A method based on multi-source feature detection for counting people in crowded areas

Songchenchen Gong, El-Bay Bourennane

A method based on multi-source feature detection for counting people in crowded areas

1st Gong songchenchen
University of Burgundy
Laboratoire ImVia
Dijon, France
gscc19@hotmail.com

2nd El-Bay Bourennane
University of Burgundy
Laboratoire ImVia
Dijon, France
ebourenn@u-bourgogne.fr

***Abstract*— We propose a crowd counting method for multisource feature fusion. Image features are extracted from multiple sources, and the population is estimated by image feature extraction and texture feature analysis, along with for crowd image edge detection. We count people in high-density still images. For instance, in the city's squares, sports fields, subway stations, etc. Our approach uses a still image taken by a camera on a drone to appraise the count in the population density image, using a kind of sources of information: HOG, LBP, CANNY. We furnish separate estimates of counts and other statistical measurements through several types of sources. Support vector machine SVM, classification and regression analysis, along with obtain a high density population, reasonable early warning, to ensure the safety of the population. This method can achieve bon results in scenes where people are extremely crowded.**

Keywords- —HOG, LBP, SVM, CANNY

I. INTRODUCTION

Counting the number of people in these years has become a very hot topic because people are paying more and more attention to safety awareness. With the improvement of living standards, a kind of entertainment activities are increasing, and it also brings hidden dangers to people's safety. Then, it is extremely important to accurately appraise the number of people in the public area for security control, to prevent people in some areas from exceeding a certain level, and to give early warning. On December 31, 2014, it was the New Year's Eve event. As many tourists gathered in the Shanghai Bund to greet the New Year, some people lost their balance and fell on the viewing platform, which led to many people falling and stacking, causing a crowded stampede. For preventing such accidents, when some areas exceed a certain number of people, the number of people can be accurately estimated in real time, and the safety measures can be quickly taken to evacuate the crowd through the form of drone warning or large-scale broadcast broadcasting.

In this work, we offer a crowd counting approach for multi-source feature fusion. Image features come from multiple sources of information [1], [6], and the population is estimated by image feature extraction and texture feature analysis, along with for crowd area image edge detection. We count crowd people in high-density still images. The remainder of this article is expressed as follows. In Section 2, we briefly review the related work. In Section 3, we outline the proposed approach. In Section 4, we evaluated our benchmarks, experimental data, and databases. In Section 5, we summarize the paper.

II. RELATED WORK

There are many methods that have been used for crowd detection and counting. Some of the efficient techniques are:

The HOG feature is used to detect the human head. Dalal et al. first use the HOG feature for the pedestrian detection in the static image. The main thought used the gradient histogram of the local region to describe the target feature. HOG features combined with SVM classifier for head detection, SVM classifier training and head detection [1,2].

A new method for population counting and density estimation. The CNN-based approach is further categorized according to the training process and network attributes. Select a representative subset of the latest methods for detailed analysis and review. In addition, we observed that combining the ratio and context information in the CNN-based approach greatly improved the estimation error [3].

Based on different framework structures, individuals are detected by grouping local blocks in the region [4-6].

To used multiple sources of information to compute a measure of the number of individuals present in a severely dense crowd visible in a single image. Such as texture elements (using SIFT), and frequency-domain analysis to estimate counts [7-10].

An adaptive smoothing algorithm in accordance with Canny operator is combined with global and local edge detection to extract the edge features of the target. The local region detection method is selected for edge extraction. The complete image edge is achieved by the edge detection method, which combines global and local edge features [11].

III. PROPOSED SYSTEM

We propose a population counting method for feature fusion and edge detection. By image feature extraction and texture feature analysis, and for crowd head edge detection, the fusion of counts from multiple sources is used to estimate the count. We use high-density still images to calculate the number of people in the crowd.

A. HOG based Head Detections

HOG initially proposed by Dalal et at for Pedestrian Detection is essentially gray gradient statistics of local image regions [1],[2]. As HOG feature sand Haar-like features utilize gradient information on targets, they are robust to noise and illumination changes. But because basic description, features including HOG and Haar-like ones are relatively simple, in many cases, they are weak in describing complex image regions. To improve on this, we propose a population counting method for feature fusion and edge detection. By image feature extraction and texture feature analysis, and for crowd head edge detection, the fusion of counts from multiple sources is used to estimate the count.

Assume the pixel located at (x, y) has a gray value I, a gradient magnitude G, and a gradient direction θ. For simplicity, we adopt a one-dimensional center gradient operator indicated as [-1, 0, 1]. More precisely, the horizontal gradient and vertical gradient are defined as follows:

$$G_x(x,y) = I(x+1,y) - I(x-1,y) \quad (1)$$

$$G_y(x,y) = I(x,y+1) - I(x,y-1) \quad (2)$$

And the gradient of the pixel located at (x, y) calculated as follows:

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

$$\theta(x,y) = \tan^{-1}\left(\frac{G_X(x,y)}{G_y(x,y)}\right) \quad (4)$$

Afterwards, a HOG is constructed for each cell unit. We divide the image into several "cells"[2-3], for each cell consisting of 8x8 pixels the gradient magnitudes are weighted according to their gradient direction and then accumulated in their bin direction, resulting in a gradient histogram. We choose a directional range of 0-360 degree for HOG features of pedestrians, with 9 segments of 40 degree and correspondingly 9 bin directions.

Finally, the obtained HOG feature vectors are normalized so as to uniformly express features of different blocks.

We consult the normalized block descriptors (vectors) as HOG descriptors [10]. Through this method, the HOG descriptor becomes a vector consisting of the histogram components of all cell units in each interval. A vector with HOG feature dimensions.

B. LBP feature

The LBP were first proposed by T. Ojala, M.Pietikäinen, and D. Harwood. LBP is a simple but very effective texture operator [5]. It compares each pixel with its nearby pixels and saves the result as a binary number. The most major 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 essential LBP operator is defined as the 3*3 window. Using the value of the center pixel of the window as the threshold, the gray value of the adjacent 8 pixels is compared with it. If the surrounding pixel value is greater than the center pixel value, the pixel value is of the location is marked as '1'. Otherwise, it is '0'. In this fashion, 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 essential LBP feature for a given pixel is taking shape by thresholding the 3x3 neighbourhood with the center pixel value as the threshold, where(Xc,Yc) is the center pixel, ic be the intensity of the center pixel and in (n=0,1,2....7) pixel intensities from the neighborhood. The LBP is given by:

$$LBP(X_C, Y_C) = \sum_{n=0}^{P-1} 2^n (i_n - i_c) \quad (5)$$

Where P is the number of sample points and:

$$s(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & else \end{cases} \quad (6)$$

The LBP could be interpreted as an 8-bit integer. The essential LBP concept is presented in Figure 1

Figure 1.Illustration of the standard LBP operator. Image taken from the Daimler pedestrian dataset.

C. Canny based head edge detection

Analysis of the principles of the Canny edge detection algorithm of the Canny operator in the Gaussian filtering process were in-depth study [11], and to use first order partial derivatives of the finite difference to calculate the magnitude and direction of the gradient is extended, are calculated by using the four directions, effectively improve the precision of Canny edge detection.

First of all, denoise by using, Gaussian filtering, Its purpose is to smooth the original image and remove or weaken the noise in the image. Assuming two dimensional Gauss's function:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2+y^2}{2\sigma}\right] \quad (7)$$

Gradient vector:

$$\nabla G = \left[\frac{\partial G/\partial x}{\partial G/\partial y}\right] \quad (8)$$

To improve the speed of the decomposition method, the 2 filters at convolution template G is decomposed into 2 one-dimensional filters:

$$\frac{\partial G(x,y,\sigma)}{\partial x} = kxe^{\frac{x^2}{2\sigma^2}}e^{\frac{y^2}{2\sigma^2}} = h_1(x)h_2(y) \quad (9)$$

$$\frac{\partial G(x,y,\sigma)}{\partial y} = kxe^{\frac{x^2}{2\sigma^2}}e^{\frac{y^2}{2\sigma^2}} = h_1(y)h_2(x) \quad (10)$$

Where k is a constant, σ is Gaussian filter parameters. It controls the degree of smoothing. The σ filter, although the positioning accuracy is high, but the signal to noise ratio is low; σ is the opposite, so according to the need to adapt to the selection of Gaussian filter parameter.

We use a new 3x3 neighborhood in calculating the gradient amplitude. The procedure is as follows:

The partial derivatives of the 4 directions are calculated at first:

Partial derivative of x direction:

$$P_x(x,y) = G(x,y+1) - G(x,y-1) \quad (11)$$

Partial derivative of y direction:

$$P_y(x,y) = G(x+1,y) - G(x-1,y) \quad (12)$$

45 degree directional partial derivative:

$$P_{45}(x,y) = G(x-1,y+1) - G(x+1,y-1) \quad (13)$$

135 degree directional partial derivative:

$$P_{135}(x,y) = G(x+1,y+1) - G(x-1,y-1) \quad (14)$$

Difference in horizontal direction:

$$f_x(x,y) = P_x(x,y) + [P_{45}(x,y) + P_{135}(x,y)]/2 \quad (15)$$

Difference in vertical direction:

$$f_y(x,y) = P_y(x,y) + [P_{45}(x,y) - P_{135}(x,y)]/2 \quad (16)$$

The gradient magnitude is obtained:

$$M(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \quad (17)$$

The gradient direction is:

$$\theta(x,y) = \arctan\left(\frac{f_x(x,y)}{f_y(x,y)}\right) \quad (18)$$

The method takes into account the pixel diagonal direction, increase the pixels to computer the partial derivative direction, improves the traditional canny operator, uses differential mean value calculation, and improve the accuracy of the positioning of the edge.

Then, non-maxima suppression of gradient amplitude: the gradient of the non-maxima suppression [9], only according to the gradient magnitude image is not enough to determine the edge, in order to determine the edge must be in the refinement of the gradient magnitude image ridge belt, so as to produce the refinement edge. Non-maxima suppression through inhibition of gradient direction on the non-roof peak of the gradient magnitude to refine the gradient magnitude of the roof.

Finally, dual threshold detection and edge linking: non-maximum suppression, the gradient array processing after thresholding.

Marginal discriminant: every edge intensity above a high threshold of edge points; every edge strength less than low threshold is definitely not the edge points; if the edge strength is higher than the lower threshold than the high threshold, then look at the pixel adjacent pixels in no more than the high threshold of edge points and if there is, it is the edge point, if not, it is not the edge points.

Through the Canny operator's detection of the head image, we can get a clear head outline shown in figure 2:

Figure 2.The original image (left) and the output (right) of the Canny algorithm.

By optimizing the Canny operator, the effect of noise on the image can be reduced, and the image data can be better preserved, making it easier to find the outline of the head image for extraction.

D. Training of joint HOG-LBP-CANNY 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, thereby 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 provided in Figure 3:

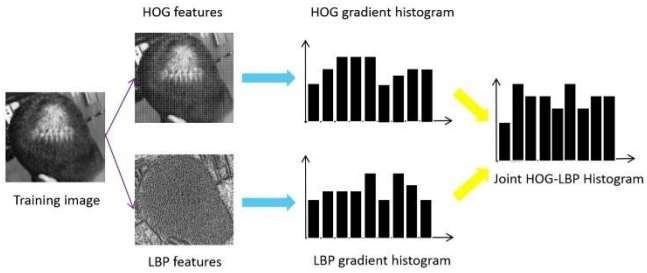

Figure 3.Joint HOG-LBP Histogram

Here, we use SVM [1], [7] to achieve optimal classification of linearly separable data. For a linear SVM with the training samples $((x_i,y_i)), 1 \le i \le N$), 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:

$$\omega \cdot x + b = 0 \quad (19)$$

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, 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:

$$f(x) = sgn\left\{\sum_{i=1}^{k} a_i y_i (x_i \cdot x) + b\right\} \quad (20)$$

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 classification 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 of the input training samples.

It is shown by the study of the human visual system, image boundary is particularly important, often a rough outline of the line alone can by identifying an object [8]. This fact for machine vision research provides important enlightenment, namely: objects available the boundary represented by the grayscale image discontinuous points consisting of basic original carrying the original image of the vast most useful information.

Here, we obtain a clear head contour, fusion feature extraction and texture feature analysis by edge detection of the header image, and combine the information sources from multiple sources to estimate the counting population. The process is as follows:

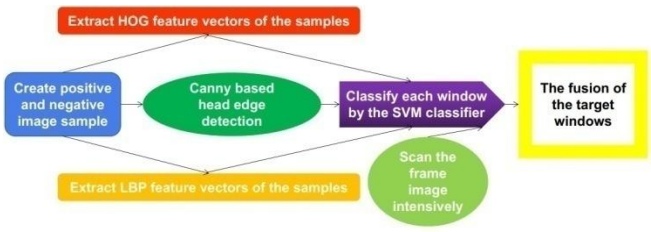

Figure 4.Crowd counting

E. Experimental Results

In order to objectively verify the performance of the proposed algorithm, some experiments and some frequently used algorithms were performed. Our experiment is mainly divided into two parts: the first part is for pedestrian detection, the second part count the population of pictures of still people.

The first part of the experiment: The training set we used to contained 5000 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 5000 images with or without pedestrians, including about 4,500 apparent head face 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, LBP texture features and CANNY head contour edge detection:

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 measurement 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 measurement of the block is 16*16, and the slide step size is the width of a Cell. Count all the blocks in the tandem image and calculate the dimensions of the HOG feature vector.

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 statistical histogram of all windows in the series image, and the calculated dimension of the LBP feature vector.

Sample CANNY head contour edge detection: use high and low thresholds to find edge points, reduce the number of false edge segments, and get the sample head contour image.

Then, the sample HOG feature vector and the LBP feature vector are connected in series to form a joint feature vector. Add CANNY edge detection to extract the outline of the head. 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 for a 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 I. Algorithm performances shown by 4 experiments.

Detection algorithm	Test sequence	False number	Detection rate
Dalal HOG	1	39	92.2%
T.Ojala LBP	2	32	93.6%
Joint HOG	3	17	96.6%
Joint HOG+LBP+CANNY	4	14	97.2%

In the second part of the experiment: we will process the image of the crowd and divide it into small pieces. For example, a set of 256 * 256 pixel pictures, each of which is defined by a size of 16 * 16 pixels, 32 * 32 pixels, 64 * 64 pixels, and 128 * 128 pixels, as provided in figure 5:

Figure 5.This figure shows the size of different pixel blocks used in this paper.

Through the feature fusion of HOG-LBP, CANNY is used to extract the edge of the head contour and combined with the SVM classifier, the population density is estimated and counted for each small block image. The training set contains 5000 head facial images that were manually cut from the INRIA sample set. The experimental test image size is 600*400, and the experimental results are as figure 6:

Figure 6. This figure show one arbitrary image from the dataset used in this paper.

We have tried many times and got the results given in Table II. In table II, the number of people detected, number of people actually present in the scene, the difference between the detected number of people and the actual number of people and time. Based on the above results, the precision calculated is 92.85\%.

Table II. Result summary.

Test	Number of people detected	Actual	Difference	Time
1	220	280	60	63.13s
2	231	280	49	82.56s
3	260	280	20	104.25s
4	257	280	23	103.13s
5	245	280	35	92.51s

F. Conclusion

In this paper, we propose a feature fusion method for population counting. A plurality of analytical detection methods, such as extraction of image features, edge detection of target feature images, and the like, so that data obtained by a plurality of information sources is used for calculating a population. Therefore, we use a variety of sources of information, namely HOG, LBP and CANNY. These sources provide separate combinations of estimates and statistical measurements. Using SVM classification techniques and regression analysis, we calculate high-density populations. The approach adopted is easy and fast in processing. Compared to the joint HOG, our experiments showed the method gives good results in crowded scenes.

G. 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. Moreover, we installed it on the drone to count high-density people and provide early warning.

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] Y. Zhang, C.L. Zhou, F.L. Chang and A.C. Kot, "Attention to Head Locations for Crowd Counting", IEEE Conference On Computer Vision And Pattern Recognition (CVPR) 2018.

[3] V. A. Sindagi and V. M. Patel,"A survey of recent advances in cnn-based single image crowd counting and density estimation," Pattern Recognition Letters, 2017.

[4] 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.

[5] Y. Hou and G. K. H. Pang, "Human detection in crowded scenes," 2010 IEEE International Conference on Image Processing, Hong Kong, 2010, pp. 721-724.

[6] H. Fradi and J. Dugelay, "Low level crowd analysis using frame-wise normalized feature for people counting," 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Tenerife, 2012, pp. 246-251.

[7] T.Le and C. Huynh, "Human-Crowd Density Estimation Based on Gabor Filter and Cell Division," 2015 International Conference on Advanced Computing and Applications (ACOMP), Ho Chi Minh City, 2015, pp. 157-161.

[8] D.Kim, Y. Lee, B. Ku and H. Ko, "Crowd Density Estimation Using Multi-class Adaboost," 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, Beijing, 2012, pp. 447-451.

[9] A. N. Marana, S. A. Velastin, L. F. Costa and R. A. Lotufo, "Estimation of crowd density using image processing," IEE Colloquium on Image Processing for Security Applications (Digest No: 1997/074), London, UK, 1997, pp. 11/1-11/8.

[10] 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.

[11] L.Yuan and X. Xu, "Adaptive Image Edge Detection Algorithm Based on Canny Operator," 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS), Harbin, 2015, pp. 28-31.