Haar Classifier based Identification and Tracking of Moving Objects from a Video Sequence

Visakha K[1], Sidharth S Prakash[1]

[1]*M.Tech in Software Systems, Department of Information Technology School of Engineering, CUSAT, Kochi, India*

Abstract - *Video surveillance and computer vision applications depend on precise detection and tracking of moving objects. This paper explains the various procedures followed to ensure the correct detection of a unique object of any chosen class and tracking it throughout the length of an entire video sequence. A video surveillance system consists of three phases: moving object recognition, tracking, and decision making. Automating the video surveillance process will help in efficient monitoring of the sensitive areas with less human resource utilization. With proper training, the system can be made sufficient to identify moving objects belonging to a particular class such as human beings, vehicles, etc. according to the user needs.*

***Key Words*: Object detection, Object tracking, Haar Feature-based Cascade Classifier, Surveillance systems, Video processing**

1. INTRODUCTION

Detection and tracking of moving objects for video surveillance involves taking the video, identifying particular entities, tracking and understanding their actions and responding appropriately. Automatic video surveillance is becoming increasingly important in many areas, including traffic control, urban surveillance, home security and health care [1][2]. Statistical analysis can be made using the collected data for commercial and research purposes. The goal of this paper is to arrive at a software solution which automatically detects and tracks unique objects from images captured by video cameras. This system can also be used for security applications - for example, surveillance in parking areas, to detect and to provide warning against abnormal activities such as human presence in a shopping complex after normal working hours or a vehicle moving into a restricted area. This is also used in computer vision applications [3][4]. Automating the video surveillance process will help in effortless monitoring of the sensitive areas with less human resource utilization [5].

This paper breaks down the entire process into two main modules - Object detection and Object tracking. Real time video feed or a pre-recorded video sequence can be used as the input from which individual frames are extracted for further processing. Haar Feature-based Cascade Classifier is used to detect a particular object from a given frame. After a classifier is trained, it can be applied to a region of interest in an input frame. The term "cascade" in the name of the classifier implies that the resultant classifier consists of multiple simpler classifiers that are applied to a particular region of interest in a series of stages, until at some stage the candidate is rejected or it passes all the stages. Compared to other features, using Haar-like feature brings about an increase in the calculation speed [6].

Object tracking requires keeping track of each unique object that was detected in the detection phase. The next position where that particular object can be found is predicted and then verified. This can be represented visually using coloured rectangles enclosing the object, with the color being unique to that particular object.

The rest of the paper is organized as follows. Section 2 presents the methodology in which the object detection and tracking are performed. The implementation is described in detail in Section 3. Section 4 contains the conclusion of the paper.

2. METHODOLOGY

For detection of an object, the functions of a Haar classifier are used. The Haar classifier is trained using hundreds of sample views of the object of various shapes and sizes from different angles, called positive samples. For instance, for tracking different cars in a video, the positive samples would include images of cars varying in their model, colour, shapes and taken from different angles. Arbitrary images without the presence of the object, called negative samples, all scaled to the same size are also used [7]. An xml file resulting from this is used to detect the objects from the input video. After the training, it can be applied to a region of interest in each of the extracted frames. The output for this process will be a "1" if the region is likely to contain a sample of the object; and "0" otherwise. The entire frame can be searched for the presence of objects by the application of this process subsequently all over the entire area. However, objects detected in a frame will vary in size based on their position with respect to the camera, their height and shape, etc [8]. To find out objects of different sizes, the classifier's design permits it to resize accordingly. For each frame, this procedure should be done multiple times at varying scales to ensure full coverage.

In object tracking, every unique object detected is tracked throughout the multiple frames of the video sequence. A structure can be used to store the various features of the detected ones. This structure will contain attributes like center(x,y) coordinates, weight, bounding rectangle's color, histograms, small increments dx, dy (that are used in tracking), etc. For each unique object detected, a vector of the type of the structure is created. For each such object, a prediction step is used to find out the next position. Then, different random samples of other objects are generated and their histograms are created along with the given sample. They are assigned weights according to their likeness with the given sample and the sample with the best weight is chosen. According to the weight, each sample is again sampled and, one with highest weight is chosen as new position. This process is repeated throughout the entire video sequence for each object and thus, the individual objects are tracked accordingly. This process follows the sampling-resampling algorithm discussed in detail in the next section. Thus, the true locations of each object throughout the entire video can be found.

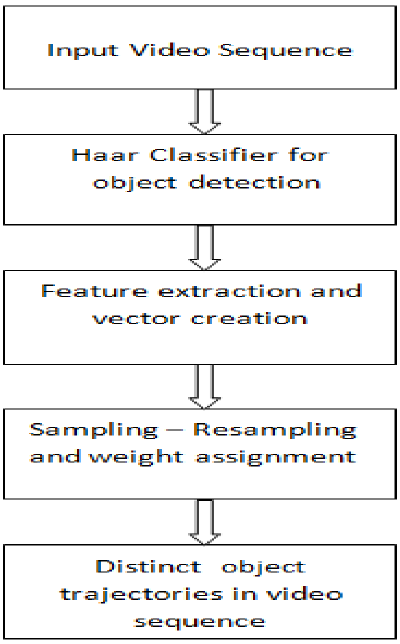

Fig -1: Phases of object detection and tracking from a video input

3. IMPLEMENTATION

3.1 Haar Classifier Training

A Haar Classifier can be trained using positive and negative samples [9][10]. Positive samples contain images of objects of varying size, shapes, etc. taken at different angles which are cropped so that only the desired object is visible. Negative samples are background images without object presence. Within the positive samples, the precise location of the object can be indicated using the BMP file location, number of rectangles, the x/y coordinate of the upper left corner and width/height from this point x/y of each rectangle [11]. The classifier is trained using the ".vec" file that is built by packing in the specifics of the positive samples, and the ".txt" file containing the details of the negative samples. The trained classifier is represented as a ".xml" file which can be implemented in the program.

```
rawdata/FudanPed00001.bmp 2 143 166 175 297 403 157 142 341
rawdata/FudanPed00005.bmp 1 174 38 151 302
rawdata/FudanPed00007.bmp 2 76 51 157 308 386 69 140 312
rawdata/FudanPed00008.bmp 1 237 152 139 294
rawdata/FudanPed00009.bmp 2 150 96 153 301 304 131 158 305
rawdata/FudanPed00010.bmp 1 256 71 150 311
rawdata/FudanPed00011.bmp 1 252 108 194 296
rawdata/FudanPed00012.bmp 2 151 59 159 308 312 49 141 292
rawdata/FudanPed00013.bmp 1 358 170 208 311
rawdata/FudanPed00015.bmp 1 36 33 158 291
```

Fig -2: Representation of positive samples using filename and co-ordinates

3.2 Detecting distinct objects from image frames

Once the Haar Classifier is trained using positive and negative samples, it can be used to detect objects from the input frames. Using as many samples as possible will ensure better probability of identifying target object forms. Once detected, the features of the object are extracted and stored in a vector of type structure with attributes representing the features of that object group. The summary of steps for this procedure is as follows:

For each element in the detected sequence, a check is performed to see whether it overlaps with any element of type structure. If it does, it is ignored as it is already been detected. If not, it means a new object has been detected. The R, G, B histograms for the newly detected object are calculated, along with the coordinates for drawing a bounding rectangle [12]. A random unique colour for the outline of the bounding rectangle is also stored in the vector.

These steps are repeated till the very last frame of the video resulting in the detection of all unique objects. As the number of image frames increases, accuracy of object detection also increases [13].

3.3 Tracking detected object throughout video

For each object being detected by the implementation of Haar Classifier, each step of tracking involves the prediction of the next step of that particular object in the next frame [14]. Instead of repeating this for every frame, this can be done for every n^{th} frame depending on the required accuracy for increasing efficiency. The prediction of location is done using weight assignment to random samples of objects and thus tracing their individual trajectories through the repeated application of this procedure.

For every distinct object in a frame, a prediction step is used to find out the next position. This is done by adding increments dx and dy to the centre co-ordinate (x,y) of the current object. Another vector for samples is created which is also of type structure that represents object features. Sampling and resampling algorithm is applied. Different random samples of other objects are generated. The increments dx and dy are calculated by generating two normally distributed random numbers using Box-Muller transform. The R, G, B histograms are created for comparison along with the given sample. They are assigned weights according to their likeness with the given sample and the sample with the best weight is chosen. According to the weight, each sample is again resampled and the one with highest weight is chosen as new position. Let the centre co-ordinate of this best sample be best_sample(x) and best_sample(y). Given the current position of the object is denoted by the coordinate (x,y), the new increments can be calculated as:

$$new\_dx = (best\_sample(x) - x)$$
$$new\_dy = (best\_sample(y) - y)$$

This process is repeated throughout the entire video sequence for each object.

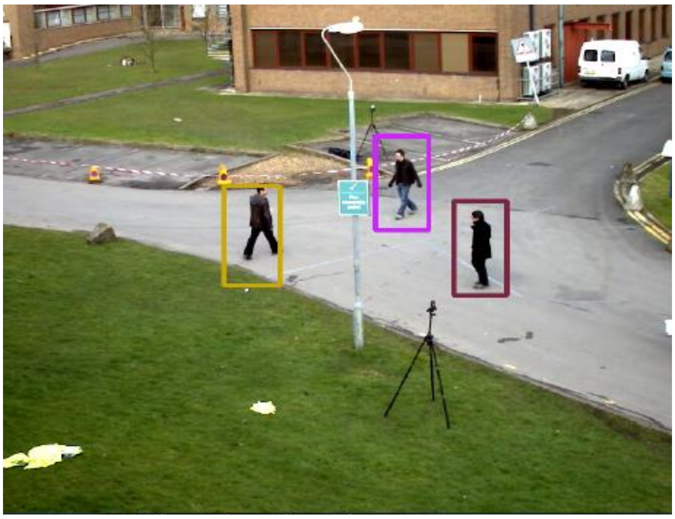

Fig -3: Human detection and tracking from a video

4. CONCLUSIONS

This paper focuses on the detection and tracking of objects from a video footage which can be pre-recorded or live. Detection is made possible by using the functions of Haar classifier. After training the classifier, it can be applied to a region of interest in an input frame. The resultant classifier consists of multiple simpler classifiers that are applied to a particular region of interest in a series of stages, until at some stage the candidate is rejected or it passes all the stages. Compared to other features, using Haar-like feature brings about an increase in the calculation speed and this strategy can be done for any object, by training the Haar classifier accordingly.

Tracking of objects is done using sampling and resampling algorithms. The next position where that object can be found is predicted with accuracy. Positions are represented visually using unique coloured rectangles enclosing the object.

The proposed system is important in many applications, including traffic control, urban surveillance, home security and health care. Scope of the system can be further extended to other applications such as control application areas with certain enhancements like an alarm trigger [15].

REFERENCES

[1] Bashirahamad. F. Momin, Tabssum. M. Mujawar. "Vehicle detection and attribute based search of vehicles in video surveillance system". International Conference on Circuit, Power and Computing Technologies (ICCPCT), March 2015.

[2] Jong Sun Kim, Dong Hae Yeom, and Young Hoon Joo. "Fast and Robust Algorithm of Tracking Multiple Moving Objects for Intelligent Video Surveillance Systems". IEEE Transactions on Consumer Electronics, 57(3), 2011.

[3] Shilpa, Prathap H.L, Sunitha M.R. "A Survey on Moving Object Detection and Tracking Techniques". International Journal Of Engineering And Computer Science, 5(4): 16263-16269, April 2015.

[4] Ali Sharifara, Mohd Shafry Mohd Rahim, YasamanAnisi. "A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection". International Symposium on Biometrics and Security Technologies (ISBAST), pp. 73 - 78 August 2014.

[5] Helly Patel, Mahesh P. Wankhade. "Human Tracking in Video Surveillance". International Journal of Emerging Technology and Advanced Engineering. 1(2), December 2011.

[6] Paul Viola, Michael Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features". Conference on Computer Vision and Pattern Recognition (CVPR), 2001.

[7] Philip Ian Wilson, Dr. John Fernandez. "Facial feature Detection using Haar Classifiers". Journal of Systems, Circuits and Computers(JCSC), 21(4), April 2006.

[8] Open CV Documentation. "Haar Feature-based Cascade Classifier for Object Detection". http:// docs.opencv.org/2.4.13.2/ modules/ objdetect/ doc/cascade_classification.html#haar-feature-based-cascade-classifier-for-object-detection.

[9] Xuezhi Xiang, Wenlong Bao, Hanwei Tang, Jiajia Li, Yimeng Wei. "Vehicle detection and tracking for gas station surveillance based on AdaBoosting and optical flow". 12th World Congress onIntelligent Control and Automation (WCICA), pp. 818 – 821, June 2016.

[10] Ravi Kumar Satzoda, Mohan Manubhai Trivedi. "Multipart Vehicle Detection Using Symmetry-Derived Analysis and Active Learning". IEEE Transactions on Intelligent Transportation Systems, 17(4), April 2016.

[11] Florian Adolf. "How-to build a cascade of boosted classifiers based on Haar-like features". http:// lab.cntl.kyutech.ac.jp /~kobalab/ nishida/ opencv/OpenCV_ObjectDetection_HowTo.pdf. September 2003.

[12] Shiying Sun, Ning An, Xiaoguang Zhao, Min Tan. "Human recognition for following robots with a Kinect sensor". IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1331 - 1336 December 2016.

[13] Murat Ekinci, Eyup Gedikli. "Silhouette Based Human Motion Detection and Analysis for Real-Time Automated Video Surveillance", 13(2): 199-229, 2005.

[14] Burcu Kır Savaş, Sümeyya İlkin, Yaşar Becerikli. "The realization of face detection and fullness detection in medium by using Haar Cascade Classifiers". 24th Signal Processing and Communication Application Conference (SIU), pp. 2217 – 2220, May 2016.

[15] Suzaimah Bt Ramli, Kamarul Hawari B Ghazali, Muhammad Faiz Bin Mohd Ali, Zakaria Hifzan B Hisahuddin. "Human Motion Detection Framework". IEEE International Conference on System Engineering and Technology, pp. 158-161, June 2011.