Hindawi
Journal of Electrical and Computer Engineering
Volume 2017, Article ID 2580860, 12 pages
https://doi.org/10.1155/2017/2580860

Research Article

Crowd Density Estimation of Scenic Spots Based on Multifeature Ensemble Learning

Xiaohang Xu,[1,2] Dongming Zhang,[1,2] and Hong Zheng[1,2]

[1]*School of Electronic Information, Wuhan University, 129 Luoyu Road, Wuhan, Hubei 430072, China*
[2]*Shenzhen Institute of Wuhan University, Shenzhen, Guangdong 518057, China*

Correspondence should be addressed to Hong Zheng; zh@whu.edu.cn

Received 25 December 2016; Revised 30 March 2017; Accepted 8 May 2017; Published 8 June 2017

Academic Editor: Panajotis Agathoklis

Estimating the crowd density of public territories, such as scenic spots, is of great importance for ensuring population safety and social stability. Due to problems in scenic spots such as illumination change, camera angle change, and pedestrian occlusion, current methods are unable to make accurate estimations. To deal with these problems, an ensemble learning (EL) method using support vector regression (SVR) is proposed in this study for crowd density estimation (CDE). The method first uses human head width as a reference to separate the foreground into multiple levels of blocks. Then it adopts the first-level SVR model to roughly predict the three features extracted from image blocks, including D-SIFT, ULBP, and GIST, and the prediction results are used as new features for the second-level SVR model for fine prediction. The prediction results of all image blocks are added for density estimation according to the crowd levels predefined for different scenes of scenic spots. Experimental results demonstrate that the proposed method can achieve a classification rate over 85% for multiple scenes of scenic spots, and it is an effective CDE method with strong adaptability.

1. Introduction

With increased living standard and the constant acceleration of urbanization progress, collective activities in large scale public places are becoming more and more frequent. In recent years, frequent accidents have been caused by dense human crowds. Therefore, using computer vision to intelligently monitor human crowds, make timely warnings, and take effective measures plays an essential role in social stability and population safety.

Current human crowd density estimation (CDE) methods are mainly divided into two categories.

(1) Direct Methods. Certain classifiers are used to directly segment or detect each human body in crowds. Then the bodies are counted to obtain human crowd density. These methods can be further divided into two subtypes.

(i) Model Based Methods. Detection or segmentation is performed according to models or shape contour of human bodies. For example, Lin et al. proposed a pedestrian detection method based on extracting human head contour feature by Haar wavelet transform combined with support vector machine [1]. Felzenszwalb et al. proposed a deformable parts model (DPM) detection method based on parts and improved histogram of oriented gradient (HOG) features [2]. Gall and Lempitsky proposed a method using Hof forest framework to detect and grade each part of pedestrians to determine pedestrians and their positions [3]. Gardzinski et al. used cameras in multiple view angles for 3D foreground modeling [4] and extract the human bodies and determine the number of people according to the shape of the human bodies.

(ii) Trajectory Clustering Based Methods. Each body is detected according to the long time tracking and clustering of interesting points on human bodies. For example, Rabaud and Belongie proposed a method for clustering the trajectories and inferring people counts in scenes using Kanade-Lucas-Tomasi (KLT) tracker and a series of

low-level features [5]. Rao et al. [6] obtained crowd contours by optical flow methods and screen out the human trajectories from movement information. Then human crowd density is clustered and analyzed. Direct methods perform well in situations where there are small numbers of people. However, the drawback is clear; that is, in crowded situations where humans seriously overlap, the performances of direct methods fall sharply.

(2) Indirect Methods. These methods treat the crowd as a whole and estimate the crowd density by extracting features of the crowd such as textures along with the use of regression models. Indirect methods can also be divided into three subtypes.

(i) Pixel Based Analysis. These methods first remove scene backgrounds and then use some very low-level features to estimate crowd density. Davies et al. [14] analyzed crowd foregrounds and edge pixels by extracting foregrounds. Perspective correction was added and people counts were estimated by linear relation. Hussasin et al. [15] corrected the perspective distortion of foreground pixels by zooming, extracted low-level features, and used backward neural networks for supervised training. The trained model is accurate for sparse crowd, but the estimation error sharply increases with the increase of density and occlusion.

(ii) Texture and Gradient Based Methods. Compared with pixel based methods, texture and gradient can better represent the number of people in a scene. Texture and gradient features are used in CDE, including gray-level cooccurrence matrix (GLCM) [16], local binary pattern (LBP) [17], HOG, and gradient orientation cooccurrence matrix (GOCM) [18].

(iii) Interest Points Based Methods. Interest points are feature pixels of interest, such as corner points detected in images. For example, Conte et al. [19] used speeded-up robust features (SURF) to detect corner points, and the number of moving corner points was used to estimate crowd density. Liang et al. [20] used three-frame differential algorithm and binarization to form foreground templates. Then SURF was used to extract feature points. Finally optical flow was applied to determine the direction of crowd moving and density. Kishor et al. [21] detected features for accelerated segment test (FAST) corner points in optical flow images. Then the density estimation image was formed according to the corner points.

However, in actual scenic spots, the existing method is difficult to make accurate prediction due to problems such as illumination change, camera angle and height inconsistency, pedestrian continuous crowding, and severe occlusion. To deal with these problems, the unique character of scenic spot monitoring is considered, and a CDE method based on multifeature ensemble learning (EL) is proposed. Experiments demonstrate that compared with people count estimation method commonly used in recent years, the proposed method not only achieves relatively good results for public data sets, but also performs well in experimental scenic spot scenes, showing strong adaptability for scenes.

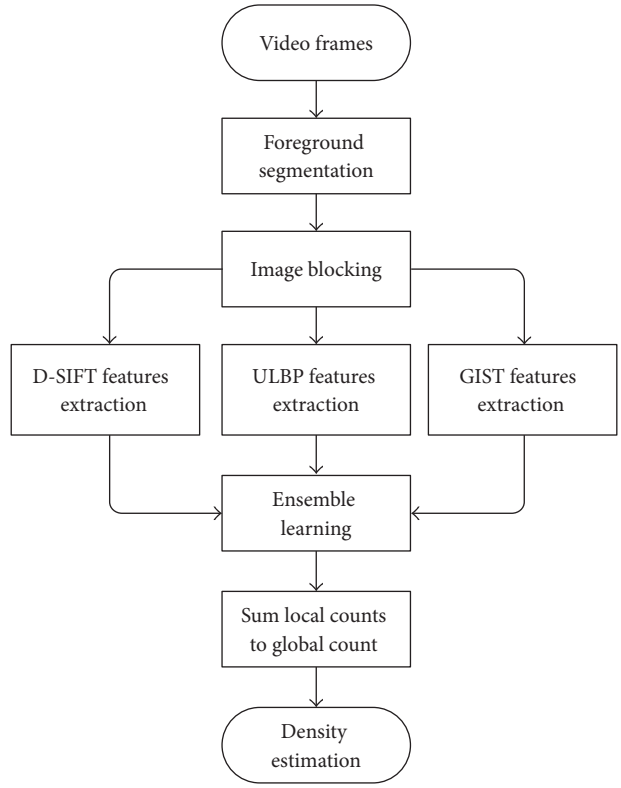


Figure 1: Flow chart of the methodology.

The structure of this paper is as follows:

(1) Section 1 introduces the development of CDE.

(2) Section 2 introduces the algorithm flow of this paper.

(3) In Section 3, an image blocking method is introduced to solve the perspective projection change problem, and then feature descriptors used in this paper are presented. Finally, the EL method is proposed and the grade standard is introduced.

(4) In Section 4, experiments are conducted to verify the effectiveness of this method.

(5) In Section 5, the method is summarized and advantages and disadvantages of it are expounded.

2. Algorithm Description

In summary, CDE methods should solve the following problems. (1) How to solve the perspective projection change problem caused by camera tilt? (2) What features should be chosen to represent the people count of a crowd? (3) What classifier should be used for features to classify different crowd density or what regression method should be used to associate features with the people count of a crowd?

To address the above problems, a density estimation method based on multifeature EL is developed, with its flow chart shown in Figure 1. The detailed steps are as follows.

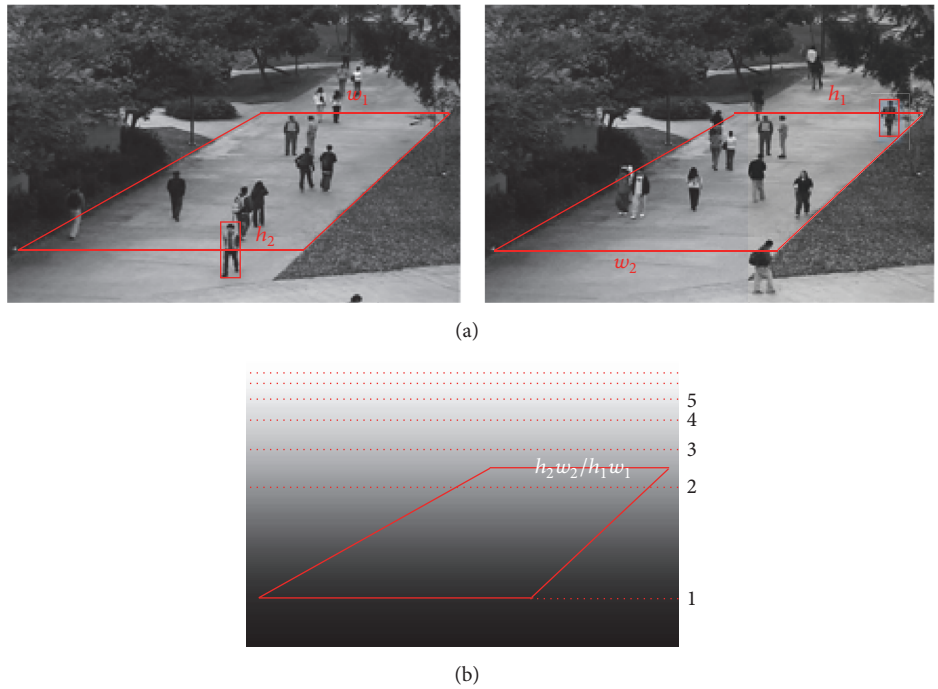


FIGURE 2: (a) Reference persons at the nearest and farthest point in the labeled ground plane area and (b) the calculated weight map.

(1) Foreground segmentation is a widely used method in density estimation because it can reduce the background interference. Ordinarily, for example, mixture of Gaussians-based technique [22] or mixture of dynamic textures-based method [23] is used to obtain the foreground.

(2) To solve the problem of camera perspective projection change, image blocking method is adopted in this study. The separated blocks (subimages) are resized to a uniform size.

(3) In order to better describe the people count of crowds, three descriptors, including dense-SIFT [24], uniform LBP [25], and GIST [26], are used in this paper.

(4) Support vector regression (SVR) is used to fit the subimage features and people counts, and an EL method is proposed, which uses two-level SVR model to predict the local people counts of subimages.

(5) Finally, the local people counts of all subimages are added, and the density is estimated according to the people counts set for different scenic spot scenes.

3. CDE Method for Scenic Spots Based on EL

3.1. Solving Projection Perspective Change by Image Blocking. In scenic spot monitoring environment, monitoring cameras are often installed in high places, with a certain tilt angle to the horizontal plane. Therefore, perspective projection effects exist in the collected images. There are two effects of perspective projection. (1) The farther the object is from the camera, the smaller the object appears, and thus it has a smaller area and fewer edge pixels in the image. (2) Crowds farther away are denser with more occlusion, which poses new difficulties for counting people or density estimation.

Currently, methods solving perspective projection change are divided into the following types.

(1) Perspective Transformation Method. The parameters of the camera are first acquired, including focal length, pixel size, and the angle between the principle optic axis and the horizontal plane. These parameters can also be obtained by camera calibration methods [27]. Then the perspective weighting model can be calculated. Finally, the images are transformed into vertical perspective [5] or mapped into geographic information system (GIS) space [28] for processing. In [1], human heads sizes in different positions were estimated by perspective model for detection and screening.

(2) Feature Weighting Method. First, ground plane areas are labeled in images, usually with the shape of a trapezoid. Assume the lengths of the upper base and the lower base are measured as w_1 and w_2, respectively. Then, for the same person in a video, the heights of the person at the top and the bottom of the region are measured as h_1 and h_2, respectively, as shown in Figure 2(a). The weights of the upper and bottom bases are from h_2w_2/h_1w_1 to 1. The weights of other positions can be obtained by linear interpolation. Finally, a weight map can be obtained, as shown in Figure 2(b). Multiplying the weight map with the extracted features can solve the problem of perspective projection effect [8, 11, 29].

(3) Blocking Methods. The sizes of the nearest and farthest blocks of the region of interest are first selected. The body height of a reference pedestrian is used as the side length of

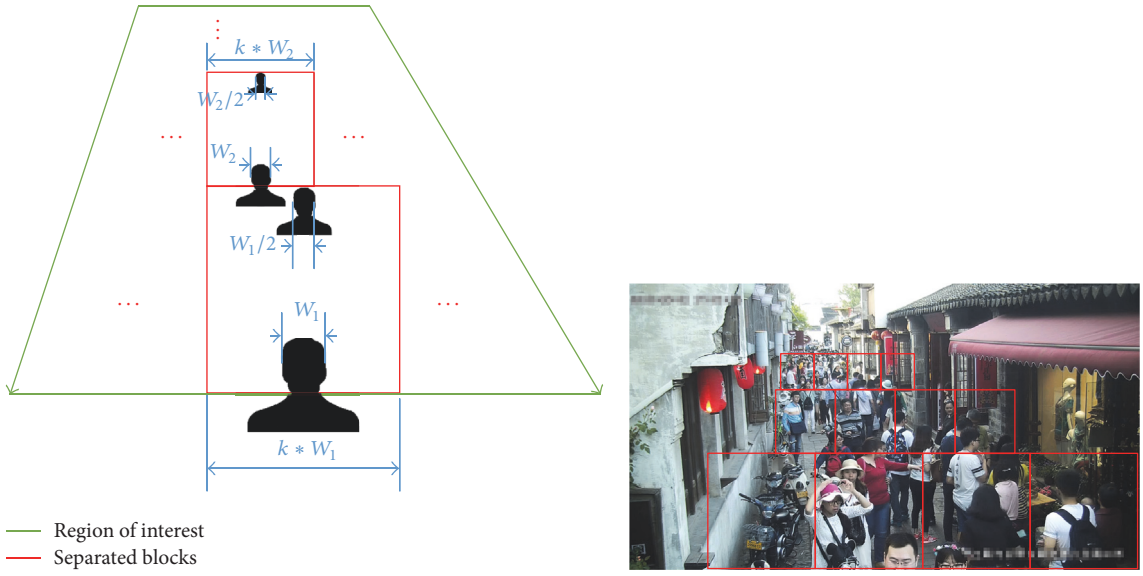


Figure 3: (a) Blocking method diagram and (b) blocking of a scenic spot scene.

a square block. The sizes of intermediate blocks are obtained by linear interpolation [18, 30].

Perspective transformation methods or feature weighting methods require training the models for different scenes. Traditional blocking methods [18, 30] use pedestrian height as the reference standard of blocking, and problems exist in real applications where blocks of bottom parts are too big, which makes the overall block hierarchy less strong. To solve the above problems, repeated tests and parameters adjustment are carried out in different scenic spot scenes, and human head width is eventually used as a reference in this study. The blocking method established from near to distant is developed and shown in Figure 3(a), and the detailed steps are as follows.

Select a reference pedestrian. When his/her head enters the region of interest, the head width is measured as W_1 pixels, and the width of block in the bottom part is determined as $k * W_1$. Then the reference pedestrian continues walking until the head width becomes $W_1/2$ pixels, and the distance from the head top to the bottom of the interested region is the height of bottom block. Repeat this blocking method to determine the size of blocks in the middle and distant part of a scene. Finally, repeat blocking in every level to cover the entire interest field. Figure 3(b) shows the blocking of a scenic spot scene.

After resize to a uniform size, the separated blocks (subimages) can be trained as unified samples. This method not only solves the problem of perspective projection, but also increases the scalability and adaptability in other scenes.

3.2. Feature Descriptor for People Count of Crowds. Since crowds usually exhibit typical texture and dense corner points, three descriptors, including D-SIFT, ULBP, and GIST, are used in this study to describe the people count of crowds.

3.2.1. Dense-SIFT Feature Descriptor and Dimensionality Reduction. SIFT feature is a method for detecting local features [31]. This algorithm extracts extreme points (feature points) by constructing difference of Gaussian (DOG) scale space and extracts the scale and direction parameters for extreme points, forming the descriptor. Due to its scale invariance, SIFT is widely applied in image retrieval. Moreover, D-SIFT feature [24] is the improvement of SIFT, which directly samples the region at specified feature points and calculates SIFT feature. The main process is as follows.

(1) Read a gray-level image, and use a patch to slide on the image with a sampling step. This patch is the sampling area of the descriptor. The size of the patch is $4 * \eta_x * 4 * \eta_y$, where the size of each cell, $\eta_x * \eta_y$, can be predefined. The patch moves over all areas it can reach, that is, the region enclosed by the bounding box which can be described by the feature. The overall geometric process is shown in Figure 4.

(2) In each patch, the gradient of each pixel is calculated. The gradient histogram of the pixels in 8 directions in each cell is calculated, resulting in a feature dimension of $4 * 4 * 8$ for each patch. Concatenating the features of all patches generates D-SIFT feature.

For an image with width W and height H, let the horizontal and vertical steps be δ_x and δ_y, respectively. The dimensionality calculation equation is then

$$D_x = \text{ceil}\left(\frac{W - (4-1) * \eta_x}{\delta_x}\right),$$
$$D_y = \text{ceil}\left(\frac{W - (4-1) * \eta_y}{\delta_y}\right), \quad (1)$$
$$D = D_x * D_y,$$

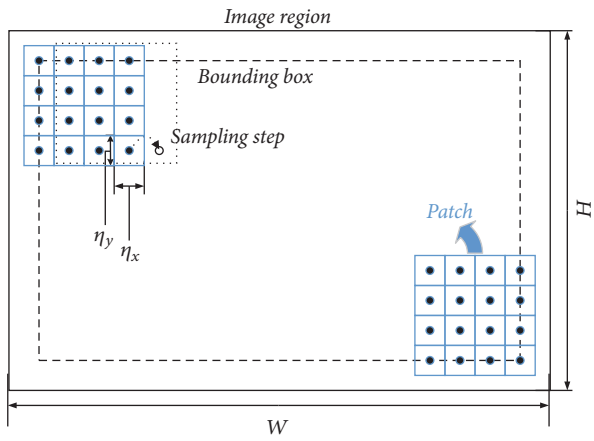


Figure 4: Dense-SIFT descriptor geometry.

where

(i) D is the total dimensionality;

(ii) D_x and D_y are directional dimensionalities;

(iii) ceil(x) is right rounding function.

For an image of size 128 ∗ 128, let $\delta_x = \delta_y = 1$, and $\eta_x = \eta_y = 3$, and the dimensionality is then 119 ∗ 119 ∗ 128.

Due to the high dimensionality of D-SIFT feature, it is unsuitable for direct use. In this study, BOF method [32] is adopted. BOF model imitates bag of words method in the texture retrieval field. It describes an image as an unordered set of local features, as shown in Figure 5(a). Then, it uses a clustering method (such as K-means) to cluster local features, as shown in Figure 5(b). Each clustering center is viewed as a visual word, and all clustering centers form the visual vocabulary composed of all the visual words. Such corresponding relation is also called codebook. Each local feature of an image is mapped to a certain word in the visual vocabulary. This mapping can be determined by calculating the distance between the feature and center points. Then the occurrence frequency of visual words is counted, as shown in Figure 5(c). The features of each image can be represented by a histogram with its dimensionality being the cluster number, as shown in Figure 5(d).

After processing the original D-SIFT features by BOF, the dimensionality is determined by the number of cluster centers, that is, the predefined word number. In the proposed scheme, 500 or 1000 words are usually set. Therefore, it is clear that BOF can significantly reduce the dimensionality of D-SIFT features.

3.2.2. ULBP Feature Descriptor. LBP (local binary pattern) is an operator that describes the local texture feature of an image. It has significant advantages such as rotation invariance and grayscale invariance. For a circular area that has a center of (x_c, y_c) (gray value is g_c) and radius of R, with P pixels (gray values are g_p $(p = 0, 1, \ldots, P)$) in it, the LBP operator can be calculated as shown in [17] by

$$\mathrm{LBP}_{P,R} = \sum_{p=0}^{P-1} s\left(g_p - g_c\right) 2^P, \tag{2}$$

where

$$s(x) = \begin{cases} 1, & x > 0, \\ 0, & x \le 0. \end{cases} \tag{3}$$

Obviously, an LBP operator $\mathrm{LBP}_{P,R}$ can produce 2^P kinds of binary mode. In order to solve this problem, Ojala et al. [25] proposed a method called uniform pattern to reduce the dimension of it. Uniform pattern refers to a binary sequence that jumps from 0 to 1 or from 1 to 0 no more than twice. In this way, the LBP operator is calculated as

$$\begin{aligned} U\left(\mathrm{LBP}_{P,R}\right) &= \left|s\left(g_{P-1} - g_c\right) - s\left(g_0 - g_c\right)\right| \\ &\quad + \sum_{p=1}^{P-1} \left|s\left(g_p - g_c\right) - s\left(g_{p-1} - g_c\right)\right|. \end{aligned} \tag{4}$$

Ojala et al. held that the LBP value jumps no more than twice in an actual image. Under this consideration, let $U(\mathrm{LBP}_{P,R}) \le 2$ for an original LBP operator. This adjustment greatly reduces the number of the binary modes from 2^P to $P(P-1)+2$, while retaining most of the image information. This is ULBP, which is good in terms of computational speed and classification accuracy. It is robust to rotation and scale transformation and illumination change. And it is suitable for classification.

3.2.3. GIST Feature Descriptor. A large number of psychological studies have proved that humans can extract information from an image within 100 ms to determine the class of a scene or obtain the global feature of a scene, which is also called the GIST of a scene [33].

In later development, people aim to use algorithms to implement such GIST features. Oliva and Torralba proposed to use multiscale and multidirection Gabor filters to filter scene images, extract contour information of scenes, and form GIST features [26]. This method achieved good results in scene classification.

Multiscale multidirection Gabor filters are based on normal Gabor filter $g(x, y)$, with scale and rotation transformation introduced, and belong to the category of self-similar Gabor wavelet processing. The formulae [33] are

$$\begin{aligned} g_{mn}(x, y) &= a^{-m} g\left(x', y'\right), \quad \alpha > 1, \\ x' &= a^{-m}\left(x\cos\theta + y\cos\theta\right), \\ y' &= a^{-m}\left(-x\cos\theta + y\cos\theta\right), \\ \theta &= \frac{n\pi}{(n+1)}, \end{aligned} \tag{5}$$

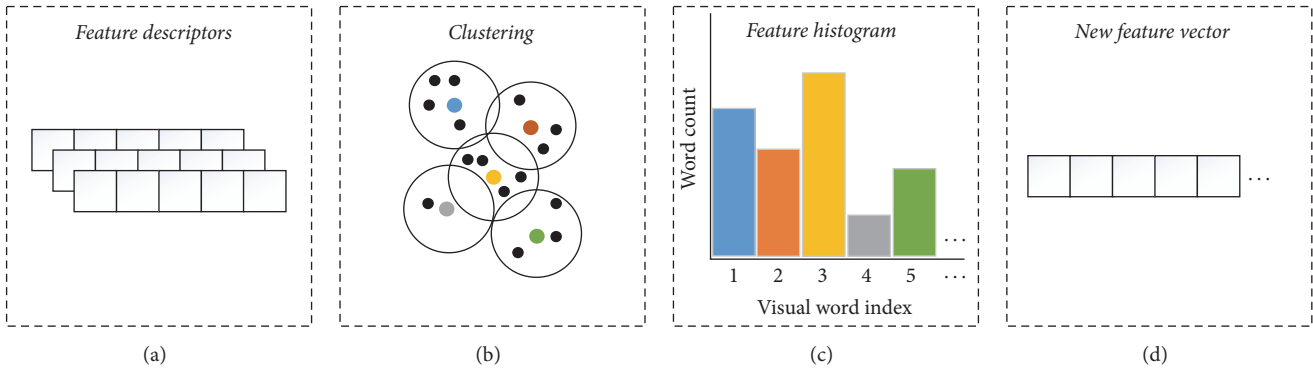


FIGURE 5: The bag of features diagram: (a) the feature descriptors extracted from images, (b) clustering, (c) the formation of word frequency statistics histogram, and (d) the new feature from the feature histogram.

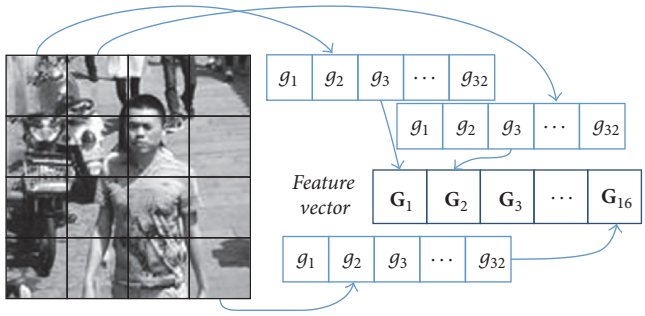


FIGURE 6: GIST feature extraction diagram.

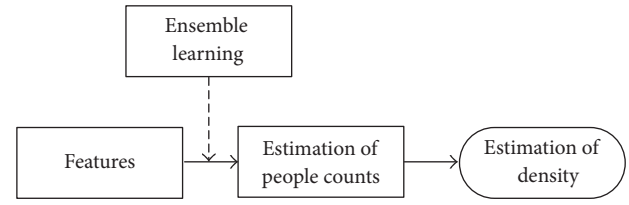


FIGURE 7: The process from features to density estimation.

where

(i) a^{-m} is the scale factor of mother wavelet expansion;

(ii) θ is the rotation angle, that is, the filter direction;

(iii) m is the number of scales;

(iv) n is the number of directions.

Divide an image with $w * h$ pixels into $n_g * n_g$ grids, with each grid denoted as $P_1, P_2, \ldots, P_{n_g^2}$. A total of $m * n$ channels of Gabor filters are used for convolution filtering of the image. Then the average values of the filtered image are calculated. All the average values are concatenated to form a vector, and the vector is GIST feature, with dimensionality $n_g * n_g * m * n$.

In this study, $4 * 4$ grids are used, as shown in Figure 6. In each grid, filtering is performed for 4 scales and 8 directions, and then the average of the filtered image is calculated, forming features with 32 dimensions $(g_1, g_2, \ldots, g_{32})$. The features of all grids are concatenated, resulting in $(G_1, G_2, \ldots, G_{16})$, that is, GIST feature, with $4 * 4 * 4 * 8 = 512$ dimensions.

3.3. EL Method for CDE. After features are extracted from subimages, we adopt a procedure where people count is estimated first followed by density estimation, as shown in Figure 7.

By EL, the features are converted into the estimation of the people counts in crowds. Then, the people counts of all subimages of an image are summed and classified, finally obtaining the density estimation.

3.3.1. People Number Estimation of Crowds. SVR is used in this study to establish the relation between features and people counts in subimages and then determine the people count of a crowd. Support vector machines search the optimum between model complexity and learning ability according to limited sample information and obtain good generalization ability. In SVR, the relation between prediction values and image features can be expressed as follows [34]:

$$f(x) = \sum_{i=1}^{n} (a_i - a_i^*) K(x_i, x) + b, \tag{6}$$

where

(i) $f(x)$ is the prediction value;

(ii) x is the feature vector;

(iii) x_i is the ith support vector;

(iv) a_i and a_i^* are the Lagrange multipliers of the ith support vector;

(v) $K(x_i, x)$ is the kernel function;

(vi) n is the number of support vectors.

Radial basis function (RBF) is usually used for kernel function [35]:

$$K(x_i, x) = \exp\left(-\lambda \left\| x - x_i \right\|^2\right), \tag{7}$$

where λ is the kernel parameter.

In this study, an EL method combining SVR is proposed to improve the prediction accuracy. EL accomplishes learning tasks by establishing and combining multiple learning

Input:
Initial training set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p), (x_{p+1}, y_{p+1}), \ldots, (x_m, y_m)\}$;
where $x = [x_{i\text{D-SIFT}}, x_{i\text{ULBP}}, x_{i\text{GIST}}]$, $1 \le i \le m$;
Primary SVR algorithm ξ_1, ξ_2, ξ_3;
Secondary SVR algorithm ξ.
Process:
for each $t \in [1, 3]$ do
//use part of the training set
let $D_t = \{(x_i[t], y_i), 1 \le i \le p\}$;
//primary learning machines training
$h_t = \xi_t(D_t)$;
end for
$D' = \varnothing$; //secondary traing set
for $i = p + 1, p + 2, \ldots, m$ do
for each $t \in [1, 3]$ do
//predict with primary learning machines
$z_{it} = h_t(x_i[t])$;
end for
$D' = D' \cup ((z_{i1}, z_{i2}, z_{i3}), y_i)$;
end for
//secondary learning machine training
$h^i = \xi(D')$;
Output:
$H(x) = h'(h_1(x_{\text{D-SIFT}}), h_2(x_{\text{ULBP}}), h_3(x_{\text{GIST}}))$.

Algorithm 1: The learning based combination strategy.

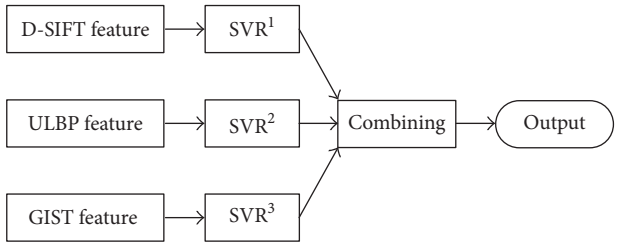


Figure 8: EL diagram.

machines. Currently, EL methods are categorized into two major classes [36]: (1) individual learning machines having strong dependent relation, which must be serialized into a serialization method; (2) individual learning machines not having strong dependent relation, which can form a parallel method.

The three features selected in this study have certain differences. Therefore, the parallel method is adopted, as shown in Figure 8.

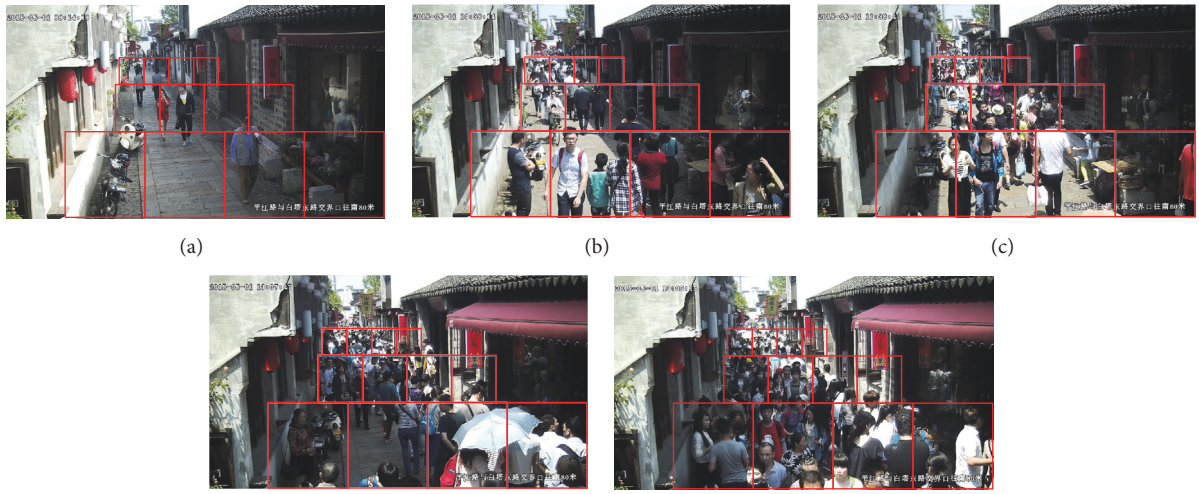
The three features extracted in Section 3.3 are denoted as $x_{\text{D-SIFT}}$, x_{ULBP}, and x_{GIST} and are used for training individual learning machines by SVR algorithm, also called basic regression machines, denoted by h_1, h_2, and h_3. Their goals are consistent, that is, to approximate the actual function f. Combining learning machines can avoid the problem of weak generalization ability of a single learning machine, and the combining strategy is also important. Common combining strategies include averaging method, simple voting, and Bayesian voting.

The method of learning is used as the combining strategy, that is, using a learning machine for combination. Here, the individual learning machines are called primary learning machines, while the learning machine used for combination is called the secondary learning machine. The learning based combining algorithm is shown in Algorithm 1.

At training stage, the secondary training set is generated by primary learning machines. If the secondary train uses the same training set as that of the primary training, the risk of overfitting is high. Therefore, only part of the training set is used for training the primary learning machines, and prediction results from the primary learning machines are used as the training set of the secondary learning machine. In this way, the overfitting problem is solved. As shown in Algorithm 1, the initial training set D is divided into two parts. One part is divided into D_1, D_2, and D_3 according to D-SIFT, ULBP, and GIST features, and SVR algorithm with different parameters is used to train the primary learning machines based on the three features, respectively. The coarse prediction results z_{it} of the primary learning machines are combined with their labels y_i from the other part of D to form the training set of the secondary learning machine. Finally, secondary SVR algorithm is used for training the secondary learning machine.

At the prediction stage, $x_{\text{D-SIFT}}$, x_{ULBP}, and x_{GIST} are coarsely predicted by the primary learning machines, generating results $h_1(x_{\text{D-SIFT}})$, $h_2(x_{\text{ULBP}})$, and $h_3(x_{\text{GIST}})$. Then, the results are combined and predicted by the secondary learning machine h', resulting in the fine prediction result $H(x)$.

Considering that different features have different sensitivity to crowd density, we adopt two levels of regression to



(a) (b) (c) (d) (e)

Figure 9: The different density images on scene 1: (a) very low, (b) low, (c) medium, (d) high, and (e) very high.

compensate the drawbacks of each of them and improve the prediction accuracy.

3.3.2. Density Estimation. For a test image, the predicted people counts of all blocks are added to generate the estimation of the people count of this image.

It is worthily noted that adding the predicted people count of all blocks results in quadratic error. And an estimation of density attracts more of our interest than an exact number of people counts. Therefore, a classification method is adopted after adding the predicted people count of all blocks.

In this study, common five-level classification is used to convert the people count into density estimation.

In this paper, different classification standards are set for different scenes. The maximum number of people $n_{\max}$ contained in the scene is divided into 5 equal intervals: $[0, n_{\max}/5]$, $[n_{\max}/5, 2n_{\max}/5]$, $[2n_{\max}/5, 3n_{\max}/5]$, $[3n_{\max}/5, 4n_{\max}/5]$, and $[4n_{\max}/5, \infty)$, corresponding to 5 levels of crowd density denoted as VL (very low), L (low), M (middle), H (high), and VH (very high). Figure 9 shows the frames of different density levels.

4. Experiment Results and Analysis

To verify the efficacy of feature selection for prediction accuracy improvement, blocking method, EL, and the adaptability to scenes, comparison tests have been made on multiple scenes in Pingjiang scenic spot of Suzhou City and UCSD public crowd data set [37].

In the data set of Pingjiang scenic spot of Suzhou City, scene 1 shown in Figure 9 is used to establish the data set, and a total of 1500 foreground frames in different periods are extracted from the monitoring video. Among them, 900 images are used as the training samples, and the other 600 images are test samples. First, the images are blocked, people count for all blocks and all images are labeled, and the density levels of all images are annotated. The corresponding grading standard is used (too few: 0~15 persons, few: 16~30 persons, medium: 31~45 persons, many: 46~60 persons, and too many: 61~75 persons) to calibrate the density levels.

In UCSD public crowd data set, a video with a total of 2000 frames is contained, which was captured in University of California of San Diego, along with the people count annotations for all frames. Similar to other algorithms [7–13], in this study, the 601st to 1400th frames are used as training samples, and the remaining 1200 frames are test samples. Likewise, the foreground of the 2000 frames is extracted and blocked, and the people count of each block is obtained through the annotations.

For the comparison of people count estimation, mean absolute error (MAE) and mean relative error (MRE) are used as the criteria [7–13].

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{n}\sum_{i=1}^{n} |P(i) - G(i)|, \\ \mathrm{MRE} &= \frac{\sum_{i=1}^{n} |P(i) - G(i)|}{\sum_{i=1}^{n} G(i)} \times 100\%, \end{aligned} \tag{8}$$

where

(i) n is the number of tested samples;

(ii) $G(i)$ is the calibrated people count, which is for reference;

(iii) $P(i)$ is the people count predicted by the algorithm.

4.1. Comparison of Different Features and Combined Feature. The performances of different individual features in people count prediction in block samples are compared. The experiment is conducted on the block samples from the data set of Pingjiang scenic spot of Suzhou City. Six features including HOG, original LBP (LBP), GLCM, D-SIFT, ULBP, and GIST are used. The subimages of the 900 training samples are used

Table 1: Comparison between different individual features.

Feature	HOG	LBP	GLCM	ULBP	GIST	D-SIFT
MAE	1.28	1.15	1.12	1.06	0.97	**0.85**
MRE (%)	21.7	19.5	19.0	18.0	16.4	**14.4**

Note. Bold indicates the best performance.

Table 2: Comparison between multifeature combinations.

Combination method	MAE	MRE (%)
ULBP + GIST	0.92	15.6
ULBP GIST EL	0.85	14.4
ULBP + D-SIFT	0.82	13.9
GIST + D-SIFT	0.81	13.7
ULBP + D-SIFT + GIST	0.78	13.2
ULBP D-SIFT EL	0.78	13.2
GIST D-SIFT EL	0.72	12.2
D-SIFT ULBP GIST EL	**0.64**	**10.8**

Note 1. "+" indicates cascading combination, others for EL. *Note 2. Bold* indicates the best performance.

Table 3: The classification accuracy of global method (%).

Scene 1	VL	L	M	H	VH
VL	**89.86**	10.14			
L	10.32	**73.02**	16.67		
M		13.22	**77.69**	8.26	0.83
H			8.40	**68.07**	23.53
VH			4.17	25.00	**70.83**

to train the model. The remaining subimages of the 600 test samples are used to predict and compare with the ground truth. The results are shown in Table 1.

It is clear from Table 1 that, for individual features, D-SIFT, GIST, and ULBP perform better, which is the reason for choosing these three features in this study.

Then, different combinations of D-SIFT, GLCM, and GIST are compared. Two-feature combinations are compared with the three-feature combination, and cascading combination is compared with EL. In order to ensure the fairness of the comparison, for cascading combination, the training and test sample is exactly the same as that of the individual feature. For EI, since it uses two-level SVR model, half of the training samples are selected as each level's regression training samples, and the test sample is the same as cascading combination. Table 2 shows the comparison result of different feature combinations.

It is clear from Table 2 that EL method is more accurate than simple cascading. Using three features in EL is more accurate than using two features. Therefore, EL method is effective.

4.2. Comparison of the Blocking Method and the Global Method. In order to verify the accuracy of the proposed blocking method, a global estimation method is designed to contrast.

This global method differs from the proposed method in that it removes the blocking step by extracting the ULBP, GIST, and D-SIFT features directly from the foreground of the region of interest. 450 training samples are used to train the first level of SVR model along with the people count annotations. Then the second level of SVR was trained with the output of the first level with another 450 training samples. Finally, 600 test samples are predicted by using the two-level SVR, and the classification results are compared with the ground truth. Results are shown in Table 3.

In the proposed method, the 600 test samples are divided into blocks, and the model trained by EL with the three features in Section 3.2 is used to predict the people count of each block. Then, the predicted people count of each block is summed together. The grading standard is used to obtain the density estimation, which is compared with the ground truth. The classification accuracy of the proposed method is shown in Table 4.

Experimental results show that using the proposed method can achieve an average accuracy of 91.67%, while using the local analysis method only achieves an average accuracy of 76.5%. This indicates that the proposed blocking method is indeed capable of solving the perspective effect problem.

4.3. Comparison with Other Algorithms. To further verify the effectiveness of the proposed method, algorithms in [7–13] and the proposed method are tested on UCSD public data set.

Table 4: The classification accuracy of our method (%).

Scene 1	VL	L	M	H	VH
VL	**99.28**	0.72			
L	1.59	**96.03**	2.38		
M		7.44	**87.60**	4.96	
H			8.40	**85.71**	5.88
VH				12.50	**87.50**

Table 5: Comparison with other algorithms on UCSD dataset.

Method	MAE	MRE (%)
Wu et al. [7]	2.60	14.2
Chan et al. [8]	2.30	12.6
Zhang et al. [9]	2.08	11.3
Chen et al. [10]	2.07	11.3
Proposed	1.95	10.7
Lempitsky and Zisserman [11]	**1.70**	**9.3**
Hu et al. [12]	1.98	10.8
Zhang et al. [13]	**1.60**	**8.7**

Table 6: The accuracy of our method on scene 2 (%).

Scene 2	VL	L	M	H	VH
VL	**97.10**	2.90			
L	4.76	**87.30**	7.94		
M		4.92	**88.52**	6.56	
H			6.78	**83.05**	10.17
VH				18.75	81.25

Table 7: The accuracy of our method on scene 3 (%).

Scene 3	VL	L	M	H	VH
VL	**95.83**	4.17			
L	4.92	**88.52**	6.56		
M		6.15	**90.77**	3.08	
H			5.17	**84.48**	10.34
VH			2.27	15.91	**81.82**

The comparison of the accuracy of people count prediction is shown in Table 5. It should be clarified that results of method in [7–13] have all been reported in the original papers.

It is clear in Table 5 that the proposed method achieves preferable accuracy in the UCSD database, which is second only to the method in [11]. It is still considerably acceptable even compared to the neural network method in [12, 13].

4.4. Scene Adaptability Test. To further verify the adaptability of the model, other scenes are tested.

Figure 10 shows experiment result of scene 2 from 08:00 to 18:00 in a single day. It claims that the classification accuracy reaches around 90%.

The 5-class classification accuracy of 300 test samples of two selected scenes is shown in Tables 6 and 7. The diagram of the two scenes is shown in Figure 11.

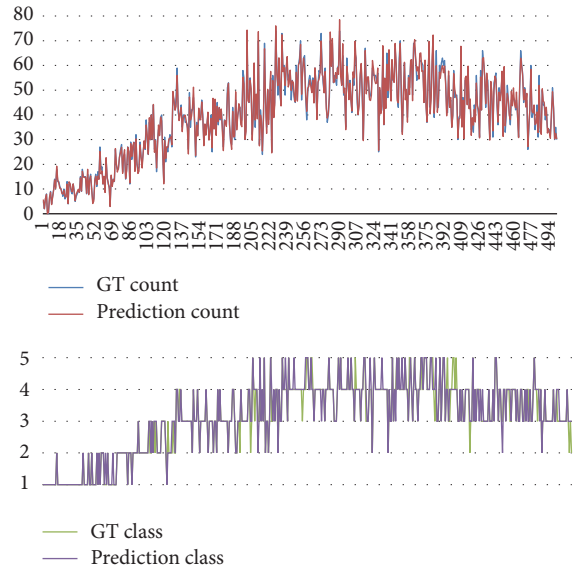


Figure 10: Ten-hour prediction data of scene 2.

Experimental results show that, under different scenes, the same method can achieve accuracy over 85%. Particularly, the average classification accuracy for scene 2 and scene 3 is 88% and 89%, respectively, indicating that the proposed method has relatively strong scene adaptability.

5. Conclusion

In this study, a scenic spot crowd density estimation algorithm based on multifeature ensemble learning is proposed. The algorithm solves the perspective effect problem by introducing a new blocking method for scenic spot scenes. In each block of an image, the coarse regression prediction of people count is made by a layer of SVR model for the extracted multiple features. Then, another layer of SVR model is used on the coarse regression results for fine regression prediction. The people counts of all subimages are summed up and graded for density estimation according to the standards defined for different scenes. Experiments demonstrate that the proposed algorithm is highly robust and effective for crowd density estimation.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Figure 11: The images of scene 2 and scene 3.

Acknowledgments

This work was supported by the Shenzhen Basic Science & Technology Foundation of China (Grant no. JCYJ20150422150029095) and the Suzhou Industrial Technology Innovation Foundation of China (Grant no. SS201616).

References

[1] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.*, vol. 31, no. 6, pp. 645–654, 2001.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[3] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Decision forests for computer vision and medical image analysis*, pp. 143–157, Springer, London, UK, 2013.

[4] P. Gardzinski, K. Kowalak, L. Kaminski, and S. Mackowiak, "Crowd density estimation based on voxel model in multi-view surveillance systems," in *Proceedings of the 22nd International Conference on Systems, Signals and Image Processing (IWSSIP '15)*, pp. 216–219, September 2015.

[5] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 705–711, June 2006.

[6] A. S. Rao, J. Gubbi, S. Marusic, and M. Palaniswami, "Estimation of crowd density by clustering motion cues," *Visual Computer*, vol. 31, no. 11, pp. 1533–1552, 2014.

[7] X. Wu, G. Liang, K. K. Lee, and Y. Xu, "Crowd density estimation using texture analysis and learning," in *Proceedings of the International Conference on Robotics and Biomimetics (ROBIO' 06)*, pp. 214–219, China, December 2006.

[8] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, USA, June 2008.

[9] Z. Zhang, M. Wang, and X. Geng, "Crowd counting in public video surveillance by label distribution learning," *Neurocomputing*, vol. 166, pp. 151–163, 2015.

[10] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR' 13)*, pp. 2467–2474, USA, June 2013.

[11] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS '10)*, pp. 1324–1332, December 2010.

[12] Y. Hu, H. Chang, F. Nian, Y. Wang, and T. Li, "Dense crowd counting from still images with convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 530–539, 2016.

[13] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR' 15)*, pp. 833–841, USA, June 2015.

[14] A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," *Electronics and Communication Engineering Journal*, vol. 7, no. 1, pp. 37–47, 1995.

[15] N. Hussain, H. S. M. Yatim, N. L. Hussain, J. L. S. Yan, and F. Haron, "CDES: A pixel-based crowd density estimation system for Masjid al-Haram," *Safety Science*, vol. 49, no. 6, pp. 824–833, 2011.

[16] A. Marana, L. Costa, R. Lotufo, and S. Velastin, "On the efficacy of texture analysis for crowd monitoring," in *Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision (SIBGRAPI '98)*, pp. 354–361, Rio de Janeiro, Brazil.

[17] T. Ojala, M. Pietik Inen et al., "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[18] W. Ma, L. Huang, and C. Liu, "Crowd density analysis using co-occurrence texture features," in *Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT '10)*, pp. 170–175, Korea, December 2010.

[19] D. Conte, P. Foggia, G. Percannella, and M. Vento, "Counting moving persons in crowded scenes," *Machine Vision and Applications*, vol. 24, no. 5, pp. 1029–1042, 2013.

[20] R. Liang, Y. Zhu, and H. Wang, "Counting crowd flow based on feature points," *Neurocomputing*, vol. 133, pp. 377–384, 2014.

[21] P. V. V. Kishore, R. Rahul, K. Sravya, and A. S. C. S. Sastry, "Crowd Density Analysis and tracking," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '15)*, pp. 1209–1213, India, August 2015.

[22] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[23] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 5, pp. 909–926, 2008.

[24] D. Forsyth, P. Torr, and A. Zisserman, *Sift flow: Dense correspondence across different scenes*, Springer, Berlin, Heidelberg, Germany, 2008.

[25] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[26] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[27] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[28] S. Hongquan, L. Xuejun, L. Guonian et al., "A Cross-camera Adaptive Crowd Density Estimation Model," *China Safety Science Journal*, vol. 23, no. 12, article 139, 2013.

[29] J. Li, L. Huang, and C. Liu, "Robust people counting in video surveillance: dataset and system," in *Proceedings of the 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '11)*, pp. 54–59, September 2011.

[30] Q. Xinhui, W. Xiufei, and X. Zhou, "Counting people in various crowed density scenes using support vector regression," *Journal of Image and Graphics*, vol. 18, no. 4, pp. 392–398, 2013.

[31] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1150–1157, Kerkyra, Greece, September 1999.

[32] G. Csurka, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision (ECCV' 04)*, vol. 1(1-22), pp. 1-2, 2004.

[33] Y. Zhao, G. Jun, and X. Zhao, "Scene categorization of local Gist feature match kernel," *Journal of Image and Graphics*, vol. 18, no. 3, pp. 264–270, 2013.

[34] J. A. Smola and B. Lkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[35] B. Schölkopf, K. Tsuda, and J. Vert, "A Primer on Kernel Methods," in *Kernel Methods in Computational*, pp. 35–70, MIT Press, 2004.

[36] Z. Zhihua, *Machine Learning*, Tsinghua University Press, 2016.

[37] http://www.svcl.ucsd.edu/projects/peoplecnt/.