International Journal of Future Generation Communication and Networking
Vol. 11, No. 1 (2018), pp.37-46
http://dx.doi.org/10.14257/ijfgcn.2018.11.1.04

Developing a Robust Framework to Reduce the Size of a Recorded Video Surveillance Systems

Sabri Mohammed Alamin[1] and Othman O Khalifa[2]

[1]College of Computer Science & Information Technology, Sudan University of Science and technology, Khartoum, Sudan
[2]Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[1]sabri_alamin@hotmail.com, [2]Khalifa@iium.edu.my

Abstract

Most of the video surveillance strategies take a significant amount of space for storage as surveillance camera's unexceptionally recorded everything during camera – on time. Whereby, it leads to consuming the storage capacity of the device of the system. In fact, many algorithms have been proposed solving in the dilemma to object recognition and compress the video to reduce the size whenever it save's data. Nevertheless, the technology deprived efficient methods to reducing the storage of space for consummation.

The Idea of this paper is to propose a framework on how to possibly can be reduce the size of a recorded video of the surveillance system via recording only the part of the video that contains the motion, and ignore the other parts based on the motion detection. The result shows that the framework give an outstanding results on the uncompressed surveillance video recorded from a single fixed camera. The proposed framework enables to save 30% more of playback time and can provide more than 50% of storage of space saving.

Keywords*: motion detection, video compression, motion estimation*

1. Introduction

Recently, video surveillance systems have widely been used in a commercial and residential environments for safety purposes in, monitoring people's behavior, crime prevention and other activities. The basic surveillance activity is to detect and track important objects such as people and exhibiting suspicious behavior. In order to achieve these goals, It requires a large amount of camera to be used [1]. Increasing of the processing power along with its low costs' and affordability, the monitoring camera's helping the technology of the video surveillance in becoming the most active area to a vast research in academic field of industry and to other areas.

In video surveillance systems, however, found out that a problem occurs on object detection, many objects are available in one frame of the video and yet, when the target objects are small, then, the appearance tends to be less distinguished or noisy. More often, problem can be more complicated when tracking many objects are required in one video especially when the target part of an object is not visible as it's hidden behind from the other object [2, 3].

Extensive research has been conducted for object detections and tracking. Accordingly, we focus on studies that is transparently the same and has the common features on it. In [4] was developed an algorithm to real time detection and tracking of moving target objects in terrestrial scenes using a mobile camera. Although, the algorithm used a gray images for object detection. The architectural features of an automatic real-time video

Received (January 29, 2017), Review Result (August 15, 2017), Accepted (January 8, 2018)

ISSN: 2233-7857 IJFGCN

surveillance system was presented in [5]. This architectural design is capable of autonomously detecting anomalous behavioral events. However, the proposed system use of simple spatial information from the center-of-mass position of the object, without considering other important features (*e.g.* color).

Another detection technique is applied in video surveillance and monitoring system, it is an improved motion detection that algorithm [6] based on an integrated algorithm consisting of the temporal frame differencing, optical flow, double background filtering, and morphological processing methods. However, it is a very complicated algorithm and it takes high time for processing. In [7] developed a new method for vehicle detection, tracking and classification based on color. The method can detect and track only one car that approaches the camera.

Most of the video surveillance system's used a huge amount of space for storage as it would record everything that has been captured in the surveillance cameras. Having that said, It leads to generate a large of amount of videos every day that has been stored for archiving purposes. According to IDC report[8], fifty present of huge data came from video surveillance camera in 2010. The report also estimated that sixty five present of all big data's came from video surveillance data in 2015, and it says that the trend will continue for the coming years.

Many researchers have proposed a different type of methods for video compression, references No: [9] [10] [11] [12].

However, most of its work focuses on compressing the original video size and ignore the time of consumption for playback and viewing the recorded videos. Also, a non-standard resolution of the compressed video is considerable as another drawback of those methods.

In [13] has proposed a method to records only the part of the video that contains an important information. As expected, the proposed method solves two problems: Firstly, time consumed to record, second, time taken to view videos, in addition to excessive storage space required to store the videos. This goal has been achieved with a digital video camera and digital signal processing algorithm that detects motion; the surveillance system that was developed based on TI DSP 'C54' and their algorithm called block-based MR-SAD (Mean Reduced – Sum Average Difference). However, this method needs a special hardware and does not work with exist surveillance system camera.

Motion sensed real time video recording algorithm based on Mean Reduced Sum of Average Difference Method (MRSAD) was proposed in [14]. The algorithm starts by video recording only when motion is detected. However the main drawback of proposed algorithm is the false triggering problem, Due to different intensity of pixels at night as it compared to intensity of pixels by day.

A prototype for Intelligent Data Recorder and Transmitter for Surveillance was proposed in [15], this prototype enables the data acquisition and interfaces to the processor and transmits the data. Recorder is used to provide audio - video information by which can be transmitted to the relevant authorities using wireless network. However prototype was proposed not to deploy on the existing surveillance.

In this paper, we propose an efficient framework to reduce the archiving size of the videos which is based on the motion detection and tracking. Usually, surveillance videos that contains a huge number of sequential frames which refer to the same place without changing it. Our framework aims to reducing the storage space that required the surveillance video by recording only the part of the video that contains the motion captured, and ignore the rest of videos. Also, the proposed framework aims to reducing time consumption for playback and viewing the recorded videos. Proposed framework's assigned to work with video recordings from static digital cameras.

In general overview, the proposed framework consists of two modules: (a) Movement Detection, and Tracking module, and (b) Recording and Storing module. As shown in Figure 1. Movement detection and tracking module are consist of five phases. The first

phase is: Object segmentation which is used to identify the foreground objects from the background using a background subtraction on Gaussian Mixture Models. Second Phase is: Object recognition, is used to identify the foreground objects that should be tracked using a simple blob detection. The third Phase is: Object representation, it computes a representation for each recognized object to be tracked. Phase four is: Object tracking, it is used to track detected objects by using Kalman's filter. Recording and storing module is responsible to keep a list of frames of the output from module and store it on a new video file.

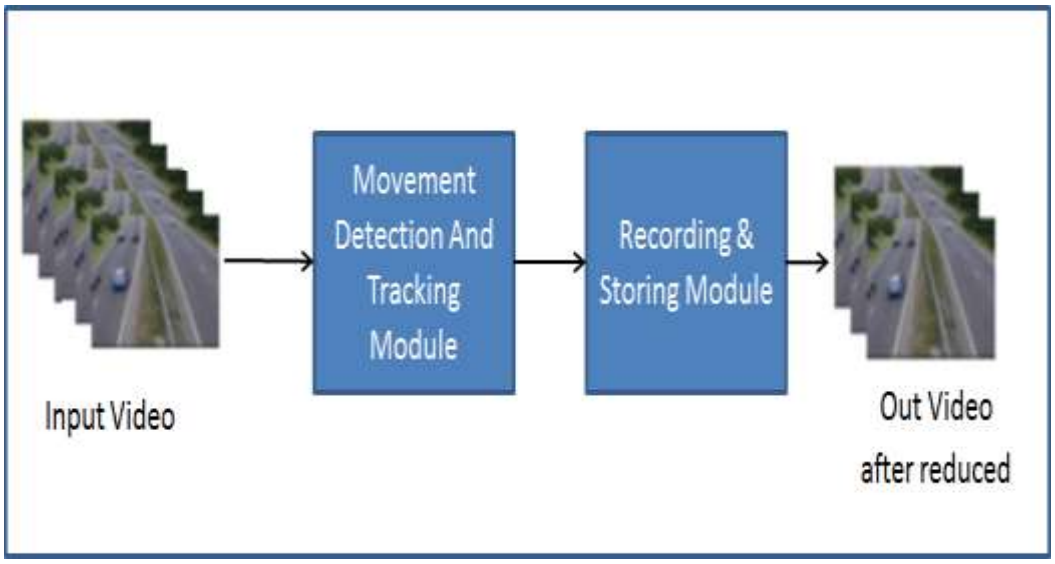

Figure 1. The General Overview of the Proposed Framework

This paper is organized as follows: In Section 2, the researcher demonstrates the designs of the proposed framework. Implementation and Results via Matlab software which is shown in Section 3. And Section 4 that includes the conclusion of this paper.

2. Proposed Framework

The proposed framework consists of two modules as shown in Figure 1. Movement detection module is used to determine the frames of the video that contains only the motion and extract important frames. Recording and storing module are used to record and store selected frames in video file. Figure 2 shows the structural design of the proposed framework.

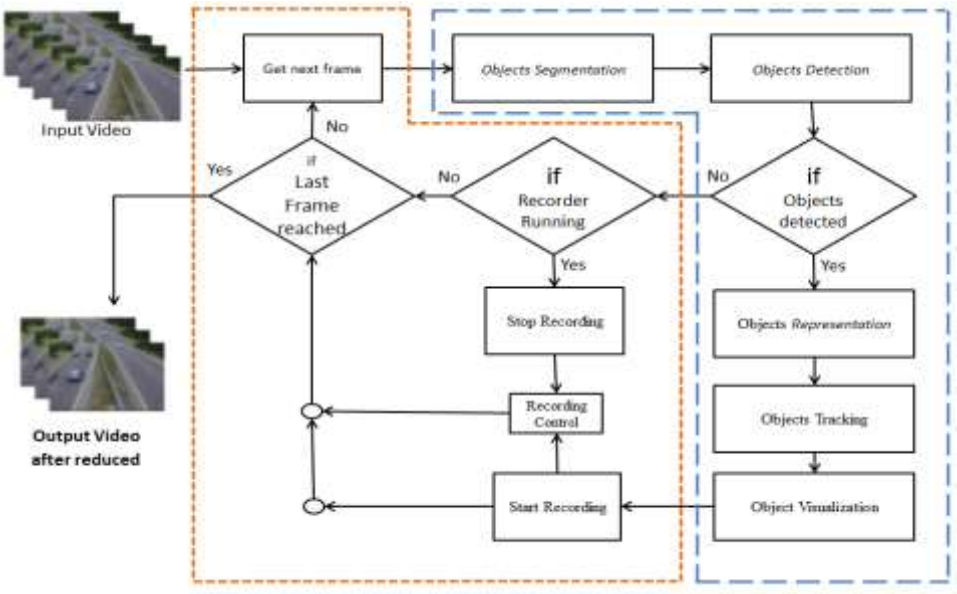

Figure 2. Structural of the Proposed Framework

2.1. Movement Detection and Tracking Module

This module aims to identify frames that contain important information, such as frames containing objects' movement in a video and ignore the others frame. (This does not contain movements, or contain fixed and redundant frames). The following subsections will describe the module phases briefly.

2.1.1. First Phase: Object Segmentation

Segmentation is the process of separating the foreground objects from the background of the video sequence by using simple background subtraction based on Gaussian Mixture Models [16].

Background subtraction is a common technique used for motion segmentation in static scenes. To detect moving regions, this technology subtracts the current image pixel-by-pixel from a reference background image that is created by averaging images over time in an initialization period [17]. The pixels where the difference is above a threshold are classified as foreground. After creating a foreground pixel map, some processing operations are performed to reduce the effects of noise and enhance the detected regions. With new images over time the reference background is updated to adapt to dynamic scene changes [17].

The simple version of this scheme where a pixel at location (x, y) in the current image It is marked as foreground if

$$|I_t(x,y) - B_t(x,y)| > T \quad (1)$$

Is satisfied where T is a predefined threshold. The background image Bt is updated by the use of an Infinite Impulse Response (IIR) filter as follows [16]:

$$B_{t+1} = \alpha I_t + (1 - \alpha)B_t \quad (2)$$

The foreground pixel map creation is followed by morphological closing and the elimination of small-sized regions.

In Gaussian Mixture Model, every pixel in a frame is modeled into Gaussian distribution. First, every pixel is divided by its intensity in RGB color space. Every pixel is computed for its probability whether it is included in the FG or BG with:

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t}.\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (3)$$

- X_t : current pixel in frame t
- K: the number of distributions in the mixture
- $\omega_{i,t}$: the weight of the kth distribution in frame t
- $\mu_{i,t}$: the mean of the kth distribution in frame t
- $\Sigma_{i,t}$: the standard deviation of the kth distribution in frame t
- Where $\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$ is probability density function (pdf):

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\sum|^{\frac{1}{2}}} exp^{-\frac{1}{2}(X_t - \mu)\sum^{-1}(X_t - \mu)} \quad (4)$$

2.1.2. Second Phase: Object Recognition or Detection

Detection or Recognition is the process of which of the segmented foreground objects is interesting to be tracked based on simple blob detection [4]. Blob detection is mathematical methods that is aimed to detect regions in a digital image that vary in properties, such as color or brightness, compared to areas surrounding those regions [7, 18]. Blob detection was used to obtain regions of interest as the "recognized" object to be tracked.

2.1.3. Third Phase: Object Representation

The results of the second phase was used as an input for this phase. This phase also used to extract the bounding box and the size of each detected object, and store it into array where every element of array is a data structure that used to store properties and information about objects that are tracked in a video.

This phase is responsible for maintaining the list of objects being tracked and the establishment of the correspondence between the recognized objects and those that currently tracked. The recording and storing to check the size of the list; if list size is greater than zero, then recording is started and will be continued until the list size is equal to zero (recording stop).

2.1.4. Fourth Phase: Object Tracking

This phase uses Kalman's filter method to follow and trace the detected objects depending on the results from third phase to track the position of the objects being tracked[4]. The Kalman's filter method works by appreciating an unobservable state which is updated in time with a update of additive Gaussian noise and linear state.

Kalman's filter is a Point Tracking method that is used to detect an object in a sequence of frames that has been detected based on the rule: if f and h are linear functions and the initial state X and noise have a Gaussian distribution then the optimal state estimate is specified by the Kalman's Filter: prediction and correction

Prediction: Below are the steps used on the state model to predict the new state of the variables:

$$\bar{X}^t = DX^{t-1} + W$$
$$\bar{\Sigma}^t = DX^{t-1}D^T + Q^t \quad (5)$$

Where D is the state transition matrix which defines the relation between the state variables at time t and t-1, And Q is the covariance of the noise W.

Correction: The following steps are used on the current observations Z^t to update the object's state:

$$K^t = \bar{\Sigma}^t M^T [\, M\bar{\Sigma}^t M^T + R^t \,]^{-1} \quad (6)$$
$$X^t = \bar{X}^t + K^t \underbrace{[\, Z^t - M\bar{X}^t \,]}$$

$$\upsilon$$
$$\Sigma^t = \bar{\Sigma}^t - K^t M \bar{\Sigma}^t \quad (7)$$

Where: K is a Kalman's gain, M is a measurement matrix and V is called the innovation.

2.1.5. Fifth Phase: Object Visualization

This phase is implemented by drawing and displaying a yellow rectangle around of the detected objects in current frame. In addition, It has a display label on the left top corner of the video which is used to describe the number of objects that are currently being tracked. This phase, in turn, is useful to give a minimal visualization to show and follow up the tracking phase performance.

2.2. Recording and Storing Module

This module aims to record only frames that contain information by getting results from movement detection module and storing all frames in a new video file. This can be achieved by one object or more than one, as the object has been detected in a frame; this frame will be stored in a new video file, and will continue storing process. When it happens to detect a frame that does not contain any object; in such a case, storing process automatically stops, whereas, empty frames will not be shown in the concerned file. The process of storing frames goes on until reaching the last frame on an input video.

To check the quality of our framework, we calculated the storage for space savings and Playback savings as follows. The Storage space savings can be defined as the reduction in size relatively to the uncompressed size based on the following equations:

$$Storage\ space\ savings = 1 - \left(\frac{Compressed\ size}{Uncompressed\ size}\right) \quad (8)$$

Where the uncompressed size is the size of original video and the compressed size is the size of an output video after the frame work has been applied.

Playback savings can be defined as the reduction in time relatively to the uncompressed playback time based on the following equations:

$$Playback\ savings = 1 - \left(\frac{Compressed\ playback\ time}{Uncompressed\ playback\ time}\right) \quad (9)$$

3. Implementation and Results

The proposed framework is validated and tested on several video sequences from visual surveillance system datasets. In order to implement the proposed framework we used Matlab's 2015a with Windows 7 64 bits on Intel(R) Pentium(R) Dual CPU T2370 @1.73 GHz and all videos were encoded using mpeg-4 encode.

Table 1. The Comparative Results for Storage Space and Playback based on our Framework

Experiment	Original video			Output video			Storage space savings	Playback savings
	Size	Num of frames	Playback duration/ minutes	Size	Num of frames	Playback duration/ minutes		
First video	3.23 MB	2358	1.56	1.3 MB	807	0.55	59.8%	64.7%
Second video	453 MB	20655	11.63	224 MB	14748	8.11	50.6%	30.3%
Third video	3.84 GB	940,577	627.5	534 MB	182,359	126.63	99.9%	79.8%

The first video was taken from CAVIAR Test dataset[19]. We use a video recorded from frontal view of the surveillance camera of the Shopping Center. The video contains a

couple walking down the corridor, and people are going in and out of the stores. The number of frames in the original video is 2358 frames which accumulated the size of 12.9 MB; As we applied Our frameworks frame, as expected the frames decreases to 807 that only accumulated the size of becoming 1.3 MB by 59.8%. Playback was reduced by 64.7% that equals to 0.55 min. The results in Figure 3.a Shows the sample of an empty frames that does not contain any objects on the original videos. The Figure 3.b shows the frames in the output video of the sequence; each frame contains an object with yellow rectangle around the detected object.

Figure 3.a. The Original Frames of Sequence

Figure 3.b. The Output Video Frames of the Sequence

The second video was taken from VIRAT data set [20]. We use a recorded video from outdoor surveillance camera at the parking area. The video contains a number of people walking, and moving vehicles. The number of frames on the original videos are 20655 frames, and accumulated the size of 453 MB; and when our framework applies, frames are then decreased to 14748 frames, and only accumulated the size of 224 MB by 50.6%. Playback was reduced by 30.3% that equals to 8.11 min. The results in (Figure 4.a) Shows a sample of empty frames that does not contain any objects on the original videos. The (Figure 4.b) show frames in the output video of the sequence; each frame has contain an object with yellow rectangle around the detected object

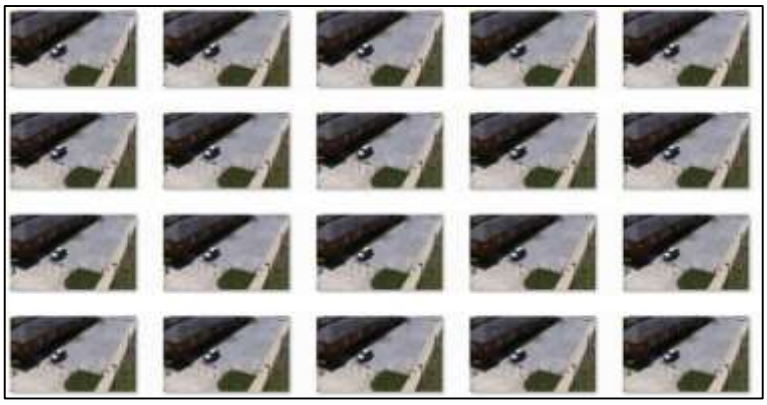

Figure 4.a. The Original Surveillance Video Frames of the Sequence

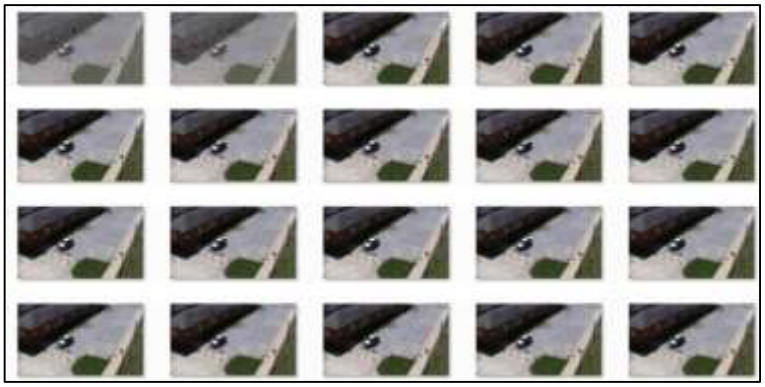

Figure 4.b. The Results Output Video Frames Sequence after Our Frameworks Apply

As we seek for more validating and testing of our framework; we used the third video from real surveillance camera that provides a large surveillance time[21]. Below are the recorded video of the stairs to one of the buildings in the University. The numbers of frames on the original videos are 940,577 frames and accumulated the size of 3.84 GB; after we apply our framework, frames are decreased to 182,359 and the size becomes 534 MB. The results in Figure 5.a Shows that the sample of an empty frames that does not contain any objects on the original videos. The Figure 5.b shows the frames in the output video sequence; And each frame has an object that looks like a yellow rectangle around it upon object detection.

Figure 5.a. The Original Indoor Surveillance Video of Frame Sequence

Figure 5.b. The Results of the Output Video after Our Framework Applies

4. Conclusion

In this paper, we have proposed a framework to reducing the size of a recorded video on a smart surveillance system. The proposed framework mainly focuses on the two goals: (a) to reduce the storage of space that required to store video by recording the parts of the video that contains only the motion and to ignore the rest of video. (b) To reduce the consumption time for playback and viewing videos of the surveillance system. So, that, this framework gives more quality as a results for the uncompressed surveillance video (*e.g.* with fixed camera). In the future, our proposed framework can be extended to work with the video from motion camera.

References

[1] T. Huang, "Surveillance video: the biggest big data", Computing Now, vol. 7, no. 2, **(2014)**, pp. 82-91.
[2] E. Maggio and A. Cavallaro, Video tracking: theory and practice: John Wiley & Sons, **(2011)**.
[3] S. M. Ahmed and O. O. Khalifa, "Vision-based detection and tracking of moving target in video surveillance", pp. 16-19.
[4] A. Behrad, A. Shahrokni, S. A. Motamedi and K. Madani, "A robust vision-based moving target detection and tracking system",
[5] A. Mecocci, M. Pannozzo and A. Fumarola, "Automatic detection of anomalous behavioural events for advanced real-time video surveillance", pp. 187-192.
[6] N. Lu, J. Wang, Q. Wu and L. Yang, "An improved motion detection method for real-time surveillance", IAENG International Journal of Computer Science, vol. 35, no. 1, **(2008)**, pp. 1-10.
[7] H. Rabiu, "Vehicle Detection Tracking and Colour-based classification in Video", in IJAI, vol. 2, no. 1, **(2013)**.
[8] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. December 2012", URL http://www. emc. com/collateral/analyst-reports/idc-the-digital-universe-in-2020. pdf, **(2014)**.
[9] S. Kuzmin, "Video compression for panoramic video surveillance systems", pp. 51-53.
[10] G. Qiang, L. Yue and F. Yu, "An region of interest based video compression for indoor surveillance", pp. 157-160.
[11] S. Wang, Y. Chen and Y. Bai, "A surveillance video compression algorithm based on regional dictionary".
[12] L. Tian, H. Wang, Q. Tang and Y. Zhou, "Surveillance Source Compression with Background Modeling for Video Big Data", pp. 105-110.
[13] C.-K. Huang and T. Chen, "Motion activated video surveillance using Ti Dsp", DSPS FEST, vol. 99, **(1999)**, pp. 4-6.
[14] S. Badnerkar and P. Ingole, "Motion sensed video storage algorithm for surveillance recording", pp. 431-434.
[15] G. Sasikala and M. Varadarajan, "Intelligent Data Recorder and Transmitter for Surveillance: A Survey Report", International Journal of Machine Learning and Computing, vol. 3, no. 1, **(2013)**, pp. 1.
[16] C. L. Devasena, R. Revathí and M. Hemalatha, "Video surveillance systems–a survey", International Journal of Computer Science (IJCSI), vol. 8, no. 4, **(2011)**.

[17] A. M. Cheriyadat, B. L. Bhaduri and R. J. Radke, "Detecting multiple moving objects in crowded environments with coherent motion regions", pp. 1-8.
[18] Y. C. Wang, C. C. Han, C. T. Hsieh and K. C. Fan, "Vehicle type classification from surveillance videos on urban roads", pp. 266-270.
[19] R. Fisher, J. Santos-Victor and J. Crowley, "CAVIAR: Context aware vision using image-based active recognition", **(2005)**.
[20] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee and L. Davis, "A large-scale benchmark dataset for event recognition in surveillance video", pp. 3153-3160.
[21] I. A. Taj-Eddin, M. Afifi, M. Korashy, D. Hamdy, M. Nasser and S. Derbaz, "A new compression technique for surveillance videos: Evaluation using new dataset", pp. 159-164.

