I J C T A, 9(14) 2016, pp. 6691-6700

Efficient Human Detection Technique for Intrusion Detection Systems

Abel Jaba Deva Krupa[1], Dhanalakshmi Samiappan[2], Niraimathi Ponnusamy[3], V. Hemalatha[4] and S. Chithira[5]

ABSTRACT

The research in intrusion detection system demands for efficient human detection algorithm. The challenge is to detect the intruder from the given video data with reduced false alarms. Several algorithms exist for tracking the people in a given frame which deals with outdoor images in most cases. In this paper, we propose a technique to detect the intruder presence in a given indoor image frame. A complete framework to extract the object from the given video data and to determine the extracted object is a human or not is provided. The proposed algorithm is an iterative technique which reads the frames from the input video and processes each frame using a robust object extraction. Unlike existing sliding window techniques, we propose a method which first extracts the object from given image before processing for recognition. This is found to reduce the overall processing time. A well-known descriptor, Histogram of Orientation gradients (HOG) is used for object description. This descriptor is used together with Linear Support Vector Machine (SVM) Classifiers for object classification purpose. By using the proposed object extraction technique, we conclude that the human detection algorithm proposed in this paper performs better than the state-of-art sliding window based detection method.

***Index Terms*:** Histogram of Orientation Gradients (HOG), Intrusion detection systems, Object detection, Object extraction, Support Vector Machine (SVM).

I. INTRODUCTION

Intrusion detection systems are being widely employed because of their increasing need in providing security in homes, banks and many other commercial applications. The primary components of the system are processing unit and the sensors. Different types of sensors can be employed which are application specific and performs functions like detecting intruders, fire accidents etc. Most of the intrusion detection systems are employed with cameras which either continuously operate or can be triggered by a sensor after the detection of suspicious events. These images taken from the camera are processed for human detection.

Human detection is pervasive in different areas like pedestrian detection, person tracking and identification, person detection in dense crowds and people counting. In most of the applications, the technique adopted for human detection is the Object recognition process. Object recognition is a two stage process involving object description and object classification. Object description deals with describing an object in a way the computer understands. The descriptors describing the objects otherwise called as feature vectorsare given to the classifiers for detection of particular class of object.

Our problem of interest is associated with human detection where the classifier has to classify whether the given object is human or not. The overall performance of the system is based on the descriptor chosen. Thus a careful attention is paid while selecting the descriptors.Several object descriptors are found in literature. The descriptor extracted from the object has to hold most relevant information about the object and should be invariant to changes in illumination, viewpoint etc. A detailed focus on existing object detection techniques is provided in Section II.

[1,2,3,4,5] SRM University, Kattankulathur - 603 203, Kancheepuram - Dist., Tamil Nadu, India, *E-mails: jabadevakrupa.a@ktr.srmuniv.ac.in, dhanalakshmi.s@ktr.srmuniv.ac.in*

A well-established technique for human detection is proposed by Navneet Dalal [2]. This technique is based on sliding window algorithm, where a window is chosen in the image and a descriptor called Histogram of Oriented Gradients (HOG) is computed over the window. Classifier is run on HOG computed in this window for human detection. This process is repeated over all the imageregions by sliding the window in horizontal and vertical direction. This sliding window method is proved to perform well for pedestrian detection application. But, unlike Dalal's work which is meant for pedestrian detection, we do not use the sliding window technique, as our application is intrusion detection. This is because of the fact that there is large difference in the number of persons available in the images obtained in pedestrian detection and intrusion detection problems. So instead of scanning all the regions in an image and processing it for object recognition task, we first perform object extraction process on the images obtained which results in the single or multiple objects. We then consider each object individually and process it for object recognition. Thus the novelty of the proposed human detection technique is that it eliminates the need for scanning each image region which will speed up the overall detection process.

The paper is organized as follows. In Section II, we present a review on existing object detection techniques. Following that we focus on the object descriptor and classifier used in our approach. In Section III, we discuss the details of the proposed object extraction technique. The overall algorithm of human detection in a given video is discussed in Section IV. The results obtained for the image frames extracted from a real time video is provided in Section V. A Comparison is made between the proposed algorithm and Dalal's approach in terms of run time of the algorithm. Finally we conclude our work in Section VI.

II. REVIEW OF OBJECT DETECTION ALGORITHMS

The performance of any object detection algorithm depends on the object descriptor used. Several objects descriptors are found in literature. In this Section, we review some of them and then focus on the object descriptor called Histogram of Oriented Gradients (HOG) [1] used in our approach. Broadly categorizing, the object descriptors can fall under three types. Model-based object descriptors [18], Example-based [19] andfinally based on image pattern relationships that determines the object uniquely [20]. Example based descriptors learn the salient features of a class using the set of positive and negative examples. They have been successfully used in the areas of object recognition and other computer vision algorithms.

Such example based descriptor is proposed by Papageorgiou in his work for pedestrian detection problems [20][21]. Leibe *et al.* [6] used a key point detector which extracts the relevant information by considering only the local regions in the object. Lowe in his paper [7] proposed one such key point detector which is based on image gradients called Scale Invariant Feature Transformation (SIFT) descriptor. SIFT descriptors computes the feature vectors which are scale and rotation invariant. Other such similar descriptor is shape context descriptor proposed by Belongie et al.[5] Both SIFT and Belongie shape context computes local histograms of image gradients. SIFT uses rectangular grids and shape context uses log polar grids for computing the histogram.

As a motivation from these approaches, NavneetDalal proposed an object descriptor called Histogram of Orientation gradients (HOG) descriptor [1]. In his work, Dalal used this HOG descriptor for human detection particularly for pedestrian detection problem. In [1], it is shown that the grid of HOG outperforms the existing gradient based descriptors.

The HOG descriptor proposed by Dalal *et al.* has been used as a feature vector in several algorithms Cascade of HOG is used for fast human detection in [11]. HOG is extensively used for pedestrian detection problems in many articles [12], [15], [16]. Recent work proposed by Haghighat, Mohammad et al[14] uses HOG for single sample face recognition techniques. Thus it is evident that HOG is a powerful descriptor successfully used. With this motivation, we use HOG for detecting humans in intrusion detection systems.

Fig. 1 shows the brief steps involved in computing HOG. The detailed description and need for each step is provided in [1], [2].

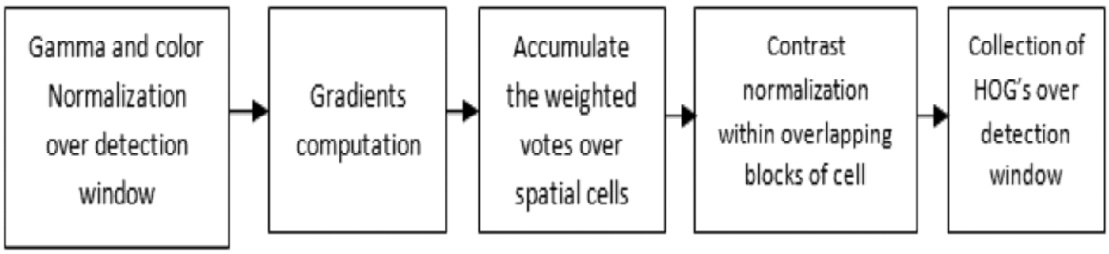

Figure 1: Computation of Histogram of Oriented Gradients over a detection window.

In Fig. 1, we have shown the HOG computation over a given detection window. However in our algorithm the HOG will be computed over a single detection window instead of sliding through the image. This simplification is due to the proposed object extraction technique to be discussed in Section III.

Thus, the proposed algorithm for human detection involves three stages. First task is to extract the moving object from the images using object extraction technique. Second task is to find the HOG for the objects obtained in object extraction process. Finally, the extracted HOG is given to the classifiers for classification process. The classifier we use is linear SVM [7] as it is proved to give better results when used together with HOG descriptors for human detection problem.We provide detailed explanation on classifiers in section IV.

III. PROPOSED OBJECT EXTRACTION TECHNIQUE

Object extraction is the process of finding the moving objects from the images. Given the input image and the background image, the state-of-art technique for finding the moving object is Background Subtraction. Simple technique involves direct subtraction of the pixel values to obtain the foreground mask or the moving object. This is because background image will have only the static objects and the input image will have the moving object present in the background scene. Thus, subtracting these two images results in the foreground object. But the disadvantage with this method is that it will not guarantee the extraction of complete object. This occurs in cases where the pixel value of the moving object and the background scene are same.

In this paper, we present a robust object extraction technique which will not only consider the intensity difference between the background and the input image but also takes into account other property differences like texture, luminance and chrominance. By considering all these differences together, we can guarantee a better object extraction.

According to the proposed object extraction process, for the given input and background image, we find the Luminance, Chromatic and Texture differences. We call them as L-map, C-map, and T-map respectively. All these maps are ORed together to result in a moving object. In this section, we discuss the procedure for finding these maps.

(A) Texture Difference (T-Map)

We often relate the texture property by the correlation coefficients. For each pixel in the image, we consider a block of size $(2M+1)\times(2N+1)$. The texture description of this image block is calculated using the following autocorrelation function R,[6]

$$R(x,y) = \frac{(2M+1)(2N+1)}{(2M+1-x)(2N+1-y)} \times \frac{\sum_{m=0}^{2M-x}\sum_{n=0}^{2N-y} B(m,n)B(m+x,n+y)}{\sum_{m=0}^{2M}\sum_{n=0}^{2N} B^2(m,n)} \quad \frac{0 \le m \le 2M}{0 \le n \le 2N} \tag{1}$$

Here *x*, *y*are the position displacements in the *m*, *n* direction, *B*(*m*,*n*) represents the intensity value at (*m*,*n*) in the image block *B*. Using the equation(1), we compute autocorrelation for input as well as background frames. The mean square difference between these autocorrelation functions *R* of each input image block B_i with the same location of background image block B_b gives the texture difference as given by the equation(2).

$$d_T(u,v) = \frac{1}{(2M+1)(2N+1)} \sum_{x=0}^{2M}\sum_{y=0}^{2N} [R_{(u,v)i}(x,y) - R_{(u,v)b}(x,y)]^2 \tag{2}$$

Finally, the T-map is obtained by thresholding the texture difference $d_T(u,v)$.

$$T(u,v) = \begin{cases} 1 & d_T(u,v) > \tau_t \\ 0 & otherwise \end{cases}$$

We use iso data algorithm [2], [8] for finding the threshold.

(B) Luminance and Chrominance Difference (L-map, C-map)

The first step in finding L-map and C-map is to find the YCbCr model for the given images. YcbCr model is used here because it separates the luminance and the chromatic components of an RGB image. Now the luminance and the chrominance differences can be calculated using the equations (3) and (4) respectively.

$$d_y(u,v) = \begin{cases} Y_i(u,v) - Y_b(u,v) Y_i(u,v) - Y_b(u,v) > 0 \\ 0 \qquad\qquad otherwise \end{cases} \tag{3}$$

$$d_c(u,v) = [Cb_i(u,v) - Cb_b(u,v)]^2 + [Cb_i(u,v) - Cb_b(u,v)]^2 \tag{4}$$

In equation (3) and (4), *Y,Cb* and *Cr* denote respectively the Luminance component and Chrominance components. The subscript *i* denotes the input image and *b* denotes the background image.

The binary images then are computed by thresholding d_y and d_c as we did for obtaining T-map. We call the resulting images as L-map and C-map respectively.

Finally, all the maps obtained i.e. L-map ,C-map and T-map are OR'ed to obtain the foreground image.

$$ORmap = L_{map} + T_{map} + C_{map} \tag{5}$$

As given in equation (5), the OR map will give the moving object which can be further processed for recognition task

IV. HUMAN DETECTION ALGORITHM

The input data in an intrusion system will be a video usually with small frame rates. The challenge is to detect the presence of human in any of the frames[22]. The usual technique adapted for this is to process the frames separately for the human presence. Our proposed algorithm is one such technique which is iterated for all the frames of the input video. If any of the frames is found to be a positive sample, then the output of the algorithm is positive. Fig.2 shows the flow chart for the human detection in given video.

The first step is to extract the frames from the given video and then process them individually. Each extracted frame is added to the data matrix D given by

$$D = [[f_1]_{MxN}[f_2]_{MxN} \quad \ldots [f_t]_{MxN}] \tag{6}$$

Here, f_i is the i^{th} frame with *MxN* pixels and the order of *D* will be *MxtN*, t being the number of frames.

In each iteration, the algorithm considers a*MxN*matrix extracted from *D* for processing. The detailed steps involved in the proposed algorithm in i^{th} iteration are given below. Fig 2 shows the flow chart for overall human detection algorithm in given video.

Step 1: Extract the frames from the video and form the data matrix *D*.

Step 2: Extract the sub matrix $F = D(1:M, iN+1:(i+1)N),\ i = 0,1 \ldots (t-1)$

Step 3: Run the human detection algorithm for images as shown in Fig with *F* as input image.

Step 4: If the algorithm results positive for any of the sub matrix *F*, terminate the algorithm. Otherwise proceed for next iteration.

Step 5: If none of the frames yields positive result, then no human is present in the given data.

Human detection algorithm for an input image is of two fold. First is the object extraction as explained in section III and following that is the object recognition process. The two stages involved in object recognition is *object description*- determine the feature vector or descriptor containing the information about the object or it describes the object in the way a machine can understand and *object classification*- classifier run on descriptor to determine the class of the object.

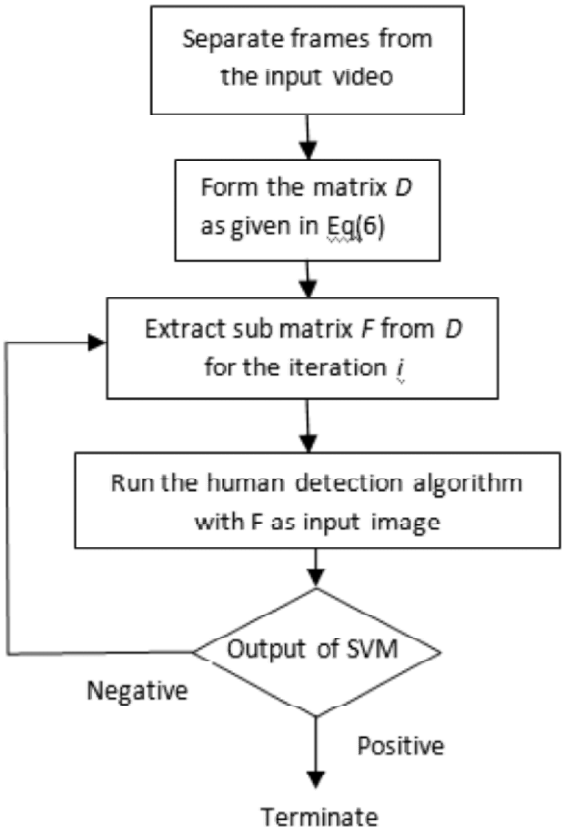

Figure 2: Iterative algorithm to find human object in video

Classifiers are basically machine learning algorithm which provides object/non-object decision. They can be either binary or multiclass classifiers. The former suffice for our problem as it provides only two decisions, human or non-human decision. Support Vector Machine (SVM) is one of the extensively used binary classifiers. They exhibit good generalization and learning with optimization [7]. We use linear SVM for classification.

We do not talk much about this part in our paper. But we provide the method for learning aclassifier in a nutshell as follows:

- Prepare a dataset of positive and negative images for training the SVM
- Find the descriptors for all the positive and negative images which can be used to train the SVM classifier [7].
- The resulting will be a trained classifier which can now make object or non-object decision.

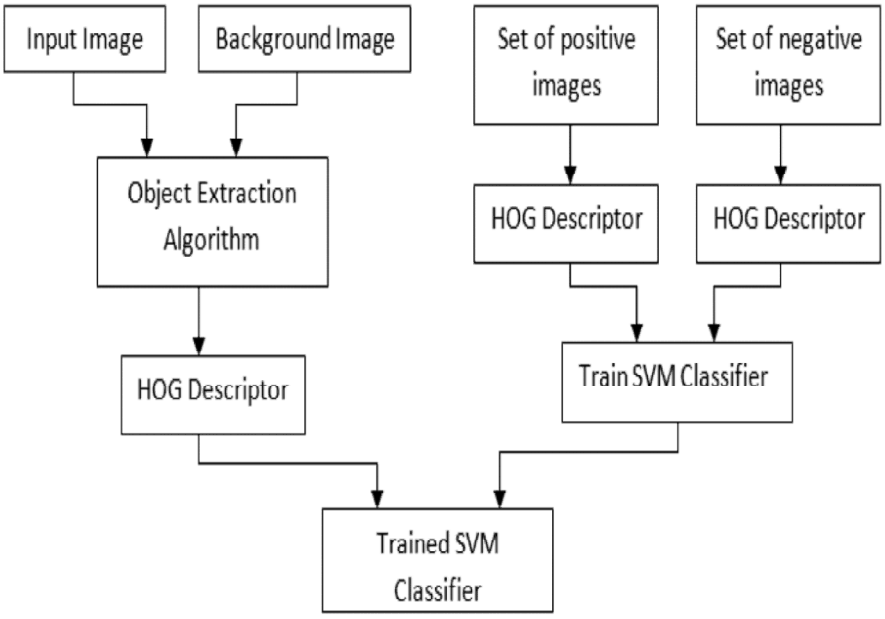

Figure 3: Flow Diagram for finding human objects in images

We now present the overall algorithm for detecting human in the images extracted from the video obtained bythe surveillance camera. The algorithm involves two independent processes, one dealing with object extraction and other deals with training the classifier as shown in Fig 3. The following are the detailed steps.

- For the object extraction algorithm, we need a background image and the input image will be the frame obtained in step 2 of algorithm shown in Fig 2.
- Using these two images, we determine the foreground image using the technique explained in section III.
- The moving object can be easily separated from the foreground image based on the intensity values (0 for background and 1 for moving object).
- Determine the HOG descriptor for this object using the method shown in Fig.1. This is fed to the trained classifier for classification purpose.
- Train the SVM classifier using the positive and negative samples of training data.

– Run the classifier on the HOG descriptor found for the moving object.

The classifiers output will be a decision function which can be +1 if the object is human or -1 if the object is not human.

V. EXPERIMENTAL RESULTS

The proposed algorithm seeks for improvement in computation time by avoiding the sliding window mechanism. The procedure adopted in [2] for pedestrian detection is that the full image with object is scanned in all regions and HOG is found in each region. Then a classifier is run on the each region for classification purpose. The process is repeated by considering the image at different scales. This can be well served for pedestrian detection application because the input images can have numerous people on roads or any other open environment. But for the human detectiontask in surveillance systems, the images obtained cannot have such crowd of human. Thus scanning all the image regions may lead to unnecessary computation burden. Thus, by using the proposed object extraction algorithm described in section II, we determine the region of interest, the object and then perform the description and classification task. This can save the time required for finding the descriptor and running the classifier trained with rich dataset every time on all regions of image at different scales.

The input to our algorithm is a video with frame rate 15fps captured by real time camera in intrusion detection system. We have shown the results of object extraction obtained for two frames taken from the video in Figure 4 and Figure 5. The background image is obtained at the time of installation of intrusion system at user's zone.

Fig 4(a) shows the input image and Fig 4(b) shows the background image. The L-map, C-map and T-map obtained for these images on applying the algorithm described in section II are as shown in Fig 4(c),

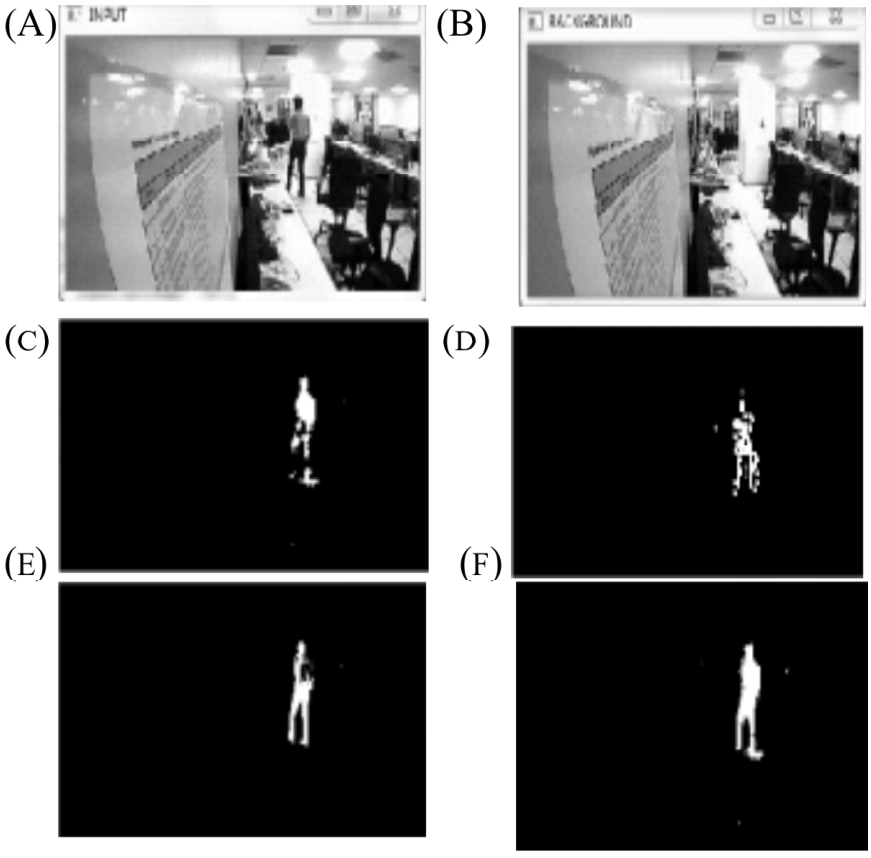

Figure 4: (a) Input Image (b) Background Image (c) L-map (d) C- map (e)T-map (f) Extracted foreground image

(d) and (e) respectively. Fig 4(f) shows the final binary image obtained with the moving object being extracted. This image is the result of performing OR operation on the images shown in Fig 4(c), (d) and (e).

It is to notice that the amount of information carried in different maps varies with images. For the given input and background image shown in fig 4(a) and (b), luminance and texture difference contribute lot while chrominance difference image carries only small information about the object. But all the three maps together contribute to the extraction of full object as shown in Fig 4(f). Once the binary image with moving object is found, it is very easy for us to extract the object from the input image based on the pixel coordinates. Fig 6 shows the object extracted from the input image based on the coordinates of the image shown in Fig 4(f). The foreground image obtained for the input image shown in Fig.5 (a) is shown in 5(f). The objects extracted from the foreground images 4(f) and 5(f) are shown in Fig.6 (a) and (b).

The frame 4(A) comes first and later comes 5(A) in the input video. Thus frame 4(A) will be first processed according to the algorithm given in Fig.2 and we extract the object 5(a). We find the HOG descriptor for this and feed it to the SVM classifier.

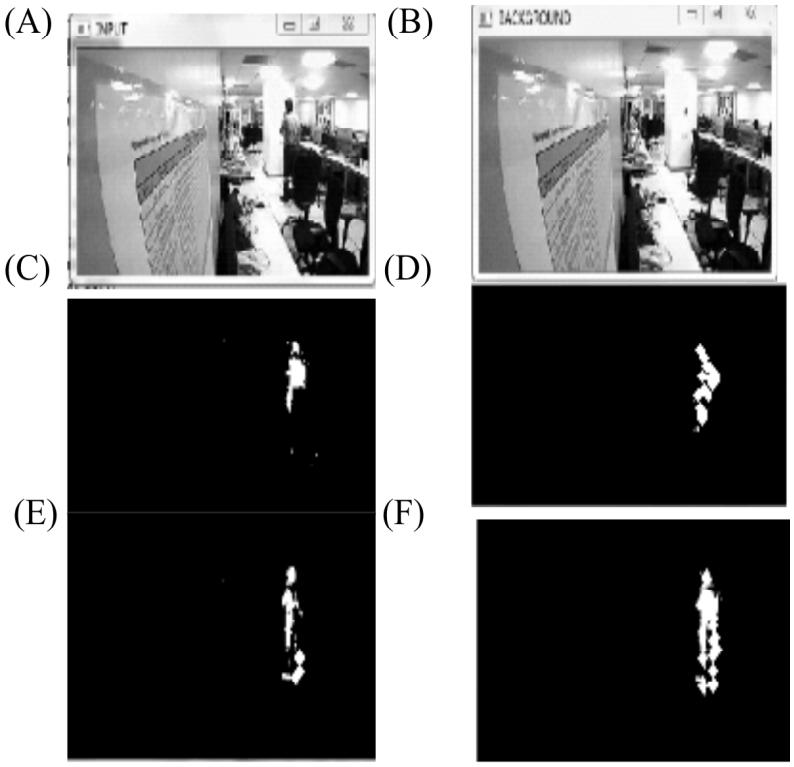

Figure 5: (a) Input Image (b) Background Image (c) L-map (d) C- map (e)T-map (f) Extracted foreground image

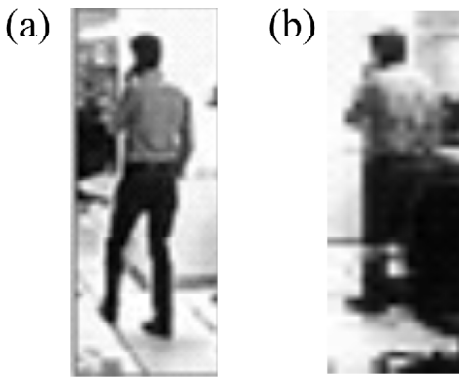

Figure 6. Extracted object from the image (a) For input image 4(A) (b) For input image 4(B)

For training the classifier, we used set of human images taken under different luminance conditions and also human with varying poses as positive samples. While training the SVM, the label given for positive sample is +1 and the label for negative sample is -1. We have taken the images from INRIA person dataset [9]. For negative samples, we took the images from indoor environment CVPR dataset [10]. Some of the positive and negative samples are shown in Fig 7.

For the object in 5(a), the result of SVM classifier is +1, i.e. the label associated with positive sample which means that the input object is human. Now the algorithm terminates with a decision that intruder has entered the armed zone. We have tested the algorithm for negative cases also, where SVM classifier output was -1. This shows that the given input object does not belong to the human class.

In terms of computation, our proposed method can perform faster than Dalal's approach. This is because, in Dalal's work, each region in the input image has to be processed for object recognition i.e. the input image shown in Fig 2(a) is the input to the object recognition task.Here, we consider each region in this image, find HOG descriptor and run the trained classifier for decision. But by using our proposed technique, we take out only the region of interest through the object extraction task and then perform recognition process which will eliminate the unnecessarycomputations.

Figure 7: Samples from positive and negative training dataset.

The Table 1 shows the time taken by Dalal algorithm and our proposed algorithm to find the human detection in image shown in Fig 2(a).

As we can see from the Table 1, our algorithm can perform faster than Dalal approach for finding human in images in particular for surveillance systems. It has to be noted that the data given in table depends on several factors like size of image, size of training set for building classifier etc.

Table I
Time Taken For Human Detection

Algorithm	*Time (sec)*
Dalal	126.5
Proposed	90

VI. CONCLUSION

This paper focuses on human detection in images particularly for surveillance systems. We proposed an object extraction technique which can enhance the overall detection task. The object extracted alone can be

processed for recognition while eliminating the computation burden on other regions. This can improve the system speed and makes it efficient. In our results section, we proved this by providing a comparison table showing the speed of our proposed algorithm and the state of art technique. Our approach can well serve for extracting the multiple moving objects from given image. This approach can be extended in finding the partly visible persons. We can implement part based detection for the extracted objects in this case.

REFERENCES

[1] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[2] Dalal, Navneet. "*Finding people in images and videos*", Diss. Institut National Polytechnique de Grenoble-INPG, 2006.

[3] Lam, William Wai Leung, Clement Chun Cheong Pang, and Nelson HC Yung. "Highly accurate texture-based vehicle segmentation method." *Optical engineering* 43.3 (2004): 591-603.

[4] Lowe, David G. "Distinctive image features from scale-invariant keypoints."*International journal of computer vision* 60.2 (2004): 91-110.

[5] Belongie, Serge, Jitendra Malik, and Jan Puzicha. "Shape matching and object recognition using shape contexts." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.4 (2002): 509-522.

[6] Leibe, Bastian, Edgar Seemann, and BerntSchiele. "Pedestrian detection in crowded scenes." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[7] Sonka, Milan, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[8] Vapnik, Vladimir. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[9] Bezdek, James C. "A convergence theorem for the fuzzy ISODATA clustering algorithms." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1 (1980): 1-8.

[10] http://pascal.inrialpes.fr/data/human

[11] Quattoni, Ariadna, and Antonio Torralba. "Recognizing indoor scenes."*Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[12] Zhu, Qiang, et al. "Fast human detection using a cascade of histograms of oriented gradients." *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. IEEE, 2006.

[13] Suard, Frédéric, et al. "Pedestrian detection using infrared images and histograms of oriented gradients." *Intelligent Vehicles Symposium, 2006 IEEE*. IEEE, 2006.

[14] Déniz, Oscar, et al. "Face recognition using histograms of oriented gradients." *Pattern Recognition Letters* 32.12 (2011): 1598-1603.

[15] Haghighat, Mohammad, Mohamed Abdel-Mottaleb, and WadeeAlhalabi. "Fully automatic face normalization and single sample face recognition in unconstrained environments." *Expert Systems with Applications* 47 (2016): 23-34.

[16] Bertozzi, Massimo, et al. "A pedestrian detector using histograms of oriented gradients and a support vector machine classifier." *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*. IEEE, 2007.

[17] Kobayashi, Takuya, Akinori Hidaka, and Takio Kurita. "Selection of histograms of oriented gradients features for pedestrian detection." *Neural Information Processing*. Springer Berlin Heidelberg, 2008.

[18] Yuille, Alan L. "Deformable templates for face recognition." *Journal of Cognitive Neuroscience* 3.1 (1991): 59-70.

[19] Vaillant, Régis, Christophe Monrocq, and Yann Le Cun. "Original approach for the localisation of objects in images." *IEE Proceedings-Vision, Image and Signal Processing* 141.4 (1994): 245-250.

[20] P. Sinha, Object Recognition via Image Invariants: A Case Study,Investigative Ophthalmology and Visual Science, vol. 35, pp. 1735-1740, May 1994.

[21] Mohan, Anuj, Constantine Papageorgiou, and TomasoPoggio. "Example-based object detection in images by components." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.4 (2001): 349-361.

[22] Papageorgiou, Constantine, and TomasoPoggio. "A trainable system for object detection." *International Journal of Computer Vision* 38.1 (2000): 15-33.