International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Computer Vision based System to Detect Abandoned Objects

Balasundaram A1, Chellappan C2 ,

***Abstract:** Security has become the most essential aspect in every walk of life. With ever growing technology, there has been an apparent inclination towards developing smart surveillance systems that do not stop with just monitoring and recording of events, but also possesses the ability to observe the events and alert in case of any discrepancies if observed. One of the impending threats to security in crowded places is the ignorance of un-attended objects. These un-attended objects or abandoned objects may have been left behind intentionally and may contain hazardous things which can cause huge disasters. This paper is focused towards developing a computer vision based approach that analyses the blob areas to detect any abandoned objects and instantaneously send appropriate alert without any human intervention. The uniqueness of this approach is that it handles the occlusion scenario and also quick execution time to detect abandoned objects. The approach has been tested with various benchmark video datasets and real-time video sequences as well. The performance has been measured in terms of accuracy of classifying abandoned objects in a given video sequence and also the execution time taken for computing the outcome. Results indicate good accuracy in terms of abandoned object detection under varying conditions and scenarios and also faster execution time when compared to other contemporary approaches.*

***Key words:** Abandoned object, background subtraction, blob analysis, object detection, video surveillance*

I. INTRODUCTION

With the increasing security threats posed time and again, deployment of advanced and smart surveillance systems has become the need of the hour. These smart systems should posses the ability to closely monitor the area under surveillance and at the same time should automatically detect any threats and instantaneously alert the concerned person. The smart systems should have the ability to identify objects and track them as discussed in [14] and [15], determine any anomalous behavior posed by human or objects. One such anomaly is to instantaneously detect the presence of any abandoned object that is dropped in the region of surveillance. This is an essential and probably the most critical aspect that should be available as part of any smart system especially with the various security threats posed by leaving unattended objects. These abandoned objects may carry any items which may pose a serious threat to public especially in crowded places like railway stations, airport, malls etc. Some examples of abandoned objects are shown in Figure 1. The significance of having an automatic system to detect abandoned objects has been compounded by the difficulties faced by human in continuously monitoring the system actively and the increasing threat of missing any abandoned object by mistake. This paper describes a new blob analysis based approach towards detecting abandoned objects in real-time environment.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Balasundaram A*, Assistant Professor, Department of Information Technology, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, India. E-mail: balasundaram2682@gmail.com

Chellappan C, Professor, Department of Computer Science and Engineering, G.K.M. College of Engineering and Technology, Chennai, India.

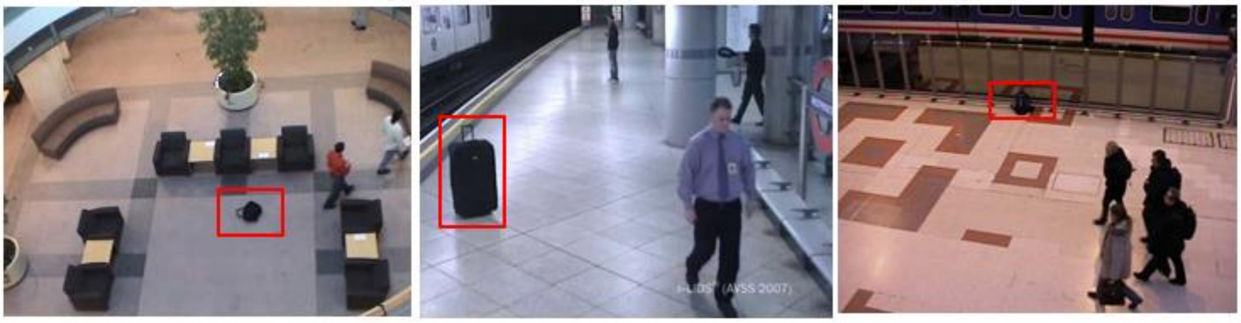

Figure 1. Some examples of abandoned objects

Previous works carried out towards abandoned object detection can be broadly classified into three different categories as shown in Figure 2 namely, background subtraction based approaches, tracking based approaches and finally model based approaches. While background subtraction yields good result and recall, it is highly reliant on the static background and is highly affected by noisy data. Though tracking based methods focus on tracking the object at pixel level, they fail miserably in case of occlusion scenarios. The model based methods are pretty accurate in cases where there is minimal noise interference.

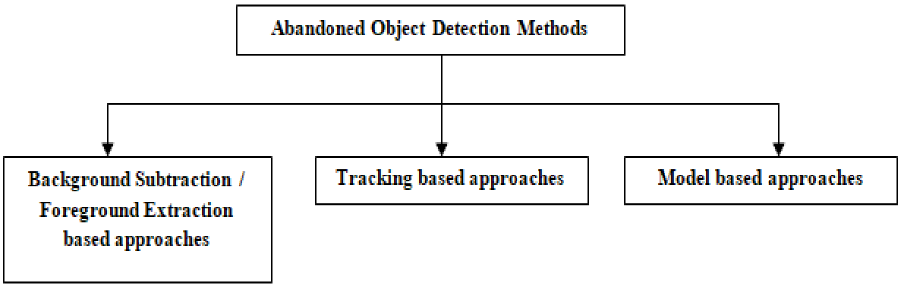

Figure 2. Taxonomy of Abandoned Object Detection Methods

Table 1 strikes a comparison between the three different abandoned object discussion methods discussed so far.

Table 1. Comparison of different methods for abandoned object detection

Approach	Description	Advantages	Limitations
Background subtraction	A Static background is fixed and objects are identified against the static background	Easy to implement Good accuracy Good recall	Does not work well with noisy data.
Tracking methods	Analyze the objects based on the motion history of objects using camera	Pixel level information tracked	Fails in clustered environment and occlusion scenarios.
Model based methods	Mathematical models are defined to identify the abandoned objects	Good accuracy and recall	Vulnerable to noisy data

II. LITERATURE REVIEW

The different aspects that were covered as part of the literature survey are depicted in Figure 3.

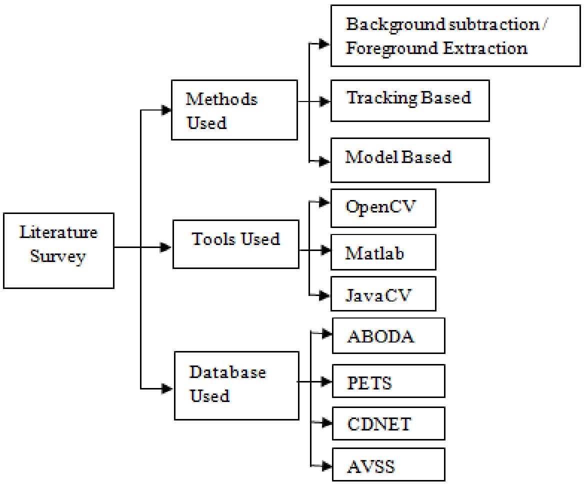

Figure 3. Taxonomy of the literature review covered

The literature review was performed by taking into consideration the following three aspects namely the methods used to detect and identify abandoned objects, the various tools that are used in aiding the development of programs and the various benchmark datasets that were used to evaluate and report the performance of the approaches that were developed for detecting abandoned objects.

Abandoned object detection was carried out by using any one of the following methods like background subtraction, foreground extraction, tracking based methods and model based methods. The implementations in the literature review used various tools such as OpenCV, Matlab and JavaCV. Also, these works used several datasets such as ABODA, PETS, CDNET and AVSS datasets. The performance of various abandoned object detection techniques was evaluated by testing the approaches with these datasets.

2.1. Related work

Table 2 shows the different works carried out towards abandoned object detection. It can be observed that most of these works would have followed any one of the approaches discussed earlier.

Table 2. Review of various works towards abandoned object detection

Reference Number	Approach used	Database	Advantages
[1]	Gaussian mixture based framework was developed for abandoned object detection.	Background subtraction and foreground extraction	PETS 2006 and i-LIDS
[2]	An edge based abandoned object detection mechanism was proposed.	Background and foreground edges	PETS2006, i-LIDS, CDnet 2014, ABODA
[3]	An approach based on short and long term background models had been used to identify abandoned objects	Background subtraction	PETS 2006 and AVSS 2007
[4]	A new strategy based on three background-foreground non parametric detectors had been developed to detect stationary objects	Model based, Foreground extraction, Finite State Machine	PETS2006 and LASIESTA
[5]	A novel approach for indexing video events was proposed, In this abandoned object detection is a sub-system	Frame Differencing and Tracking method	PETS2006
[6]	A pixel wise method using dual foregrounds to detect abandoned objects was proposed	Foreground extraction	PETS 2006 and i-LIDS
[7]	A new approach to determine abandoned object based on the proximity to owner has been proposed	Foreground mask sampling and selective tracking	PETS 2006 and AVSS2007
[8]	A static foreground blob based technique was proposed to detect abandoned objects	Blob analysis	PETS2006
[9]	A background modeling technique to classify abandoned objects has been proposed.	Background Modeling and Finite State Machines	PETS 2006, i-LIDS and CAVIAR
[10]	A fully-automatic grab-cut object segmentation approach had been proposed to detect abandoned objects.	Background Modeling, Gaussian mixtures	CAVIAR, PETS2006, CDnet 2014

2.2. Commonly used datasets for abandoned object detection

It can be seen from the above table that most of the works use PETS2006, i-LIDS AVSS 2007, ABODA, CDNET2014 and CAVIAR datasets. A thorough study of each data set and its features was carried out. Table 3 strikes a comparison among these datasets based on various parameters.

Table 3. Comparison of various datasets used for abandoned object detection

Dataset	Description	Complexity	Illumination Changes	Occlusion	Crowd density
PETS2006	This comprises of 7 video sequences which contain abandoned luggage.	Simple, medium and complex	No	Yes	Considers all cases from less to thick crowd density.
iLIDS AVSS2007	Contains 7 video sequences which were taken outdoor and contains abandoned objects.	Simple, medium and complex	No	Yes	Considers all cases from less to thick crowd density.
ABODA	Contains 11 video sequences which contains abandoned objects and were captured both in indoor and outdoor environment.	Simple, medium and complex	Yes	Yes	Considers all cases from less to thick crowd density.
CDNET2014	Contains videos depicting 11 scenarios and that includes abandoned objects as well	Simple, medium and complex	No	Yes	Considers all cases from less to thick crowd density.
CAVIAR	This dataset contains videos that cover various scenarios which left objects in public places`	Medium and complex	No	Yes	Medium density crowd observed.

2.3. Tools used

The most commonly used tools to implement vision based applications are OpenCV and Matlab. Each of these tools has its own advantages and limitations. Table 4 strikes a comparison of both these tools.

Table 4. Comparison of Matlab and OpenCV

Parameter compared	Matlab	OpenCV	Remarks
Toolboxes	Many	Less	Many toolboxes such as Computer Vision toolbox, image processing, neural network etc to support implementation are available.
Lines of Code	Less	More	Requires less lines of code when compared to OpenCV for implementing any logic.
Visualization	More	Less	Matlab provides more visualization to interpret and analyze the results.
Support documentation	More	Less	This is the major drawback in OpenCV. There is minimal support documentation which makes development using this tool complex for beginners.
Cost	High	Free	OpenCV is an open source tool and comes at no cost where as Matlab is licensed at pretty high cost.

III. PROPOSED METHOD

The proposed method uses a combination of blob analysis and frame subtraction to accurately determine abandoned or un-attended objects. There are two major contributions as part of this paper namely a faster approach to detect these abandoned objects and the person who has introduced the abandoned object inside the surveillance area and the other being an efficient algorithm to detect abandoned objects even under occlusion scenarios. Figure 4 shows the block diagram of the proposed method. The continuous feed from the surveillance camera is converted into frames and these are in turn subjected to preprocessing for noise removal. The Blob analysis segment involves identifying static blobs which are caused by objects that remain stationary over a period of time. The static blobs are segregated and the respective frame from when an object is identified as abandoned is indexed and recorded in a database. The indexing is done mainly to retrieve the data related to the abandoned object at a fast rate. The events that happen in the area under surveillance are continuously monitored. The occlusion handling module is responsible for tracking the abandoned objects even when an occlusion has occurred such that the abandoned object is not visible due to an obstruction. The occlusion handling module refers to the database to fetch the details of abandoned objects that were identified earlier and also updates the same with the recent detections. As soon as an abandoned object is detected, the alert module is triggered which raises an alarm.

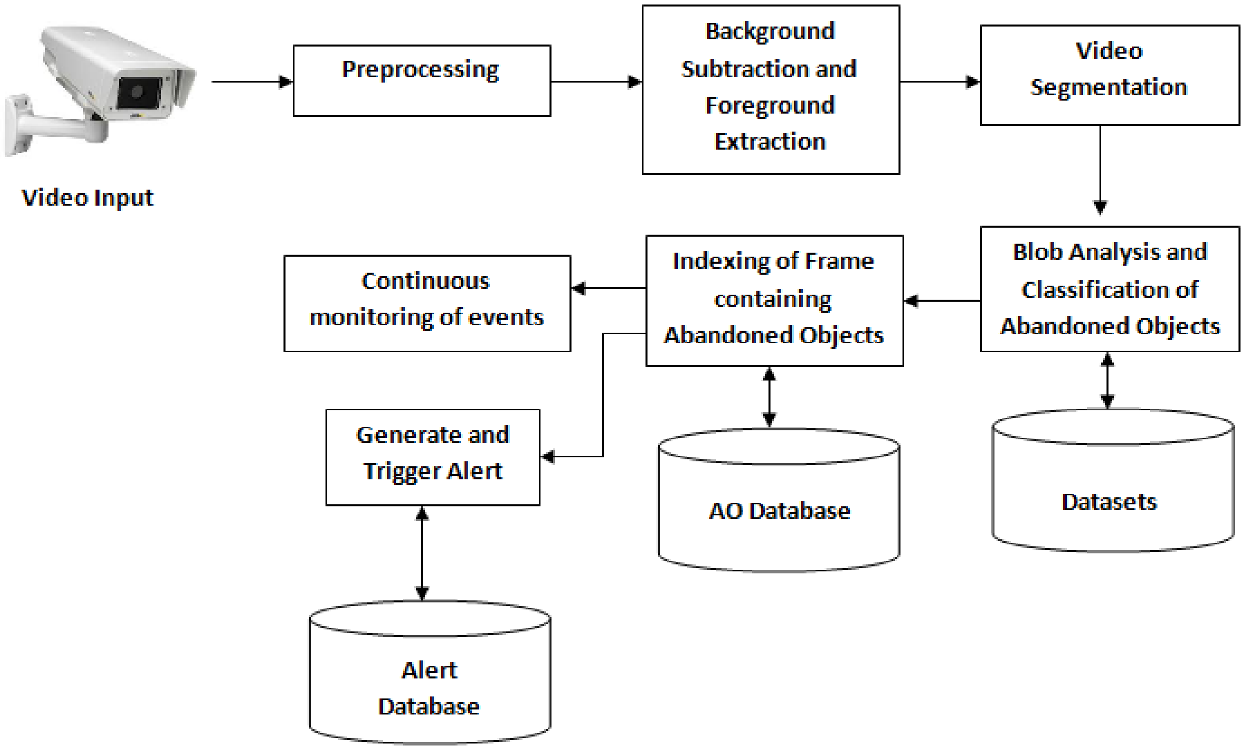

Figure 4. Block diagram of the proposed method

The flow diagram of the proposed method is shown in Figure 5. The input video is converted into frames. As part of the pre-processing stage, the input frames are converted from RGB to YcbCr format to provide contrast enhancement. The enhanced images are then subject to noise reduction where in they are passed through various filters such as Gaussian, median and linear filters to provide image smoothening. The pre-processed frames are then considered for back ground subtraction. In this work, Gaussian Mixture Model (GMM) is used to perform background subtraction. GMM assumes that the pixel intensity at any given instant is a combination of foreground and background processes and these processes can be modeled individually by a Gaussian Probability distribution function.

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta\left(X_t, \mu_{i,t}, \Sigma_{i,t}\right) \tag{1}$$

In equation (1), $P(X_t)$ provides the probability of observing the current intensity in the given frame. K denotes the number of distributions. In this work, the K value is set to 3. ωi, is the weight associated with the i^{th} iteration at a given time t while $\mu i,t$ and Σi,t are the mean and co-variance matrix of this distribution. η is the exponential Gaussian probability density function.

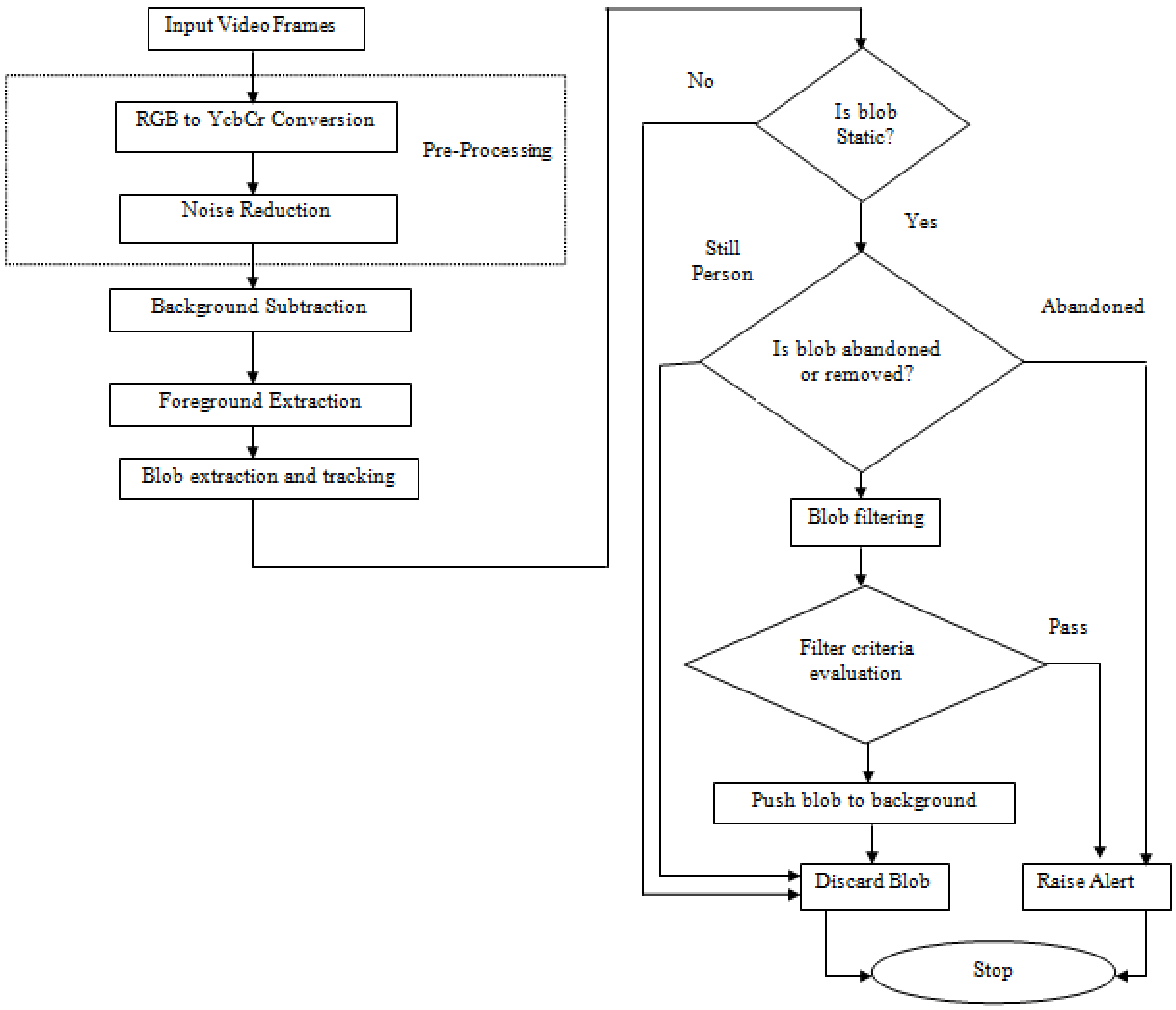

Figure 5. Flow diagram of the proposed method

The output of background subtraction may sometimes contain false foregrounds that may be created due to occlusions and lighting effect changes. So, the foreground extraction step is carried out to remove the noisy or unwanted foreground pixels that were created during background subtraction stage. The next step following the foreground analysis is blob extraction. The refined foreground mask obtained after the previous step is subjected to the proposed algorithm. This extracts meaningful and relevant blobs while discarding small blobs that are negligible by comparing the blob size with a standardized threshold value that is customizable according to the nature of the application. Using the customized functions available in the algorithm, the boundary box, centroid and area of each of the relevant blob is calculated. For the purpose of tracking these blobs, all the details pertaining to individual blobs are written in a matrix. By comparing the centroid position and size with the adjacent frames, the alarm is incremented. If the centroid position and size is unaltered, the alarm count is incremented which indicates that the object is stationary without any movement. In the case of occlusion, some other object overlaps the un-attended object thereby hiding the un-attended object. In such cases, the previous blob values are retained over a certain frame duration. If the occlusion is momentary, the abandoned object will get resurfaced in the subsequent frames. Thereby the blob comparison will indicate that the stationary object's presence and subsequently the alarm count is incremented. If the alarm count increases after a particular threshold value defined in the program, an alarm is triggered. If the stationary object has been removed by some person, another counter called miss count is increased. If the miss count for a blob reached a certain threshold, the blob is deleted from the list. The algorithm pertaining to the implementation of this approach has been discussed in the subsequent section.

IV. ALGORITHM FOR VISION BASED ABANDONED OBJECT DETECTION

Algorithm name: Vision based intelligent abandoned object detector (VIAOD)

Step 1: Receive the input video stream from surveillance camera.
Step 2: Convert the video into individual frames starting from i=1…..n where n denotes the last frame in the video sequence.
Step 3: Initialize the alarm counter (ac) and miss counter (mc) values. Alarm counter threshold (ac_{th}) is set to 50.
Step 4: Initiate a loop with starting index i= 1 and ends with index n

Step 5: Convert color contrast from RGB to YCbCr using vision.colorSpaceConvertor.
Step 6: Remove noise and smoothen the frame using median filter.
Step 7: Stop the loop when index i=n.
Step 8: Perform background subtraction using Gaussian Mixture Model which uses Gaussian probability distribution function listed in (1).
Step 9: Exclude the small and noisy blobs by using morphological operations. vision.MorphologicalClose('Neighborhood', strel('square',10));
Step 10: Find all the blob properties (Blob area, centroid and bounding box) pertaining to each blob in the image frame.
Step 11: Store the properties of each blob in a blob matrix.
Step 12: Inside the blob matrix, initiate another loop with index j = 1……n1
Step 13: Compare the centroids of each blob across adjacent frames. Check if the position of the blob has changed by comparing the boundary box of the blob, blob area and centroid present in the current frame with the background frame.
Step 14: If Step 12 returns true ac = ac+1 else mc = mc+1.
Step 15: Check if ac= ac_{th}.If true, classify as abandoned object and raise an alarm signal by making bounding box red and raising a signal.
Step 16: Store the abandoned object frame in a file and send an alert the through email.
Step 17: If the blob completely disappears, replace the blob in previous frame and retain for 20 frames to cover occlusion scenario. If the original blob resurfaces, occlusion is negated and the abandoned object still exists. Else the abandoned object has been removed from scene.
Step 18: Close the matrix loop when j=n1.
Step 19: Free up all the objects used.
Step 20: Wait for the next video sequence.

V. EXPERIMENTAL SETUP

The proposed algorithm was developed using Matlab2017b and the performance of this system was evaluated using Abandoned Object Database (ABODA) and another real time dataset. Initially, the experiment was evaluated by considering the ABODA dataset. The same test was also performed over another set of videos which were captured in indoor and outdoor conditions under un-controlled environment. Table 5 lists the various parametric values and threshold values that was set as part of the experiment.

Table 5. Experimental parameters

Parameter	Value
Alarm counter threshold	50 frames
Occlusion handling counter	20 frames
Maximum number of objects that can be handled in a frame	250 objects
Volume of video sequence in ABODA	11 videos
Volume of video sequence in real time dataset	11 videos

The alarm counter threshold value was set to 50. This meant that if an un-attended object remains in a place for 50 continuous frames, it will be marked as an abandoned object and a red bounding box is drawn around the object and an immediate alert sound will be produced. Also the frame image which contains the abandoned object will be sent as an email alert to the configured email address. The performance of this approach has been gauged and reported in terms of sensitivity and misclassification rate.

VI. RESULT DISCUSSION

As discussed in the previous section, the proposed algorithm was tested initially with ABODA dataset. Figure 6 shows the images which depict the sample output of the approach. It can be seen that the object that is left back un-attended is bounded in red and highlighted.

Figure 6. Sample output sequence from ABODA dataset highlighting an abandoned bag

The proposed approach was also tested using a real time dataset which was specifically created to test the working of this approach. As part of creating this dataset, several aspects such as indoor and outdoor environments, illumination effects and lighting changes have been covered and 11 video sequences were grouped to for this real time data set. Figure 7 shows the sample output of how the abandoned objects have been identified in real time dataset. The performance of this proposed approach has been measured in terms of sensitivity and misclassification rate which are computed as shown in (2) and (3) respectively. Table 6 and Table 7 provide the performance metric details pertaining to ABODA and real time dataset respectively.

Sensitivity = True positive / (True positive + False negative) (2)

Misclassification rate = 1 – True positive rate (3)

Figure 7. Sample output sequence from real time dataset highlighting an abandoned bag and backpack

Table 6. Results for ABODA dataset

Video Sequence	Conditions	True Positive (TP)	False Positive (FP)	False Negative (FN)
Video 1	Indoor	1	0	0
Video 2	Outdoor	1	0	0
Video 3	Outdoor	2	0	0

Table 7. Results for Real time dataset

Video Sequence	Conditions	True Positive	False Positive	False Negative
Video 1	Outdoor	1	0	0
Video 2	Outdoor	2	0	0
Video 3	Outdoor	3	0	0

Retrieval Number: A1095109119/2019©BEIESP
DOI: 10.35940/ijeat.A1095.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Video 4	Outdoor	1	0	0
Video 5	Night	1	0	0
Video 6	Lights switched on	2	1	0
Video 7	Lights switched on	1	0	0
Video 8	Lights switched on	1	0	1
Video 9	Indoor	1	0	0
Video 10	Indoor	1	0	0
Video 11	Crowded	2	1	0
Total		14	2	1

Video 4	Outdoor	1	0	0
Video 5	Outdoor	3	0	0
Video 6	Outdoor	2	1	0
Video 7	Indoor	1	0	0
Video 8	Indoor with lights	1	0	0
Video 9	Indoor with lights	3	1	1
Video 10	Indoor	1	0	0
Video 11	Indoor, crowd	4	1	0
Total		22	3	1

It can be seen from Table 6 that the sensitivity of the proposed approach when tested with ABODA dataset is computed as below.

$$Sensitivity_{ABODA} = TP/(TP+FN) = 14/15 = 0.934 \quad (4)$$

$$Misclassification\ rate_{ABODA} = (1-TPR) = 0.066 \quad (5)$$

From table 7, the sensitivity of the proposed approach when tested with the real time dataset is shown below.

$$Sensitivity_{RT} = TP/(TP+FN) = 22/23 = 0.957 \quad (6)$$

$$Misclassification\ rate_{RT} = (1-TPR) = 0.043 \quad (7)$$

Figure 8 shows the comparison of sensitivity and misclassification rate for ABODA dataset while figure 9 shows the comparison of sensitivity and misclassification rate for the real time dataset.

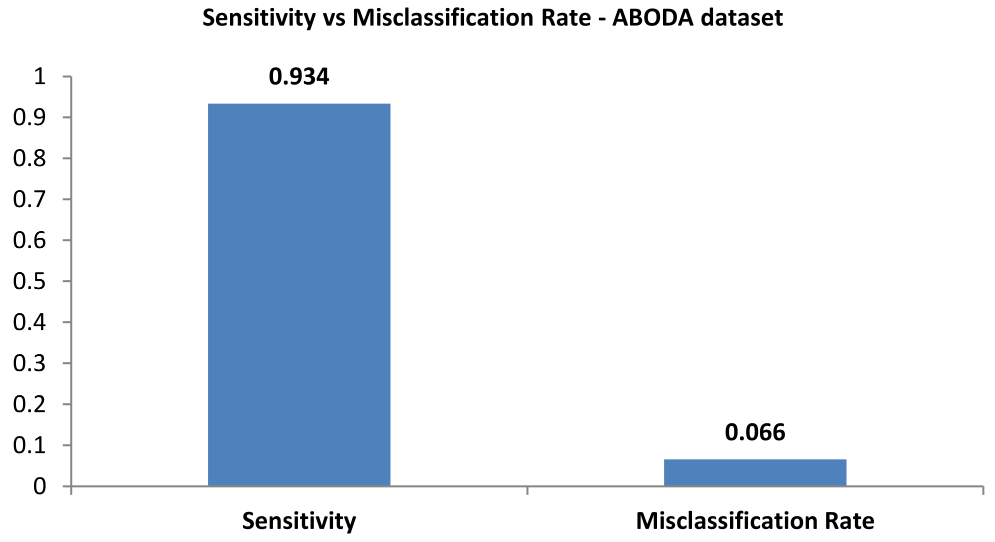

Figure 8. Sensitivity Vs Misclassification rate for ABODA dataset

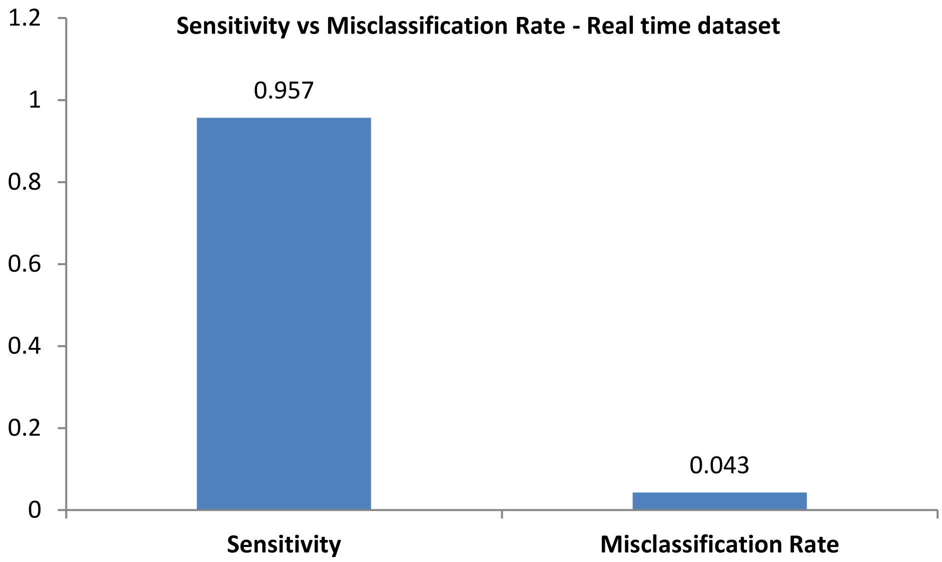

Figure 9. Sensitivity Vs Misclassification rate for Real time dataset

An approach can be considered to be performing well if the sensitivity is high and the misclassification rate is very minimal. From the computations performed in (4) and (6), it can be seen that the sensitivity values for ABODA dataset and real time data set are 0.934 and 0.957 respectively which is very high when compared to the misclassification rate values computed in (5) and (7). Hence it can be said that the proposed approach is robust and accurate in classifying abandoned objects. As part of the experiment, the processing time of the proposed approach was compared with other contemporary methods. It was observed that the processing speed of the proposed approach for a 320x240 resolution was 112 frames per second which was a better when compared to the other methods. The processing speeds of various approaches compared as part of this experiment is tabulated in table 8 and is depicted in figure 10.

Table 8. Comparison of processing time of different methods

Method	Processing speed in Frames per second
Proposed method	112 Fps
[11]	108 Fps
[12]	51 Fps
[13]	32 Fps

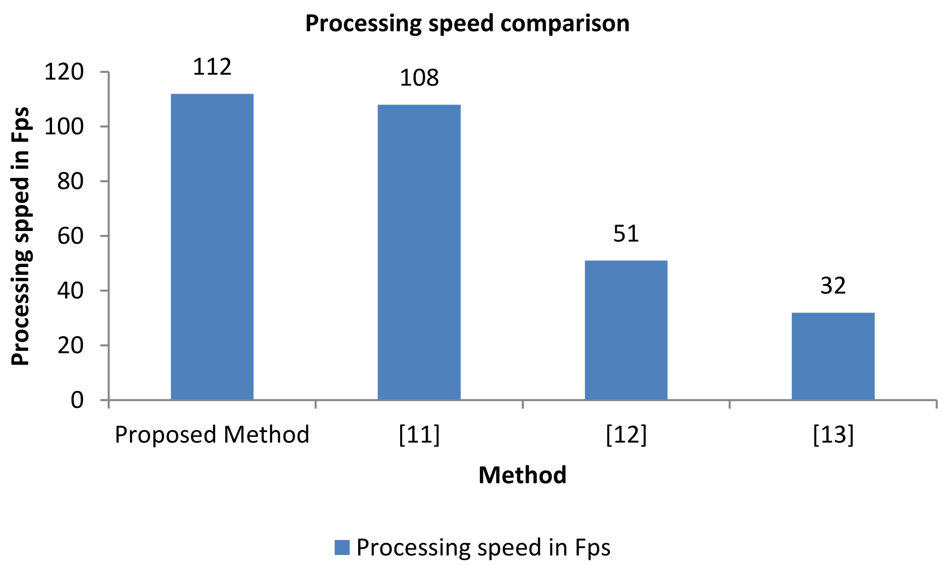

Figure 10. Processing speed comparison of various methods.

VII. CONCLUSION AND FUTURE WORK

In this paper, a new approach based on blob analysis has been proposed to detect abandoned objects with good accuracy and faster processing time. The proposed algorithm has been tested with a standard database namely ABODA and has also been tested with a contemporary real time data set that was created specifically to test abandoned objects. The experimental results indicate high sensitivity to detecting abandoned objects across ABODA and real time datasets. Also, the misclassification rate was very minimal. The processing speed of the proposed approach was compared with other existing methods and it was observed that the processing speed of the proposed algorithm was slightly better than existing methods. This approach has also been able to address the tracking of abandoned objects even during occlusion scenarios where the abandoned objects become invisible due to other object overlapping them. Future work will be in the direction to integrating this algorithm into a sophisticated surveillance system that can detect anomalies as well along with detecting abandoned objects.

ACKNOWLEDGMENT

The authors wish to express their thanks the people involved in creating the ABODA dataset which has been used as part of this experiment in detecting abandoned objects.

REFERENCES

1. Y. Tian, R. S. Feris, H. Liu, A. Hampapur, M. T. Sun. Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2011; 41: 565-576.
2. Ilias Dahi, Miloud Chikr El Mezouar, Nasreddine Taleb, Mohamed Elbahri. An edge-based method for effective abandoned luggage detection in complex surveillance videos. Computer Vision and Image Understanding 2017; 158: 141-151.
3. K. Lin, S. C. Chen, C. S. Chen, D. T. Lin, Y. P. Hung. Abandoned Object Detection via Temporal Consistency Modeling and Back-Tracing Verification for Visual Surveillance. IEEE Transactions on Information Forensics and Security. 2015: 10: 1359-1370.
4. C. Cuevas, R. Martínez, D. Berjón, N. García. Detection of Stationary Foreground Objects Using Multiple Nonparametric Background-Foreground Models on a Finite State Machine. IEEE Transactions on Image Processing. 2017; 26: 1127-1142.
5. L. Foresti, L. Marcenaro, C. S. Regazzoni Automatic detection and indexing of video-event shots for surveillance applications. IEEE Transactions on Multimedia. 2002: 4: 459-471.
6. Fatih Porikli, Yuri Ivanov, Tetsuji Haga. Robust Abandoned Object Detection Using Dual Foregrounds. EURASIP J. Adv. Signal Process. 2008; 2008:1-11.
7. Jing-Ying Chang, Huei-Hung Liao, Liang-Gee Chen. Localized Detection of Abandoned Luggage. EURASIP J. Adv. Signal Process. 2010; 2010: 1-9.
8. Alex Pereira, Osamu Saotome, Daniel Sampaio. Patch-based local histograms and contour estimation for static foreground classification. EURASIP Journal on Image and Video Processing. 2015; 39: 1-11.
9. Ruben Heras Evangelio,Thomas Sikora. Static Object Detection Based on a Dual BackgroundModel and a Finite-StateMachine. EURASIP Journal on Image and Video Processing. 2011; 2011: 1-9.
10. Chih-Yang Lin, Kahlil Muchtar, Chia-Hung Yeh, Robust techniques for abandoned and removed object detection based on Markov random field. Journal of Visual Communication and Image Representation 2016; 39: 181-195.
11. Ilias Dahi1, Miloud Chikr El-Mezouar, Nasredddine Taleb, Mohamed Elbahri. An edge-based method for effective abandoned luggage detection in complex surveillance videos. 2017; 158: 141-151
12. G. Szwoch. Extraction of stable foreground image regions for unattended luggage detection. Multimedia Tools and Applications. 2016; 75: 761-786.
13. K. Lin, S.C. Chen, C.S. Chen, D.T. Lin, Y.P. Hung. Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance. IEEE Transactions on Information Forensics and Security. 2015; 10:1359–1370.
14. A. Balasundaram and C. Chellappan, "Vision Based Motion Tracking in Real Time Videos," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, 2017, pp. 1-4.
15. Balasundaram, A. & Chellappan, C. J Real-Time Image Proc (2018). https://doi.org/10.1007/s11554-018-0840-6