MAK HILL Publications Journal of Engineering and Applied Sciences

Research Article
doi: jeas.2022.44.51

Iot Surveillance System Based on Moving Average Subtraction Technique

Oluyemi Adetoyi and Oladayo Olakanmi
University of Ibadan, Ibadan, 200282, Nigeria

ABSTRACT
Most algorithms for computer vision-based surveillance systems involve high computation overload, making them unsuitable for real-time application. Also, some of these algorithms cannot handle abrupt illumination changes in the image without causing estimation errors. This work employs a background subtraction algorithm model, which is robust against illumination variations and less computation intensive. The design consists of a robot surveillance agent and a control center. The robot was equipped with a camera and Raspberry-pi board to continually capture environment scenes. The video frames were pre-processed to enhance image quality and a moving average algorithm was applied to detect motion. The frames containing motion are stored locally and also transmitted via the internet to a web application where further analysis can take place and necessary action can be taken. The web application was built on the flask framework, using Python language. The control center for monitoring the robot was a personal computer but could be a mobile device too. The implemented algorithm was able to detect small movements and it can continuously track an object's motion. The algorithm run time is less than the Gaussian mixture models' algorithm. Analysis of video frames to detect motion enables savings in storage and transmission requirements to be made.

OPEN ACCESS

Key Words
Video surveillance, IOT, computer vision, object tracking, remote monitoring, running average, gaussian kernel

Corresponding Author
Oluyemi Adetoyi
University of Ibadan, Ibadan, 200282, Nigeria

Received: 14 September, 2022
Accepted: 20 September, 2022
Published: 23 September, 2022

Citation: Oluyemi Adetoyi and Oladayo Olakanmi, Iot Surveillance System Based on Moving Average Subtraction Technique. J. Eng. Applied Sci., 17: 44-51, 2022 doi: jeas.2022.44.51

INTRODUCTION

Since the advent of Closed Circuit Television (CCTV) there has been tremendous development in video surveillance to enhance performance issues[1]. Surveillance systems are no longer restricted to the fixed camera but can rotate about a fixed position or are part of the onboard system of robotic devices. The three widely used camera technologies are CCD, thermal and night vision devices. The use of networked multi-camera system in dynamic scenes are becoming increasingly popular in an attempt to detect, recognize and track certain objects from image sequences; but more importantly to overcome single-camera ambiguity problems[2-4]. Unlike traditional CCTV that does not produce any notification and warning to a user whenever it captures any suspicious object, many surveillance systems today are capable of sending SMS messages to a mobile or streaming video to a website in the occurrence of an event[5,6]. The streaming of video in the event of any suspicious object or activity in the environment is made possible through advancements in video compression and transmission techniques. This results in considerable savings in bandwidth, storage media and overall cost. Artificial intelligence has now found usefulness in video analysis to understand and describe object behaviors[7-9]. The lowest level task in video surveillance analysis is motion detection, which aims at segmenting areas corresponding to moving objects from the rest of the image. Subsequent processes, such as object tracking, behavior analysis and recognition are highly dependent on it[10]. Some of the algorithms developed to solve detection problems in visual surveillance systems can be found in 2019[5,11-17]. These algorithms can be classified into background subtraction, spatio-temporal filter and optimal flow method of motion detection[4,18-20]. However, few of these algorithms have been implemented as visual surveillance systems. More so, the accuracy of such implemented algorithms is often low, while high complexity makes them unsuitable for real time applications. This work presents a surveillance system, which employs low computational and efficient algorithm to determine background image from corresponding foreground image in video frames. The detected subject’s motion is stored locally and also transmitted to a web application for remote monitoring and control. The paper is arranged as follows. Related works presents similar works in video surveillance field and provides justification for the work. The system design was presented next, which highlights the developmental process. Test results were presented and discussed and finally the concluding remarks about the work.

Related works: An optical flow approach of estimating motion vectors from images was presented by Hernandez *et al.*[21]. The images are divided into non-overlapping macroblocks of 16×16 pixels and an algorithm looks for the region that closely matched the macroblock under analysis in the next image frame. The solution of the search algorithm is a distance and direction that minimized Mean Absolute Error (MAE) of reference frame macroblock and actual frame macroblock. A three-level hierarchy estimation was then applied to reduce the computational complexity. At decomposition level 2, the image size is 1/16 of the original image, while image size is 1/4 of original image at decomposition level 1. The motion vector estimation is then carried out at decomposition level 0 with macroblocks of size 16×16 with search range [-1,1] around the origin pixel. The obtained motion vectors consist of horizontal and vertical components, which are used to estimate the motion vector magnitude and angle. A significant motion is deduced if the motion vector angle is between 200 and 340 degrees relative the camera position. The disadvantages of this method are that accurate estimation is dependent on camera position and absence of camera vibration. Also, abrupt illumination changes in the image can cause wrong estimation, which necessitate the use of a median filter.

Puttegowda and Padma *et al*. detected motion by taking absolute difference of two consecutive images to model the background[22]. Then for each image point of the difference image, the standard deviation and mean were computed; the threshold was obtained as sum of the mean and twice standard deviation. Motion is detected if the absolute difference is greater than the threshold, otherwise there is no motion. Tracking was accomplished by using spatio-temporal interest points to select object of interest. An edge tracking algorithm was then used to extract features from the selected object of interest. These features are used to train K-Nearest Neighbors algorithm to classify the detected human body motion. However, the method of background estimation is computational intensive, since it has to be computed on two consecutive frames repeated, which may not be suitable for real time application. More so, it is not robust to handle light variations and noise.

A visual surveillance system based on adaptive background modeling was presented by Zaman *et al.*[23]. A threshold of 25 was applied to the preprocessed image to convert it to black and white image. Filter was applied to remove noise from the resulting binary image. The number of white pixels in the image that correspond to motion level was counted using histogram technique. Motion is considered to have taken place if the counted white pixels is greater than 10. In the implementation of the adaptive background, the background frame was moved slightly in the direction of the current frame. The pixel intensities in the background frame are changed by one level per frame so that the background is slowly adapted to the current image. While the result is considered better than simple subtraction technique, the computation overhead involved is quite high, which makes it unsuitable for real time application.

Another adaptive background modelling approach was presented by Al-Asadi and Joda[24]. The proposed algorithm has the ability to update threshold when illumination changes and removal of moving cast shadow is possible. The foreground was detected by thresholding the difference between current frame and the background reference. Initial background was obtained by computing the median of N frames and this was updated frequently from the current background, current frame and foreground mask. The weighted averaging of the color components (H, S, V) correlations was then obtained; if this is greater than 0.7, the pixel is classified as shadow, otherwise it is foreground. However, the inclusion of foreground mask in the computation of the background reference leads to increase in background update's time. The correlation computation to discriminate between the object and the shadow also elongate the detection time and affects the real-time applicability. An improvement over this method based on multi-modal background modelling was proposed by Al-Asadi and Joda[25]. The background was constructed from histogram of Average Centre Symmetric Local Binary Pattern (ACS-LBP) and histogram of hue channel for each pixel. The similarity measure of a new frame and the background was obtained by computing the weighted average similarity of the hue and ACS-LBP. The foreground is detected if the overall similarity is less than the threshold value. Although the method was efficient compared with the benchmark methods, the computation complexity is high.

The Channel State Information (CSI) acquired at the receiver from a radio transmitter was used to estimate human motion by Liu *et al.*[26]. The channel was assumed relatively stable when no motion occurs, but motion will generate scattered signals with significant channel distortions in amplitude attenuation and phase shift. Human motion can then be recognized by building a map between the pattern of CSI signal variation and human motion. However, multipath effect due to reflection from walls, ceiling, floor and obstacles, changes with time and affects the CSI. In addition, the human motion CSI pattern is affected by whether the human is within the transmitter and receiver Line of Sight (LOS) or not.

Motion was detected in the surveillance system by Kumbhar and Bhaskar[20] and Huang *et al.*[27] by the frames' gray level relational method. The objective is to detect moving objects by comparing the extracted gray level feature of two consecutive frames. The extracted gray levels are optimized through clustering technique and the Euclidean distance between the two gray levels are estimated. Motion is said to be detected if the estimated distance is greater than the selected threshold. The comparison was based on the principle that similar background intensity values have a higher gray relational rate, whereas dissimilar background will have low gray relational rate. Though the simulated result shows an increase in the level of detection with increased motion, the choice of threshold can significantly alter the result and high level of computation involved.

Previous works on motion detection, which are based on CSI are thus unreliable, since CSI pattern is greatly influenced by reflection from background objects and whether the target person is in the LOS or NLOS condition. Optical flow detection approach is unreliable too under abrupt illumination changes, unless special filters are incorporated. The major drawback of existing background modelling approach is high computation overhead, which makes them unsuitable for real time application. In this work, simple moving average was used to model the background; this requires less computation and reduced memory resource allocation, since it is recursively updated. Hence, makes it suitable for real time application.

System design: The design stages of the wireless surveillance and monitoring system is as shown in Fig. 1. The servo motors acts on the signal from Raspberry-pi to control the camera view and snapshots of the environment were taken periodically. The captured images are processed by the Raspberry-pi according to the stored algorithm. In the event of detected motion, such frames are saved locally and streamed to a website.

Image capturing: The camera for capturing images is an onboard camera module of Raspberry-pi 4. The camera module is mounted on a pan and tilt support provided by two servo motors to allow for precise control of camera angular position. The pulse width modulated signal from the Raspberry-pi controls the servo motors, which in turn controls the vertical and horizontal movement of the camera. The camera resolution was set to a low value of 320×240 to prevent storage overflow. The camera capture mode is continuous at a rate of 16 frames per second to allow for image processing.

Image processing: A good image quality video file "atrium.avi" comprising of 1,140 image frames of size 360×640 was used in the MATLAB environment for the algorithm simulation. The video was read frame by frame and the frames were converted to grayscale to reduce the effect of lighting condition variations. The intensity image is then converted to double precision

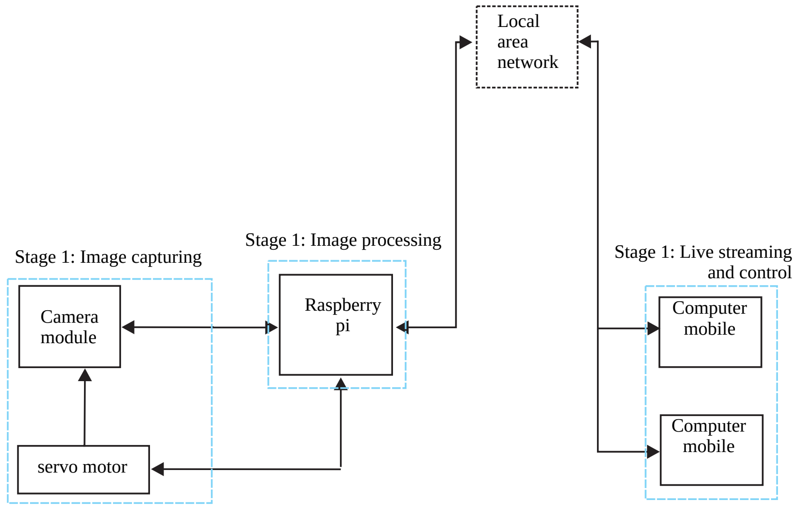

Fig. 1: System design block diagram

format to represents a wide dynamic range of numeric values. To deploy on the Raspberry-pi, the captured videos from the Raspberry-pi camera are resized to 224×224, to reduce storage requirements. The images were converted to grayscale and passed to the Gaussian deblur function to remove noise, thereby enhancing the image. The Gaussian deblur feature act as a nonuniform low-pass filter that preserves low spatial frequency and crops out uninformative areas from the image. The filtering was achieved by convolving images with a Gaussian kernel. This Gaussian kernel in 2-D form is expressed as:

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

where, σ is the standard deviation of the distribution, ×and y are the location indices of the image pixel. The value of σ controls the variance around a mean value of the Gaussian distribution, which determines the extent of the blurring effect around the pixel. In the proposed image segmentation, a sigma value of 3.5 was used to satisfy the condition of 6σ-1 values for Gaussian kernel size. This was followed by dilation, to restore object size that has shrunk during noise removal. The image processing steps for Raspberry-pi deployment are as follows:

- Resize the image frame to 224×224
- Convert the RGB frame to grayscale
- Deblur the frame by convolving images with a Gaussian kernel

Motion detection and tracking: For the detection of motion between frames in the surveillance camera output, a background subtraction technique was used to separate foreground elements from the background, by generating a foreground mask. The form of background modeling technique that was used is the moving average. The video sequence was analyzed for motion detection at a rate of 16 frames per second, during which the moving average is recursively computed as:

$$B_t(x, y) = (1-\alpha)\ B_{t-1}(x, y)+\alpha(P_t(x, y)) \quad (2)$$

where $B_t(x,y)$ is the background image pixel value at time t, $B_{t-1}(x,y)$ is the background image pixel value at time t-1, $P_t(x,y)$ is the pixel value of current frame at time t and is the averaging parameter that determines the major determinant of current background out of previous background and current frame.

The background model average is initialised to zero and any new object introduced during the sequencing of the video becomes part of the foreground. Thus, the current frame holds the newly introduced object with the background. Then the computation of the absolute difference between the background model $B_t(x,y)$ the current frame $P_{t+1}(x,y)$ is a measure of motion at time t+1 and is given as:

$$\Delta = |P_{t+1}(x,y)-B_t(x,y)| \quad (3)$$

A global threshold value of 0.1 was applied to convert the delta frame to binary. The binarized image was dilated to fill in the holes. The delta computation was identically zero if there was no significant motion in the current frame compared to the last updated background, otherwise motion is detected and flagged with a bounding box. The frames in which motion is detected are stored as images on the Raspberry-pi storage and later upload to Dropbox. The algorithm for the detection process is as follows:

- Initialise average to starting background frame
- Input next frame and compute the moving

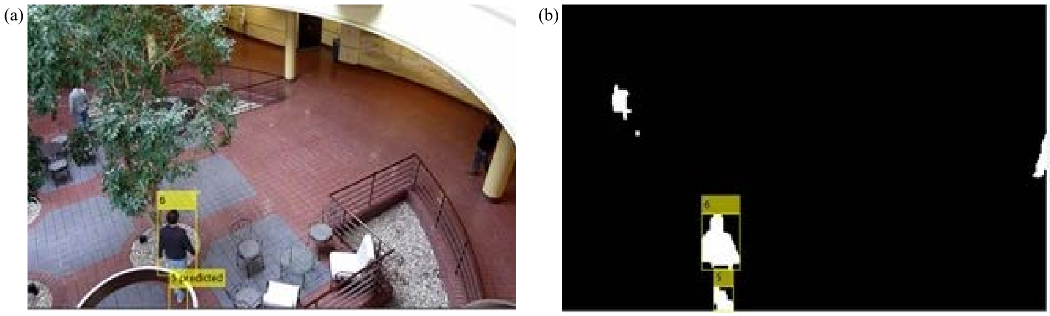

Fig. 2(a-b): (a) Two subjects out of three are obscured by background and (b) Only one subject detected

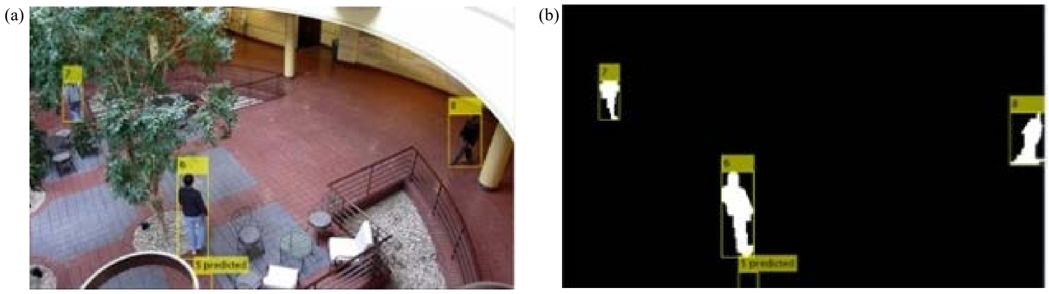

Fig. 3(a-b): (a) Obscured subjects are now visible and (b) All three subjects detected

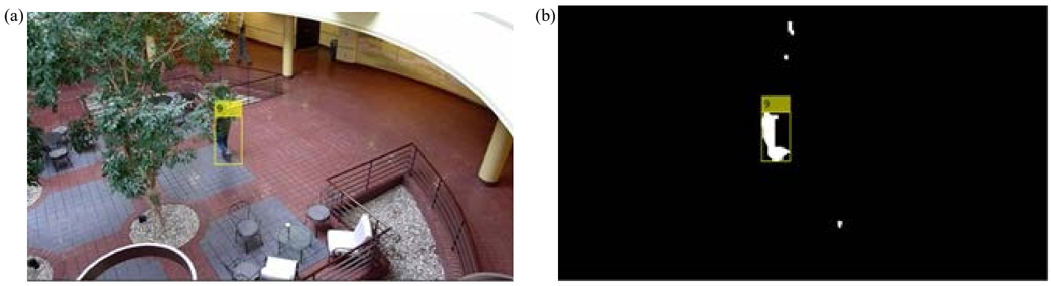

Fig. 4(a-b): (a) One moving subject out of two obscured by background (b) Only one subject detected

average between the current frame and previous frames as given by Eq. 2

- Compute the difference between the current frame and accumulated moving average of step 2 according to Eq. 3
- Apply 0.1 global threshold to the difference image in step 3
- Dilate the threshold image to fill in holes
- Perform Blob analysis on the binary image to determine Centroid and bounding box×coordinates
- Include timestamp and detection text on the frame
- If detection occurs, check if elapsed time after last upload Is equal or greater than set minimum upload time
- Check if the number of frames with consistent motion is Equal or greater than set minimum motion frames
- Upload the image to Dropbox×and display on the security feed if steps 9 and 10 are true, if not write the image to temporary file
- Update the last upload timestamp and reset the motion counter
- Cleanup the temporary file in preparation for the next captured frames

Multi-object tracking of the detected object was accomplished by configuring a Kalman filter object for each physical object. The Kalman filter algorithm involves two steps, prediction and correction (update). In the first step, previous states were used to predict the current state, while the second step uses the current measurement of object location to correct the state. The Kalman filter implements a discrete time, linear State-Space System. However, the object (s) must be moving at constant velocity or constant acceleration in order to be trackable.

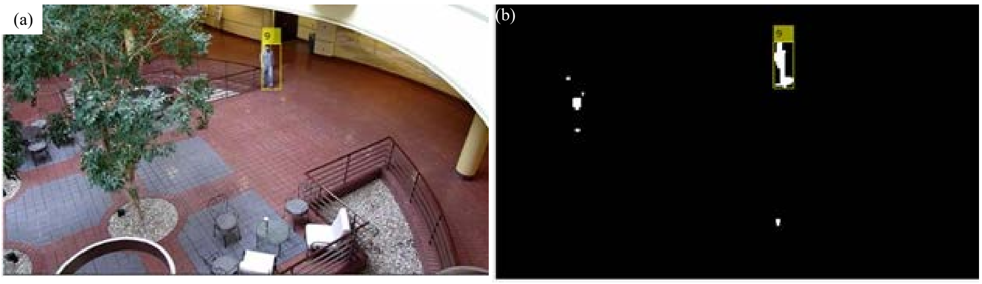

Fig. 5(a-b): (a) Obscured subject now visible, initial visible subject obscured (b) Two subjects detected as one

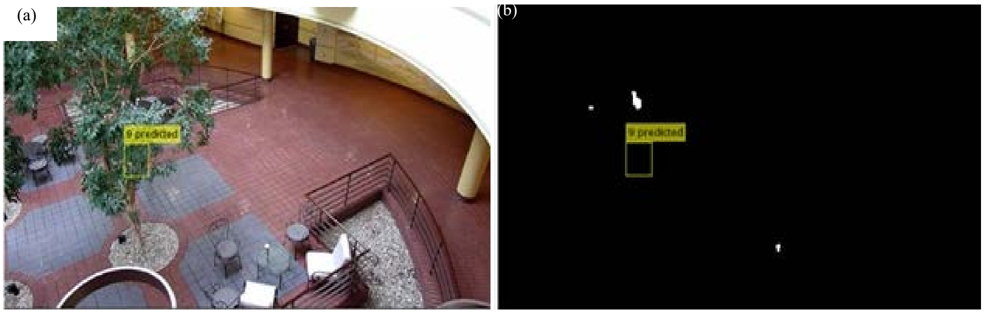

Fig. 6(a-b): (a) Two out of two subjects obscured by background (b) Only one subject predicted

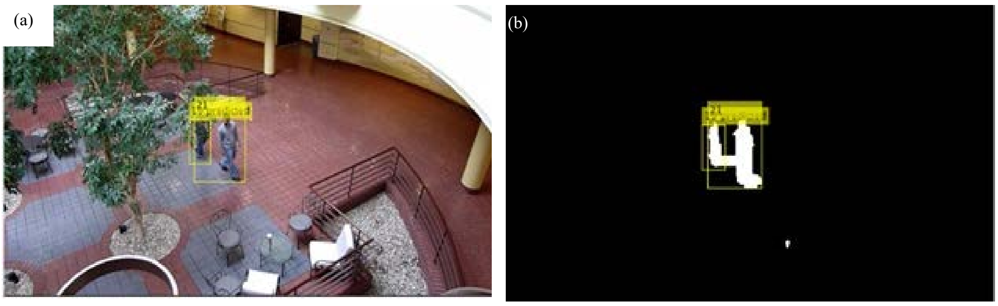

Fig. 7(a-b): (a) Two obscured subjects became visible (b) Two subjects detected

Fig. 8(a-b): Subject tracking by the deployed Raspberry-pi surveillance system

Live streaming and control: This consists of the user's computer and mobile devices moving the web application and connected to the same local area network (LAN) as the Raspberry-pi video surveillance. It is the output stage, where images of the surveillance system are viewed and monitored by the user. The

Fig. 9: Web interface for streaming videos

user is also able to take control of actions on the surveillance system. The web application was built on the Flask framework using Python language to live stream the output of the surveillance camera. Flask provides native support for streaming responses through the use of generator functions that can be interrupted and resumed. The Flask, streams video frames captured from the camera in Motion JPEG format (MJPEG). The framework ensures the web application reliability, scalability and maintainability by providing reusable code or extensions for common operations.

RESULTS AND DISCUSSION

The performance of the moving average algorithm on atrium.avi video as evaluated in the MATLAB environment is as presented in Figures. Subjects 7 and 8 are not detected in Fig. 2, because they are stationary and are thus seen as part of the background, but their motion are detected in Fig. 3. In Fig. 4, a subject is not detected because the motion is obscured by the terrace, the subject is now detected in Fig. 5 but wrongly identified as subject 9 who just disappeared among the trees. The original subject 9 and wrongly identified subject 9 are both hidden among trees in Fig. 6 but only one subject is predicted to be there. When they both emerges from the trees in Fig. 7, they are seen as new subjects 19 and 21 with overlapping motion area. Although the system is quite sensitive to motion but if the motion is obscured by the background, there will be missed detection similar to that of subject 8 in Fig. 2. The tracking was solely based on the assumption that all objects move in a straight line with constant speed. When the motion of an object significantly deviates from this model, tracking errors may result as was the case with subject 9 in Fig. 4-7. The total simulation time for the moving average algorithm is 68.055s while the simulation time for the same atrium.avi video using background subtraction algorithm based on Gaussian mixture models is 71.347s. The tracking of a subject by Raspberry-pi implementation of the surveillance system is shown in Fig. 8. The web interface to view video streaming is shown in Fig. 9. The control of the camera viewing area is also possible via this interface, which allows for system flexibility.

ACKNOWLEDGMENT

The author wishes to acknowledge Damilola Oluwarore who carried out the experimental implementation.The author(s) received no specific funding for this study.The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCE

1. Chamasemani, F.F. and L.S. Affendey, 2013. Systematic review and classification on video surveillance systems. Inf. Technol. Comput. Sci., 7: 87-102.
2. Banu, V.C., I.M. Costea, F.C. Nemtanu and I. Badescu, 2017. Intelligent video surveillance system. Proceding of the 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), October 26-29, 2017, IEEE, Constanta, Romania, pp: 208-212.
3. Wang, X., 2013. Intelligent multi-camera video surveillance: A review. Pattern Recogn Lett., 34: 3-19.
4. Ko, T., 2011. A Survey on Behavior Analysis in Video Surveillance Applications. In: Video Surveillance. Lin, W. (Ed.)., IntechOpen, London, UK, pp: 1-18.

5. Elhoseny, M., 2019. Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. Circuits. Syst. Signal. Process., 39: 611-630.
6. Kim, J.S., D.H. Yeom and Y.H. Joo, 2011. Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems. IEEE Trans. Consum. Electron., 57: 1165-1170.
7. KUNCAN, F., Y. KAYA and M. KUNCAN, 2019. A novel approach for activity recognition with down-sampling 1D local binary pattern features. Adv. Electr. Comp. Eng., 19: 35-44.
8. Shidik, G.F., E. Noersasongko, A. Nugraha, P.N. Andono and J. Jumanto *et al*., 2019. A systematic review of intelligence video surveillance: Trends, techniques, frameworks and data sets., IEEE Access, 7: 170457-170473.
9. Konwar, L., A.K. Talukdar and K.K. Sarma, 2021. Robust real time multiple human detection and tracking for automatic visual surveillance system. Wseas Trans. Signal Process., 17: 93-98.
10. Suliman, C., C. Cruceru and F. Moldoveanu, 2010. Kalman filter based tracking in an video surveillance system. Adv. Electr. Comput. Eng., 10: 30-34.
11. Gajjar, V., Y. Khandhediya and A. Gurnani, 2017. Human detection and tracking for video surveillance: A cognitive science approach. Proceeding of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW) October 22-29, 2017, IEEE, Venice, Italy, pp: 2805-2809.
12. Shehzed, A., A. Jalal and K. Kim, 2019. Multi-person tracking in smart surveillance system for crowd counting and normal/abnormal events detection. Proceeding of the 2019 International Conference on Applied and Engineering Mathematics (ICAEM), August 27-29, 2019, IEEE, Taxila, Pakistan, pp: 163-168.
13. Huang, H. and A.V. Savkin, 2019. Reactive 3D deployment of a flying robotic network for surveillance of mobile targets. Comput. Netw., 161: 172-182.
14. Lin, J., Y. Yumei, Z. Maosheng, C. Defeng and W. Chao *et al*., 2020. A multiscale chaotic feature extraction method for speaker recognition. Complexity, 2020: 1-9.
15. Sreenu, G. and M.A.S. Durai, 2019. Intelligent video surveillance: A review through deep learning techniques for crowd analysis. J. Big Data, Vol. 6, No. 48, 10.1186/s40537-019-0212-5
16. Febin, I.P., K. Jayasree and P.T. Joy, 2019. Violence detection in videos for an intelligent surveillance system using MoBSIFT and movement filtering algorithm. Pattern. Anal. Applic., 23: 611-623.
17. Mrabti, W., K. Baibai, B. Bellach, R. O. Haj Thami and H. Tairi *et al*., 2019. Human motion tracking: A comparative study. Procedia Comput. Sci., 148: 145-153.
18. Paul, M., S.M Haque and S. Chakraborty, 2013. Human detection in surveillance videos and its applications-a review. EURASIP. J. Adv. Signal Process., 2013: 1-16.
19. Sehairi, K., F. Chouireb and J. Meunier, 2017. Comparative study of motion detection methods for video surveillance systems. J. Electron. Imaging., 26: 22967-023025.
20. Kumbhar, A. and P.C. Bhaskar, 2018. Motion Detection for Video Surveillance System. In: Advances in Computing and Data Sciences. Singh, M., Gupta, P. Tyagi, V. Flusser and J.T. Oren, (Eds.)., Springer, Singapore, pp: 523-534.
21. Hernandez, J., H. Morita, M. Nakano-Miytake and H. Perez-Meana, 2009. Movement detection and tracking using video frames. Progress in Pattern Recognition, Image Analysis, Computer Vision and Applications (CIARP 2009), November 15-18, 2009, Springer, Berlin, Heidelberg, 1054-1061.
22. Puttegowda, D. and M.C. Padma, 2016. Human motion detection and recognising their actions from the video streams. Proceedings of the International Conference on Informatics and Analytics (Year 2016, ICIA), August 25, 2016, ACM, New York, pp: 1-5.
23. Zaman, F.H.K., M.H. Ali, A.A. Shafie and Z.I. Rizman, 2017. Efficient human motion detection with adaptive background for vision-based security system. Intl. J. Adv. Sci. Eng. Inf. Technol., 7: 1026-1031.
24. Al Asadi, T.A. and F.A. Joda, 2017. Adaptive background modelling and shadow removal for video surveillance system. J. Eng. Appl. Sci., 12: 7689-7695.
25. Al-Asadi, T.A. and F.A. Joda, 2018. Object detection based on fusion of multi-feature for video surveillance system. J. Eng. Appl. Sci., 13: 8159-8166.
26. Liu, J., L. Wang, J. Fang, L. Guo and B. Lu *et al*., 2018. Multi-target intense human motion analysis and detection using channel state information. Sensors, 18: 1-3379.
27. Huang, S.C., H. Liu, B.H. Chen, Z. Fang and T.H. Tan *et al*., 2019. A gray relational analysis-based motion detection algorithm for real-world surveillance sensor deployment. IEEE Sensors J., 19: 1019-1027.