International Journal of Engineering & Technology, *7 (4) (2018) 5829-5843*

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.11865
Research paper

Key-frame extraction for summarization of surveillance footage by analysis of colour histograms

Gauri Tyagi [1] *, Bhrigu Kapur [1], Sathyaraj R [1]

[1] Vellore Institute of Technology, Vellore
**Corresponding author E-mail: gauritg@gmail.com*

Abstract

Everyday a plethora of video content is generated by surveillance cameras all over the world. This footage has two major problems, it takes a lot of space even for parts that are not important (empty rooms, night time recording etc) and also takes a lot of time to review the same. Our objective through this paper is to reduce the spatial and temporal redundancies of the video through a process known as video summarisation. This paper proposes a summarization algorithm for surveillance footage via key-frame extraction, based on comparison of consecutive frames of the video over certain frame descriptors. The algorithm avoids exhaustive comparison by using K-means on the colour bins of each temporal shot to extract dominant colour bins to extract relevant sections of the surveillance footage. Experiments are performed on the i-Lids dataset for AVSS (Advanced Video and Signal based Surveillance) 2007 and EC Funded CAVIAR project' dataset on city surveillance. Ground truth was used as the metric to judge the validity of the proposed algorithm. The obtained values from the algorithm are evaluated for precision, recall and F-measure. We found out that our algorithm satisfies the ground truth of the all video datasets and is also fast enough to perform the action. We conclude by showing the results and their comparisons of how our algorithm performs by highlighting various metrics of precision and accuracy.

Keywords: *Frame descriptors; Key frames extraction; Surveillance; Video Summarization; Visual Summary evaluation*.

1. Introduction

In the contemporary world surveillance cameras are ubiquitous, continuously monitoring and recording human activity as it happens. The immense data that is generated is stored on hard drives to be kept for referencing it at a later point of time. High security areas such as museums, government buildings and prisons have a team of people constantly monitoring the feed. The status quo is largely inefficient in the way surveillance happens both spatially and temporally. Surveillance cameras record continuously and often there is no activity happening in the frame of view either due to the location of the camera (pointing towards a desolate area) or during hours of inactivity (night time recording) making a large portion of the footage "uninteresting" to the surveillance. When the raw footage from a camera is stored on a hard drive, hundreds and thousands of gigabytes of "uninteresting" footage is stored with it occupying unnecessary space. Storing of the footage also becomes extremely costly which is something small business and home owners cannot afford so they have to write over the previous data, permanently erasing the earlier records. The problem is not limited only to space occupied but extends to the large amount of human time and effort needed to monitor/examine the footage either in real time or post facto the recording.

To remove unnecessary frames and retaining the pith while maintain the chronological order of the constituent frames of the input video is called video summarization. The summary provides a gist with minimum loss of information in and also considerably reduces the playback manifolds. Theoretically it can be seen as a context aware content-based image and video retrieval algorithm [1]. Surveillance video footage is highly redundant both in terms of space occupied by the video and the length of the video. Video summarization aims to address both these problems [3]. We need such software to automate the process of finding clips of interest in the footage, which otherwise would be carried out manually via skimming through the entire lengths of the video. Hence considerably reducing payback time and storage requirements without loss of true content. The process of summarisation eliminates the need of excessive manual browsing to retrieve significant events of interest from the footage by automating the process via key frame extraction. Our proposed algorithm is purely software based, and not aided by any hardware supplements such as PIR sensor or motion detection cameras. Making the algorithm sustainable and cost effective and hence scalable. To arrive at the summary, we use the technique of key frame extraction.

To better understand the mechanism of keyframe extraction and the different algorithms that can be used we must first be familiar with the anatomy and structure of any video sequence.

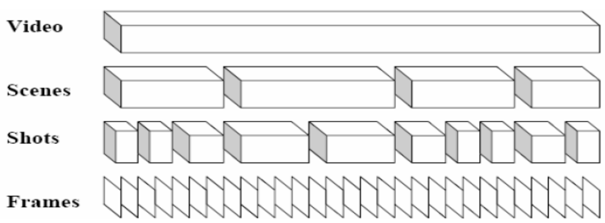

Fig. 1: Structural Anatomy of a Video Sequence.

The number of key frames extracted per shot are accurate representation of the convolution of the video fed to the algorithm.

Literature classifies two approaches, namely cluster based and sequential. Here we discuss, sequential based approach, which makes use of space and time bound features to estimate the importance of a frame. Whenever the algorithm detects significant amount of change (based on a threshold) in comparison to preceding and succeeding frames the frame in question is deemed to be a key frame. The unsupervised approach based on a method known as clustering produces a congregation of frames that are semantically similar. The centroid of each of these individual clusters marks the key frame of that particular congregation. Unlike the conventional uses of clustering where the output doesn't have to respect the sequence of events in the input provided to it, over here preservation of global time sequence of the utmost priority. Comparing the histogram of a particular frame to it's nearby frames is the approach that has been adopted by the proposed framework.

2. Related works

Liu et al in [10] has extensively worked on various key frame extraction and proposes the following methods. Video shot method: It used the histogram and frame average method. Key frame extraction is achieved by calculating the farthest possible distance from the vector grid. Semantic analysis: Properties of the frames such as hue, spatial smoothness and indicators of visual information are monitored and when changed significantly represents a key frame. Method applying clustering: This method is simply making the clusters of all frames and the selection if the key frames is done by choosing the frame that is closest to the center. Motion Based Analysis: Movement of objects in the frames are observed and monitored over various frames. The proposed algorithm by Liu et. al. [10] provides insight into the film and summaries as they hypothesize that directors often put the most important part of the story in the center. Using this assumption, they provide more weight to the central frames. But this tendency to centralise important frames is not holistic and falls short in many applications.

Zeinalpour in [1] applies genetic algorithm in the process of generating summaries of the video files. They use a multistep procedure which is: Probing: Frames that are adjacent to each other will be the most similar, they reduce the number of frames by eliminating the frames that look similar. Encoding: They then assign the weights which would indicate whether that particular frame is included or not. 0 would indicate that the frame is to be eliminated but 1 indicates that we need to include and take that frame. Fitness Function: The fitness function described the selection criteria of the frames that are ultimately included. Crossover and Mutation: The genetic algorithms then select pairs of the chromosomes corresponding to the measure of their fitness in the sample space. In the later stages, two chromosomes will exchange the genome value from a point of split chosen randomly. The boundary condition estimates the mean of the chromosome values across the entire population. This approach is accurate but is too complicated to set up and works only in certain domains.

Sony et al. [11] in their research use the Euclidean distance after unsupervised learning to get the final frames that will form the summary. The aforementioned algorithm is again majorly built upon on the pruning of the duplicate frames and makes sure only a finite user defined number of frames are there in the summary. The frames that are visually similar are grouped together using the Euclidean measure. Following the clustering procedure frames with larger distance between them are taken from every cluster to form a pattern which would be the resultant output.

Doulamis et. al. with their research [12] examined the method of cross correlation criteria that is applied by the formation of a histogram that is multidimensional in nature and fuzzy to take out relevant key frames from the input video. Their research provides an important starting point to examine the key frame extraction which forms the basis of any video summarizer.

Raikwar et al. [13] provide an interesting method to extract key frames, they posit that rather than use techniques such as clustering etc. to select the best possible key frame, they directly pick up the starting frame of each temporal sample of the video as the de facto key frame. Doing so saves lot of time in the extraction process making the system extremely fast but this may not be the correct choice as the first frame might not be reflective of the best/most important detail of the video.

Zhao et al.[14] proposed a key frame extraction technique built on the idea of segmenting curve of the video representation. Frame representation is done by calculation colour histograms wherein the CIE UV color space values with 256 bins denote each frame. The frames are then compared for similarity, a distance measure is estimated for neighboring frames which is plotted in a 2-D vector space. Examining this curve for sharp edges and corners gives us the keyframes of the temporal shot in question. This approach of color space representation of every single frame and working with differentials of curve to locate regions of interest is inherently brute-force and hence computationally expensive.

Mukherjee et al. in [15] ideated a framework that robustly estimates key-frames from footage segments based on the prediction of randomness of the consequent frames. This estimation is followed by wavelet calculation, namely Har and spatial wavelet. For the case, representing par intensity, busyness of intensity is denoted by a 3x3 mask matrix for the image frames. While in the case of Har transform wavelets, the frequency points in space are estimated at varied resolution. Also frame dependent characteristics are proved to be a good measure of randomness across a neighborhood of frames, wherein frames with a randomness value that breaches a threshold are selected as keyframes.

Zhuang et. al. in [16] made use of unsupervised clustering to achieve key frame extraction in the research. Color histograms of each frame of video were taken out with HSV so as to have minimum effect of illumination on the algorithm as the HSV color space separates the chroma (color) from the luma (illumination). After that a threshold was added to monitor the clustering density of the video. Only clusters that are considered to be large enough are considered for key frame extraction in the first place. Each big enough cluster contributes a key frame from their respective cluster. This method of representing an entire cluster through it's centre frame fails because the spatial centre frame may not necessarily also be a keyframe with respect to the region of interest.

Gong and Liu in [17] exploited the idea of singular value decomposition to achieve video abstraction. In the RGB color space the proposed methodology computed the color histogram of the video frames. Each frame is divided into 3x3 block to better use the spatial information so as to generate a three-dimensional histogram for the respective blocks. Concatenation of all the sub histograms leads to the formation of array space containing features. Feature vectors are pivotal in forming the clusters. The closest to the center of each cluster that is formed by feature vector is selected as the key frame.

Clustering is another popular method to extract key frames. Xiaolin and Peng [18] put in extensive research into making the color histogram that provided the basis to cluster the frames upon which the most prominent features were taken to form key frames in each cluster. The clustering algorithm that they used for clustering prioritized frames based on the cluster and when the clusters were compared against each other the temporal order of the frames was lost meaning that video was out of not in the order of sequence of events but randomized to an extent.

3. Event detection strategies

3.1. Change detection

This method uses the fluctuation in the color vector across subsequent frames as the bedrock foundation for change detection. We presuppose under reasonable boundaries, for CCTV footage, that the scene will remain static for the greater part of the footage. Which implies, that each frame under evaluation that the color vector does not change across the immediate frame neighborhood. Quantitative verification is done via comparison of the color histograms of consecutive frames. Correspondingly, if a significant event happens in the footage, it is reasonable to expect that the change in the hue of the frames will be quite significant.

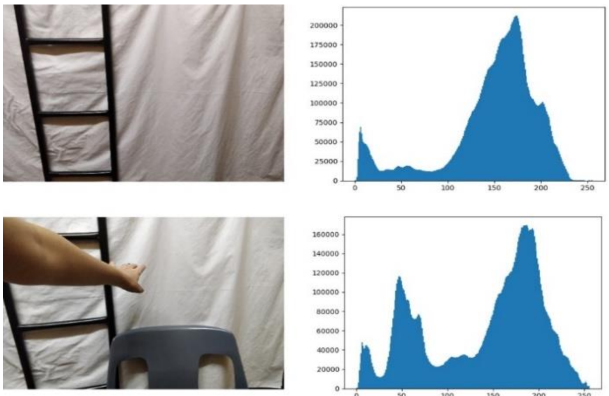

Fig. 2: Effect on Colour Histogram with Change in the Image.

Figure 2 is an illustration of the effect on the colour histogram with a change in subsequent frames. The similarity between histograms is representative of the degree redundancy in the frames, while the variance between them is a measure of the proportional significance of the frames.

3.2. Colour histograms

Every frame is itself made up of the most basic elements called pixels that collectively form the image that is represented on the screen. Each pixel represents the variation of three primary colors red, green and blue in different proportions to form the color for that individual pixel. The number of colors after discretizing the color space make the color histogram of an image [6]. This way of discretizing the images as opposed to pixel value comparison between the frames under evaluation will directly circumvent problems triggered by noise in the video—Pixel values are susceptible to fluctuation between frames even for static frames in a still camera footage, particularly if the video is recorded in low lighting conditions. The images can be refined by the application of contrast enhancement on the so obtained color and grayscale images[20]. Figure. 2 illustrates how a change in scene corresponds change in the colour histogram.

3.3. Transformation of the colour space to HSV

RGB color model is used to represent the raw input video that is captured via almost all camera devices. Switching color space model from RGB to HSV is a pragmatic step to ease computing hue difference among frames because the HSV color model separates chroma and luma[21]. Chroma in an image provides us with the color information while luma is responsible for telling us about the intensity of an image. For the problem of key-frame extraction at hand, changes in the intensity can be ignored and only changes in the color information are needed. Doing so, makes the algorithm not affected by intensity changes such as the natural lighting and even shadows of objects and people.
Conversion from the RGB color model to the HSV color model is step one. The HSV is an alternative representation to the RGB color space. For thi s particular application, the most optimum choice was the HSV model on the grounds that it provides us with a perfect solution to the problem of separating various components of colour representation, namely luma and chroma. Also this transformation between these two models is simple enough to calculate.
RGB Conversion to HSV done by the following: [7]:

$$V \leftarrow \max(R, G, B) \tag{1}$$

$$S \leftarrow \begin{cases} \frac{V-\min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 \text{ otherwise} & \end{cases} \tag{2}$$

$$H \leftarrow \begin{cases} 60\frac{(G-B)}{V} - \min(R,G,B) & \text{if } V = R \\ 120 + \frac{60(B-R)}{V} - \min(R,G,B) & \text{if } V = G \\ 240 + 60\frac{(R-G)}{V} - \min(R,G,B) & \text{if } V = B \end{cases} \tag{3}$$

If $H < 0$ then $H \leftarrow H + 360$.

3.4. Histogram comparison

There are three methods to compare the histograms for an image which are as follows:
Histogram Correlation is mathematically defined as:

$$d_{correl}(H_1, H_2) = \frac{\sum_i (H_1(i)-\bar{H}_1)(H_2(i)-\bar{H}_2)}{\sqrt{\sum_i (H_1(i)-\bar{H}_1)^2 \sum_i (H_2(i)-\bar{H}_2)^2}}$$

where

$$\bar{H}_k = \frac{1}{N}\sum_j H_k(j) \tag{4}$$

Where the total count of histogram bins is represented by the letter N. The number 1 signals an exact match, while a-1 represents incongruity.
Chi Squared comparison is defined as follows: [9]

$$\chi^2(H_1, H_2) = \sum_i \frac{(H_1(i)-H_2(i))^2}{H_1(i)} \tag{5}$$

Any value apart from zero suggests that the histograms are disparate, while perfectly matching histograms produce zero as the chi-square distance.
Bhattacharyya distance is defined as follows:

$$d_{Bhattacharya} = \sqrt{1 - \frac{\sum_i \sqrt{(H_1(i).H_2(i))}}{\sqrt{\sum_j H_1(j).\sum_j H_2(j)}}} \tag{6}$$

The perfect match would give a score of 0 and 1 denotes not a match.
As mentioned in section one there are three methods to compare histograms, chi squared was chosen as the method for histogram comparison as it had the lowest value of normalization. This low value indicates that it is the least susceptible to noise among the three methods.

Table 1: The Normalized Histogram Comparison Results

Comparison Method	Same frame	No activity	Activity
Correlation	0.0	0.0623	1.0
Chi-Square	0.0	0.0567	1.0
Bhattacharyya Distance	0.0	0.3418	1.0

Table 1 illustrates the results of the histogram comparison across a neighborhood of frames for three specific scenarios, these results are normalized.

3.5. Optical flow

Optical flow deals with the distance that a pixel has travelled through various frames and designates the amount of "change". This in its very basic essence is the tracking of various pixels throughout the screen at different intervals of time and taking cognizance of their motion.
There are two approaches to Optical Flow that we will explain over here:

3.5.1. Fireback algorithm

A polynomial expansion was used hypothesized by Farnebäck to estimate the motion between two frames. Farnebäck's method assumes that the brightness and texture remain constant across the subsequent images. The following Quadratic polynomial helps us approximate the neighborhood of pixels:

$$f(x) \sim x\ TAx + b\ T\ x + c \tag{7}$$

In the formula above, the scalar quantity is denoted by 'c'. 'A' the coefficient is reflective of a symmetric matrix and the coefficient 'b' suggests a vector quantity. Iteratively solving the equation mentioned above over the input frames gives us the optical flow value.

3.5.2. Lucas kanade optical flow

The movement of a pixel is monitored by a movement vector (u,v). Movement vector is associated with all "interesting" pixels which is arrived at by the juxtaposition of a frame with either its succeeding or preceding frame.
There are two assumptions that the algorithm follows:

a) Δt is the time difference by which two images are told apart.
b) Objects feature a natural scene containing textured objects that have various intensity levels and change smoothly.

The Lucas Kande method makes the most reasonable approximation of the displacement of a neighbourhood. In the initial stage where the values of both u and v are unknown with their equations, we require a neighbourhood in order to get more equations. This helps in the algorithm forming the change and we then look for a minimum squared solution.

3.6. Evaluation metrics for accuracy

The output will be evaluated on primarily three metrics, Precision(P), Recall(R) and F-Measure (F). These are defined as follows:

$$Precision(P) = \frac{True\ Positive}{True\ Positive + False\ Positve}$$

$$Recall(R) = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F - Measure(F) = \frac{2 \times P \times R}{P + R} \tag{8}$$

4. Dataset

Experiments have been performed on videos from i-Lids dataset for AVSS (Advanced Video and Signal based Surveillance) 2007 and EC Funded CAVIAR project' dataset on city surveillance.and results are compared with the standard ground truth. These are datasets for event detection in CCTV footage and general surveillance.

Table 2: Dataset Details Sample (Video 1)

No.	Attribute	Value
1.	Location of recording	Various locations in the UK
2.	Number of sequences	7
3.	Total number of images	35000
4.	Image size	8-bit colour MOV
5.	Format of images	720 x 576 pixels
6.	Video sampling rate	25 Hz

5. Proposed algorithm

As seen in the literature, numerous approaches have been adapted to perform key-frame extraction, but the following limitations were identified.

- Brute force comparison of each frame in the video sequence is computationally expensive therefore not suitable for our application for surveillance systems where time is of value.
- Majority of the key-frame approaches use some sort of a static threshold for similarity measurement. Which cannot accommodate to changing degree of activity in the footage subsequently resulting in loss of key events.

The proposed algorithm is an efficient variation on the traditional, more intuitive histogram comparison technique. In the latter technique, the footage to be summarized is spilt into all of its component frames. In this sample space, all the frames are exhaustively compared with its prior and subsequent frames. This approach is not only based on an often-undesirable brute-force approach and relies on static values. Many existing algorithms also need training before they can give satisfactory results on benchmark or test data. Keeping these gaps in mind, here we propose a more inclusive framework that is based on clustering and dynamic statistical measures to prune out "uninteresting" frames to ultimately check for key

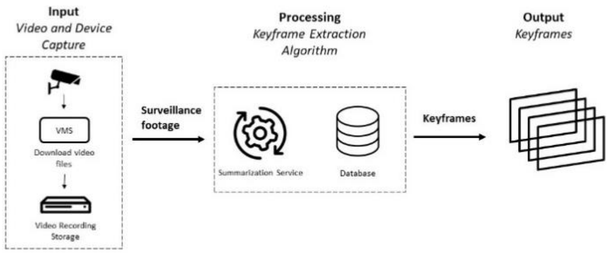

Fig. 3: Input-Output framework of the Summarisation Algorithm.

substantial changes in the video and provide frames only from the "interesting" sections. Figure 3 clearly highlights the input-output of our algorithm.

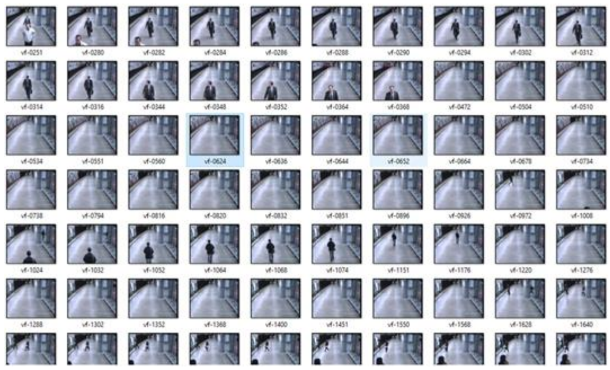

Fig. 4: Converting Video to Component Frames.

1) The first step is frame skipping, i.e. to prepare the video sequence in a manner that it can be fed into the key-frame selection algorithm. Unlike many of the algorithms in literature which perform similarity check on each frame of the footage with a naïve approach. This step reduces the quantity of frames(images) in the subset of the video by extracting only a certain number of frames per a given unit

of time. Trial and error proved the value of one frame per second to be the most optimum. Extract one frame for every second of video. Effectively compressing a video down into much less data. The output a set of frames that represent transition points in the video. This module of the algorithm returns output the frames as well as a metadata file with data collected during the process. This data will include dominant colours in the images for the key frames.

2) The second step is to extract dominant colours from the frames from the previous step. K-Means is applied on the pixels of the frames from step 1 and then returns the centroids of the colours. Clustering parameters are set by trial and error. Calculates change in data one frame to the next one. Convert to grayscale, scale down and blur to make calculating image differences more robust to noise. Compute colour states i.e. bin mean, median and standard deviation. Select a set of frames as key frames (frames that represent a significant difference in the video i.e. potential scene changes). Key frames are selected as those frames where the

number of pixels that changed from the previous frame are more than 1.85 standard deviations times from the mean number of changed pixels across all inter frame changes.

6. Results and discussions

Table 2 shows the details of the dataset for the video in illustration. Table. 3 shows the experiment test values gained by the proposed video summarization algorithm. The key-frames extracted by the proposed scheme covers most of the frames in the ground truth. Resulting in comprehensive summaries of all 5 videos considered for experiment.

Table 4: Key Frame Results

Video	Frame count	fps	Height	Width	Number of Keyframes
Abandoned baggage	5474	25	576	720	174
Parked vehicle detection	6104	25	576	720	80
People leaving bag behind chairs	1075	25	288	384	56
Two people fighting	4735	25	375	420	126

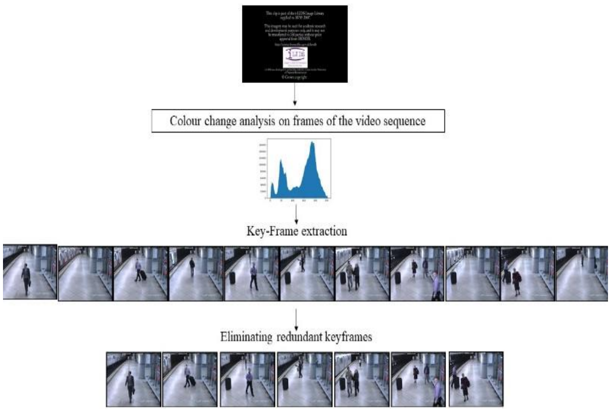

Fig. 5: Overall Framework.

Table 3: Evaluation of the Proposed Algorithm

No.	Proposed Algorithm			Frames Blocks Feature method		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1.	1	0.75	0.85	0.5	0.5	0.5
2.	0.82	0.81	0.81	0.71	0.44	0.54
3.	0.69	0.56	0.62	0.90	0.63	0.74
4.	1	1	1	0.78	0.60	0.68
5.	0.96	0.91	0.93	1	1	1
6.	1	0.85	0.92	0.80	0.80	0.80

It can be easily observed, that the scenes detected via the key frames extracted by our proposed algorithm cover majority of the ground truth and do not miss any significant scene from the surveillance footage. The proposed framework is also compared with the algorithm given by Liu et al.19. The proposed algorithm accommodates majority of the frames representing the ground truth when compared to the second technique. Subsequently, further study was done on the soundness of the performance of the individual descriptors and overall precision, recall and f-measure. It was also noticed that the chronological structure of the video was preserved in the generated summary, and the global essence of the footage wasn't lost

7. Conclusion

A framework for an intelligent surveillance system via video summarization is proposed, which works by extracting the key frames of the input footage. Our system has two other modules, namely face detection/recognition and alert sending. The former ensures authentication for the location being secured and alert sending prevents the exhaustive activity of browsing through the recorded footage and avoid polling by notifying the client with timely summaries of the recorded footage.

The results obtained from the experiments on the video from the datasets from CAVIAR and i-Lids dataset show that the extracted key frames by using the algorithms proposed result in satisfactory summaries as can be seen from the values of the f-measure, precision and recall.

Acknowledgement

We would like to express our heartfelt gratitude to the Computer Science department of Vellore Institute of Technology for their continued support and encouragement to pursue this research work.

References

[1] Zeinab Zeinalpour, Behrouz Minaei Bidgoli, Mahmud Fathi, "Video Summarization Using Genetic Algorithm and Information Theory" Computer Conference, 14th International CSI, 2009: 158-163.

[2] Clive Norris. A review of the increased use of CCTV and video-surveillance for crime prevention purposes in europe. Briefing Paper for Civil Liberties, Justice and Home.

[3] Vikas Choudhary and Anil Kumar Tiwari. Surveillance video synopsis. In ICVGIP, pages 207–212. IEEE, 2008. https://doi.org/10.1109/ICVGIP.2008.84.

[4] Zhonglan Wu and Pin Xu, "Research on the technology of Video key-frame extraction based on clustering", IEEE Fourth international conference on Multimedia Information networking and security, 2012, p. 290-293

[5] Naveed Ejaz et al., "Adaptive key-frame extraction for video summarization using an aggregation mechanism", Journal of Visual Communication 23 (2012), p.1031-140. https://doi.org/10.1016/j.jvcir.2012.06.013.

[6] Michael J. Swain and Dana H. Ballard. Colour indexing. International Journal of Computer Vision, 7(1):11–32, November 1991. https://doi.org/10.1007/BF00130487.

[7] Linda G. Shapiro and George C. Stockman. Computer Vision. Prentice Hall, 2001.

[8] H.D. Cheng, X.H. Jiang, Y. Sun, and Jingli Wang. Colour image segmentation: advances and prospects. Pattern Recognition, 34(12):2259 – 2281, 2001. https://doi.org/10.1016/S0031-3203(00)00149-7.

[9] Ofir Pele and Michael Werman. The quadratic-chi histogram distance family. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, Computer Vision - ECCV 2010, volume 6312 of Lecture Notes in Computer Science, pages 749–762. Springer Berlin Heidelberg, 2010. https://doi.org/10.1007/978-3-642-15552-9_54.

[10] Huayong Liu, Wenting Meng, Zhi Liu, "Key Frame Extraction of Online Video Based on Optimized Frame Difference". 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012: 1238-1242. https://doi.org/10.1109/ICCT.2012.6511333.

[11] Aju Sony, Kavya Ajith, Keerthi Thomas, Tijo Thomas, Oeepa P. L., "Video Summarization by Clustering Using Euclidean Distance". Proc. International Conference on Signal Processing, Communication, Computing and Networking Technologies, 2011: 642-646. https://doi.org/10.1109/ICSCCN.2011.6024630.

[12] Anastasios D. Doulamis, Nikolaos D. Doulamis and Stefanos D. Kollias" Efficient Video Summarization Based on A Fuzzy Video Content Representation". IEEE International Symposium on Circuit and Systems,2000:301-304.

[13] Suresh C Raikwar, Charul Bhatnagar and Anand Singh Jalal, "A frame work for key-frame extraction from surveillance Video", 5th International Conference on Computer and Communication Technology", IEEE, 2014, p. 297-300. https://doi.org/10.1109/ICCCT.2014.7001508.

[14] Zhao et al. "Key-frame extraction and shot retrieval using nearest feature line",Proceedings of ACM Workshop on Multimedia, 2000,p. 217- 220. https://doi.org/10.1145/357744.357942.

[15] Mukhargee et al., "Key-frame estimation in video using randomness measure of feature point pattern", IEEE transactions on circuits on systems for video technology, vol.7, no.5, May 2007, p. 612-620. https://doi.org/10.1109/TCSVT.2007.895353.

[16] Zhuang Y, Rui Y, Huang T.S and Mehvotra S, “Adaptive key-frame extraction using unsupervised clustering”, Proceedings of International conference on Image Processing, 1998, p 866-870.
[17] Gang Y and Liu, “Video summarization using singular value decomposition”, Proceedings of Computer Vision and Pattern Recognition, 2000, p 347-358.
[18] J. Peng and Q. Xiao-Lin, Keyframe-Based Video Summary using Visual Attention Clues, IEEE MultiMedia, (2), pp. 64–73, (2009). https://doi.org/10.1109/MMUL.2009.65.
[19] H. Liu and T. Li, Key Frame Extraction based on Improved Frame Blocks Features and Second Extraction, In IEEE 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 1950–1955, (2015).
[20] Yogamangalam, R., & Karthikeyan, B. (2013). Segmentation techniques comparison in image processing. *International Journal of Engineering and Technology (IJET)*, *5*(1), 307-313.
[21] A A, M., & G.R Sathiaseelan, J. (2018). Contrast Enhancement of Grayscale and Color images using Adaptive Techniques. *International Journal of Engineering & Technology*, *7*(2.22), 1-4. https://doi.org/10.14419/ijet.v7i2.22.11798.