Implementation of real time reconfigurable embedded architecture for people counting in a crowd area

Songchenchen Gong, El-Bay Bourennane

Implementation of real time reconfigurable embedded architecture for people counting in a crowd area

Gong songchenchen and El-Bay Bourennane

Université Bourgogne Franche-Comté, Laboratoire LE2I, France
gscc19@hotmail.com, ebourenn@u-bourgogne.fr

Abstract. We propose a feature fusion method for crowd counting. By image feature extraction and texture feature analysis methods, data obtained from multiple sources are used to count the crowd. We count people in high density static images. Most of the existed people counting methods only work in small areas, such as office corridors, parks, subways and so on. Our method uses only static images to estimate the count in high density images (hundreds or even thousands of people), for example, large concerts, National Day parade. At this scale, we can't rely on only one set of features for counting estimation. Therefore, we use multiple sources of information, namely, HOG and LBP. These sources provide separate estimates and other combinations of statistical measurements. Using the support vector machine (SVM) classification technique, and regression analysis, we count the crowd with high density. The method gives good results in crowded scenes.

Keywords: HOG · LBP · SVM

1 Introduction

In recent years, due to the awareness of social security, much more attention has been paid on people counting and human detection area. With the improvement of living standard, there are various recreational activities, some of which will bring potential dangers to human safety. In fact, it is important to accurately estimate the number of people in public areas to ensure safety and prevent the overcrowding. A stampede occurred in a "love parade"electronic music festival in Germany in 2010, and another accident occurred at the Bund of Shanghai China in 2014. In order to prevent the occurrence of such accidents in overcrowded areas, it is important to accurately estimate the number of people.

2 Related work

* The HOG function is used to detect human heads. Dalal et al. firstly used this function for pedestrian detection in static images. The main idea is to estimate the gradient histogram of the local region, which is used to describe the target characteristic. Then, the HOG function is combined with Support Vector Machine (SVM) classifier to detect heads. SVM classifier and head detection are divided into two stages[1].

Fig. 1. This figure shows four arbitrary images from the dataset used in this paper

* A method extracted texture features using Gabor filters and Least Squares vector support machine. The local texture characteristics of each cell on the grid mask are more efficient than the overall characteristics with the same multichannel Gabor filters[2].

* A method is proposed that can be used to detect and count people. For feature extraction, local binary texture descriptors are applied and classified using a support vector machine (SVM)[3].

* A recognition method for head and shoulders in a complex background. It starts with Head and shoulder contour features like $'\Omega'$, gose through the coarse filtering using Haar classifiers as well as precise verification using SVM cascade classifiers based on Joint HOG features,and detection and identification of head-shoulders.It has a good detection rate and is suitable for population statistics and video surveillance, especially in public places[4].

* Dense crowds can be thought as a texture that corresponds to a harmonic pattern at fine scales. Counts from texture analysis methods: crowds are repetitive in nature since all human appear similar from a distance. We will adopt three different texture analysis methods which separately give an estimate of the count[5][6].

* Detect the crowd, use the point of interest detector to specify the image of the head area, and then count the number of people in the crowd[7][8].

* A novel integrated framework for the analysis of crowds including all relevant aspects as simulation, detection and tracking of pedestrians and dense crowds and event detection. It utilizes an appearance-based approach for object detection since this method has been successfully applied for very small objects. Track the pedestrian using the Bayesian tracking method. Graph-based event detection using hidden markov models (HMM)[9].

* A deep-learning approach to estimate the number of individuals in mid-level crowd visible in a still image. The proposed deep-learning framework is used to learn a feature-count regressor, which can estimate the number of people within each part, and the crowd estimation in whole image is therefore the sum of that in all parts[10][11].

* To use local information at pixel level substitutes a global crowd level or a number of people per-frame. The proposed approach consists of generating fully automatic and crowd density maps using local features as an observation of a probabilistic crowd function[12][13].

* According to the people counting method based on head detection and tracking to evaluate the number of people who move under an over-head camera. The main methods are: foreground extraction, head detection, head tracking, and crossing-line judgment[14][15][16].

* Based on MID foreground segmentation module, it provides active areas for the head-shoulder detection module to detect head and count the number of people[17][18].

* Human detection in crowded scenes using a Bayesian 3D model based method[19].

3 Proposed system

We propose a feature fusion method of crowd counting. By image feature extraction and texture feature analysis methods, data obtained from multiple sources are used to count the crowd. We count people in high density static images.

3.1 HOG based Head Detections

HOG is used for head detection. Locally normalized HOG descriptors perform better than existing feature sets (including wavelets). Compared to the edge direction histograms and SIFT descriptors, the HOG is calculated on grid-sized uniform cell units, and in order to improve performance, overlapping local contrast normalization is also used.

In a static image, the representation and shape of the local target can be well described by the gradient or edge direction density distribution. Its basic is the gradient of statistical information, and the gradient mainly exists on the edge of the space. In order to reduce the effect of light factors, we firstly normalize the image. For the texture intensity of the image, the proportion of the local surface exposure is larger. Therefore, the compression process can effectively reduce the image of the local shadow and light changes. Because the function of color information does not work well, images are usually converted to grayscale. We calculate the gradient of the image abscissa and the ordinate direction, as well as the gradient direction value for each pixel position. The guidance operation captures the edge, silhouette and some texture information, further weakening the lighting effects. The gradient of the pixel (x, y) in the image is:

$$G_x(x,y) = H(x+1,y) - H(x-1,y) \tag{1}$$

$$G_y(x,y) = H(x,y+1) - H(x,y-1) \tag{2}$$

Where G_x, G_y represents the horizontal gradient, vertical gradient, and $H(x,y)$ pixel values at the pixel (x, y) in the input image.The gradient and gradient directions at the pixel (x, y) are:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{3}$$

$$\partial(x,y) = tan^{-1}(\frac{G_x(x,y)}{G_y(x,y)}) \tag{4}$$

A gradient orientation histogram is constructed for each cell unit.We divide the image into several "cells", each cell consisting of 8*8 pixels and each block consisting of 2*2 cells. Suppose we use nine bin histograms to count the 8*8 pixel gradient information as shown in Figure 2:

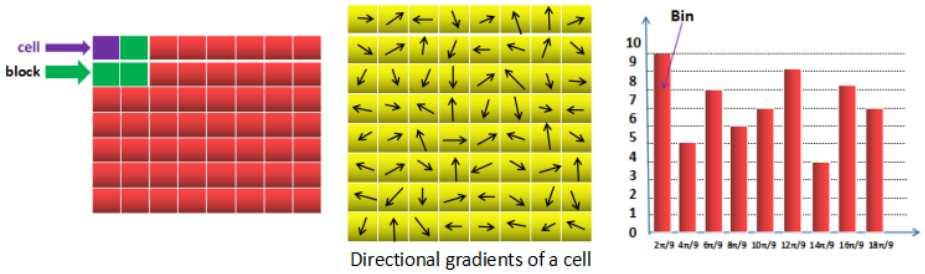

Fig. 2. The definition of a cell, a block and a bin.

We divide the gradient of the cell by 360 degrees into nine directions.Due to changes in local illumination and changes in foreground and background contrast, the range of gradient intensity is very large. This requires normalizing the gradient intensity. Normalization can further illuminate light, shadows, and edge compression. The approach we take is to combine the individual cell units into large blocks and spatially connected intervals. Normalizing the HOG feature vector in the block and introducing v to represent a vector that has not been normalized. It contains all histogram information for a given block. According to the $||V_k||$ standard, where k is 1 or 2, and e is a small constant. At this point, the normalization factor can be expressed as follows:

$$L2-norm, f = \frac{v}{\sqrt{\|V\|_2^2 + e^2}} \tag{5}$$

We refer to the normalized block descriptors (vectors) as HOG descriptors. In this way, the HOG descriptor becomes a vector consisting of the histogram components of all cell units in each interval. A vector with HOG feature dimensions.

3.2 LBP feature

The Local Binary Patterns (LBP) were first proposed by T. Ojala, M.PietikÃd'inen, and D. Harwood. LBP is a simple but very effective texture operator. It compares each

pixel with its nearby pixels and saves the result as a binary number. The most important advantage of LBP is its robustness to changes in grayscale such as illumination changes. Its other important feature is its simple calculation, which makes it possible to analyze the image in real time. The basic LBP operator is defined as the $3*3$ window. Using the value of the center pixel of the window as threshold, the gray value of the adjacent 8 pixels is compared with it. If the surrounding pixel value is greater than the central pixel value, the pixel value is of the location is marked as '1'. Otherwise it is '0'. In this way, the 8 points in the $3*3$ neighborhood can be compared to produce 8-bit binary numbers (usually converted to decimal numbers, is 256 types of LBP codes), which is to get the LBP value of the pixel in the center of the window, and use this value to reflect the texture information of the area, for example: 00010011. Each pixel has 8 adjacent pixels, and 2^8 possibilities.

The basic LBP feature for a given pixel is formed by thresholding the 3x3 neighbourhood with the centre pixel value as the threshold,where(Xc,Yc) is the center pixel,ic be the intensity of the centre pixel and in (n=0,1,2....7) pixel intensities from the neighborhood.The LBP is given by:

$$LBP(X_c, Y_c) = \sum_{n=o}^{P-1} 2^n s(i_n - i_c) \tag{6}$$

Where P is the number of sample points and:

$$s(x) = \begin{cases} 1 & if \quad x \geqslant 0 \\ 0 & else \end{cases} \tag{7}$$

The LBP could be interpreted as an 8-bit integer. The basic LBP concept is presented in Figure 3:

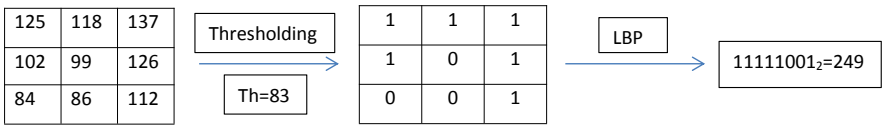

Fig. 3. Illustration of the standard LBP operator. Image taken from the Daimler pedestrian dataset.

When the LBP operator is used for texture classification or face recognition, the statistical histogram of the LBP mode is often used to express the image information, and more pattern types will make the data volume too large and the histogram too sparse. Therefore, it is necessary to reduce the dimension of the original LBP mode so as to best represent the image information in the case of a reduced data amount. In order to solve the problem of too many binary patterns and improve statistics, we proposed a "Uniform Pattern" to reduce the dimension of LBP operator's pattern. In an actual image, most LBP patterns only contain at most two transitions from '1' to '0' or from '0' to '1'. When a loop binary number corresponding to an LBP changes

from '0' to '1' or from '1' to '0' at most twice, the binary corresponding to the LBP is called an equivalent pattern class. For example, 00000000 (0 jumps), 00000111 (only one transition from '0' to '1'), and 10001111 (first jump from '1' to '0', then '0' to '1' and two jumps in total) are all equivalent modes class. The modes other than the equivalent mode class are classified as another class, called a mixed mode class, such as 10010111 (a total of four transitions). With this improvement, the variety of binary patterns is greatly reduced without losing any information. The number of modes is reduced from the original 2P to P (P-1)+2, where P represents the number of sampling points in the neighborhood set. For the 8 sampling points in the 3*3 neighborhood, the binary pattern is reduced from the original 256 to 58. This allows the feature vector to have fewer dimensions and can reduce the effects of high-frequency noise.

3.3 Training of joint HOG-LBP classifiers

Feature extraction is one of the most critical aspects of human head detection. Extracting features with distinguishing significance plays an important role in the accurate detection of the human head. Our work integrates the features of HOG and LBP, which not only combines the effective identification information of multiple features, but also eliminates most of the redundant information, there by realizing effective compression of information, saving information storage space, and facilitating the acceleration of operations and real-time processing of information. Here we use a serial fusion approach, as shown in Figure 4:

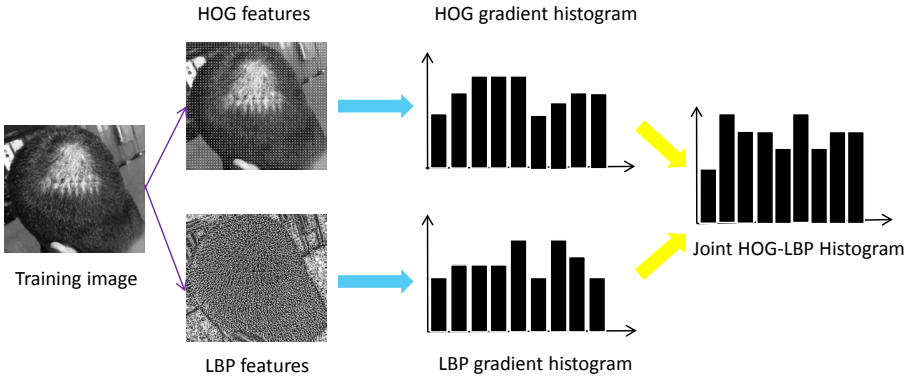

Fig. 4. Joint HOG-LBP Histogram

Here, we use support vector machines to achieve optimal classification of linearly separable data. For a linear SVM with the training samples $\{(x_i, y_i)), 1 \leqslant i \leqslant N\}$, where x_i is the ith instance sample, y_i is the corresponding category labels(i.e.,the expected response),its decision surface equation can be expressed as:

$$\omega \cdot x + b = 0 \tag{8}$$

Where x is the input vector, ω is the dynamically variable weight vector,and b is the offset. In essence to find an optimal classifier is to find an optimal hyper plane according to formula(8), which can not only separate two classes correctly but also maximize the between-class distance. Accordingly,support vectors refer to the training sample points located in the classification boundaries, which are the key elements of the training sample set. Based on these theories and concepts, the following formula is used to classify the input samples:

$$f(x) = sgn\left\{\sum_{i=1}^{k} a_i y_i (x_i \cdot x) + b\right\} \tag{9}$$

where a_i is the weight coefficient corresponding to the support vector x_i.

The next step, we connect the sample HOG feature vector and the LBP feature vector in series to form a joint feature vector input SVM. Here, in the classifier process, the linear inseparable low-dimensional space is converted into a linearly separable high-dimensional space mainly through SVM kernel functions and use the cross-validation method to select the SVM optimal parameters, so that the classifier has the highest classification accuracy for the input training samples. In accordance with the above method, the training process for joint HOG-LBP SVM classifiers is shown in Figure 5:

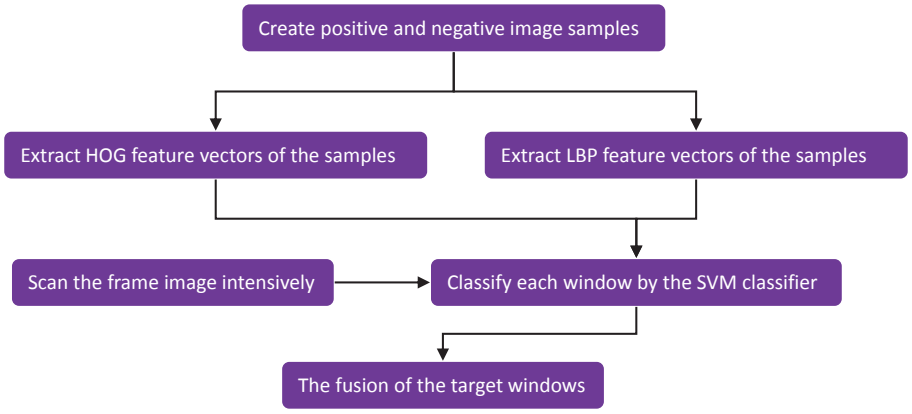

Fig. 5. Training process for joint HOG-LBP SVM classifiers

4 Experimental Results

In order to objectively verify the performance of the algorithm proposed, some experiments were made for it and some frequently-used algorithms. Our experiment is mainly divided into two parts: the first part is the pedestrian detection, and the second part is the counting of the crowd pictures.

The first part of the experiment: The training set we used contains 500 head face images clipped manually and enough non-head face images from some sample sets including INRIA, PETS2000, and MIT. During training negative samples can be selected as needed automatically from the background images. The test set contains 500 images with or without pedestrians, including about 1,500 apparent head faces and covering various scenes, angles, postures, and clothing, etc.The algorithm is embodied in a program developed with Matlab 2017a function library.

Extract sample HOG and LBP features:

Sample HOG feature calculation steps: For each positive and negative sample set, each size of 32*32 grayscale pictures (in this case, the grayscale picture is used to consider the effect of the size of the calculation, and the final detection result has little effect), and the rectangular HOG feature is calculated. Descriptor: The set cell size is 8*8, the size of the block is 16*16, and the slide step size is the width of a Cell. The specific process of HOG feature calculation is as follows: To reduce the influence of lighting, the sample is first Gamma standardization of images. Then calculate the gradient of x and y directions for each pixel in the grayscale image, and use the [-1,0,1] template to calculate the direction and amplitude of the gradient. In each Cell, set the projection direction to 9 bins, and use the gradient magnitude of each pixel as the weight, and vote to count the weighted histograms of gradient directions of each Cell. The dimension of this histogram is 9. Four cells in a Block (with overlaps between blocks) are normalized using L2-norm, and the gradient histograms of four cells are counted, and the dimension is 36. Finally, all blocks in the image are concatenated, and the dimension of the obtained HOG feature vector is calculated.

Sample LBP feature calculation steps: For each positive and negative sample set, each size is a 32*32 grayscale image, and LBP feature extraction based on a sliding window is used. The general description of the sliding window for the image algorithm is as follows: In an image of size W*H, the w*h window (W»w, H»h) is moved according to a certain rule, and a pixel in the window is performed. In the series operation, the window moves one step to the right or down after the operation is completed, until the entire image is processed. Set the size of the window to 16*16, and set the window's horizontal and vertical sliding steps to half the width of the window. The specific process of the LBP feature calculation is as follows: For one pixel in each window, an LBP feature value is calculated using an operator LBP (representing a radius 1, a ring containing 8 neighborhoods, a uniform mode). According to the LBP eigenvalue calculated in the window, the histogram of each window is calculated, that is, the number of occurrences of each LBP eigenvalue, and then L2-norm is used for normalization. The statistical histograms of all windows in the tandem image, the dimensions of the resulting LBP eigenvectors.

Finally, the sample HOG feature vector and the LBP feature vector are connected in series to form a joint feature vector. We use the SVM classifier to transform the linearly indivisible, low-dimensional space into a linearly separable, high-dimensional space by using a kernel function. The cross-validation method is used to select the optimal SVM parameters so that the classifier pairs the input training samples. The highest classification accuracy. The experimental test image size is 384*288. The algorithm is

modified based on Matlab 2017a and runs on an Inter Core i5-5250 (1.60 GHz), 4 GB RAM computer. The experimental results are shown in table 1:

Table 1. Algorithm performances shown by 3 experiments.

Detection algorithm	Test sequence	False number	Detection rate
Dalal HOG	1	39	92.2%
T.Ojala LBP	2	32	93.6%
Joint HOG+LBP	3	17	96.6%

The second part of the experiment: we will process the image of the crowd, and divide it into multiple small pieces. For example, a picture of a crowd of 256 * 256 pixels, the size of each small piece is defined as: 1*1 pixel, 2*2 pixels, 4*4 pixels, 8*8 pixels. Through the feature fusion of HOG-LBP, combined with the SVM classifier, population density estimation and counting are performed on each small block image. The training set contains 500 head face images that were manually cut from the INRIA sample set. The experimental test image size is 595*350, and the experimental results are as table 2:

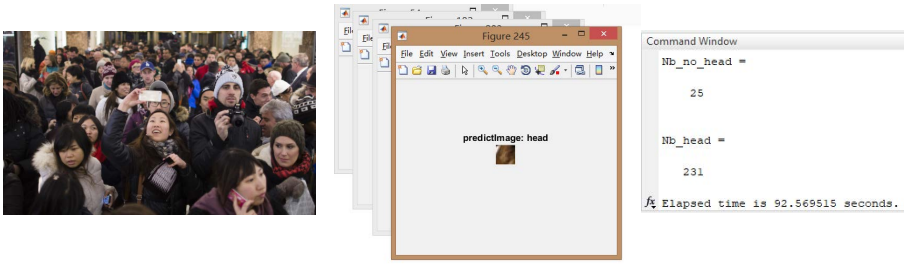

Fig. 6. Crowd counting

We have tried many times and got the results. In the table, number of people detected, number of people actually present in the scene , difference between detected number of people and actual number of people and time.Based on the above results, precision calculated is:91.923%.

5 Conclusion

In this paper, we propose a feature fusion method for crowd counting. By image feature extraction and texture feature analysis methods, data obtained from multiple sources are

Table 2. Result summary.

Test sequence	Number of people detected	Actual	Difference	Time
1	236	260	24	94.32s
2	231	260	29	92.56s
3	220	260	40	89.41s
4	228	260	32	90.28s
5	239	260	21	96.21s
6	226	260	34	90.03s

used to count the crowd. Therefore, we use multiple sources of information, namely, HOG and LBP. These sources provide separate estimates and other combinations of statistical measurements. Using the support vector machine (SVM) classification technique, and regression analysis, we count the crowd with high density. The approach adopted is easy and fast in processing. Our experiments showed the method gives good results in crowded scenes.

6 Future work

In order to improve the detection efficiency and apply it to the crowd testing in the real-world, we plan to use the FPGA cards which is famous for its performance in real-time crowd image processing.

References

1. M. Li, and Z. Zhang,"Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection", 19th International Conference on Pattern Recognition, pp. 1-4, Tampa, FL, 2008.
2. T. S. Le, and C. K. Huynh, "Human-Crowd Density Estimation Based on Gabor Filter and Cell Division," International Conference on Advanced Computing and Applications (ACOMP'15), pp. 157-161, 2015.
3. T. Kryjak, M. Komorkiewicz, and M. Gorgon, "FPGA implementation of real-time head-shoulder detection using local binary patterns, SVM and foreground object detection," Proceedings of the 2012 Conference on Design and Architectures for Signal and Image Processing, Karlsruhe, pp. 1-8, 2012.
4. L. Chen, H. Wu, S. Zhao and J. Gu, "Head-shoulder detection using joint HOG features for people counting and video surveillance in library," 2014 IEEE Workshop on Electronics, Computer and Applications, Ottawa, ON, 2014, pp. 429-432.
5. Rohit, V. Chauhan, S. Kumar and S. K. Singh, "Human count estimation in high density crowd images and videos," 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, 2016, pp. 343-347.
6. H. Idrees, I. Saleemi, C. Seibert and M. Shah, "Multi-source Multi-scale Counting in Extremely Dense Crowd Images," 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, 2013, pp. 2547-2554.
7. V. B. Subburaman, A. Descamps and C. Carincotte, "Counting People in the Crowd Using a Generic Head Detector," 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, Beijing, 2012, pp. 470-475.

8. D. Conte, P. Foggia, G. Percannella and M. Vento, "A Method Based on the Indirect Approach for Counting People in Crowded Scenes," 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, Boston, MA, 2010, pp. 111-118.
9. M. Butenuth et al., "Integrating pedestrian simulation, tracking and event detection for crowd analysis," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, 2011, pp. 150-157.
10. Yaocong Hu,Huan Chang,Fudong Nian,Yan Wang,Teng Li,"Dense crowd counting from still images with convolutional neural networks,"2016 Journal of Visual Communication and Image Representation,2016,Pages 530-539.
11. Cong Zhang, Hongsheng Li, X. Wang and Xiaokang Yang, "Cross-scene crowd counting via deep convolutional neural networks," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 833-841.
12. H. Fradi and J. Dugelay, "Crowd density map estimation based on feature tracks," 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), Pula, 2013, pp. 040-045.
13. Z. Wang, H. Liu, Y. Qian and T. Xu, "Crowd Density Estimation Based on Local Binary Pattern Co-Occurrence Matrix," 2012 IEEE International Conference on Multimedia and Expo Workshops, Melbourne, VIC, 2012, pp. 372-377.
14. Krishna A.N et al.,"A People Counting Method Based on Head Detection and Tracking,"2016 International Journal of Innovative Research in Computer and Communication Engineering,2016,2320-9801.
15. H. Yang and H. Zhao, "A novel method for crowd density estimations," IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012), Shenzhen, 2012, pp. 1-4.
16. Z. Ma and A. B. Chan, "Crossing the Line: Crowd Counting by Integer Programming with Local Features," 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, 2013, pp. 2539-2546.
17. M. Li, Z. Zhang, K. Huang and T. Tan, "Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection," 2008 19th International Conference on Pattern Recognition, Tampa, FL, 2008, pp. 1-4.
18. M. Rodriguez, I. Laptev, J. Sivic and J. Audibert, "Density-aware person detection and tracking in crowds," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2423-2430.
19. L. Wang and N. H. C. Yung, "Bayesian 3D model based human detection in crowded scenes using efficient optimization," 2011 IEEE Workshop on Applications of Computer Vision (WACV), Kona, HI, 2011, pp. 557-563.