

An automatic people counting method of hotel dining with occlusion

Dong Ling^{1,a}, Chen Xianqiao¹

¹College of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei
430063, China

^a1059675029@qq.com

Keywords: people counting; scenes with occlusion; Adab_SVM algorithm; head detection; eigenvector

Abstract: Video image has the advantage of large amount of information, good real-time performance and low cost, so automatic people counting based on video image has very high practical value, and many scholars have done a large number of experiments and studies on this and achieved certain achievements. But for scenes with more occlusion and background changing quickly and without obvious rules, it's difficult to count accurately. In order to improve the counting accuracy in the above scenes, to provide the number of customers for hotel managers to efficiently organize and work, based on pictures, a automatic people counting method using SVM as weak classifiers, train intensively in learning by Adaboost algorithm(i.e. Adab_SVM algorithm) of hotel dining is proposed. The method is mainly aimed at the hotel scenes with occlusion too much to complete the segmentation of human body region. Firstly, traversing the entire picture to get the preliminary head areas and the number of people, then merge these head areas to get the exact number of people, to complete the statistical work on the number of people of the entire picture. Experimental results show that the method has higher counting accuracy in the hotel scenes with occlusion.

1. Introduction

The people counting system is an video intelligence application by analyze video images to get the statistical information, widely used in hotels, shopping malls, airports, stations, science and technology museums and other public places. The user can get the exact number of people and the crowd flow data on site.

In ref[1], the HOG feature is used to describe the individual feature of target by calculating and statistic gradient histogram of the image, then the whole human body region is extracted to achieve the purpose of people counting. For scenes that are more difficult to obtain for the entire human body area, this method will not detect the target area. In ref[2], the LBP feature is used to extract head area, and the Adaboost algorithm is used to train a classifier to achieve the purpose of statistics. It is very limited to extract the head information for scenes with more target types, and the Adaboost algorithm is very sensitive to noise and anomaly data. When the samples are difficult to distinguish, there will be over-learning, leading to the increase of error. In order to improve the accuracy, it's necessary to train a lot of samples in the process of training the classifier using Adaboost algorithm, which requires much time, the system is inconvenient to expand, so there are limitations. In ref[3], background difference method is used to eliminate the

background, and the selective updating of background model is added, which can adapt to slow change of the background to a certain extent. However, it is difficult to extract the target area by background difference and background image update for scenes with fast background and no obvious order.

This paper is aim to statistics the number of hotel dining, to allow the hoteliers to grasp the number of hotel guests, efficiently and orderly arrange guests dining. In the hotel dining scene, the movement and change of tables, chairs, and tableware, and so on, have uncertainty, causing the background image to be difficult to determine; The occlusion between person and person and between chair and person, makes it difficult to separate the individual, the number of people can not be measured by the human body; People’s entering and leaving is random, so the real-time requirement is higher. Therefore, the people counting methods presented in the above literature are inadequate and limited to meet the needs of the study in this paper. In this paper, we propose a method based on Haar-like feature to describe the head area, SVM classifier as the weak classifier, and Adaboost algorithm to further strengthen and integrate learning (Adab_SVM algorithm) to get the final classifier. To achieve the purpose of people counting through gathering statistics the number of heads in the the hotel scenes with occlusion.

2. Counting Model of Dining for Occlusion Scene

2.1 Extraction and Calculation of Head Features.

Firstly, by the window traversal to get all the areas to be detected, and then calculate eigenvalue of each area to be detected.

(1)Window Traversal. Head detection needs to find all areas that may exist head area, firstly.In this paper, the sub-window to be detected (ie, the area where the head area may exist) is obtained by window traversal[5]. The eigenvalues of these sub-windows can be calculated to determine whether or not the head area exists.In this study, the initial window size is 20*20, the step size is 5px, the magnification is 1.5 (that is, the detection window size from 20*20 to 30*30), the final size of the detection window is 67.5*67.5.

(2)Feature Extraction. In this paper, we use the Haar-like feature to describe the head area and calculate the eigenvalue in each sub-window to be detected. The eigenvalue is calculated as the difference between the black rectangle area pixel and the white rectangle area pixel sum.This feature reflects the grayscale variation of the image, the basic type shown in Fig.1 .

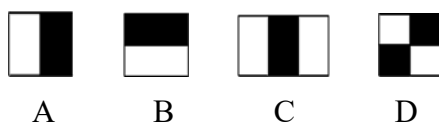


Fig.1 Haar-like feature

The eigenvalue of templates A, B, D is calculated as:

$$V = \text{Sum}(\text{black}) - \text{Sum}(\text{white}) \quad (1)$$

The eigenvalue of templates C is calculated as:

$$V = 2 \times \text{Sum}(\text{black}) - \text{Sum}(\text{white}) \quad (2)$$

The commonly used Haar-like feature types have edge feature, linear feature, point feature (center feature), diagonal feature. In this paper, according to the different classifiers, different forms of Haar-like feature and their combinations are used to construct the eigenvectors to describe the different azimuths of the head image.

(3)Eigenvalue Calculation. In this paper, we use the integral image[6] to compute the eigenvalue of each sub-window to determine whether it is the target feature area.The integral image at the point (x, y) is defined as the sum of all the pixels in the upper left corner of the point, as shown in the following equation:

$$ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y') \quad (3)$$

Where, $I(x', y')$ is the pixel value of point (x, y) in the image. It can be seen from the above, the eigenvalue of the area ABCD in Fig. 2 is calculated as follow:

$$V_{ABCD} = Sum(black) - Sum(white) = [(ii4 + ii1) - (ii2 + ii3)] - [(ii6 + ii3) - (ii4 + ii5)]$$

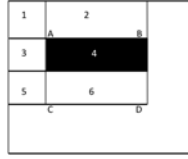


Fig.2 The eigenvalue of area ABCD

2.2 Design and Implementation of Head Classifier.

Firstly, train the head detection classifier with Adab_SVM algorithm,then use the final classifier to get the preliminary head areas in image to be detected.

(1)Design of Adab_SVM Algorithm. In order to increase the accuracy of the classifier and improve the training speed, in this paper, use the Gaussian kernel SVM classifier as weak classifier, and further enhance and integrate the learning by Adaboost algorithm. Adaboost[7] is an adaptive iterative algorithm, obtaining a lot of weak classifiers(basic classifiers) by different classification operations of the same sample set with updated sample weights, and then calculates the new weight of each weak classifier according to the classification error. Finally, these weak classifiers are weightedly merged to form a strong classifier (the final classifier with higher accuracy). Adaboost has a very high practical value because it is easier to obtain multiple classification methods with better accuracy than random guessing. SVM[8] is a classifier defined by classification hyperplane, mapping linearly indivisible sample data in low-dimensional space into high-dimensional feature space through kernel, to avoid "dimensional disaster" and realize linear segmentation of data in high-dimensional space. The core idea of SVM method is to maximize the classification boundary and minimize the empirical risk. In this paper, Adab_SVM algorithm is proposed based on the advantages and disadvantages of Adaboost algorithm and SVM algorithm.

The main idea of the Adab_SVM algorithm is as follows: Firstly, a weak classifier is obtained according to the sample distribution of the given initial training sample set, so that the whole sample has less classification error in the classifier. Then, according to the principle that reduce the weight of a sample if it is classified correctly, increase the weight of a sample if it is classified wrongly, adjust the weight of each sample. The training sample set with the updated weights is used as the sample set of the new training, to get the next classifier. The above steps are repeated until there is no change in the sample weights or the preset iterative number is reached. The Adab_SVM algorithm is described as follows:

2.2.1 Initialization

Given a sample set with label $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$, where $X_k \in X$, $k = 1, 2, \dots, n$ (X denotes

the feature space, $X \subseteq R^d$, X_k is the sample eigenvector, $Y_k (Y_k \in \{-1, +1\})$ is the corresponding label; n is the total number of samples, each $w(k)$ is the weight of a sample and the initial value of them are $1/n$; T is the preset iterative number. In this study, the dimension of feature space is low, the total number of samples is much larger than the dimension, so the Gaussian kernel function is selected.

2.2.2 For $i=1$ to T

1) According to the distribution of samples, the weak classifier is obtained by Gaussian kernel function

$$G_i = P(x, y, w) \quad (4)$$

2) Calculate the error rate of the weak classifier

$$e_i = \sum_{k=1}^n [G_i(x_k) \neq y_k] \times w_k \quad (5)$$

3) Calculate the weight of the weak classifier

$$\alpha_i = (1/2) \ln[(1 - e_i) / e_i] \quad (6)$$

4) Update the sample weights of the training sample set

$$w_{i+1}(k) = [w_i(k) / Z_i] \times \exp(-\alpha_i y_k G_i(x_k)), k = 1, 2, \dots, n \quad (7)$$

Where, $Z_i = \sum_{k=1}^n w_i \exp(-\alpha_i y_k G_i(x_k))$ is the normalization factor.

2.2.3 Combining all the weak classifiers

$$f(x) = \sum_{i=1}^T \alpha_i G_i \quad (8)$$

2.2.4 Get the final classifier

$$G = \text{sign}(f(x)) = \text{sign}\left(\sum_{i=1}^T \alpha_i G_i\right) \quad (9)$$

(2) Implementation of classifier. On the basis of Haar-like feature, the head discriminator is composed of a crown of head classifier, a face classifier, a left-face classifier, a right-face classifier, and a back-brain classifier. The detection result of area to be detected is a logical or operation of the results of the five classifiers, that is, the result of any one classifier is true, then the area to be detected is judged as head area. The final classifier is shown in Fig.3.

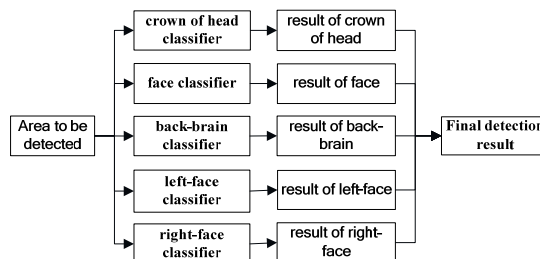


Fig.3 Head detection classifier

2.3 Target counting.

To accurately count the number of dining, it is need to merge the detected head, removing the same head in order to avoid repeated count, to ensure that a head count only once. The main idea in this paper is to merge the rectangles(head areas) with enclosing relationships into a new rectangle. The average of the coordinates of the old rectangles is as the coordinate of the new rectangular. Specific merger process is as follows:

```
BEGIN
1.For  $i=1$  to  $n$ 
    Calculate the center point  $C_i$  of the rectangle  $Rect_i$  and mark the  $Rect_i$  as inactive;
2.  $j=0$ ; //  $j$  is the number of new rectangles that are merged
3.For  $i=1$  to  $n$ 
    If( $Rect_i$  is inactive){
        Search all the center points in  $Rect_i$ , and add them into the point set  $S$ ;
        The average of coordinates of the rectangles corresponding to the center points in  $S$  is calculated
as the coordinate of the new rectangle  $NewRect_j$ ;
        Mark all the rectangles corresponding to the center points in  $S$  as active;
         $j = j + 1$ ; Clear  $S$ ;} //End If
End For
4.Output the total number of new rectangles  $j$ , and the coordinate of all  $NewRect_j$ , that is, the final head
area information.
END
```

If one area to be detected is determined as a head area by the head discriminator, the counter plus one, and record the coordinate information of the area. After all of the areas to be detected are detected, the total number n of rectangles initially detected as the head area is obtained, then merge the n rectangles using the above-mentioned head area merging method to obtain the total number j of merged new rectangles. So j is the final number of heads.

3. Experimental Results and Analysis

In order to verify the people counting methods proposed in this paper, relevant experiments were done, the Scene 1(Video 1) is downloaded from the network, Scenes 2(Video 2) is shot in a hotel in Wuhan. Experiment results are shown in Fig.4, Fig.5, Fig.6, Fig.7, Fig.8, Fig.9, where, (a),(b),(c) in scene 1 represents the 347th frame, the 392th frame, the 446th frame in Video 1, respectively; (a),(b),(c) in scene 2 represents the 1076th frame, the 1104th frame, the 1179th frame in Video 2, respectively. Experimental system environment is Windows 7, using the VS2015, OpenCV3.0 and other software tools.

Scene 1(Fig.4、 Fig.5、 Fig.6):



Fig.4 Statistical results of Bagging algorithm



Fig.5 Statistical results of Adaboost algorithm



Fig.6 Statistical results of Adab_SVM algorithm
Scene 2(Fig.7、 Fig.8、 Fig.9):



Fig.7 Statistical results of Bagging algorithm



Fig.8 Statistical results of Adaboost algorithm



Fig.9 Statistical results of Adab_SVM algorithm

It can be seen from the above example scenes, Adab_SVM, Adaboost, and Bagging have the same detection results for less occlusion scenes (eg, 446th frame in Video 1, 1179th in Video 2). Adab_SVM is superior to Adaboost and Bagging in more occlusion frames(such as 392th frame in Video 1, 1079th and 1104th frame in Video 2), the Adab_SVM detected the occluded headers that were not been detected by Adaboost and Bagging:the head area in the intermediate position by the rear of 392th frame in Video 1, the head area in left front of 1079th frame in Video 2, and the head area in the middle of 1104th frame in Video 2.This shows the effectiveness of the classifier constructed by eigenvectors in occlusion scene. Adab_SVM algorithm has better detection performance than Adaboost and Bagging when there are more occlusion. From a large number of experiments, we can get the experimental results in Table 1. The Adab_SVM algorithm basically meets the needs of the detection.

Table 1 Experimental result of some classifiers

	correction rate	omission rate	mistake rate
Adab SVM	91.5%	8.5%	1.0%
Adaboost	86.1%	13.9%	3.3%
Bagging	75.3%	24.7%	5.2%

4. Conclusions

In this paper, the Adab_SVM algorithm is used to process the captured video image to obtain the head areas in the image, and then merge the areas to get the information of the people in the image. This method can solve the problem of occlusion between people and people, between people and background items, and the problem of background changing quickly and irregularly in video images. It can be seen from the experiment, the people counting methods has better detection performance with a high accuracy rate. But when the features of the back of head are not obvious, the detection error may occur. So further extracting features of the back of head will be the focus in the future research and improvement.

Acknowledgements

This work was financially supported by National Science and Technology Supporting Project, China (No.2015BAG20B02, No.2015BAG20B00).

References

- [1] Nguyen D, Huynh L, Dinh T B, et al. Video Monitoring System: Counting People by Tracking[C]// IEEE Rivf International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future. IEEE, 2012:1-4.
- [2] Zebin Cai, Yu Zhu Liang, Hao Liu, et al. Counting people in crowded scenes by video analyzing[C]// IEEE, Conference on Industrial Electronics and Applications. 2014:1841-1845.
- [3] Tongyi Sun. Research on People Counting in Real-time Video Surveillance[D]. National University of Defense Technology, 2011.(in Chinese)
- [4] Wen W, Ho M, Huang C. People tracking and counting for applications in video surveillance system[C]// Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on. IEEE, 2008:1677 - 1682.
- [5] Bingkun Zhang. The research of algorithm of people counting base on the head detection[D]. Xi'An University of Science and Technology, 2013.(in Chinese)
- [6] <http://blog.csdn.net/byxdaz/article/details/4328415>
- [7] Ying Cao, Qiguang Miao, Jiachen Liu, et al. ACTA AUTOMATICA SINICA, 2013, 39(6):745-758.(in Chinese)
- [8] Jianming Cui, Jianming Liu, Zhouyu Liao. Research of Text Categorization Based on Support Vector Machine[J]. Computer Simulation, 2013, 30(2):299-302.(in Chinese)
- [9] Wennuan Ou, Xuhong Tian, Tonglin Zhu. JOURNAL OF GRAPHICS, 2012, 33(2):113-118.(in Chinese)
- [10] Chen C C, Lin H H, Chen T C. Tracking and counting people in visual surveillance systems[C]// International Conference on Acoustics. 2011:1425-1428.
- [11] Kuo C H, Nevatia R. How does person identity recognition help multi-person tracking?[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2011:1217-1224.
- [12] Hong Liu, Ru Song, Baoxing Wang. Journal of Anhui University(Natural Science Edition), 2015(3):47-50.(in Chinese)