2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)

Crowd Detection and Management using Cascade classifier on ARMv8 and OpenCV-Python

[a]Dr. S. Syed Ameer Abbas, [b]Dr. P. Oliver Jayaprakash, [c]M. Anitha, [c]X. Vinitha Jaini
[a]Professor, Electronics and Communication Engineering
[b]Professor, Civil Engineering
[c]Students, Electronics and Communication Engineering
Mepco Schlenk Engineering College, Sivakasi

***Abstract*— The steady increase in population and overcrowding has become an unavoidable factor in any public gathering or on the street during any festive occasions. The intelligent monitoring technology has been developing in recent years and human tracking has made a lot of progress. In this paper, we propose a method to manage the crowd by keeping in track the count of the people in the scene. In our study, we develop a system using Raspberry Pi 3 board that consists of ARMv8 CPU that detects the human heads and provide a count of humans in the region using OpenCV-Python. A Haar cascade classifier is trained for human head detection. Human tracking is achieved by indicating the direction of movement of the person. The results of the analysis will be helpful in managing the crowd in any area with high density of crowds.**

***Index Terms*— cascade classifier, Haar features, Adaboost algorithm, head detection, tracking**

I. Introduction

Public safety has become a major problem in areas like malls, railway stations, streets during festive seasons, etc. The massive disasters that happen worldwide include numerous instances of fatality where people gather in crowds. An efficient automated system to manage the crowd count is essential. Human tracking provides a way to detect the position, to obtain the motion trail and to maintain identities of persons in the scene.

Managing a crowd of varying densities involves detection of the individual humans in the crowd. In a high density crowd, because of inter-object occlusion, detection and tracking of humans in the crowd be a challenge in computer vision. Existing tracking algorithms present challenges due to difficulty of background modelling. Inter-object occlusion also is a big challenge to body shape models and detection of body parts. Researchers have proposed and studied the tracking methods based on features like motion blobs, texture and color. There have been studies for head detection based on background modelling, head-shoulder detection using omega model and so on.

This study focusses on training a cascade classifier for human head detection by taking positive samples and negative samples. The trained cascade is then used to process the video frames in which the human heads are detected and the count of the humans in the scene is provided. Training the cascade for head detection provides efficient detection in areas subjected to inter-object occlusion. The detected human heads are then tracked using optical flow algorithm. This tracking provides the direction of motion of the persons in the scene.

II. Literature Survey

Detection of objects by using a cascade of simple features was mainly introduced by researchers for face detection [1]. The role of Haar features extracted from an integral image in object detection is elaborated in their work. Some researchers explain a framework that can be trained for object detection [2]. From a set of wavelet basis functions, an object class is derived and is used as an input to support vector machine.

Many studies have been carried out to detect humans in a scene by using part-based human detector [3][4][5]. Parts may not be clearly visible as they depend on factors such as inter object occlusion, illumination, etc. Some other researches have been carried out to detect presence of human based on foreground segmentation [6][7][8]. The background is subtracted from the frame and some types of filtering, for example Gaussian filtering, is done to extract human motion blobs out of the frame. This process is usually done in a grayscale image. The motion blobs are then processed to determine its similarities with that of humans. Illumination variation in different regions give rise to split motion blobs and hence it becomes challengeable to formulate a shape for human detection.

Density based head detection has been demonstrated by some researchers [9]. The density of the crowd is initially estimated from which the heads of the individual persons are detected. Multi-human tracking has also been studied based on head detection using particle filter [10]. Energy functions are constructed which can obtain minimum value for an optimal strategy. Head detection for tracking passengers in subway has also been a major field of study for many researchers. Many work has been done to analyze passengers' moving direction through space coordinate information [11][12].

III. Proposed Work

Our proposed work is to develop a system for crowd management and human tracking using Raspberry pi. Initially, a cascade classifier is trained for head detection with the

978-1-5090-3294-5/17/$31.00 ©2017 IEEE

samples collected from the scene. Haar features are used to train the cascade classifier through OpenCV. The set of rectangular features (Haar) provides an image representation which provides effective learning [4].

The proposed system uses a camera to record the crowded scene. Raspberry pi 3 uses a quad-core ARMv8 central processing unit which processes the video frame by frame to detect the presence of humans by using the previously trained cascade classifier for head detection. This process gives the count of humans present in the scene. To manage the crowd, the count of the persons is kept in track and if the count goes beyond a threshold, prevention of more people entering the scene can be done.

The detected human heads are then tracked by using the optical flow algorithm and the direction of motion of each human is indicated on the display connected to the processor.

The model of our proposed system is as shown in Figure 1. The first block indicates a video frame of the scene captured by the camera. The arrows in the block represent the direction of flow of the crowd. The video recorded is stored in memory of the processing unit. It is then processed and the results are displayed on the display screen.

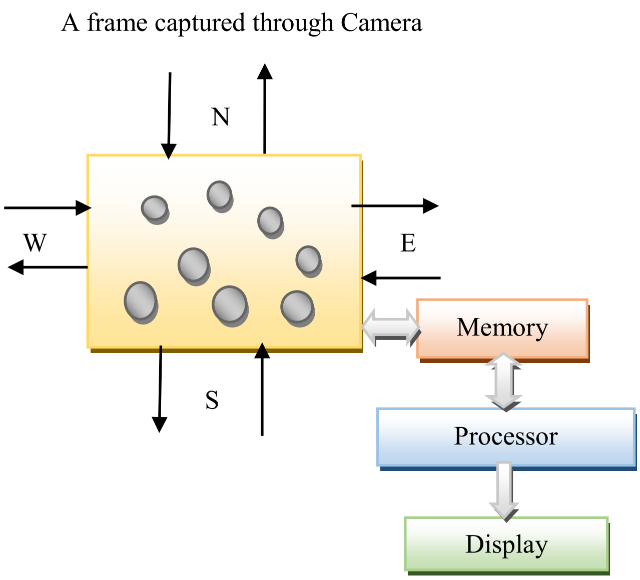

Figure 1. Proposed model of the system

The proposed work is given in Figure 2. It mainly consists of two processes: training and detection.

Training is a preprocessing step for the process of crowd management. Training using the video frames shot in the scene where the surveillance camera is to be implemented will fetch accurate results. This helps in minimizing false detection. We train our cascade classifier using the real time video captured in our institution.

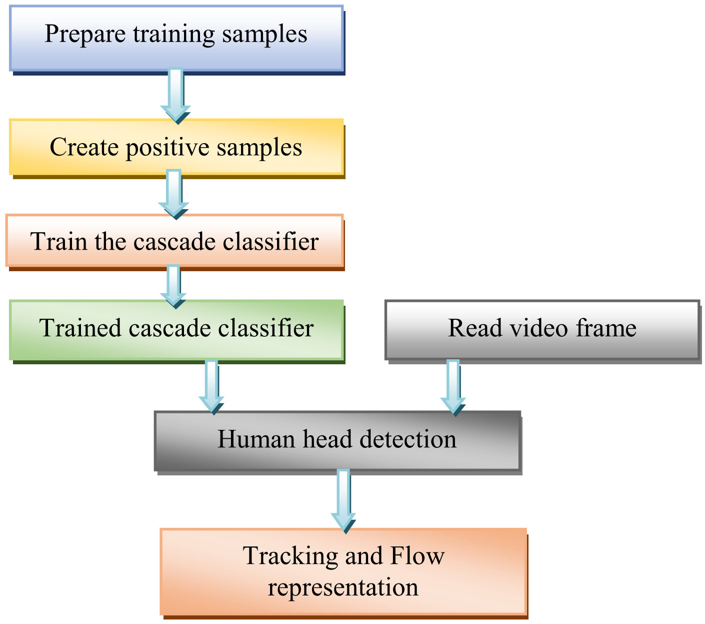

Figure 2. Proposed work of our system

IV. Hardware and Software Tools

For hardware implementation of our study, we use Raspberry Pi board. It has a 1.2GHz 64-bit quad-core ARMv8 CPU, 802.11n Wireless LAN, Bluetooth 4.1 and Bluetooth Low Energy (BLE). It also has a 1GB RAM, 4 USB ports, 40 GPIO pins, Full HDMI port, Ethernet port, Camera interface, Display interface and a Micro SD card slot. The Raspberry Pi 3 board with camera interface is shown in Figure 3.

The software used here is OpenCV-Python. OpenCV is a library of programming functions. It is mainly aimed at real-time computer vision. OpenCV supports a wide variety of programming languages like C++, Python, Java, etc. OpenCV-Python is the Python API of OpenCV. To support various applications, OpenCV includes a statistical machine learning library.

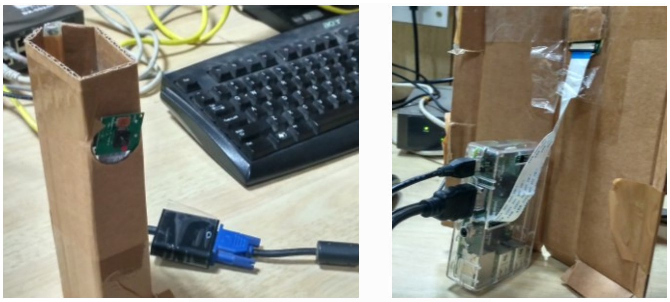

Figure 3. Hardware implementation

Figure 4. (a-e)Positive samples

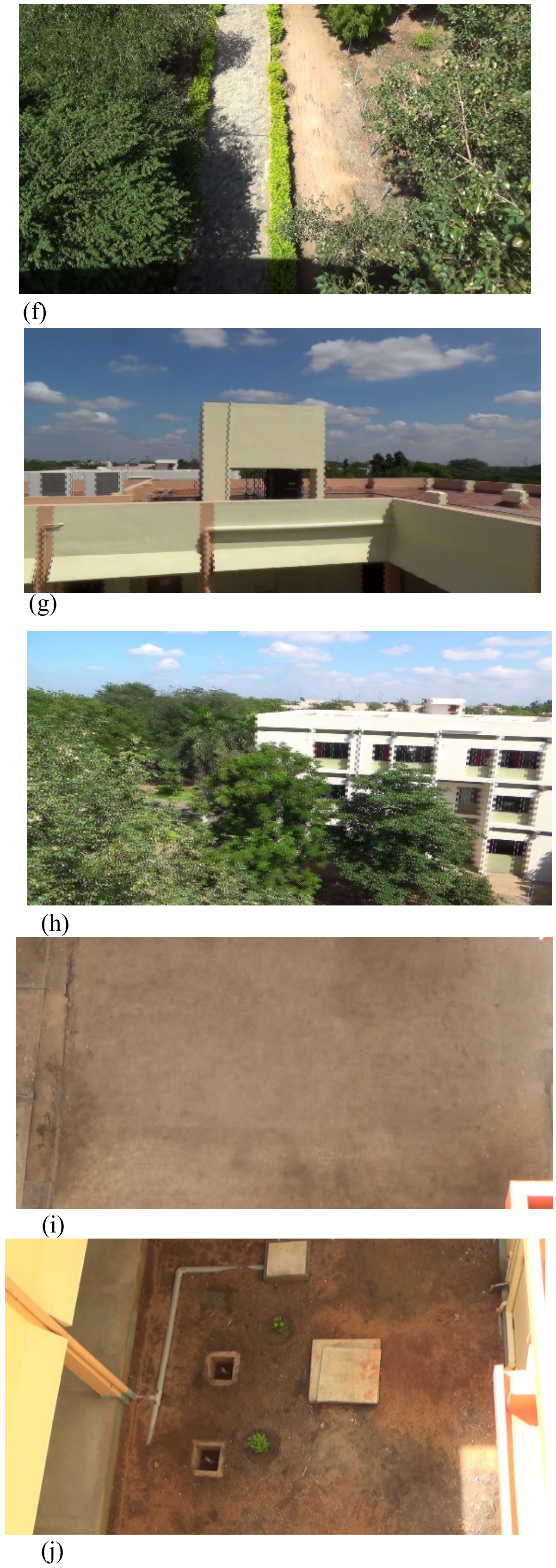

Figure 4. (f-j) Negative samples

V. Implementation

The platform we chose for our study is OpenCV-Python compatible with the Raspberry Pi 3. We implement our study using the tiny computer, Raspberry Pi 3. It's 1.2 GHz CPU provides results in a short time. The video frames are recorded by the camera interfaced with the Raspberry pi.

A. *Training*

Our first step is to train a Haar cascade classifier to detect human heads in a crowd. Cascade classifier training involves collection of positive samples and negative samples. The positive samples are the images which contain human heads and the negative samples (background images) are the images that do not contain human heads. Separate videos are taken in an area with and without humans. The videos are then processed and the frames are segmented. Some of the samples we used for our training purpose are shown in Figure 4.

The frames with the humans are then subjected to an annotation tool which allows us to mark the human heads. The process of marking human heads consumes time but larger the number of images with marked human heads given as input better is the accuracy of head detection. Annotation tool is an OpenCV application which, when executed, opens the images one after the other contained in the mentioned directory. The path to the directory containing the images and the directory wherein the output is to be saved are mentioned in the command line.

A sample of our image with human heads marked using annotation tool is shown in Figure 5. The tool outputs a text file containing in each line the path of each image followed by the pixel locations of the heads marked, each separated by a space. A line of the output text file (annotations.txt) is shown in Figure 6.

Figure 5. Marking human heads using annotation tool

```
/Python27/pos\pos(2).png 3 1101 195 47 49 775 330 57 69 616 332 65 57
/Python27/pos\pos(200).png 2 993 515 64 62 1100 525 60 57
```

Figure 6. Annotation tool output text file (annotations.txt)

Positive samples are then created using the annotation tool output containing the pixel locations of marked human heads in the frame and the background images. The cascade classifier is then trained to build a cascade of weak classifier stages (twenty in our work). The classifier uses the Haar features extracted from the positive samples. A cascade classifier consists of a cascade of weak classifiers each classifying the sub-window on the basis of certain features. The stages in the trained cascade are sorted based on the Adaboost algorithm. The trained cascade is output as an .xml (Extensible Markup Language) file.

B. *Detection*

The video is read frame by frame and each frame is made to pass through all the stages of the trained cascade classifier. A sequence of increasingly complex classifiers is called a cascade. The cascade is shown in Figure 7.

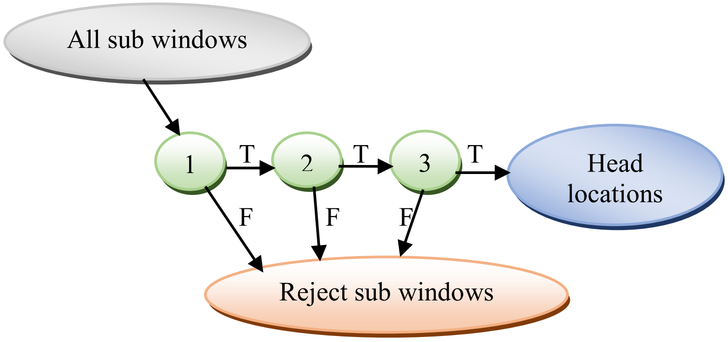

Figure 7. The cascade detection

Each frame is analyzed by several sub windows. The analyzed window is either passed to the next stage or refused by the current stage of the cascade. A window must pass through all the stages. Any stage in the classifier may reject it. After a window passes through all the stages, it is detected as the human head.

C. *Tracking*

The detected human heads are then tracked for their position in successive frames to compute the direction of motion of any person. The tracking is done by optical flow concept. Using optical flow concept, the corners of the bounding box indicating the human heads in a frame are fetched and the corresponding pixels in the successive frames are then found. The direction of movement of a person is determined by comparing the pixel values in each frame. Each person is tracked individually.

VI. Results and Discussion

In our experimental study, we tested our cascade detector on a dataset from Mall dataset and on a real time video recorded in our institution.

A. Using mall dataset:

In our training of the cascade classifier for head detection, we use 530 positive training samples from different video frames and 2000 negative training samples without any human heads. Initially, we mark the head regions in each frame by using the annotation tool. The output of this process is a text file containing the pixel location of each head region in each frame.

We train a cascade classifier of 20 stages. The time taken for the cascade classifier training is about 12 hours. The trained cascade classifier is used for head detection. Frames are tested against the trained cascade classifier. Not every head in the frame are detected, however, satisfactory results are obtained. The result of this trained classifier is shown in Figure 8.

Figure 8. Head detection result – mall dataset

B. Using real time video

We capture two real time videos in our institution using a 12 mega pixel camera one consisting of humans and the other with no persons in the scene. Surveillance cameras in crowd are specifically positioned in a particular angle and hence video is recorded in a specified angle and then trained using that video. We then segment each video into frames and save each in separate directories as positive and negative. The positive images are then marked for human heads using annotation tool.

In the training of our cascade classifier using real time video frames, we use 1000 positive samples and 4400 negative samples. The training of 20 stages cascade classifier takes about 5 days. The trained classifier is then used for head detection. Video frames taken in the same scenario are tested against the trained cascade classifier.

In the frames, after passing through the twenty stages of the classifier, the head regions are detected and results are obtained. The results are more accurate when compared to that using dataset. A sample of the results obtained is shown in Figure 9.

Figure 9. Head detection result – real time video (in our institution)

The count of the persons in the scene is displayed on the display screen. It is seen that training using the images captured in the region where the surveillance camera is to be installed will fetch more accurate results.

C. Tracking using real time video

Tracking a human using OpenCV is done by using optical flow concept. Optical flow involves selecting the features to track and then computing the flow of the features. In our experimental study, the heads detected are indicated by a bounding box. Multiple bounding boxes are detected for multiple humans in the scene. In order to track the persons, the pixel locations of the bounding boxes are fetched. In each frame, the pixel values of the corners of the bounding box are read and stored. The pixel value of the corner of the bounding box in a frame is then compared with the one in the previous frame. The comparison between the two provides the direction of movement of each person in the frame.

We experimented our study in a scene with two entrances where people head towards either the north or the south direction. As soon as a person enter the scene, the head is detected and the corners of the bounding box are fetched. In each successive frame, the values are compared and the direction of motion of each person is determined. The direction is then displayed near the bounding box as 'North' or 'South'. Satisfying results are obtained for different video frames.

Some of the samples of our results obtained are shown in Figure 10.

(a)

(b)

(c)

Figure 10 (a-c). Tracking and Flow results – real time video (in our institution)

VII. Conclusion

Crowd management using head detection is realized using computer vision in our study, implementing our study using video taken from our institution. We use Haar features and Adaboost algorithm to detect the person's head region. We track the human using optical flow concept. Using increased number of samples, the results are found to be efficient. The human detection and tracking can generally be used in surveillance tasks.

References

[1] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", Mitsubishi Electric Research Labs, Cambridge, 2001, IEEE.

[2] P.Papageorgiou, Micheal Oren and Tornaso Poggio, "A General Framework for Object Detection", Center for Biological and Computational Learning Artificial Intelligence Laboratory, Cambridge.

[3] Ashfin Dehghan, Haroon Idrees, Amir Roshan Zamir and Mubrarak Shah, "Automatic Detection and Tracking of Pedestrians in Videos with Various Crowd Densities", Computer Vision Lab, University of Central Florida, Orlando, USA, 2014, Springer International Publishing Switzerland.

[4] Fuentes O, Ramirez G A, "Multi-Pose face detection with asymmetric Haar features", Proceeding of the 15th IEEE Workshop on Apllications of Computer Vision, Copper Mountain, 2008, pp. 1-6.

[5] J.Yang, A. Wailbel, "A real-time face tracker," Proceedings of the third IEEE Workshop on Applications of Computer Vision, WACV96, 1996.

[6] A.Jaysri Thangam, Padmini Thupalli Siva, B.Yogameena, "Crowd Count In Low Resolution Surveillance Video using Head Detector and Color based Segmentation for Disaster Management", 2015, IEEE.

[7] Min Li, Zhaoxiang Zhang, Kaiqi Huang and Tieniu Tan, "Estimating the Number of People in Crowded Scenes by MID Based Foreground Segmentation and Head-shoulder Detection," Institute of Automation, Chinese Academy of Sciences.

[8] K.Kraus, M.Uiberacker, R.Reda and O.Martikainen "Hot-Spot Blob Merging for Real-Time Image Segmentation," World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:2, No:10, 2008.

[9] Mikel Rodriguez, Ivan Laptev, Josef Sivic and Jean-Y ves Audibert, "Density-aware person detection and tracking in crowds," Imagine, LIGM, Universite Paris-Est.

[10] Dongping Zhang, Yafei Lu, Liwei Hu and Huailiang Peng, "Multi-human Tracking in Crowds Based on Head Detection and Energy Optimization," College of Information Engineering, China Jiliang University, Hangzhou 310018, China, 2103 Asian Network for Scientific Information.

[11] Yingjie Chen , Liquan Zhang and Jia Wang, "Passenger Detection for Subway Transportation Based on Video," Beijing University of Civil Engineering and Architecture, Beijing. China. A247, 2014, IEEE.

[12] Chongwen Wang and Meijie Tian, "Passenger Flow Direction detection for Public Transportation Based on Video." 2010 International Conference on Multimedia Communications, Krakow, 2010, pp. 198-201.

[13] Tiancang Du and Songyan Ma. "Improved Adoboost face detection," International Conference on Mechatronics Automation and Measuring Technology, Changsha, 2010, pp. 434-437.

[14] Songyan Ma and Tiancang Du. "Improved Adoboost face detection," International Conference on Measuring Technology and Mechatronics Automation, Changsha, 2010.