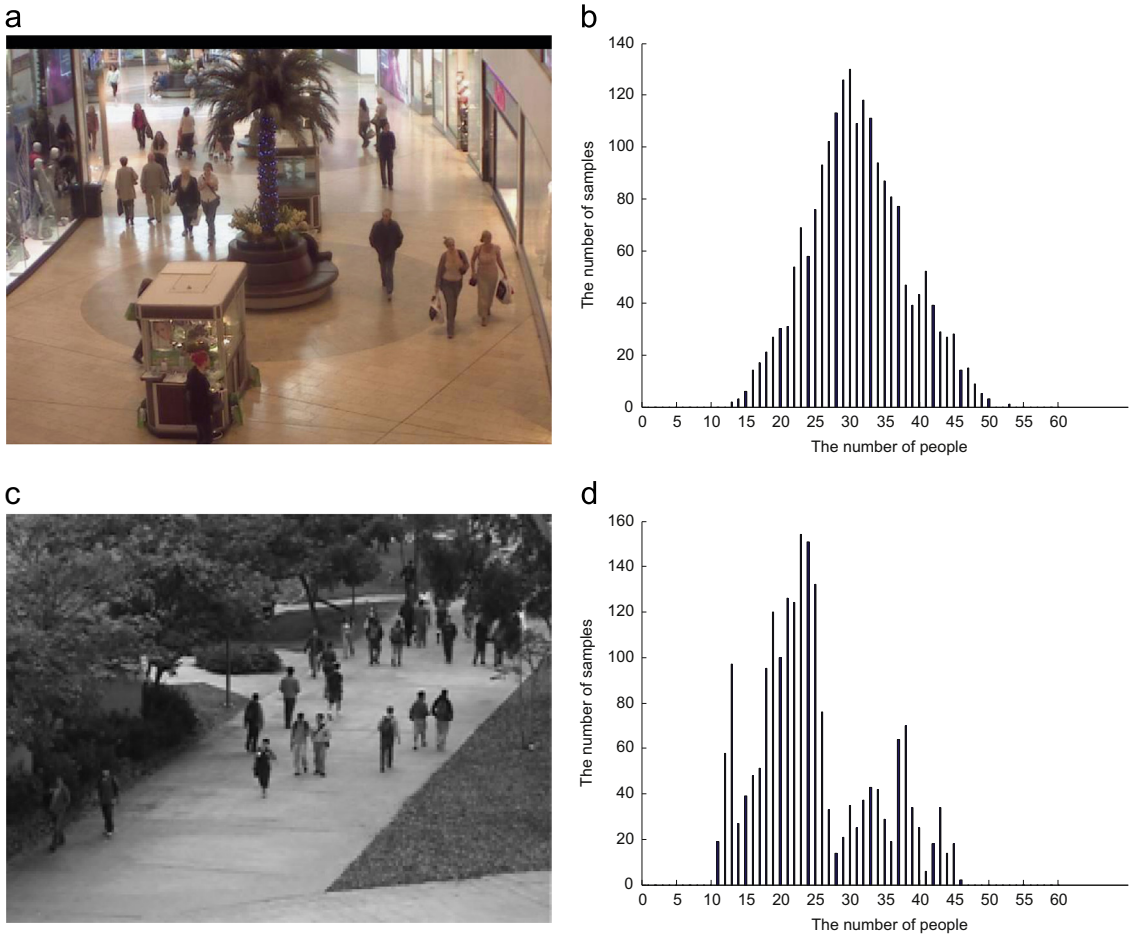


Fig. 1. Both Mall dataset and UCSD dataset suffer from insufficient and imbalanced training data. (a) A sample of Mall dataset, (b) the distribution of Mall dataset, (c) a sample of UCSD dataset, and (d) the distribution of UCSD dataset.

shown in Fig. 1. The insufficiency of the dataset reflects in its limited number of samples belonging to certain classes, while the imbalance of the dataset means the samples of different classes have a great quantity variance. Both insufficiency and imbalance of training data have significantly negative effect in crowd estimation. To dispose this challenge, Chen et al. [12] propose a cumulative attribute based ridge regression (CA-RR) method for crowd counting. The experimental results of CA-RR on Mall dataset show its superior performance to the state-of-the-arts, but the solution seems complicated and not straightforward. Besides the attribute based solution, we can consider a more essential solution based on information reuse. According to the discussions in [12], the problem of crowd counting has a characteristic that features of crowd images which contain adjacent number of people are strongly correlated. In another word, the number of people varies continuously on features while discretely on labels. For instance, the crowd image containing 30 people shows the similar features to the one containing 29 people, while is significantly different from the one with 10 people. As we can see, apart from the real number of people, one crowd image can also contribute to the learning of its neighboring people-count. In this way, it can reuse the information of samples for insufficient and imbalanced training data.

The most popular strategy to combine regression and information reuse is label distribution learning [14], which learns a regression function between multi-dimensional features and a label distribution, instead of the conventional single label. Label distribution learning has been successfully used in facial age estimation, which has the same characteristic as the problem of crowd counting that samples vary continuously on features while discretely on labels. In our opinion, the label distribution learning algorithm can also be employed for crowd counting, and dispose the insufficiency and imbalance of training data by reusing the information of samples.

In this paper, we assign a label distribution to each crowd image rather than a traditional single label. The label distribution of each crowd instance covers numerous class labels with a probability model, which is utilized to represent the degree that each class label describes the instance. In this way, a crowd instance will contribute to not only the learning of its real class, but also the learning of its neighboring classes. Hence the training data are increased significantly, and the classes with insufficient training data are supplied with more training data, which belong to their adjacent classes. Then, a regression function between feature set and label distributions is learned by the IIS-LDL algorithm, where the optimization uses a strategy similar to Improved Iterative Scaling (IIS). The IIS is a well-known algorithm for maximizing the likelihood of the maximum entropy model, which makes the IIS-LDL be an iterative optimization process. Finally, given an unseen instance, the regression function will generate a label distribution. The predicted value is estimated by the weighted average of the label distribution.

When designing the label distribution of the training instance, the real class label must be the leading description. In other words, the highest probability in the label distribution is assigned to the real class label, which ensures its leading position in the class description. While the probability of other class labels decreases with the distance from the real class label, which makes the

classes closer to the real class contribute more to the class description. Consequently, in the process of prediction, labels in a predicted distribution are also correlative, which proves the theory evidence to synthetically use the predicted distribution for estimating results.

Our framework is illustrated in Fig. 2. Firstly, the label set of a dataset is transformed into label distributions by allocating different probabilities to each label within a certain range. Secondly, the normalization is adopted to remove the effect of perspective before extracting features from dataset. Thirdly, three types of features are extracted, including the global features and the local texture features. In the end, a regression function utilized to predict is learned by the IIS-LDL algorithm. Experimental results on benchmarking datasets show that label distribution learning for crowd counting can effectively reduce the effect caused by the insufficient and imbalanced training data, thus improving the accuracy generally over the state-of-the-arts.

2. Related work

Existing crowd counting techniques are classified into three categories: counting by pedestrian detection, counting by trajectory clustering, and counting by feature-based regression. Various approaches for crowd counting have been proposed [1].

Counting by detection detects the instances of pedestrian by using a trained detector to scan the image space. This paradigm includes several different detection methods.

Monolithic detection method [3,15,16] is a typical pedestrian detection approach. In this method, a pedestrian classifier is trained by the full-body appearance extracted from a set of pedestrian training images. The full-body appearance is represented by some common features, such as Haar wavelets [17], gradient-based features (e.g. histogram of oriented gradient (HOG) feature) [3], edgelet [18], and shapelet [19]. The choice of classifier often requires a balance between the speed and quality of detection. It commonly used linear classifiers such as boosting [20], linear SVM, or Random/Hough Forests [21] rather than non-linear classifiers such as RBF Support Vector Machines (SVM) since non-linear classifiers offer a good quality but suffer from a low speed. A trained classifier is then applied in a sliding window style across the whole image space, which aims to detect pedestrian candidates. Monolithic detector can generate reasonable detections in sparse scenes, but suffers from crowd scenes with the frequent occlusion.

Part-base detection method [4,22,23] is proposed to handle the partial occlusion problem. Even in the occlusion situation, some specific body parts such as head and shoulder can easily appear in the video. Consequently, boosted classifiers are constructed for these body parts to estimate the number of people in a monitored area [24]. It is found that the head region alone is not sufficient for reliable detection because of its shape and appearance variations. While adding the shoulder region to form an omega-like shape pattern, it tends to perform better in real-world scenes.

Shape matching method [5,25] employs shape prototypes to describe and detect the pedestrians. A set of parameterized body shapes composed of ellipses is defined to represent the pedestrians [5], then applied in a stochastic process to estimate the number and shape configuration that explains a given foreground mask greatly in a scene. This idea is extended by using more flexible and realistic shape prototypes in [25], where it learns a mixture model of Bernoulli shapes from a set of training images to search for maximum a posteriori shape configuration of foreground objects. And it reveals not only the count and location, but also the pose of pedestrians in a scene.

Multi-sensor detection methods are based on multiple cameras, and can further incorporate multi-view information to resolve visual ambiguities caused by the inter-object occlusion. For instance, the foreground human silhouettes are extracted from a network of cameras to establish bounds on the number and possible locations of people in [6]. Similarly, the multi-view geometric constrains are utilized to estimate the number of people and their spatial locations in [26]. Benefitting from the multi-sensors, these approaches improve the accuracy and speed of detection. However, they are restricted since a multi-camera setup with overlapping views is not always available in many cases.

Counting by trajectory clustering counts people by tracking visual features over time. A crowd can be regarded as a mixture composed of many individual entities, each of which has its own feature trajectory. These trajectories which exhibit coherent motion are clustered, and the number of clusters approximates the number of moving people. Typical examples include [7], which uses a Kanade–Lucas–Tomasi (KLT) tracker to extract the low-level tracked features, and clusters the trajectory to estimate the number of pedestrians. Another example is proposed in [8], which relies on an unsupervised Bayesian approach. Such a trajectory clustering based algorithm works well with the video that has enough high frame rate, in order to extract the trajectory information reliably. Otherwise, it will suffer from an unsatisfied result. In addition, this paradigm assumes the coherence of pedestrian motion. Hence the false estimation may arise when pedestrians remain static in a scene, or two objects sharing common feature trajectories over time.

Counting by regression avoids explicit segmentation or feature point tracking but estimates the crowd density based on the

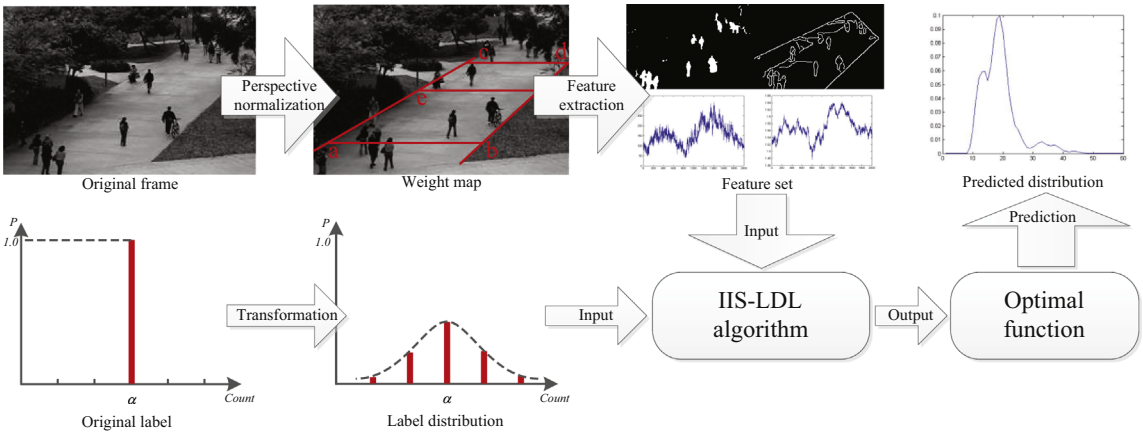


Fig. 2. The processing pipeline of our framework: perspective normalized features are extracted from segmentation, then the label distribution associated with an instance is transformed by a probability model, finally, the IIS-LDL algorithm is utilized to train a regression function for prediction.

holistic description of crowd instances. Consequently, this paradigm becomes a feasible method for crowded environments where detection and tracking are severely limited. In detail, feature-based regression methods aim to learn a direct mapping between multi-dimensional features and class labels, which is used to predict the number of pedestrian. One of the earliest regression algorithms for crowd counting is proposed by Davies et al. [9]. They first extract low-level features such as foreground segmentation and edge features from each video frame. Then the holistic properties such as foreground area and edge count are generated from these raw features. Finally, a linear regression model is used to establish a mapping between the holistic properties and the actual people-count. In this way, when given the feature set extracted from an unseen video frame, conditional expectation of the people-count can be predicted. This work opens up a new path in crowd counting, and various algorithms following this idea have been proposed. These algorithms present either the improved feature set or more sophisticated regression models, but still share a similar processing pipeline as in [9]. A few popular regression models are shown as follows.

Linear regression is first used in crowd counting, and the simplest approach is to form a linear regression function that involves a linear combination of the input variables. In a sparse scene with smaller crowd size and fewer inter-object occlusion, the aforementioned linear regressor [27,9,28] may suffice since the mapping between the observations and class labels presents a linear relationship. However, when given a more crowded environment with severe inter-object occlusion, it has to employ a nonlinear regressor to capture the nonlinear trend in the feature space adequately [29]. A special example of this model is polynomial regression function considered in [30], which uses a form of powers of input variables to structure the basis functions. Gaussian basis function and sigmoidal basis function are other possible choices of basis functions. One of the key limitations of linear model is that the model will get unnecessarily complex high-dimensional observed data, some of which are useless for prediction. Furthermore, part of the extracted data may be highly co-linear, which may lead to an unstable estimation [31] and cause the severe over-fitting.

Partial least squares regression is a way to dispose the multicollinearity problem [10], by projecting both input variables and target variables to a latent space. In this space, the lower-dimensional latent variables explain the covariance between input variables and target variables as much as possible.

Ridge regression (RR) is another method for mitigating the multicollinearity problem [13]. A regularization term is added to the error function of the ridge regression, which is used to estimate the sum of squared errors. Chen et al. [11] present a multi-output ridge regression (MORR) for crowd counting, which is based on a multivariant ridge regression. Later, they map low-level visual features onto a cumulative attribute space where each dimension has clearly defined a label that captures how the scalar output value changes continuously and cumulatively, named as cumulative attribute based ridge regression (CA-RR) [12]. A nonlinear version of the ridge regression is the kernel ridge regression (KRR) [32], which can be achieved by a kernel trick [33]. In detail, a linear ridge regression model is constructed in a higher dimensional feature space induced by a kernel function defining the inner product. The kernel function has several typical choices of linear, polynomial, and radial basis function (RBF) kernels.

Support vector regression (SVR) is used for crowd counting in [34]. In contrast to KRR, the SVR utilizes the concept of support vectors to determine the solution, which can lead to a faster testing speed than the KRR that sums over the entire training set.

Gaussian process regression (GPR) is one of the most popular nonlinear methods for crowd counting, which benefits from its pivotal properties, e.g. it allows a possibly infinite number of basis functions driven by the data complexity, and it models the uncertainty in regression problems elegantly. Based on the conventional GPR, various extended approaches have been proposed. For example, Chan et al. [29] propose a generalized Gaussian process model, which allows a different parameterization of the likelihood function, such as a Poisson distribution for prediction in [35]. Lin et al. [36] utilize two GPR in their framework, one for learning the mapping between features and classes, and the other for deducing the mismatch between the predicted label and the actual label caused by the occlusion. The flexibility of kernel algorithms such as KRR, SVR, and GPR makes it possible to design different assumptions about the function we wish to learn. This property is exploited in [2] by combining a linear and a squared-exponential (RBF) covariance function, which captures both the linear trend and local non-linearities in the crowd feature space.

Random forests regression (RFR) is able to achieve the scalable nonlinear regression modelling [37]. A random forest composed of randomly trained regression trees can achieve a better generalization than a single over-trained tree [38]. Each tree in the forest splits a complex nonlinear regression problem into a set of subproblems, which can be more easily addressed by simple learners such as a linear model. And the forest is trained by optimizing an energy over a given training set and label set.

Semi-supervised regression is proposed for crowd counting when only given insufficient labelled data. Loy et al. [39] develop a unified framework for the active and semi-supervised learning of a regression model with transfer learning capability, and the frame work is formulated based on exploiting the underlying manifold structure of unlabelled crowd data to facilitate counting when the labelled samples are insufficient.

3. Feature extraction

3.1. Perspective normalization

Before extracting features from datasets, the effects of perspective cannot be neglected. Perspective makes objects closer to the camera appear larger in frames. Thus, it is important to normalize the features for reducing the effects of perspective. Since perspective conforms to linear variation, feature extracted from each pixel could be weighted by the relative location of the pixel in a frame. The weight of a pixel is dependent on the depth of object which contains the pixel. That is to say, the weight is bigger when the object is farther.

In this work, weights are computed with the proportion of depth in the scene. For the UCSD dataset, a ground plane is first marked by determining two vanishing lines $\overline{ac}$ and $\overline{bd}$, as in Fig. 3(a). The bottom horizontal line $\overline{ab}$ is assumed as the standard and weights of the pixels on $\overline{ab}$ are set to 1. Next, the lengths of $\overline{ab}$ and $\overline{cd}$ are manually measured. The length of any line $\overline{ef}$ parallel to $\overline{ab}$ can be computed by the linear interpolation, where the length represents the interpolant. Finally, each pixel on the line $\overline{ef}$ is given a weight of $|\overline{ab}|/|\overline{ef}|$.

Different from the UCSD dataset, the region of interest (ROI) of the scene in Mall dataset is curve, as shown in Fig. 3(b), which may lead to a noticeable deviation in the process of perspective normalization. Observationally, the ellipse on the ground appears larger in the scene when it is closer to the camera. However, all the ellipses are equirotal in reality. Consequently, we define two sets of vanishing lines based on the three ellipses, one for the normalization of the front scene, marked as $\overline{ae}$ and $\overline{bf}$. The other is used to the normalization of the back scene, marked as $\overline{ce}$ and $\overline{df}$. The rest of work is the same as aforementioned processes.

In addition, the aforementioned horizontal vanishing line is assumed to be parallel to the image horizontal scan lines. However, we think that the effect of perspective normalization will be better if employing an automatic approach instead of these manual processes.

In this way, objects generated by these pixels are projected onto a flat, and then features can be extracted away from the influence of perspective. When applying the weights to feature extraction, it is assumed that the foreground area changes quadratically, whilst the total edge pixels change linearly. As a consequence, each foreground segment pixel is weighted by the original weight and the edge features are weighted by square-roots of the weights. For the features based on the gray-level co-occurrence matrix (GLCM) [40], we normalize them by weighting the occurrence of each pixel pair when accumulating the co-occurrence matrix.

3.2. Feature extraction

As the pivotal part of input for a regression model, feature representation concerns the extraction, selection, and transformation of low-level visual properties in the frame. A ROI is manually selected in two datasets in order to exclude spurious foreground segments from other regions, as shown in Fig. 4. A popular approach as in [2] is to combine several features to form a large bank of features.

Foreground segment features: The foreground segment is the most common or arguably descriptive representation for crowd counting, which is obtained by the background subtraction. The sample of foreground segment is shown in Fig. 5. Various holistic features can be derived from the extracted foreground segment, for example

- Area – the total number of pixels in the foreground segment.
- Perimeter – the total number of pixels belong to the foreground segment perimeter, computed with the dilation and erosion in morphological operators.
- Perimeter edge orientation – the orientation histogram of the foreground segment perimeter, generated by finding the maximum response to a set of Gabor filters at that point.
- Perimeter–area ratio – the ratio between the foreground segment perimeter and area, which approximates the complexity of the foreground segment shape.

Edge features: Foreground features capture the global properties of the segment, while edge features inside the segment abstract complementary information about the local and internal patterns. Intuitively, the low-density crowds tend to present coarse edges, while the high-density crowds tend to present complex edges. We employ the Canny edge detector [41] to extract the edges from the segment. Fig. 5 shows the sample of the edge image. Some common edge-based features are listed as follows:

- Total edge pixels – the total number of pixels in the edge image.
- Edge orientation – the orientation histogram of edges, generated in the same way as the perimeter edge orientation.
- Minkowski dimension – the Minkowski fractal dimension or box-counting dimension of the edges, which counts how many predefined structuring elements such as square are required to fill the edges.

Texture features: The crowd texture contains significant information about the number of people in a scene. Compared with the low-density region, the high-density crowd region tends to exhibit a stronger texture response, which has a distinctive local structure. One of the common texture features is based on GLCM. A typical process to obtain GLCM is that first quantizing the image into 8 gray-levels and using the foreground segment to mark it. The joint

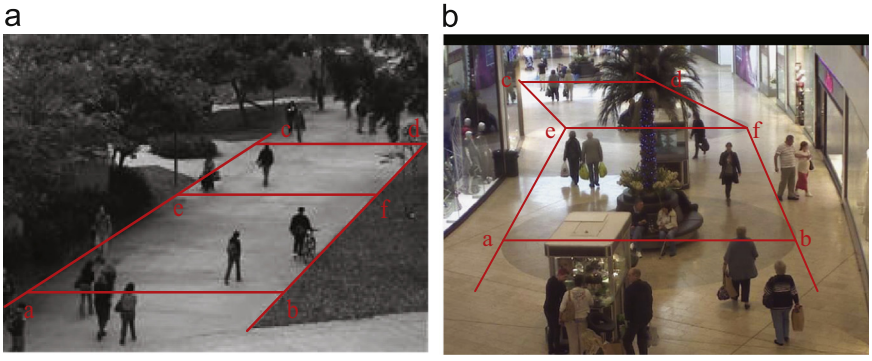


Fig. 3. The setting of perspective normalization on two datasets. (a) The normalization sketch of UCSD dataset and (b) the normalization sketch of Mall dataset.

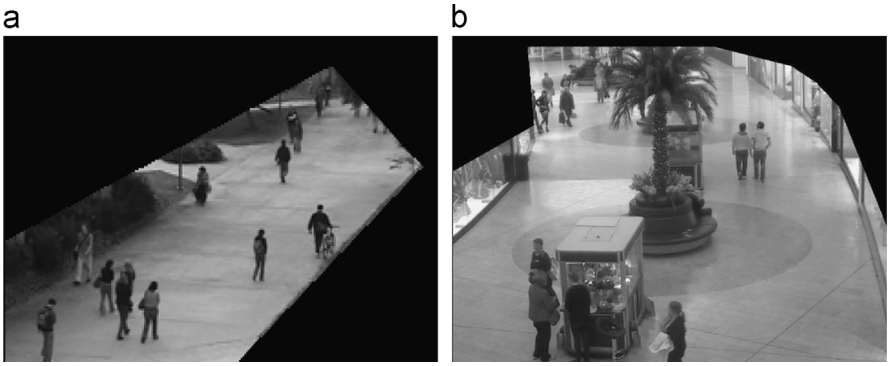


Fig. 4. The regions of interest in two datasets. (a) The ROI of UCSD dataset and (b) the ROI of Mall dataset.

probability or co-occurrence of neighboring pixel values, $p(i,j|\theta)$, is then estimated for four orientations, $\theta \in \{0°, 45°, 90°, 135°\}$. After extracting the co-occurrence matrix, a set of features can be derived, including

- Homogeneity – the smoothness of texture, measured by $\sum_{i,j} p(i,j|\theta)/(1+|i-j|)$.
- Energy – the quadratic sum of GLCM element, which reflects the uniformity and thickness degree of instances, measured by $\sum_{i,j} p(i,j|\theta)^2$.
- Entropy – the measure of information contained in the image, representing the non-uniformity and complexity degree of texture, measured by $\sum_{i,j} p(i,j|\theta) \log p(i,j|\theta)$.

4. Methodology

4.1. Label distribution

In this subsection, we cite the discussion in [14] to illustrate the formulation of label distribution, and explain how to utilize it in our framework.

Each label y in a label distribution is assigned a real number $P(y) \in [0,1]$, which represents the degree that describes the instance. The sum of these numbers assigned to all labels is 1, meaning the full description of the instance. In traditional ways, an instance is labelled with a single label or multiple labels, which can be viewed as several special cases of the label distribution. However, by the definition of the label distribution given in Section 1, the general case is considered to be the most effective representation of an instance. Fig. 6 shows some examples of typical label distributions for five class labels, including single-label, multi-label and general case. For the single-label case (a), the instance is fully described by the label y_3, so $P(y_3) = 1$. For the multi-label case (b), the instance is described by labels y_2 and y_4 together, so $P(y_2) = P(y_4) = 1/2$. Finally, (c) represents a general case of the label distribution, which only restrained by $\sum_y P(y) = 1$.

It is necessary to emphasize the meaning of $P(y)$, which is not the probability that the class y labels the instance correctly, but the degree that the class y describes the instance. Thus, any label with a non-zero $P(y)$ is a correct label to describe the instance but just with the different importance valued by $P(y)$. Based on this difference, it can be clearly distinguished the label distributions from the possible labels, where the basic assumption is that each instance only has one correct label. Obviously, $P(y)$ is not a probability by definition, but they have something in common, i.e. $P(y) \in [0,1]$ and $\sum_y P(y) = 1$. So many theories and methods in statistics can be applied to label distribution.

Furthermore. both the label distribution and the category membership used in fuzzy classification utilize the ambiguity of the instance, but they are different in principle. In fuzzy classification, the category membership aims to express an objective measure (e.g., 'the height of people = 185 cm') of the instance by a subjective category (e.g., 'tall'), which is a conceptual

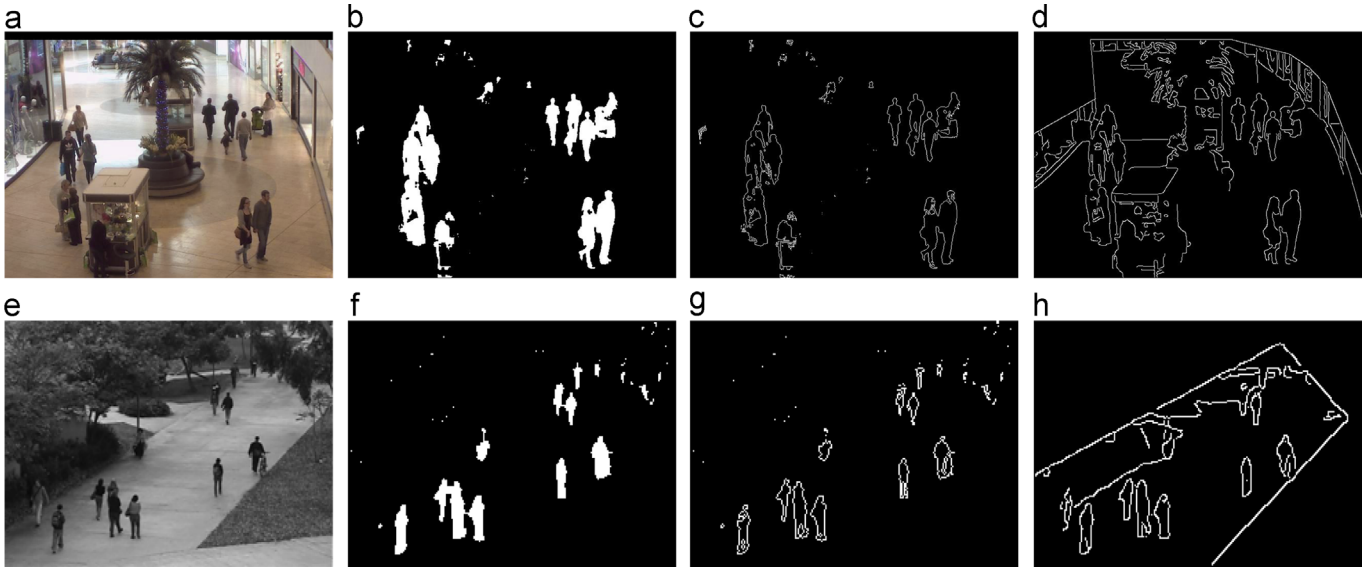


Fig. 5. Some samples of the foreground segment, the edge image of foreground segment, and the edge image on two datasets. (a) An original image in Mall dataset, (b) a sample of foreground segment in Mall dataset, (c) a sample of foreground edge image in Mall dataset , (d) a sample of edge image in Mall dataset, (e) an original image in UCSD dataset, (f) a sample of foreground segment in UCSD dataset, (g) a sample of foreground edge image in UCSD dataset, and (h) a sample of edge image in UCSD dataset.

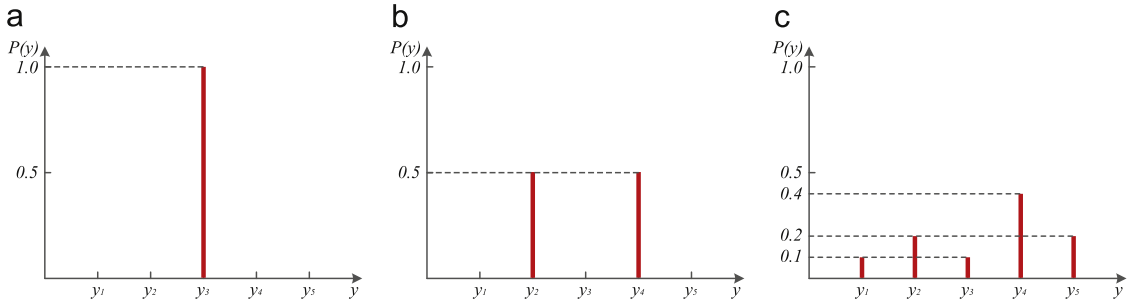


Fig. 6. Three cases of the label distribution.

characteristic of the instance. Thus the ambiguity embodied by the category membership is in the features of the instance, while the final class label of the instance is unambiguous. On the contrary, the ambiguity represented by the label distributions is in the class label of the instance, while the features of the instance are unambiguous. According to the comparison, it appears that pure fuzzy methods are usually based on the setting of the designer, while the label distribution is suitable to adopt the learning from samples.

In this work, Gaussian distribution and triangle distribution are applied to transform the label of an instance into a label distribution, due to their significant properties. As shown in Fig. 7, the concentricity of Gaussian and triangle distribution makes it possible to assign the highest probability in the distribution to real label α, in order to ensure the leading position of α in the class description. The monotonicity and symmetry allow us to decrease the probabilities of other labels with the distance away from α, which makes the label closer to the real label α contribute more to the class description. In this way, it takes full advantage of the similarity among the instances have adjacent class labels.

4.2. Label distribution learning

According to the methodology in [14], the process of label distribution learning is summarized as using a training set composed of feature set and label distributions to learn a regression function, which can generate a label distribution similar to the input label distribution given a instance. Let $\mathcal{X}=\mathbb{R}^d$ denote the input space and $\mathcal{Y}=\{y_1,y_2,\ldots,y_c\}$ denote the finite set of possible class labels. The process of label distribution learning can be formally described as utilizing the training set $S=\{(x_1,P_1(y)),(x_2,P_2(y)),\ldots,(x_n,P_n(y))\}$, where $x_i\in\mathcal{X}$ is an instance and $P_i(y)$ is the distribution of all class labels $y\in\mathcal{Y}$ corresponds to x_i, to learn a conditional p.d.f. $p(y|x)$, where $x\in\mathcal{X}$ and $y\in\mathcal{Y}$.

In order to realize the learning process, a parametric model $p(y|x;\theta)$ is converted from the $p(y|x)$ and the θ is the vector of model parameters. In this way, the goal of LDL translates into finding the θ that can generate a distribution similar to $P_i(y)$ from the training set S. And the similarity between two distributions is measured by the Kullback–Leibler divergence, then, the best model parameter vector θ^* can be computed by

$$\begin{aligned}\theta^* &= \arg\min_\theta \sum_i\sum_y \left(P_i(y)\log\frac{P_i(y)}{p(y|x_i;\theta)}\right)\\ &= \arg\max_\theta \sum_i\sum_y P_i(y)\log p(y|x_i;\theta).\end{aligned} \tag{1}$$

It is necessary to examine that the optimization criterion shown in Eq. (1), which is based on the traditional learning paradigms. For the supervised learning, an instance is associated with a single label, thus $P_i(y)=\delta(y,y_i)$, where $\delta(a,b)$ is the Kronecker function and y_i is the class label of x_i. Consequently, Eq. (1) can be simplified to the maximum likelihood criterion. According to the maximum conditional entropy of $p(y|x;\theta)$, it can be proved that such a maximum entropy model has the exponential form:

$$p(y|x;\theta)=\frac{1}{Z}\exp\left(\sum_k \theta_k f_k(x,y)\right), \tag{2}$$

where $f_k(x,y)$ is a feature function which depends on both the instance x and the label y, $Z=\sum_y\exp(\sum_k\theta_k f_k(x,y))$ is the normalization factor, and θ_k's are model parameters. In practice, the features usually depend on the instance but not the class label, thus Eq. (2) can be rewritten as

$$p(y|x;\theta)=\frac{1}{Z}\exp\left(\sum_k \theta_{y,k} g_k(x)\right), \tag{3}$$

where $g_k(x)$ is a class-independent feature function.

Substituting Eq. (3) into the optimization criterion used in Eq. (1) and considering the constraint $\sum_y P_i(y)=1$, the target function of θ can be written as

$$\begin{aligned}T(\theta) &= \sum_{i,y} P_i(y)\log p(y|x_i;\theta)\\ &= \sum_{i,y} P_i(y)\sum_k \theta_{y,k} g_k(x_i)\\ &\quad -\sum_i \log\sum_y \exp\left(\sum_k \theta_{y,k} g_k(x_i)\right).\end{aligned} \tag{4}$$

Because it cannot generate a closed form solution by directly setting the gradient of Eq. (4) w.r.t. θ to zero, the optimization of Eq. (4) uses a strategy similar to improved iterative scaling (IIS), a well-known algorithm for maximizing the likelihood of the maximum entropy model. IIS starts with an arbitrary set of parameters, then for each step, it updates the current estimate of the parameters θ with $\theta+\Delta$, where Δ maximizes a lowerbound of the likelihood changes between the adjacent steps. This iterative process, nevertheless, needs to be migrated to the new target function $T(\theta)$. Furthermore, the constraint needed for IIS on the feature functions $f_k(x,y)\geq 0$ (hence $g_k(x)\geq 0$) should be removed to ensure the freedom in choosing any feature extractors suitable for the data.

In detail, the change of $T(\theta)$ between the adjacent steps is

$$\begin{aligned}T(\theta+\Delta)-T(\theta) &= \sum_{i,y} P_i(y)\sum_k \delta_{y,k} g_k(x_i)\\ &\quad -\sum_i \log\sum_y p(y|x_i;\theta)\exp\left(\sum_k \delta_{y,k} g_k(x_i)\right),\end{aligned} \tag{5}$$

where $\delta_{y,k}$ is the increment of $\theta_{y,k}$. Applying the inequation

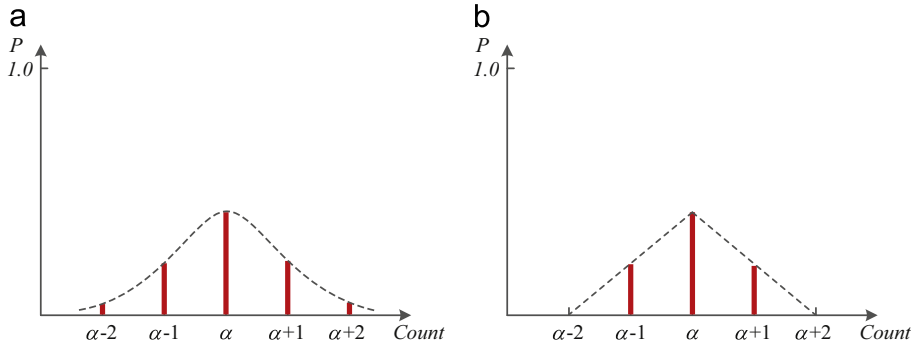


Fig. 7. Typical label distributions for people-count α used in this work. (a) Gaussian distribution and (b) triangle distribution.

$-\log x \geq 1-x$ yields

$$T(\theta+\Delta)-T(\theta) \geq \sum_{i,y} P_i(y) \sum_k \delta_{y,k} g_k(x_i) + n - \sum_{i,y} p(y|x_i;\theta) \exp\left(\sum_k \delta_{y,k} g_k(x_i)\right). \quad (6)$$

Differentiating the right side of Eq. (6) w.r.t. $\delta_{y,k}$ yields coupled equations of $\delta_{y,k}$ which are hard to be solved. To decouple the interaction between $\delta_{y,k}$, Jansen's inequality is applied here, i.e., for a p.d.f. $p(x)$:

$$\exp\left(\sum_x p(x) q(x)\right) \leq \sum_x p(x) \exp q(x). \quad (7)$$

The last term of Eq. (6) can then be written as

$$\sum_{i,y} p(y|x_i;\theta) \exp\left(\sum_k \delta_{y,k} s(g_k(x_i)) g^{\#}(x_i) \frac{|g_k(x_i)|}{g^{\#}(x_i)}\right), \quad (8)$$

where $g^{\#}(x_i) = \sum_k |g_k(x_i)|$ and $s(g_k(x_i))$ is the sign of $g_k(x_i)$. Since $|g_k(x_i)|/g^{\#}(x_i)$ can be viewed as a p.d.f., Eq. (6) can be rewritten as

$$T(\theta+\Delta)-T(\theta) \geq \sum_{i,y} P_i(y) \sum_k \delta_{y,k} g_k(x_i) + n - \sum_{i,y} p(y|x_i;\theta) \sum_k \frac{|g_k(x_i)|}{g^{\#}(x_i)} \exp(\delta_{y,k} s(g_k(x_i)) g^{\#}(x_i)). \quad (9)$$

Denote the right side of Eq. (9) as $A(\Delta|\theta)$ which is a lower-bound of $T(\theta+\Delta)-T(\theta)$. Setting the derivative of $A(\Delta|\theta)$ w.r.t. $\delta_{y,k}$ to zero gives

$$\frac{\partial A(\Delta|\theta)}{\partial \delta_{y,k}} = \sum_i P_i(y) g_k(x_i) - \sum_i p(y|x_i;\theta) g_k(x_i) \exp(\delta_{y,k} s(g_k(x_i)) g^{\#}(x_i)) = 0. \quad (10)$$

The advantage of about Eq. (10) is that $\delta_{y,k}$ appears alone, and therefore can be solved one by one through nonlinear equation solvers, such as the Gauss–Newton method. This algorithm is called IIS-LDL and summarized in Algorithm 1.

Algorithm 1. IIS-LDL.

Input: The training set $S = \{(x_i, P_i(y))\}_{i=1}^n$, the feature functions $g_k(x)$
Output: The conditional p.d.f. $p(y|x;\theta)$

1 Initialize the model parameter vector θ^0;
2 $i \leftarrow 0$;
3 **repeat**
4 $\quad i \leftarrow i+1$;
5 $\quad$ Solve Eq. (10) for $\delta_{y,k}$;
6 $\quad \theta^i \leftarrow \theta^{i-1} + \Delta$;
7 **until** $T(\theta^i) - T(\theta^{i-1}) < \varepsilon$;
8 $p(y|x;\theta) \leftarrow \frac{1}{Z} \exp(\sum_k \theta^i_{y,k} g_k(x))$.

After $p(y|x)$ is learned from the training set, given a new instance x', its label distribution $p(y|x')$ can be first calculated. The availability of the explicit label distribution for x' provides many possibilities in classifier design. To name just a few, if the expected class label for x' is single, then the predicted label could be $y^* = \arg\max_y p(y|x')$, together with an confidence measure $p(y^*|x')$. If multiple labels are allowed, then the predicted label set could be $L = y|p(y|x') > \xi$, where ξ is a predefined threshold. Moreover, all the labels in L can be ranked according to their probabilities.

5. Experiments

5.1. Datasets and evaluation settings

Datasets: Experiments are conducted on two benchmarking datasets: the UCSD dataset and the Mall dataset which represent the outdoor and the indoor scene respectively. The UCSD dataset was collected from a stationary digital camcorder overlooking a pedestrian walkway at University of California, San Diego (UCSD). The Mall dataset was captured using a publicly accessible surveillance camera in a shopping mall. There are some detailed information of the two datasets in Table 1. As shown in Fig. 8, both of these datasets contain the perspective distortion, the objects occlusion, and the objects shadow. Specifically, the abnormal pedestrians have more effect in the UCSD dataset, while the challenging lighting condition and the glass surface reflection influence more in the Mall dataset.

Evaluation protocol: For the UCSD dataset, we employed Frames 601–1400 for training and the rest for testing. For the Mall dataset, the first 800 frames are used to train and the remaining 1200 frames to test. The above settings follow the same training and testing partition as in [12].

Evaluation metrics: We utilized three evaluation metrics as in [11], namely *mean absolute error* (MAE) ε_{abs}, *mean squared error* (MSE) ε_{sqr}, and *mean deviation error* (MDE) ε_{dev}:

$$\varepsilon_{abs} = \frac{1}{N}\sum_{i=1}^{N} |v_i - \tilde{v}_i|, \quad \varepsilon_{sqr} = \frac{1}{N}\sum_{i=1}^{N} (v_i - \tilde{v}_i)^2, \quad \varepsilon_{dev} = \frac{1}{N}\sum_{i=1}^{N} \frac{|v_i - \tilde{v}_i|}{v_i},$$

where N is the total number of test frames, v_i is the actual number of people, and $\tilde{v}_i$ is the predicted number of people in ith frame.

5.2. Comparison among different distributions

Three types of distribution models are employed in our experiments. In detail, the single distribution only utilizes the real label, the triangle distribution utilizes these class labels within a triangle, and the Gaussian distribution utilizes all class labels. In addition to the coverage types of distributions, the performance of LDL method can also be affected by the parameters of label distributions. Fig. 9 shows the MAE of IIS-LDL on the Gaussian distribution with different standard deviations $\sigma = 0, 1, \ldots, 4$, the triangle distribution with different bottom lengths $l = 2, 4, \ldots, 16$, and the single distribution expressed by $\sigma = 0$ or $l = 2$. Furthermore, looking into the three distributions that IIS-LDL works on, the MAE can be ranked as triangle < Gaussian < single. The comparison between the ground truth and the optimal prediction is shown in Fig. 10.

The MAE associated with different distributions and parameters in Fig. 9 proves that different datasets have various optimal parameters, according to the insufficiency and imbalance degree of training data. In detail, with the number of people growing, the number of training samples in Mall dataset varies more smoothly than the one in UCSD dataset, as shown in Fig. 11. Whilst, the optimal parameters of Mall dataset is more concentrative than the one of UCSD dataset. For instance, the experimental results on the Mall dataset obtain the best performance when $\sigma = 1$ in the

Table 1
Dataset properties: N_f is the number of frames, R is the resolution, FPS is the frame per second, D is the density (minimum and maximum number of people in the ROI), and T_p is the total number of pedestrian instances.

Data	N_f	R	FPS	D	T_p
UCSD	2000	238*158	10	11–46	49 885
Mall	2000	640*480	< 2	13–53	62 325

Fig. 8. (a) and (b) are example samples in the UCSD dataset, and (c) and (d) are example samples in the Mall dataset.

Gaussian distribution and $l=4$ in the triangle distribution. However, the optimal parameters for the UCSD dataset are $\sigma=3$ and $l=12$. Consequently, it is obvious that the optimal distribution will be more dispersive when the insufficiency and imbalance of training data appear more serious. For a certain dataset, too concentrative or too dispersive distribution would lead to the performance deterioration, which is consistent with our imagine that the related classes are helpful but should not threaten the priority of the real class. Thus, the scale of the distribution is crucial to achieve a good performance.

5.3. Comparison with state-of-the-arts

Table 2 compares the performances of seven methods, all based on regression and using the two benchmarking datasets. For experimental results on the UCSD dataset, the MAE decreases to 2.08 when the regression function is generated by the IIS-LDL method. For the Mall dataset, the MAE further decreases to 2.69 when employing the IIS-LDL method, which is a significant improvement compared with other methods.

As for the feature-based regression approaches, a major challenge for generating a satisfactory regression function is insufficient and imbalanced training data. IIS-LDL method reuses the information of training data in the learning process, which aims to reduce the effect of insufficiency and imbalance of training data. Compared with other six methods, IIS-LDL method is able to increase the training data significantly. By labelling training samples with label distributions, the information of a training sample contributes to not only the learning of its real class, but also the learning of its neighboring classes. Because of the IIS-LDL method, the classes with insufficient training data are supplied with more training data, which belong to their adjacent classes. Thus, the IIS-LDL method can generate a better regression function, which leads to an excellent performance.

Specifically, the CA-RR method performs slightly better than the IIS-LDL method on UCSD dataset, but far less than that on Mall dataset. The result proves that both label distributions and cumulative attributes are suitable to the problem of crowd counting, however, the IIS-LDL method is more robust than CA-RR method due to its steady performance on two datasets.

5.4. Against insufficient and imbalanced training data

According to the discussion in Section 1, the insufficient and imbalanced training data is one of the major challenges for learning a good regression function. The insufficiency of training data reflects in its little number of samples belonging to some classes, while the imbalance means the samples of different classes have a great quantity variance. The utilization of label distribution learning for crowd counting aims to reduce the effect of insufficiency and imbalance, then generates an adequate regression function.

In this subsection, we conduct two extra experiments to verify the effect of label distributions against the insufficiency and imbalance of training data. The first experiment aims to confirm the effect of label distributions against the insufficiency of training data, in other words, it will show that label distributions indeed perform better than single label on insufficient training data.

In the first experiment, it picks up the most insufficient 10 classes from training data in two datasets, then abstracts the predicted results associated with these classes in testing data from aforementioned experimental results. Finally, evaluation of these extracted results and comparison of the performance of three distributions are shown in Tables 3 and 4.

Observably, the performances of Gaussian distribution and triangle distribution are better than that of single distribution, which successfully evidences the effect of label distributions on insufficient training data. Based on the theory of label distribution learning, a training sample contributes to not only the learning of its real label but also the learning of its neighboring labels. Thus, the insufficient training data can be supplied by the samples belonging to neighboring classes in the learning process. In this way, it reduces the effect of insufficiency and generates a better

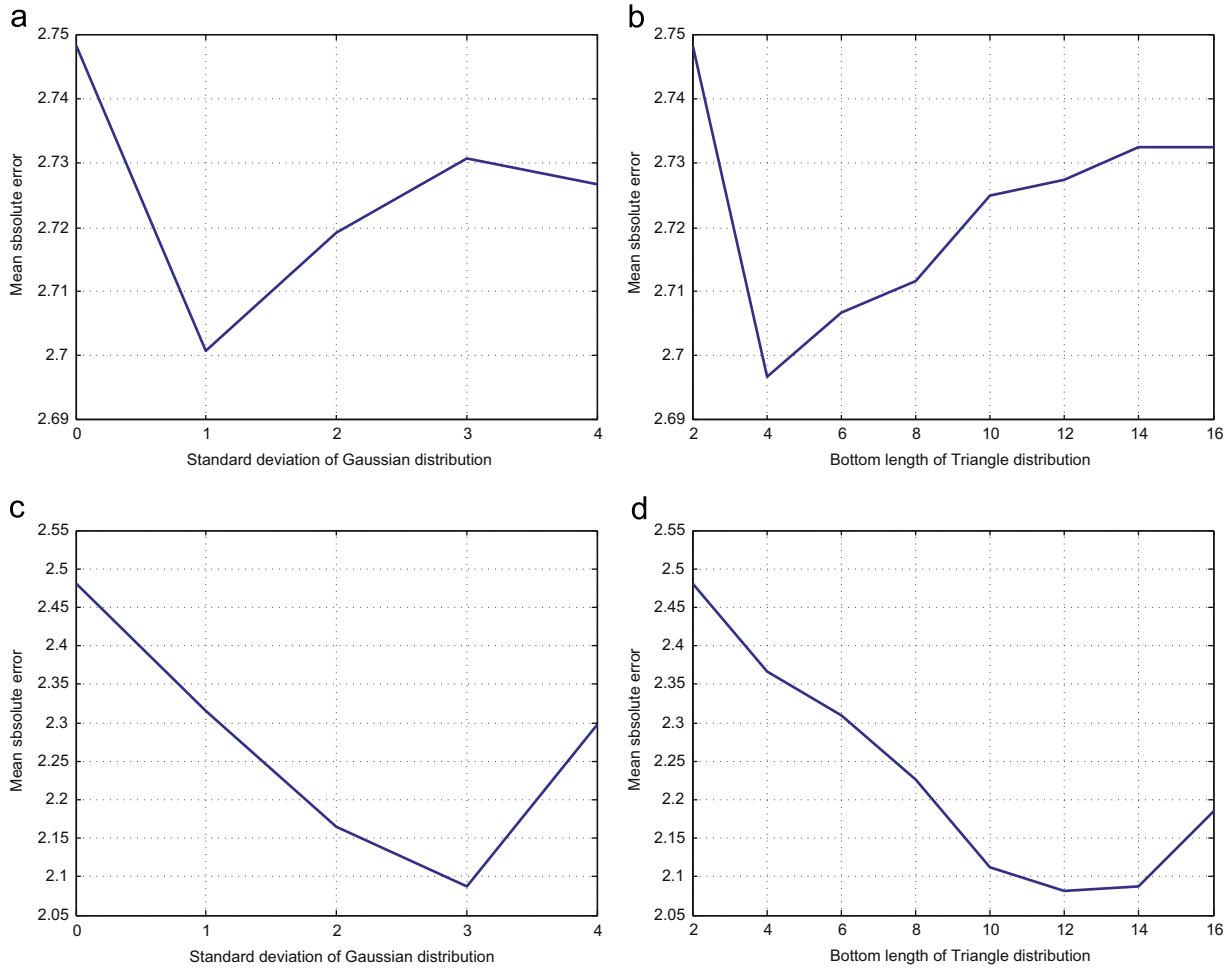


Fig. 9. The MAE of IIS-LDL on Gaussian distribution with different δ, triangle distribution with different l and single distribution on Mall dataset and UCSD dataset. (a) The performance of Gaussian distribution on Mall dataset, (b) the performance of triangle distribution on Mall dataset, (c) the performance of Gaussian distribution on UCSD dataset, and (d) the performance of triangle distribution on UCSD dataset.

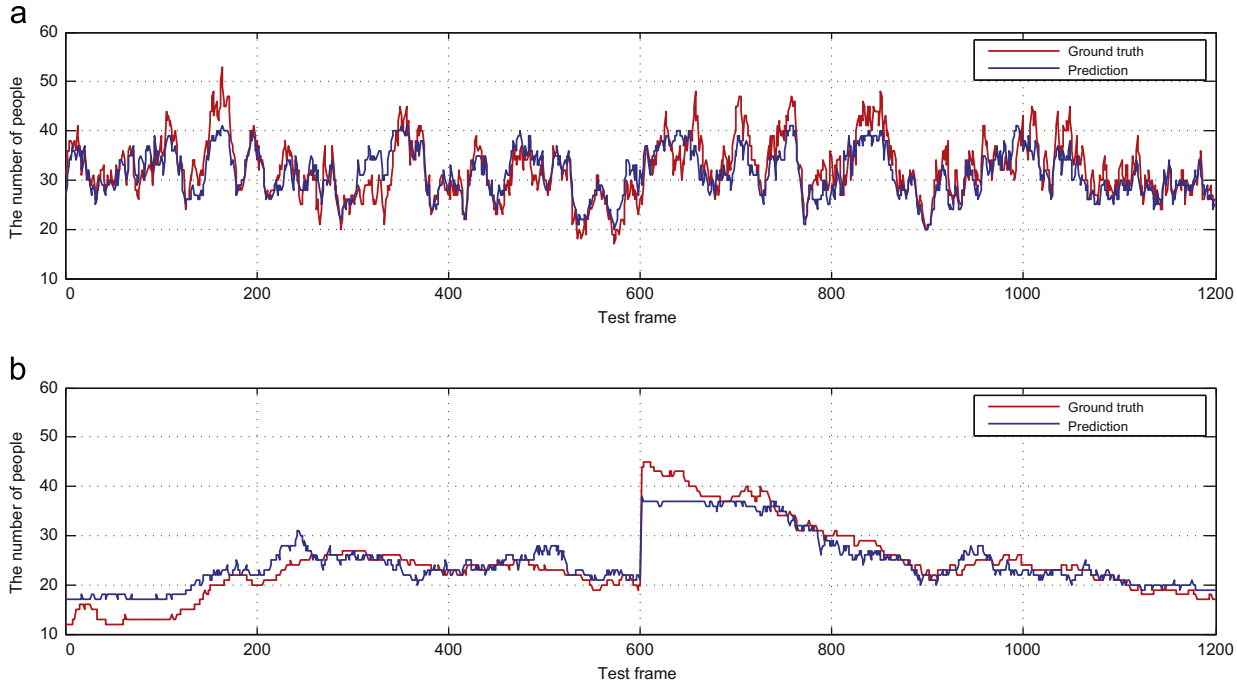


Fig. 10. Comparison between ground truth and optimal prediction. (a) Mall dataset and (b) UCSD dataset.

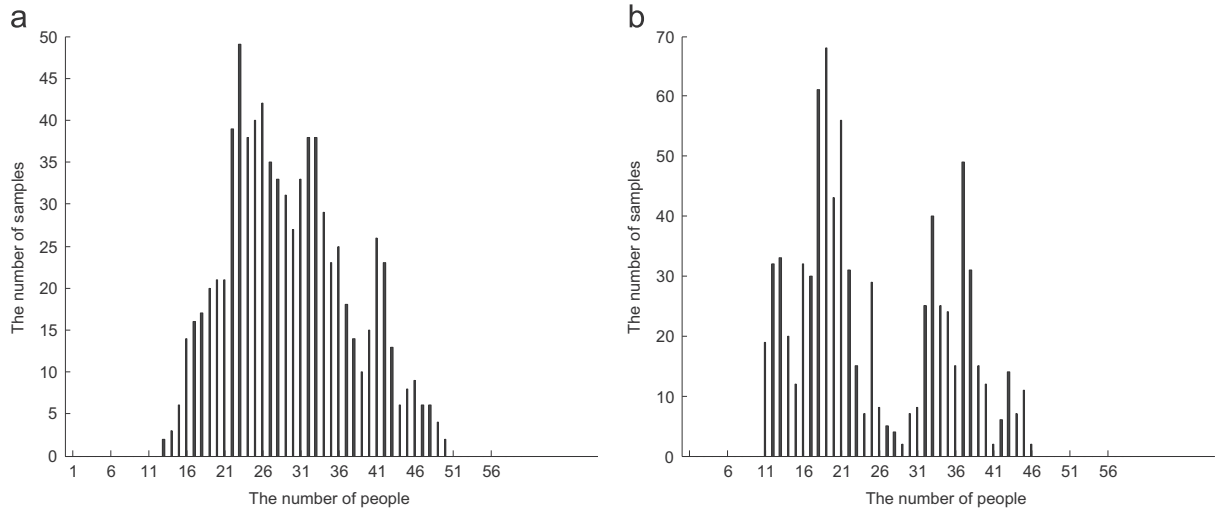


Fig. 11. Distributions of training data. (a) Mall dataset and (b) UCSD dataset.

Table 2
Performances comparison.

Methods	UCSD			Mall		
	MAE	MSE	MDE	MAE	MSE	MDE
KRR	2.16	7.45	0.107	3.51	18.1	0.108
RFR	2.42	8.47	0.116	3.91	21.5	0.121
GPR	2.24	7.97	0.112	3.72	20.1	0.115
RR	2.25	7.82	0.110	3.59	19.0	0.110
MORR	2.29	8.08	0.109	3.15	15.7	0.097
CA-RR	2.07	6.86	0.102	3.43	17.7	0.105
IIS-LDL	**2.08**	**7.25**	**0.098**	**2.69**	**12.1**	**0.082**

Table 3
Performances comparison of three distributions against insufficient training data in the UCSD dataset (N_{train} is the number of training data associated with the label, and N_{test} is the number of testing data).

Label	N_{train}	N_{test}	MAE		
			Single	Gaussian	Triangle
24	7	144	5.32	2.04	1.88
26	8	68	6.74	1.62	1.31
27	5	28	7.46	1.50	1.46
28	4	10	9.00	2.20	2.00
29	2	19	9.79	2.84	2.84
30	7	28	8.86	2.39	2.54
41	2	4	4.00	4.00	4.00
42	6	12	5.00	5.00	5.00
44	7	7	7.00	6.86	6.86
46	2	0	–	–	–

Table 4
Performances comparison of three distributions against insufficient training data in the Mall dataset.

Label	N_{train}	N_{test}	MAE		
			Single	Gaussian	Triangle
13	2	0	–	–	–
14	3	0	–	–	–
15	6	0	–	–	–
44	6	21	4.38	3.48	3.14
47	6	9	6.00	5.68	5.78
49	4	1	8.00	7.00	7.00
50	2	1	9.00	8.00	8.00
51	0	0	–	–	–
52	0	0	–	–	–
53	0	1	12.00	11.00	11.00

regression function, which makes the performance better than single label learning.

Furthermore, the effect of label distributions against insufficient training data is different in degree due to the distribution of training data. For instance, label distribution learning has an unsatisfied performance on the minimum classes and maximum classes in the dataset, due to the lack of adjacent classes and the corresponding training data. As shown in Tables 3 and 4, the comparison result indicates that label distribution learning has an outperformance on classes far from the boundary classes (the distributions of training data in Mall dataset and UCSD dataset are shown in Fig. 11), i.e. classes from 24 to 30 in Table 3. However, label distribution learning has a poor performance on classes near the boundary classes, i.e. classes from 41 to 46 in Table 3 and from 47 to 53 in Table 4.

The other experiment evaluates the effect of label distributions against the imbalance of training data. In the experiment, training data are approximately divided into seven groups according to their labels, namely, a label group is composed of each five adjacent labels. In order to make the training data more imbalanced, the label-based group is removed from training data one by one (one removed group represents 12% missing label percentage). Obviously, the performances of three distributions degrade when more training data are removed, as shown in Fig. 12. However, the Gaussian distribution and triangle distribution perform better than single label. As a result of the LDL method, even training data are removed, the corresponding classes can be learned from the training data with neighboring labels. For instance, when the label group which contains the training samples with labels 15–20 is removed, the training samples with labels 21–30 can contribute to the learning of classes 15–20 because of the label distributions. In this way, the insufficient training data can be supplied significantly, which reduce the imbalance of training data.

In a word, the insufficiency and imbalance of training data are correlative. Supplying training data for the classes which have insufficient training data reduces the imbalance of training data. Thus, the LDL method could handle the imbalance of training data effectively by increasing training data significantly in the learning process.

In addition, according to the insufficiency and imbalance degree of training data, the label distribution has various optimal parameters.

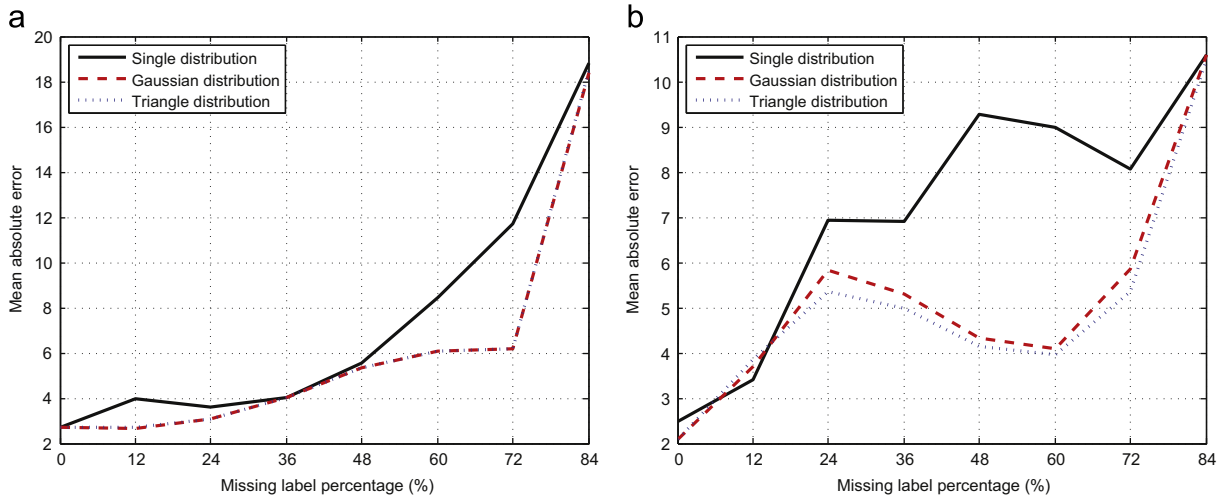


Fig. 12. Performances comparison on insufficient and imbalanced training data measured by mean absolute error (the lower the better). (a) Mall dataset and (b) UCSD dataset.

The more serious the insufficiency and imbalance of training data appear, the more dispersedly the label distribution covers class labels. Because more class labels can be covered if it uses a more dispersive label distribution to describe the class of a training sample, and the corresponding training data are supplied in the learning process. Experimental results verify our aforementioned ratiocination that label distributions are able to address the challenge of insufficient and imbalanced training data, and label distribution learning method is suitable for crowd counting.

6. Conclusions

This paper adopts the label distribution learning method into the problem of crowd counting, where it labels training data with label distributions and makes a training sample to contribute to not only the learning of its real class but also the learning of its neighboring classes. In this way, the training data are increased significantly, and the classes with insufficient training data are supplied with more training data, which belong to their adjacent classes. As for the label distribution, apart from the type of coverage, the parameters can also affect the performance of LDL method. The optimal distribution for a dataset will be more dispersive when the insufficiency and imbalance of training data appear more serious. Experimental results confirm the effectiveness of LDL method for crowd counting, and the strong robustness to different datasets contain insufficient and imbalanced training data. Though in this paper, we mainly focus on the crowd counting problem, the LDL method could be also applied to other problems which have the following two characteristics: (1) instances with adjacent classes are correlative and (2) the training data is insufficient and imbalanced.

Acknowledgment

This work is funded by the National Natural Science Foundation of China (Nos. 61375036, 61273300, 61232007), the Beijing Natural Science Foundation (No. 4132064), the Jiangsu Natural Science Funds for Distinguished Young Scholar (BK20140022), the Program for New Century Excellent Talents in University, the Beijing Higher Education Young Elite Teacher Project, the Fundamental Research Funds for the Central Universities, and the Key Lab of Computer Network and Information Integration of Ministry of Education of China. Zhaoxiang Zhang is the corresponding author of this paper.

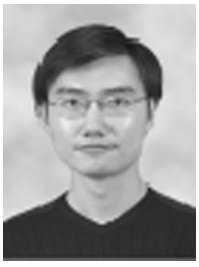
References

[1] Chen Change Loy, Ke Chen, Shaogang Gong, Tao Xiang, Crowd counting and profiling: methodology and evaluation, in: Modeling, Simulation and Visual Analysis of Crowds, Springer, 2013, pp. 347–382.
[2] Antoni B. Chan, Z.-S.J. Liang, Nuno Vasconcelos, Privacy preserving crowd monitoring: counting people without people models or tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, IEEE, 2008, pp. 1–7.
[3] Navneet Dalal, Bill Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, vol. 1, IEEE, 2005, pp. 886–893.
[4] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, Deva Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[5] Tao Zhao, Ramakant Nevatia, Bo Wu, Segmentation and tracking of multiple humans in crowded environments, IEEE Trans. Pattern Anal. Mach. Intell. 30 (7) (2008) 1198–1211.
[6] Danny B. Yang, Héctor H González-Baños, Leonidas J. Guibas, Counting people in crowds with a real-time network of simple image sensors, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003, IEEE, 2003, pp. 122–129.
[7] Vincent Rabaud, Serge Belongie, Counting crowded moving objects, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2006, pp. 705–711.
[8] Gabriel J. Brostow, Roberto Cipolla, Unsupervised Bayesian detection of independent motion in crowds, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2006, pp. 594–601.
[9] Anthony C. Davies, Jia Hong. Yin, Sergio A. Velastin, Crowd monitoring using image processing, Electron. Commun. Eng. J. 7 (1) (1995) 37–47.
[10] Paul Geladi, Bruce R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.
[11] Ke Chen, Chen Change Loy, Shaogang Gong, Tony Xiang, Feature mining for localised crowd counting, in: BMVC, vol. 1, 2012, p. 3.
[12] Ke Chen, Shaogang Gong, Tao Xiang, Chen Change Loy, Cumulative attribute space for age and crowd density estimation, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 2467–2474.
[13] Craig Saunders, Alexander Gammerman, Volodya Vovk, Ridge regression learning algorithm in dual variables, in: (ICML-1998) Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann, 1998, pp. 515–521.
[14] Xin Geng, Chao Yin, Zhi-Hua Zhou, Facial age estimation by learning from label distributions, IEEE Trans. Pattern Anal. Mach. Intell. 35 (10) (2013) 2401–2412.
[15] Bastian Leibe, Edgar Seemann, Bernt Schiele, Pedestrian detection in crowded scenes, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, vol. 1, IEEE, 2005, pp. 878–885.
[16] Oncel Tuzel, Fatih Porikli, Peter Meer, Pedestrian detection via classification on Riemannian manifolds, IEEE Trans. Pattern Anal. Mach. Intell. 30 (10) (2008) 1713–1727.
[17] Paul Viola, Michael J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[18] Bo Wu, Ramakant Nevatia, Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors, in: Tenth

IEEE International Conference on Computer Vision, 2005, ICCV 2005, vol. 1, IEEE, 2005, pp. 90–97.
[19] Payam Sabzmeydani, Greg Mori, Detecting pedestrians by learning shapelet features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, CVPR'07, IEEE, 2007, pp. 1–8.
[20] Paul Viola, Michael J. Jones, Daniel Snow, Detecting pedestrians using patterns of motion and appearance, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003, IEEE, 2003, pp. 734–741.
[21] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, Victor Lempitsky, Hough forests for object detection, tracking, and action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2188–2202.
[22] Sheng-Fuu Lin, Jaw-Yeh Chen, Hung-Xin Chao, Estimation of number of people in crowded scenes using perspective transformation, IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. 31 (6) (2001) 645–654.
[23] Bo Wu, Ram Nevatia, Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors, Int. J. Comput. Vis. 75 (2) (2007) 247–266.
[24] Min Li, Zhaoxiang Zhang, Kaiqi Huang, Tieniu Tan, Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection, in: 19th International Conference on Pattern Recognition, 2008, ICPR 2008, IEEE, 2008, pp. 1–4.
[25] Weina Ge, Robert T. Collins, Marked point processes for crowd counting, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, CVPR 2009, IEEE, 2009, pp. 2913–2920.
[26] Weina Ge, Robert T. Collins, Crowd detection with a multiview sampler, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 324–337.
[27] Yassine Benabbas, Nacim Ihaddadene, Tarek Yahiaoui, Thierry Urruty, Chabane Djeraba, Spatio-temporal optical flow analysis for people counting, in: 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2010, pp. 212–217.
[28] Jingwen Li, Lei Huang, Changping Liu, Robust people counting in video surveillance: dataset and system, in: 2011 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), IEEE, 2011, pp. 54–59.
[29] Antoni B. Chan, Daxiang Dong, Generalized Gaussian process models, in: CVPR, 2011, pp. 2681–2688.
[30] Yang Cong, Haifeng Gong, Song-Chun Zhu, Yandong Tang, Flow mosaicking: Real-time pedestrian counting without scene-specific learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, CVPR 2009, IEEE, 2009, pp. 1093–1100.
[31] Christopher M. Bishop, et al., Pattern Recognition and Machine Learning, vol. 1, Springer, New York, 2006.
[32] Senjian An, Wanquan Liu, Svetha Venkatesh, Face recognition using kernel ridge regression, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, CVPR'07, IEEE, 2007, pp. 1–7.
[33] John Shawe-Taylor, Nello Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[34] Xinyu Wu, Guoyuan Liang, Ka Keung Lee, Yangsheng Xu, Crowd density estimation using texture analysis and learning, in: IEEE International Conference on Robotics and Biomimetics, 2006, ROBIO'06, IEEE, 2006, pp. 214–219.
[35] Antoni B. Chan, Nuno Vasconcelos, Bayesian poisson regression for crowd counting, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 545–551.
[36] Tsung-Yi Lin, Yen-Yu Lin, Ming-Fang Weng, Yu-Chiang Wang, Yu-Feng Hsu, H.-Y.M. Liao, Cross camera people counting with perspective estimation and occlusion handling, in: 2011 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2011, pp. 1–6.
[37] Andy Liaw, Matthew Wiener, Classification and regression by randomforest, R News 2 (3) (2002) 18–22.
[38] A. Criminisi, J. Shotton, E. Konukoglu, Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-supervised Learning, Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114, 5(6):12, 2011.
[39] Chen Change Loy, Shaogang Gong, Tao Xiang, From semi-supervised to transfer counting of crowds, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2256–2263.
[40] Robert M. Haralick, Karthikeyan Shanmugam, Its' Hak Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. (6) (1973) 610–621.
[41] John Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1986) 679–698.

Zhaoxiang Zhang received his B.S. degree in electronic science and technology from the University of Science and Technology of China, Hefei, China in 2004 and Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009, respectively. After that he joined the School of Computer Science and Engineering, Beihang University, Beijing 100191, China, as a faculty. He is now an Associate Professor in the School of Computer Science and Engineering, the vise-director of the Department of Computer Application Technology. His research interest include computer vision, pattern recognition and image processing. He is the corresponding author of this paper.

Mo Wang received his B.E. degree in computer science and technology from Beihang University, China (BUAA) in 2013, and is a M.E. candidate in School of Computer Science and Engineering, Beihang University, China.



Xin Geng is currently a professor and the director of the PALM lab (http://palm.seu.edu.cn/) of Southeast University, China. He received the B.Sc. (2001) and M.Sc. (2004) degrees in computer science from Nanjing University, China, and the Ph.D. (2008) degree in computer science from Deakin University, Australia. His research interests include pattern recognition, machine learning, and computer vision. He has published 38 refereed papers in these areas, including those published in prestigious journals and top international conferences. He has been an Associate Editor or Guest Editor of several international journals, such as FCS, PRL and IJPRAI. He has served as a Program Committee Chair for several international/national conferences, such as PRICAI'18, JSAI'14, VALSE'13, etc., and a Program Committee Member for a number of top international conferences, such as CVPR, ICCV, IJCAI, AAAI, ACMMM, ECCV, etc.