International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 37, No. 5, October, 2020, pp. 773-784
Journal homepage: http://iieta.org/journals/ts

Key-Frame Detection and Video Retrieval Based on DC Coefficient-Based Cosine Orthogonality and Multivariate Statistical Tests

Gowrisankar Kalakoti[1,2*], Prabakaran G[1]

[1] Department of Computer Science and Engineering, Annamalai University, Chidambaram 608002, Tamil Nadu, India
[2] Department of Information Technology, RVR & JC College of Engineering, Chowdavaram 522019, Guntur, Andhra Pradesh, India

Corresponding Author Email: gowrisankar508@gmail.com

https://doi.org/10.18280/ts.370509

Received: 14 June 2020
Accepted: 12 October 2020

Keywords:
key-frame, background scenes, forefront objects, DC-coefficients, cosine orthogonality test, multivariate statistical parametric test

ABSTRACT

This paper presents a method, which is developed based on the Discrete Cosine (DC) coefficient and multivariate parametric statistical tests, such as tests for equality of mean vectors and the covariance matrices. Background scenes and forefront objects are separated from the key-frame, and the salient features, such as colour and Gabor texture, are extracted from the background and forefront components. The extracted features are formulated as a feature vector. The feature vector is compared to that of the feature vector database, based on the statistical tests. First, the feature vectors are compared with respect to covariance. If the feature vector of the key-frame and the feature vector of the feature vector database pass the test, then the test for equality of mean vector is performed; otherwise, the testing process is stopped. If the feature vectors pass both tests, then it is inferred that the query key-frame represents the target video in the video database. Otherwise, it is concluded that the query key-frame not representing the video; and the proposed system takes the next feature vector for matching. The proposed method results in an average retrieval rate of 97.232%, 96.540%, and 96.641% for CC_WEB, UCF101, and our newly constructed database, respectively. Further, the mAP scores computed for each video datasets, which resulted in 0.807, 0.812, and 0.814 for CC_WEB, UCF101, and our newly constructed database, respectively. The output results obtained by the proposed method are comparable to the existing methods.

1. INTRODUCTION

The advent of the cutting-edge technology in computer vision, notably, the development of multimedia contents and their storage devices zoomed the multimedia data on the internet as well as in off-line. The increasing of storage and the complications of the multimedia data makes difficulties in handling the data, especially the video-related information retrieval. Analysis and understanding of video sequences is an active research field because many applications in this research area, such as video surveillance (heavy traffic at signals, tolls, etc.), optical motion capture, multimedia applications (videos games, cinema video and movie retrieval), need in the first step to detect the moving objects in the scene. So, the basic operation needed is the separation of the moving objects, called foreground, from the static information, called the background, are a challenging task. Hence, the key-frame detection, background scenes and foreground object detections play a significant role in video retrieval or analyses.

Though many works have been developed to address these problems, still, they do not fulfil the requirements. Identifying a particular object in a video with dynamic background scenes, specifically, vehicles which pass in over speed, cinema movies shots taken in natural scenes, and so on. Also, it is a challenging task to identify the objects with similar colour and patterns of the static background scenes of a video. Many works have been developed to cope this kind of difficulties, which initialize the background [1-4], subtracts background scenes [5-7], and forefront detection [8-10], that supports for video retrieval and detection of moving objects. The background subtraction, background initialization, foreground segmentation, and forefront object detection play a noteworthy role in video analysis like retrieval, summarization, classification, moving object detection and so on.

The background subtraction methods can be broadly classified into (i) Statistical, (ii) fuzzy, (iii) Dempster–Schafer (iv) classification, (v) signal processing model, and (vi) machine learning models. The Statistical, Fuzzy and Dempster–Schafer models are very useful to effectively handle the imprecision, uncertainty and incompleteness of the data owing to different situations while the Machine Learning models represent the background pixels with supervised or unsupervised methods. The Classification models classify the pixels into either background or foreground categories while the Signal processing models compute the background values. The background initialization deals with video inpainting, privacy protection, and computational photography.

Motion detection is noteworthy as well as it is a challenging task for low-level processing in computer vision, especially, in video analyses. The main objective of the motion detection is to extract the moving objects from a video sequence. It is dealt with three different approaches: (i) time difference, (ii) background subtraction, and (iii) optical flow analysis. The time difference approach calculates the time taken between two or more consecutive video frames [11], but there arises a problem that the detected objects might be incomplete and poorly presented. The background subtraction method builds a model for the static scenes, called background, then compare each image sequence with this model, in order to distinguish the regions of unusual movement, called foreground or moving objects. The optical flow method [12] calculates the optical flow, which gives detailed information about the moving objects. It is observed from the literature that object detection mostly relies on background scenes because the background scenes may be either a static or a dynamic. The object detection is a challenging process in both static and dynamic background contexts. For example, the foreground objects captured by a surveillance camera could be similar to the background in terms of colour and texture pattern so that it causes some challenges in detecting the objects from video sequences. Likewise, the same problem might arise in the context of dynamic background; for instance, the video captured on travel with the background of natural scenes, and sea waves as the background scenes.

In order to detect the forefront objects from a video, many works have attempted to model the background, based on either mathematical or statistical concepts. A piece of literature has been reported here, for instance, the background scene is modelled with basic descriptive statistics, median [13], mean [14], and histogram [15]; statistical model-based approaches like Gaussian models [16, 17]; support vector model [18]; sequential cluster model [19]; neural network models-based methods [20-24]; Bayesian approach-based models [25, 26]; and Transform domain-based models [27-30].

Muselet and Macaire [31] have presented a method, based on chromatic co-occurrence matrices, which combines the colour and spatial information to detect objects under different illumination conditions. Further, they compute a pair of adapted co-occurrence matrices. One is derived from the combinations of the query frame and any one of the target video frames at a comparable level, and the other one is derived from the pairs of chromatic co-occurrence matrices; they report that the similarity measure is high when the two images are similar than they are different. They have used the histogram intersection method to recognize the objects of interest. The structure and texture components are decomposed, and the background scene is modelled using the median filter; the absolute difference is deployed to subtract the background scenes [32]. They deploy an adaptive threshold that computes the maximum variance difference between the classes. Varadharajan et al. [33] have proposed a region-based foreground detection model, based on the mixture of Gaussian model, which generates the background scenes. They have applied the expectation-maximization and stochastic approximation methods that simultaneously detect the foreground and subtracts the background. Sobral and Zahzah [34] have introduced a method, which generates a background model by reconstructing the missing entries from neighbouring pixels; the missing entries are induced from moving regions through a simple joint motion-detection and frame-selection process. Ramirez-Alonso et al. [35] present a Background Estimation and Auto-Adaptive Parallel Self Organized Maps Architecture (BE-AAPSA) method, which automatically models the background if the background initialization and update need to be reinitialized. The re-initialization is activated while the video scene has high variations, and allows the background to be defined with precision. de Geus et al. [36] have adopted a global and a local threshold for foreground detection and background subtraction.

The background subtraction is performed by a binarization process, which uses the global and local thresholds; the details of the moving object is attained by adding the results of the local threshold to the global threshold in order to combine the binarization results. The same importance has to be given for feature extraction and matching as given to the background subtraction and foreground detection because the feature extraction and matching play a significant role in video retrieval. The features characterize the videos while the matching process accurately measures the similarity of the query and reference videos. Many researchers [37] have developed feature extraction methods and similarity measures for video/image retrieval.

The remaining part of the paper is organized as follows. In Section 2, a piece of related literature has been presented. Section 3 describes the proposed method, which discusses the video denoising, shot boundary and key-frame detection, background and foreground detection, feature vector database construction, and feature matching. Section 4 illustrates the experimental results and the performance of the proposed method. In Section 5, the paper is concluded with a conclusion and future direction of the proposed method.

2. RELATED WORKS

Anjulan and Canagarajah [37] have proposed a method to retrieve videos based on local invariant region descriptors and objects. They claim that the proposed approach gives better results than the state-of-the-art methods and report that it is robust to the camera, object movements, severe illumination changes, and spatial editing. The co-occurrence matrix is computed, based on point-wise mutual information using data collected by information retrieval (PMI-IR), and Earth Mover's Distance similarity measure is employed to match and retrieve the videos [38]. Shang et al. [39] have proposed a method, which extracts conditional probability-based entropy and LBP-based spatiotemporal features. The extracted features are compared using the histogram intersection similarity method for video retrieval. In 2012, Andrade et al. [40] have presented a method, which fuses the local and global descriptors, such as colour, texture, that are encoded through image and video encoders, and applies tree-based Genetic Programming similarity framework for video retrieval. They reported that the fusion of local and global descriptors yields good results. Adami et al. [41] presented a method, which tracks the spatial attributes and the long-term motion of local regions in videos shots with dynamic backgrounds using LIFT method and constructs a Bag-of-Spatiotemporal-Words (BoSW) model for video retrieval. They have used the histogram method to match the features and retrieve the videos. Berg et al. [42] have proposed a method and got patent for multimedia retrieval, which extracts features – spatial and temporal – from each frame of both digital video and audio data; they use cross-correlation analysis for matching spatial-temporal features while performing direct bit-wise comparison for spatial frame features. They claim that the proposed technique is invariant for retrieving videos of different formats. Mironica et al. [43] have proposed a relevance feedback method, based on Gaussian mixture model-driven Fisher Kernel (FK) function, for video retrieval. They extract local spatiotemporal features and train the SVM classifier with FK function on the top retrieval results; the FK function captures the temporal variation using frame-based features. Further, they have examined the performance of similarity metrics: Euclidean, Manhattan, Canberra, Cosine Distance, Chi-Square distance, Bray Curtis, Mahalanobis, Kullback-Leibler divergence, and Earth Mover's distance; finally, they have reported that the Euclidean distance results in better than the others. Mohamadzadeh and Farsi [44] have presented a method for shot-boundary detection and key-frame extraction; the boundary of a shot is computed using Euclidean distance, Cosine distance, and histogram bins. The key-frames are extracted, based on motion features, from YUV colour space. The shot-boundary is detected using the RGB colour model. Further, they extract texture, based on the Hadamard matrix and Discrete Wavelet Transform (HDWT), and use the Euclidean distance to compare and retrieve the videos. Hao et al. [45] have proposed a method based on the stochastic multi-view hashing algorithm for near-duplicate video retrieval. They extract colour-histogram-based global features and LBP-based texture features from each key-frame; the Kullback-Leibler (KL) divergence measure is applied to match the frames and retrieve the videos.

In 2017, Lou et al. [46] have proposed a Nested InvariancePooling (NIP) method, which derives compact and robust Convolutional Neural Network (CNNs) descriptors. The CNNs descriptors are attained by deploying three different pooling operations, such as square root pooling, average pooling, max-pooling, for feature mapping of CNNs in a nested method that are robust for rotation and scaling. Kordopatis-Zilos et al. [47] have presented a Near-Duplicate Video Retrieval scheme, based on deep metric learning, which learns features at intermediate layer and generates discriminative global video representations with two fusion variations. It is trained to approximate an embedded function for calculating an accurate distance between two near-duplicate videos. Dong et al. [48] have proposed a method, which finds the sentence describing the content of a video in a visual space. Further, they have introduced a 'Word2VisualVec', called a deep neural network architecture, which learns features from textual input and predicts a visual feature representation. Liu and Sui [49] have introduced a method, based on the combination of AlexNet network model with CAFFE deep learning framework that extracts features of the public cultural videos and applies principle component analysis method for feature dimensionality reduction. Song et al. [50] report that the frame pooling, relaxed learning, and the binarization are not sufficiently examining the temporal order of video frames in a joint binary optimization model, which results in severe information loss. To overcome this problem, they have presented a Self-Supervised Video Hashing method that simultaneously encodes the video temporal and visual information using an end-to-end hierarchical binary auto-encoder and a neighbourhood structure. Wu et al. [51] have proposed a deep hashing method, called Unsupervised Deep Video Hashing, for large-scale video similarity search to learn compact and more effective binary codes. Moreover, they claim that the proposed method differs from the existing techniques in terms of (i) organizing the hash code learning in a self-taught manner; (ii) minimizes the quantization error of projecting video features to a binary hypercube; (iii) the feature clustering in the code learning enables the neighbourhood structure to be preserved; they stated that the proposed method outperforms the state-of-the-art methods. Nie et al. [52] have reported that multi-view hashing has two limitations: (i) considers local structures in multiple features and ignores the global structure; (ii) always learns the hashing functions bit by bit, which demands high computational time for hash function learning. To address these problems, they have proposed a joint multi-view supervised hashing scheme that simultaneously learns the local and global structures. They, also stated that the proposed method results in more than 5% improvement compared to the existing methods.

It is observed from the literature that most of the works have used histogram-based feature extraction and video retrieval; many researchers have adopted LBP-based texture features with distance metrics like Euclidean, Earth Movers, Cosine similarity, Manhattan, and KL distance for video retrieval. The LBP-based features are not rational, and a detailed discussion can be found in the study [53]. The histogram feature is a global representation of the frames/images. Also, most of the above-said distance metrics deal with a single one-dimensional vector; it is difficult to measure the distance between the multi-dimensional vectors using the above metrics. Furthermore, the recent works have been developed based on the deep neural networks; though it maintains high precision, which demands high computational effort. In some times, some application domains require fast accessing and retrieval of information than spend more time on maintaining high accuracy. For instance, in some situations, it is detecting an object that wrongly or illegally passes the level-crossing at the traffic signal; detecting an object crossing the border of a country through forest or hills; identifying the air-force flights or missiles flying another country's border without permission and so on. With a view of these, we have developed a video retrieval scheme which acts trade-off between accuracy and computation time complexity. The proposed approach is discussed in the following sections.

2.1 Outline of the proposed approach

Firstly, the proposed method denoising the given input query key-frame using weighted median filter, which is discussed in Section 3.1. Background scenes and foreground objects are separated from the denoised key-frame, as illustrated in Section 3.3. The background scenes and foreground objects are divided into various blocks of size, 8×8; colour and Gabor texture features are extracted from each block. The extracted features are formulated as a feature vector as depicted in Eq. (10). The feature vector of the query key-frame is compared with the feature vectors of the feature vector database, based on the multivariate statistical parametric tests. If the feature vectors of the query and target frames pass the test for equality of covariance, then one can partly be inferred that the query and target frames have been drawn from the same video. Otherwise, the testing process is dropped and takes the next feature vector of the database. If it passes the test, then one can proceed to perform the test for equality of mean vectors. If the feature vectors pass both tests, then one can conclude that the two frames have been drawn from the same video, and the video is identified as targeted video.

Based on the acceptance of the null hypothesis of the test for equality of mean vectors, one cannot decide that the two sample groups have been drawn from the same population because the mean vectors of the two sample groups may be identical. In contrast, the covariance of the two sample groups may differ. Therefore, it necessitates testing the equality of covariance first, and then the equality of mean vectors. Statistically, this is the proper procedure to compare two sample groups, whether those have been drawn from a population or not. It is the main reason behind performing the equality of covariance first, and then the mean vectors. The overall procedure involved in developing the proposed method has been diagrammatically demonstrated in Figure 1.

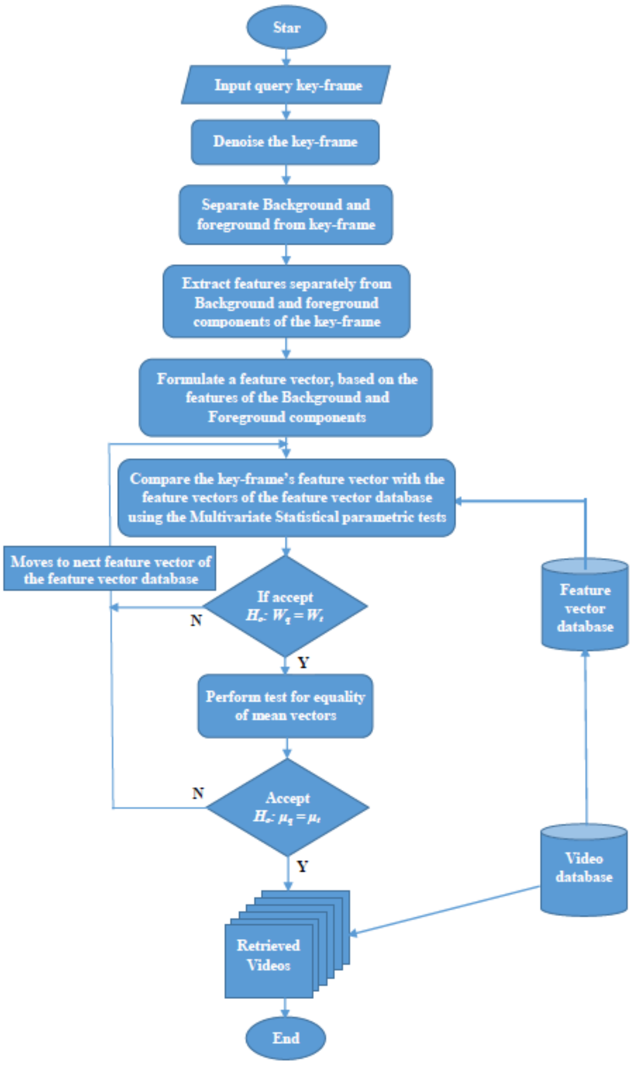
Star
Input query key-frame
Denoise the key-frame
Separate Background and foreground from key-frame
Extract features separately from Background and foreground components of the key-frame
Formulate a feature vector, based on the features of the Background and Foreground components
Compare the key-frame's feature vector with the feature vectors of the feature vector database using the Multivariate Statistical parametric tests
Moves to next feature vector of the feature vector database
If accept
H_o: $W_q = W_t$
N
Y
Feature vector database
Perform test for equality of mean vectors
Accept
H_o: $\mu_q = \mu_t$
N
Y
Video database
Retrieved Videos
End

Figure 1. Outline of the proposed video retrieval method

3. PROPOSED METHOD

3.1 Pre-processing

There are many possibilities of incorporating Gaussian noise in the videos while capturing in outdoor. So, it necessitates denoising the videos. In order to remove the noise, we deploy a weighted median filter which is more effective to remove the Gaussian noise.

$$W_{ij}= \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 3 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \qquad I_{ij}= \begin{array}{|c|c|c|}\hline 138 & 167 & 114 \\ \hline 118 & 135 & 113 \\ \hline 116 & 126 & 123 \\ \hline \end{array}$$

(a) (b)

Figure 2. (a) Weighted median filter; (b) Sub-image

The weighted median filter is better than the average filter because it is a non-linear. The speciality of the weighted median filter is that it is taking into the account of local spatiotemporal contents. Thus, it is more appropriate for the proposed video retrieval scheme in this paper. In this section, let us discuss it through a sub-image, I_{ij} (i=j=3), of size 3×3. The Weighted Median Filter assigns weights to the filter position as in the mask is presented in Figure 2. Insert each pixel within filter region W_{ij} times into extended pixel vector, EI_{ij}; and the extended vector is sorted in ascending order.

EI_{ij} = [138, 167, 167, 114,118, 118, 135, 135, 135, 113, 113, 116, 126, 126, 123]

The above vector is sorted in ascending order, and the mid-value, i.e. the 8-th element, which is highlighted with bold-face, is chosen as median value.

EI_{ij} = [113, 113, 114, 116, 118, 123, 126, 126, 135, 135, 135, 138, 167,167]

	126	

(a)

126	126	114
118	126	113
116	126	123

(b)

138	167	126
126	135	126
126	126	126

(c)

Figure 3. (a) Median Filter; (b) Low-pass Filter; (c) High-pass Filter

Figure 3 shows the filter value, 126, is higher than the simple median filter value, 123.It is observed from the result that the weighted median filter, both low-pass and high-pass filters, augments the brightness and significant feature of the image; and it sharpens local contents.

The filter value, 126, is compared to the pixels in the window. The low-pass filter is performed by replacing the pixel value with 126 if it is greater than the weighted median value, 126. Otherwise, the pixel value is maintained as it is.

In the case of the high-pass filter, the value that is less than the filter value is replaced by 126.

3.2 Shot boundary and key-frame detection

In video retrieval, it is time-consuming to process, such as feature extraction, matching, and retrieval, all the frames of a video, instead one can identify a shot and then a frame, called key-frame, can be selected from the shot. The key-frame is a comprehensive representation of the shot. One can considerably reduce the computational complexity by retrieving the video based on the feature extracted from key-frame. In literature, most existing works use low-level features such as colour, shape, size, texture, and spatial information [54, 55] that are extracted directly from the raw video. However, nowadays, almost all types of videos or multimedia data are available in the MPEG compressed format; mostly, they have been compressed using a discrete cosine technique. Therefore, the proposed work deploys the cosine transform and forms a feature vector based on the DC coefficients; then computes the cosine direction angle between the DC coefficients of the consecutive two frames of the video. By using the DC coefficient, it can be used for both compressed and uncompressed videos; in the case of compressed video, it reduces the computational time more than half of the time. If the cosine direction is higher than 60 degree, then it is assumed that the two frames are different and fixed as the starting point of the shot; two frames are different means the change of scenes.

Similarly, the endpoint of the shot is identified. Based on starting and ending frames, the shot is identified. The frames at starting and ending points are treated as key-frames of the shot. The cosine transform-based vector formulation and the cosine direction angle are expounded in the following section.

The computation of DC coefficients is performed in two different ways: (i) computes uncompressed videos; (ii) compressed videos.

In the case of uncompressed videos, the given input query-frame is divided into various sub-images of size, 8×8, and the cosine transform expressed in Eq. (1) is applied to each sub-image. The DC-based feature vector is formulated as in Eq. (2).

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1)$$

such that u, v = 0,1, …, N-1

where, $\alpha(u) = \begin{cases} \frac{1}{2} & \text{for } u = 0 \\ \sqrt{\frac{N}{2}} & \text{for } u = 1,2,\dots,N-1 \end{cases}$ and *x,y=0,1,…,N– 1.*

In the case of compressed videos, first, the compressed data are decoded until attaining the DC coefficients. After attaining the DC coefficients, the feature vector is formulated as depicted in Eq. (2).

$$FV_{kf} = \left(\overrightarrow{DC_i}\right), s.t. i = 1,2,\dots,n \quad (2)$$

where, *i* denotes the number of sub-images in the frame.

The DC-based features of the two consecutive frames are tested using the cosine-based orthogonality test, which is expressed in Eq. (3).

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{DC}^{pf} \cdot \overrightarrow{DC}^{cf}}{\left|\overrightarrow{DC}^{pf}\right| \cdot \left|\overrightarrow{DC}^{cf}\right|}\right) \quad (3)$$

where, $\overrightarrow{DC}^{pf}$ and $\overrightarrow{DC}^{cf}$ represent the feature vector of the previous frame and the current frame. The angle, *θ* represents the similarity of the two consecutive frames. After conducting rigorous experiments, *θ* is fixed at 45 degree, which is an optimal threshold value. If *θ*>45°, then the frame is regarded as a key-frame; otherwise, it is treated as similar to the previous frames and represents the same shot.

For example, two shots and their frames have been illustrated in Figure 4. The Frame-1 and Frame-8 are regarded as the key-frame of the Shot-1, and the Frame-1 and Frame-4 of the Shot-2 are regarded as the key-frames. The computed θ value between Frame-8 of the Shot-1 and Frame-1 of the Shot-2 is 78 degree; the extracted key-frames were stored in an archive. Features are derived from the key-frames and stored in the feature vector database. The feature extraction and its formulation are discussed in the following section.

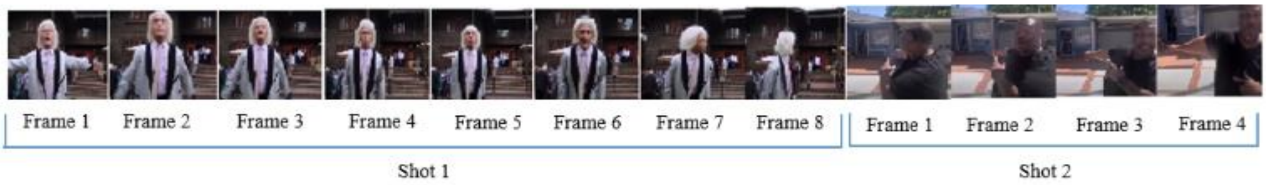

Figure 4. Sample key-frames and shots

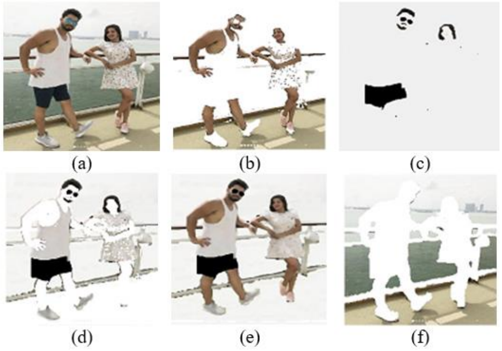

Figure 5. (a): Actual image; (b): red and red-oriented regions; (c): blue and blue-oriented regions; (d): green-blue oriented regions; (e): forefront objects; (f): background scenes

3.3 Background and foreground detection

In order to extract the features, the background scenes and forefront objects are segmented. To achieve the segmentation of forefront objects from the background scenes, in this paper, a colour intensity-based algorithm has been developed based on the expression in Eq. (4). First, the red and red-oriented; green and green-oriented; blue and blue-oriented regions are segregated. Next, the forefront objects are fused, by which the foreground objects are formulated. Now, the forefront objects are subtracted from the actual input image, by which the background scenes are derived. Finally, the background and forefront objects are separated. Figure 5 illustrates the segmentation process.

$$\text{Dominant Colour } (D_{col}) = \begin{cases} \left(2K_i - K_j - K_k\right)/2 > 8: \ K_i \text{ is the dominant color} \\ \left(2K_i - K_j - K_k\right)/2 \le 8: \text{otherwise} \\ \qquad \forall\, i,j,k; \text{and } i \ne j \ne k; \text{s.t. } i,j,k \in \{R,G,B\} \end{cases} \quad (4)$$

3.4 Feature space formulation

In order to retrieve the target video, the low-level features, such as colour and texture, are extracted from the background scenes and forefront objects. Li et al. [56] have conducted a comparative study of nine different colour models and reported that the HSV and YCbCr models result in better segmentation than the others. Wang et al. [50] have adopted the YCbCr model as it has excellent segregation characteristics of the forefront and background scenes, and they have reported that it demands minimal computational time than the HSV colour model. Thus, this paper deploys the YCbCr colour model feature extraction. Both background scene and forefront objects are converted to YCbCr colour model from the RGB model. The Cb and Cr colour components of both background scenes and forefront objects are taken into the account of features, and the texture feature is extracted from the Y component using the Gabor wavelet filter. The texture feature extraction is described below. The Gabor wavelet kernel function, which is a product of elliptical Gaussian derivative and a complex plane wave, is expressed in Eq. (5).

$$\xi_{\upsilon,\lambda}(z)=\frac{\|g_{\upsilon,\lambda}\|^2}{\sigma_2}\exp\left(\frac{\|g_{\upsilon,\lambda}\|^2\|z\|^2}{2\sigma^2}\right)\times\left[\exp\left(ig_{\upsilon,\lambda}z\right)-\exp\left(-\frac{\sigma^2}{2}\right)\right] \tag{5}$$

where, υ and λ represent the orientation and scale of the Gabor kernel function; ‖·‖ is the normalizing operator; $g_{\upsilon,\lambda}=k_v\,exp(i\varphi_v)$ with $\varphi_v=\pi v/8$ and k_v=$k_{\max}/\gamma^v$, $k_{\max}$ is the maximum frequency and γ denotes the spacing factor between kernels in the frequency domain.

The Gabor wavelet representation of the query video frame is derived by convolving the query frame with the Gabor kernel function as follows.

$$GW_{\upsilon,\lambda}(z)=qf(z)*\xi_{\upsilon,\lambda}(z) \tag{6}$$

where, $GW_{\upsilon,\lambda}(z)$ is the result of the above convolution function corresponding to the Gabor kernel at orientation υ, and scale λ; z=(k, l) represents the pixel location; * is the convolution operator. The results, $GW_{\upsilon,\lambda}(z)$, of Eq. (6) is a complex valued that represents real and imaginary parts. The real and imaginary parts are given by $R\big(GW_{\upsilon,\lambda}(z)\big)$ and $I\big(GW_{\upsilon,\lambda}(z)\big)$, respectively. The $GW_{\upsilon,\lambda}(z)$ can be written a follows,

$$GW_{\upsilon,\lambda}(z)=C_{\upsilon,\lambda}(z)exp\big(i\theta_{\upsilon,\lambda}(z)\big) \tag{7}$$

where, $C_{\upsilon,\lambda}(z)=\sqrt{R\big(GW_{\upsilon,\lambda}(z)\big)^2+I\big(GW_{\upsilon,\lambda}(z)\big)^2}$ and $\theta_{\upsilon,\lambda}(z)\;arctan\left(\frac{I(GW_{\upsilon,\lambda}(z))}{R(GW_{\upsilon,\lambda}(z))}\right)$. The real part of the Gabor wavelet performs a smoothing process while the imaginary part results the edge components. The magnitude, $C_{\upsilon,\lambda}(z)$represents the complementary information provided by $R\big(GW_{\upsilon,\lambda}(z)\big)$ and $I\big(GW_{\upsilon,\lambda}(z)\big)$ which is regarded as a stable and discriminative features [57, 58]. If we take the magnitude of all scales and orientation, and formulate the feature vector, it leads to high dimension, so that it is better to consider only the maximum values as follows [56].

$$FV=max\big(C_{\upsilon,\lambda}(z)\big). \tag{8}$$

The Gabor feature, called texture feature, is given in Eq. (9) for each pixel and the scale value of the query frame.

$$FV^T=\left(x_1,\dots x_{MN}\right) \tag{9}$$

The colour chromatic and Gabor texture features are extracted from both background scenes and forefront objects of the query frame. The extracted features are formulated as a feature vector as in Eq. (10).

$$\overrightarrow{FV}^q=\left(Cb_i^{bg},Cr_i^{bg},GF_i^{bg},Cb_j^{fo},Cr_j^{fo},GF_j^{fo}\right),\quad s.t.\,i=1,\dots n;\ j=1,\dots m. \tag{10}$$

where, FV^q stands for feature vector of the query frame; i and j represent the number of feature elements of each feature vector of the background scenes and forefront objects, respectively; $(\cdot)^{bg}$ and $(\cdot)^{fo}$ represent the features of the background and forefront objects. The size of the feature elements (sample size) might vary between the background and forefront.

Similarly, the colour chromatic and Gabor features are derived for target video frames.

3.5 Feature vector database construction

3.5.1 Video dataset

In order to validate the proposed video retrieval method, in this study, two benchmark video datasets, such as CC_WEB and UCF101, were subjected to the experiments. In addition to that, we have built a video dataset, which contains 298 videos with 1358 clips collected from online resources, namely YouTube and Metcalfe and from some movies that cover various scenarios like sports, films, advertisements, etc.

A video [59] with execution time 1 hour 43 minutes and 16 seconds (1:43:16) was divided into 40 scenes, which demonstrates the latest product and platform innovations of Google in a Keynote (Google I/O'15) press event [60], Google I/O'19 [60]) led by Sundar and his team members. Each scene was treated as a short video since they have taken time to run the scenes about 1 minute and 50 seconds (1:50) to 3 minutes 25 seconds (3:25). Boundaries identified from the short videos and key-frames were detected from each boundary. The detected key-frames have been presented in Figure 8. These short videos have also been incorporated to the new video dataset built in this paper.

Video feature vector database. Generally, there are possibilities of repeating/duplicating some scenes in a video, from other videos, owing to some referential purpose (for example, in cinema movies and serial movies). Therefore, this paper identifies the number of scenes and shots in a video and detects the key-frames for each shot, based on the methods discussed in Section 3.2. The features are extracted from each key-frame, as discussed in the previous sections. The extracted features are formulated as a feature vector as depicted in Eq. (10), which is denoted by $\overrightarrow{FV}^q$. The feature vectors are clustered into different clusters with homogeneous groups, based on the *fuzzy weighted medoids* algorithm [61].

Also, a median value is computed for each feature vector, denoted by $med\big(\overrightarrow{FV}^q\big)$, which is treated as an index of the feature-vector. A link has been established from the $med\big(\overrightarrow{FV}^q\big)$ to key-frame; and the key-frame to shot-boundary and shot-boundary to video. Now, the feature vectors of the key-frames belonging to the video database formulated as a comprehensive feature vector database, which is denoted by $\overrightarrow{FV}_{db}$. A schematic structure of the linkage between the feature-vector database and the videos in the video archive is given below. By which, the computational time complexity is considerably reduced.

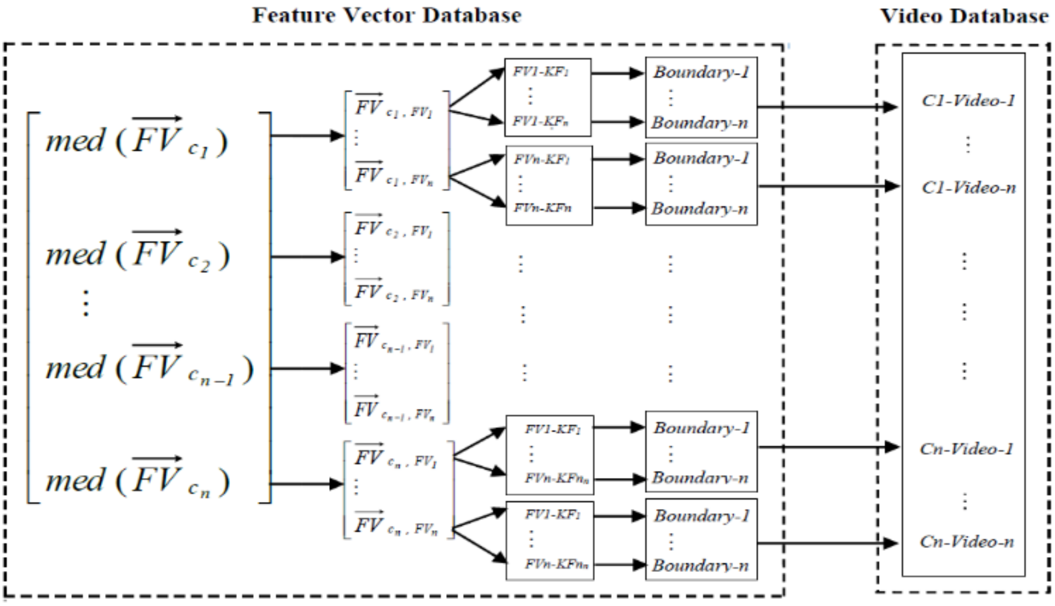
Feature Vector Database
Video Database
$med(\overrightarrow{FV}_{c_1})$
$med(\overrightarrow{FV}_{c_2})$
$med(\overrightarrow{FV}_{c_{n-1}})$
$med(\overrightarrow{FV}_{c_n})$
$\overrightarrow{FV}_{c_1, FV_1}$
$\overrightarrow{FV}_{c_1, FV_n}$
$\overrightarrow{FV}_{c_2, FV_1}$
$\overrightarrow{FV}_{c_2, FV_n}$
$\overrightarrow{FV}_{c_{n-1}, FV_1}$
$\overrightarrow{FV}_{c_{n-1}, FV_n}$
$\overrightarrow{FV}_{c_n, FV_1}$
$\overrightarrow{FV}_{c_n, FV_n}$
FV1-KF1
FV1-KFn
FVn-KF1
FVn-KFn
FVn-KFnn
Boundary-1
Boundary-n
C1-Video-1
C1-Video-n
Cn-Video-1
Cn-Video-n

Figure 6. Schematic representation of the feature vector database

3.5.2 Feature matching method

In order to match and retrieve the targeted video, the features of the query key-frame are compared to the feature vector of the key-frames of the targeted video using the multivariate statistical tests, such as the test for equality of covariance and the test for equality of mean vectors. To examine whether two multivariate samples have been drawn from the same populations; first, one has to test whether the covariance matrices of the two samples are equal or not. If the covariance matrices are equal, then it can be proceeded to test whether the mean vectors of the two samples are similar or not. If the samples pass the two tests, then it can be inferred that the samples have been drawn from the same population; otherwise, it can be assumed that the samples have been drawn from different populations. Hence, this paper, first, tests the covariance matrices of feature vectors of the query key-frame to the key-frames of the targeted videos; if the query key-frame passes the test, then it can be proceeded to test the equality of mean vectors. Otherwise, the test is dropped and takes the next feature vector of the feature vector database. The Schematic representation of the feature vector database as shown in Figure 6.

Test for equality of covariance between key-frames. Let the feature vector spaces of the query and target videos are assumed to be Gaussian random process. In the proposed work, there are only two sample key-frames, such as the query key-frame and the target key-frame, and each sample comprises six components as illustrated in Eq. (10). In this study, the goal is to test the hypothesis, H_0:W_q=W_t and the alternative hypothesis, H_a:W_q≠W_t; where W_q and W_t represent the covariance matrices of the query-frame and the key-frame of the target video. The multivariate statistical test for equality of covariance matrices of two sample points (p) with six characteristics (q) is defined as follows.

$$\Omega\psi^{-1} = \left(\left[(N_q-1)+(N_t-1)\right]\ln\left|S_q\right|\right) - \left(\left[(N_q-1)\ln(S_q)+(N_t-1)\ln\left|S_t\right|\right]\right)\psi^{-1} \qquad (11)$$

is approximately distributed to chi-square distribution with degrees of freedom (v), 1/2(q-1)p(p+1) [62], where

$$\psi^{-1} = 1 - \frac{2p^2+3p-1}{6(p+1)(q-1)}\left(\sum_{g=1}^{q}\frac{1}{n_g} - \frac{1}{\Sigma n_g}\right) \qquad (12)$$

The pooled estimate of the sample covariance matrix ***S*** can be derived as follows,

$$S = \left(\frac{1}{(N_q-1)+(N_t-1)}\right)\left((N_q-1)S_q+(N_q-1)S_t\right) \qquad (13)$$

If $\Omega\psi^{-1} > \chi^2_v(\alpha)$, then the null hypothesis H_o may be accepted, that is, the query and target key-frames have been drawn from the same population. If $\Omega\psi^{-1} > \chi^2_v(\alpha)$, then the null hypothesis H_o may be rejected, that is, the query and target key-frames have been drawn from different populations, where, v=1/2(q-1)p(p+1), and α is the level of significance. The significance level statistically means the probability of accepting the similarity of the query key-frame and the key-frames of the targeted videos, viz. it is the threshold value, by which one can decide the similarity of the key-frames.

Test for equality of mean vectors. The aim of this section is for testing the equality of the spectrum of the energy of feature vectors of the query key-frame and the targeted key-frame, i.e., average values of feature vectors. A test of hypothesis is framed to achieve the goal, which is demonstrated in Eq. (14).

$$\begin{aligned} &H_0 : \mu_q = \mu_t \quad \text{Null hypothesis} \\ &H_a : \mu_q \neq \mu_t \quad \text{Alternative hypothesis} \end{aligned} \qquad (14)$$

The above hypothetical test is performed based on the multivariate test statistic, namely test for equality of mean vectors, which is illustrated in Eq. (15).

$$t^2 = \left(\overline{FV}^q - \overline{FV}^t\right)' \left[\left(\frac{1}{N_q}+\frac{1}{N_t}\right)S_{pooled}\right]^{-1}\left(\overline{FV}^q - \overline{FV}^t\right) > T^2 \qquad (15)$$

where, the critical distance, T^2, is determined from the distribution of the two-sample Hoteling's T^2 statistic [63].

$$\mathrm{T}^2 = \frac{(\mathrm{N}_q + \mathrm{N}_t + 2)\mathrm{p}}{(\mathrm{N}_q + \mathrm{N}_q - \mathrm{p} - 1)} \mathrm{F}_{\mathrm{p}, \mathrm{N}_q + \mathrm{N}_t - \mathrm{p} - 1}(\alpha) \tag{16}$$

where, α is the level of significance. The F_{p,n_q+n_t-p-1} is refer to the table of the F distribution with degrees of freedom $p–1$ and $N_q+N_t–p$, and reject the null hypothesis mentioned in (14) at the level of α, if the observed F is greater than the critical value F_{p,n_q+n_t-p-1}.

$$S_{pooled} = \frac{(N_q - 1)S_q + (N_t - 1)S_t}{N_q + N_t - 2} \tag{17}$$

where,

$$S_q = \sum_{i=1}^{N_q}\left(FV_i^q - \overline{FV}^q\right)\left(FV_i^q - \overline{FV}^q\right)' \tag{18}$$

$$S_t = \sum_{i=1}^{N_t}\left(FV_i^t - \overline{FV}^t\right)\left(FV_i^t - \overline{FV}^t\right)' \tag{19}$$

are the sum of product of sample covariance matrices of the query key-frame and the key-frame of the target video, respectively.

$$\overline{FV}^q = \frac{1}{N_q}\sum_{i=1}^{N_q}\overline{FV}_i^q \tag{20}$$

$$\overline{FV}^t = \frac{1}{N_t}\sum_{i=1}^{N_t}\overline{FV}_i^t \tag{21}$$

are the sample mean vectors of the query key-frame and the key-frame of the target video, respectively.

Critical region: The query key-frame and the key-frame of the targetedvideo are belonging to the same video, if $t^2 \leq T^2$, where, T^2 is the upper critical value of the F-distribution with (n_q+n_t-2) degrees of freedom at significance level $\boldsymbol{\alpha}$; otherwise, it is inferred that the two key-frames have been drawn from different videos.

Finally, if the feature vectors of the query key-frame and the key-frame of the targeted video passed both test statistics, such as the test for equality of mean vectors and the test for equality of covariance matrices, then it can be inferred that the two key-frames have been drawn from the same video. Otherwise, it is assumed that they have been drawn from different videos.

4. EXPERIMENTS AND RESULTS

To validate and very the performance of the proposed method, which implemented on a system with 10-th generation Intel Core i5 processor with Windows 10 operating system and through open-source Python CV software.

The video datasets described in Section 3.5.1 were subjected to experiments to validate and verify the proposed video retrieval method. As the outcome results are in video form, it is difficult to present them in this paper. Therefore, for example, we have presented only a few of the query key-frames inputted to the system and the corresponding results obtained were given in Figure 7. The frames in column 1 of Figure 7 were inputted to the system, for which, the system retrieved the same or similar videos; some of the responses, viz., only the noteworthy key-frames of the scenes/shots of the targeted videos have been shown in columns 2–6 of the Figure 7.

First, the proposed system pre-processed the query key-frame, based on the technique discussed in Section 3.1. The background scenes and forefront objects were separated; for a sample, the segmented results have been presented in Figure 5. The feature extraction methods expounded in Section 3.4 deployed on the background scenes and foreground objects separately for feature extraction; the extracted features were formulated as a feature vector, as illustrated in Eq. (10).

The first frame of the first row of Figure 7 fed as input to the proposed system, for which, the system responded an average retrieval rate of 96.85% for correct retrieval, whereas it resulted in 3.15% the wrong retrieval at the level of significance (threshold value) 15%, i.e., α = 0.15 or α = 15%. The first frame in the second row was inputted, the rest of the frames presented in the same row are the key-frames of the retrieved videos when α fixed at the level of significance, 0.15; the system resulted in 97.12% average accuracy retrieval rate. Similarly, the proposed system resulted in average accuracy retrieval rate of 96.89% for the query key-frame in the first column of the third row. The level of significance (threshold) was fixed at 0.15 (i.e., α = 0.15 or α = 15%) after conducting experiments several times, which is the optimal threshold. According to the users' requirements, they can fix the threshold at various levels.

Also, the short videos of the Google I/O'15 and Google I/O'19 subjected to experiments. The proposed system resulted in 97.01% average accuracy of the retrieval for Google I/O'15. For the short videos of Figure 8 the Google I/O'19, it retrieved at an average accuracy rate of 97.19%.

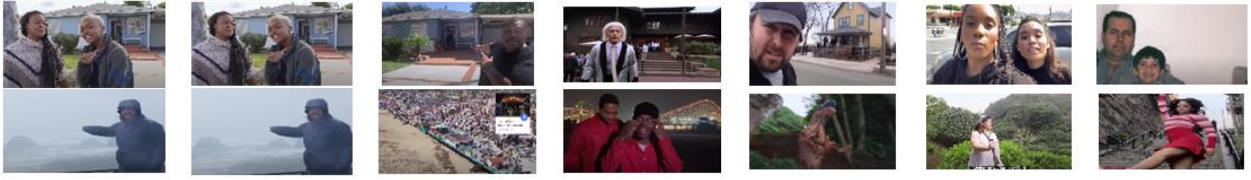

Figure 7. Column 1: query key-frames; column 2-7: retrieved videos; columns 2-6: key-frames of the correctly recognized videos; column 7: key-frames of the wrongly recognized video

Table 1. P@α versus threshold dataset-wise

Dataset	P@α in %									
	1%	5%	10%	15%	20%	25%	30%	35%	40%	Average
CC_WEB	100	99.998	98.294	97.352	97.016	96.589	95.861	95.012	94.964	97.232
UCF101	100	99.999	98.625	97.183	96.579	94.957	94.004	93.861	93.651	96.540
Ours	100	99.999	98.726	97.681	96.982	95.651	94.527	93.679	93.081	96.641

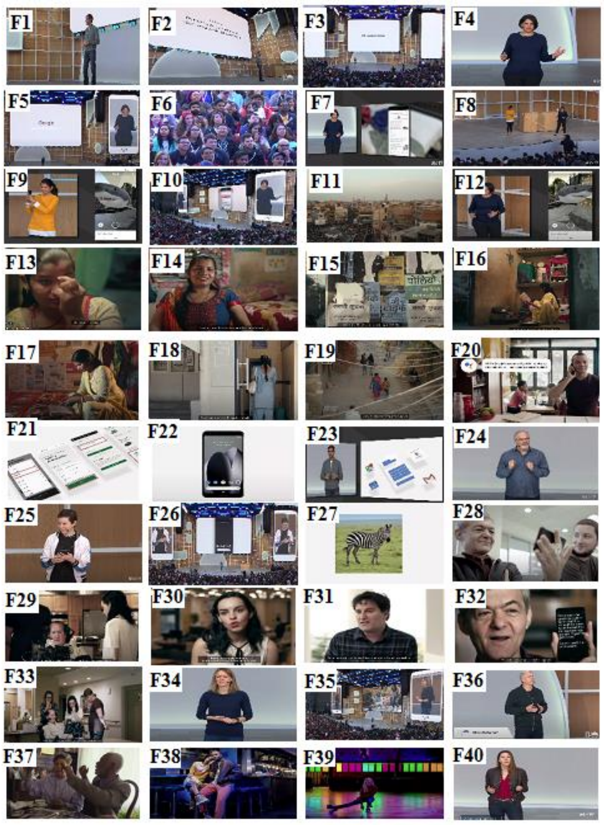

Figure 8. Key-frames detected from Google I/O'19

4.1 Performance measure

In order to evaluate the proposed method, the mean Average Precision (mAP) deployed, which is defined as follows.

$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{22}$$

where, *mAP* is the average precision of multiple queries; *AveP*(*q*) is average of the*P@K*; the *P@K* means the precision at K_i, $i = 1, \ldots, n$; *K* represents the rank position of the relevant document.

The performance of the proposed method was evaluated based on the *mAP* measure, which is defined in Eq. (22).

The *mAP* score was computed in various combinations of the query key-frames inputted to the system. The obtained results for different video sets have been presented in Table 1. A Bar-chart was drawn for the results obtained, which has been presented in Figure 9. The Bar-chart reveals that the proposed system results in cent per cent precision while α is fixed at 1% or α = 0.01. The P@α is slowly decreasing while α is increasing. Suppose, there does not exist the same target video in the database, the proposed system responds zero per cent retrieval rate when α = 1% (i.e., α = 0.01). However, it retrieves similar videos while fixing α ≥ 5.

Furthermore, the *mAP* score was computed database-wise that results in 0.807, 0.812, and 0.814 for CC_WEB, UCF101, and Ours datasets, respectively. A Bar-chart was drawn for the above mAP values, which has been illustrated in Figure 10. The computed mAP score, 0.812, for the dataset UCF101was compared to the method proposed by Dong and Li [64], the obtained results show that the proposed method gives better results than the existing method. Moreover, the proposed method requires minimal computation time than the method proposed by Dong and Li [64]. As the method [64, 65] has been developed based on the deep convolutional neural networks, it could be involved in a good number iterations so that it consumes more time for computation.

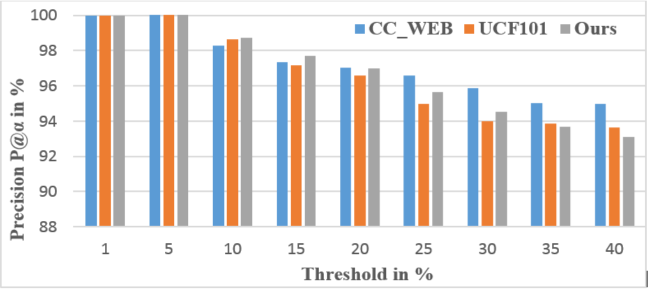

Figure 9. P@α versus threshold dataset-wise

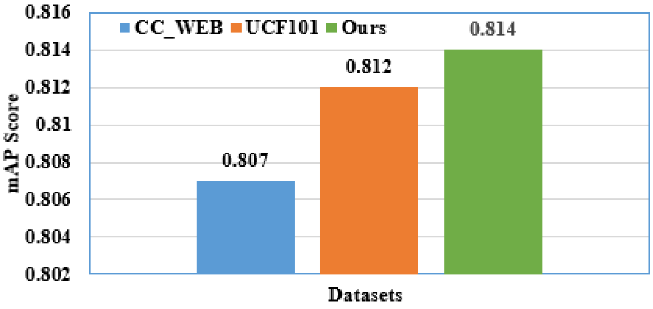

Figure 10. mAP score versus different datasets

5. CONCLUSION AND FUTURE DIRECTION

In this paper, a video retrieval method has been presented, which was developed based on the multivariate parametric statistical tests. Two different benchmark video datasets, such as CC_WEB and UCF101, subjected to verify the performance of the proposed method; in addition to that, we have constructed a new video database, which was subjected to experiments. The precision at α, i.e., P@α, and mAP score, were computed to validate the proposed method. The proposed method responded an average retrieval at the rate of 97.232%, 96.540%, and 96.641% for CC_WEB, UCF101, and our newly constructed database, respectively. Also, the mAP scores computed for each video datasets, which resulted in 0.807, 0.812, and 0.814 for CC_WEB, UCF101, and our newly constructed database, respectively. The obtained retrieval results show that the proposed method performs better than the existing methods. Moreover, the proposed method serves a trade-off between the accuracy and computational time complexity compared to that of the existing methods.

The technique adopted in this paper could also be extended for big-data analytics with multiple characteristics, both on offline and online data analytics. We also have a plan to refine this method and implement online video analyses and retrieval.

REFERENCES

[1] Bouwmans, T., Maddalena, L., Petrosino, A. (2017). Scene background initialization: A taxonomy. Pattern Recognition Letters, 96: 3-11. https://doi.org/10.1016/j.patrec.2016.12.024

[2] Sobral, A., Zahzah, E.H. (2017). Matrix and tensor completion algorithms for background model initialization: A comparative evaluation. Pattern Recognition Letters, 96: 22-33. https://doi.org/10.1016/j.patrec.2016.12.019

[3] Maddalena, L., Petrosino, A. (2015). Towards benchmarking scene background initialization. In International Conference on Image Analysis and Processing, 9281: 469-476. https://doi.org/10.1007/978-3-319-23222-5_57

[4] Ramirez-Alonso, G., Ramirez-Quintana, J.A., Chacon-Murguia, M.I. (2017). Temporal weighted learning model for background estimation with an automatic re-initialization stage and adaptive parameters update. Pattern Recognition Letters, 96: 34-44. https://doi.org/10.1016/j.patrec.2017.01.011

[5] Elharrouss, O., Moujahid, D., Tairi, H. (2015). Motion detection based on the combining of the background subtraction and the structure–texture decomposition. Optik, 126(24): 5992-5997. https://doi.org/10.1016/j.ijleo.2015.08.084

[6] Garcia-Garcia, B., Bouwmans, T., Silva, A.J.R. (2020). Background subtraction in real applications: Challenges, current models and future directions. Computer Science Review, 35: 100204. https://doi.org/10.1016/j.cosrev.2019.100204

[7] Guyon, C., Bouwmans, T., Zahzah, E.H. (2012). Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis. Principal Component Analysis, 10: 223-238.

[8] Bouwmans, T. (2014). Traditional and recent approaches in background modeling for foreground detection: An overview. Computer Science Review, 11-12: 31-66. https://doi.org/10.1016/j.cosrev.2014.04.001

[9] Varadarajan, S., Miller, P., Zhou, H. (2015). Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. Pattern Recognition, 48(11): 3488-3503. https://doi.org/10.1016/j.patcog.2015.04.016

[10] Wang, Y., Luo, Z., Jodoin, P.M. (2017). Interactive deep learning method for segmenting moving objects. Pattern Recognition Letters, 96: 66-75. https://doi.org/10.1016/j.patrec.2016.09.014

[11] Bouthemy, P., Lalande, P. (1993). Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. Optical Engineering, 32(6): 1205-1213. https://doi.org/10.1117/12.134183

[12] Beauchemin, S.S., Barron, J.L. (1995). The computation of optical flow. ACM Computing Surveys (CSUR), 27(3): 433-466. https://doi.org/10.1145/212094.212141

[13] McFarlane, N.J., Schofield, C.P. (1995). Segmentation and tracking of piglets in images. Machine Vision and Applications, 8(3): 187-193. https://doi.org/10.1007/BF01215814

[14] Lee, B., Hedley, M. (2002). Background estimation for video surveillance. Image and Vision Computing New Zealand, IVCNZ, 315-320. http://hdl.handle.net/102.100.100/196971?index=1

[15] Zheng, J., Wang, Y., Nihan, N.L., Hallenbeck, M.E. (2006). Extracting roadway background image: Mode-based approach. Transportation Research Record, 1944(1): 82-88. https://doi.org/10.1177/0361198106194400111

[16] Kim, H., Sakamoto, R., Kitahara, I., Toriyama, T., Kogure, K. (2007). Robust foreground extraction technique using Gaussian family model and multiple thresholds. In Asian Conference on Computer Vision, 4843: 758-768. https://doi.org/10.1007/978-3-540-76386-4_72

[17] Allili, M.S., Bouguila, N., Ziou, D. (2007). A robust video foreground segmentation by using generalized gaussian mixture modeling. In Fourth Canadian Conference on Computer and Robot Vision (CRV'07), pp. 503-509. https://doi.org/10.1109/CRV.2007.7

[18] Tavakkoli, A., Nicolescu, M., Bebis, G. (2006). A novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. In International Symposium on Visual Computing, pp. 40-49. https://doi.org/10.1007/11919476_5

[19] Xiao, M., Han, C., Kang, X. (2006). A background reconstruction for dynamic scenes. In 2006 9th International Conference on Information Fusion, pp. 1-7. https://doi.org/10.1109/ICIF.2006.301727

[20] Palomo, E.J., Domínguez, E., Luque, R.M., Muñoz, J. (2009). Image hierarchical segmentation based on a GHSOM. In International Conference on Neural Information Processing, 5863: 743-750. https://doi.org/10.1007/978-3-642-10677-4_85

[21] Luque, R.M., Domínguez, E., Palomo, E.J., Munoz, J. (2010). An ART-type network approach for video object detection. In ESANN. (European Symposium on Artificial Neural Networks), pp. 423-428.

[22] Maddalena, L., Petrosino, A. (2012). The SOBS algorithm: What are the limits? In 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 21-26. https://doi.org/10.1109/CVPRW.2012.6238922

[23] Maddalena, L., Petrosino, A. (2014). The 3dSOBS+ algorithm for moving object detection. Computer Vision and Image Understanding, 122: 65-73. https://doi.org/10.1016/j.cviu.2013.11.006

[24] Yousif, H., Yuan, J., Kays, R., He, Z. (2017). Fast human-animal detection from highly cluttered camera-trap images using joint background modeling and deep learning classification. In 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-4. https://doi.org/10.1109/ISCAS.2017.8050762

[25] Ding, X., He, L., Carin, L. (2011). Bayesian robust principal component analysis. IEEE Transactions on Image Processing, 20(12): 3419-3430. https://doi.org/10.1109/TIP.2011.2156801

[26] Babacan, S.D., Luessi, M., Molina, R., Katsaggelos, A.K. (2012). Sparse Bayesian methods for low-rank matrix estimation. IEEE Transactions on Signal Processing, 60(8): 3964-3977. https://doi.org/10.1109/TSP.2012.2197748

[27] Tezuka, H., Nishitani, T. (2008). A precise and stable foreground segmentation using fine-to-coarse approach in transform domain. In 2008 15th IEEE International Conference on Image Processing, pp. 2732-2735. https://doi.org/10.1109/ICIP.2008.4712359

[28] Gao, T., Liu, Z.G., Gao, W.C., Zhang, J. (2008). A robust technique for background subtraction in traffic video. In International Conference on Neural Information Processing, pp. 736-744. https://doi.org/10.1007/978-3-642-03040-6_90

[29] Baltieri, D., Vezzani, R., Cucchiara, R. (2010). Fast background initialization with recursive Hadamard transform. In 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, 165-171. https://doi.org/10.1109/AVSS.2010.43

[30] Tu, G.J., Karstoft, H., Pedersen, L.J., Jørgensen, E. (2015). Illumination and reflectance estimation with its application in foreground detection. Sensors, 15(9): 21407-21426. https://doi.org/10.3390/s150921407

[31] Muselet, D., Macaire, L. (2007). Combining color and spatial information for object recognition across illumination changes. Pattern Recognition Letters, 28(10): 1176-1185. https://doi.org/10.1016/j.patrec.2007.02.001

[32] Elharrouss, O., Moujahid, D., Tairi, H. (2015). Motion detection based on the combining of the background subtraction and the structure–texture decomposition. Optik, 126(24): 5992-5997. https://doi.org/10.1016/j.ijleo.2015.08.084

[33] Varadarajan, S., Miller, P., Zhou, H. (2015). Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. Pattern Recognition, 48(11): 3488-3503. https://doi.org/10.1016/j.patcog.2015.04.016

[34] Sobral, A., Zahzah, E.H. (2017). Matrix and tensor completion algorithms for background model initialization: A comparative evaluation. Pattern Recognition Letters, 96: 22-33. https://doi.org/10.1016/j.patrec.2016.12.019

[35] Ramirez-Alonso, G., Ramirez-Quintana, J.A., Chacon-Murguia, M.I. (2017). Temporal weighted learning model for background estimation with an automatic re-initialization stage and adaptive parameters update. Pattern Recognition Letters, 96: 34-44. https://doi.org/10.1016/j.patrec.2017.01.011

[36] de Geus, A.R., Batista, M.A., Rabelo, M.N., Barcelos, C.Z., da Silva, S.F. (2019). Maize insects classification through endoscopic video analysis. In Canadian Conference on Artificial Intelligence, 11489: 251-262. https://doi.org/10.1007/978-3-030-18305-9_20

[37] Anjulan, A., Canagarajah, N. (2007). Object based video retrieval with local region tracking. Signal Processing: Image Communication, 22(7-8): 607-621. https://doi.org/10.1016/j.image.2007.05.008

[38] Aytar, Y., Shah, M., Luo, J. (2008, June). Utilizing semantic word similarity measures for video retrieval. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. https://doi.org/10.1109/CVPR.2008.4587822

[39] Shang, L., Yang, L., Wang, F., Chan, K.P., Hua, X.S. (2010). Real-time large scale near-duplicate web video retrieval. In Proceedings of the 18th ACM international conference on Multimedia, pp. 531-540. https://doi.org/10.1145/1873951.1874021

[40] Andrade, F.S., Almeida, J., Pedrini, H., Torres, R.D.S. (2012). Fusion of local and global descriptors for content-based image and video retrieval. In Iberoamerican Congress on Pattern Recognition, 7441: 845-853. https://doi.org/10.1007/978-3-642-33275-3_104

[41] Adami, N., Cavallaro, A., Leonardi, R., Migliorati, P. (2013). Analysis, retrieval and delivery of multimedia content. In Springer Lecture Notes in Electrical Engineering, 165-180. https://doi.org/10.1007/978-1-4614-3831-1

[42] Berg, J.S., Merlino Jr, A.E., Doody, D.R. (2015). U.S. Patent No. 9,087,125. Washington, DC: U.S. Patent and Trademark Office.

[43] Mironică, I., Ionescu, B., Uijlings, J., Sebe, N. (2016). Fisher kernel temporal variation-based relevance feedback for video retrieval. Computer Vision and Image Understanding, 143: 38-51. https://doi.org/10.1016/j.cviu.2015.10.005

[44] Mohamadzadeh, S., Farsi, H. (2016). Content based video retrieval based on HDWT and sparse representation. Image Analysis & Stereology, 35(2): 67-80. https://doi.org/10.5566/ias.1346

[45] Hao, Y., Mu, T., Hong, R., Wang, M., An, N., Goulermas, J.Y. (2016). Stochastic multiview hashing for large-scale near-duplicate video retrieval. IEEE Transactions on Multimedia, 19(1): 1-14. https://doi.org/10.1109/TMM.2016.2610324

[46] Lou, Y., Bai, Y., Lin, J., Wang, S., Chen, J., Chandrasekhar, V., Gao, W. (2017). Compact deep invariant descriptors for video retrieval. In 2017 Data Compression Conference (DCC), pp. 420-429. https://doi.org/10.1109/DCC.2017.31

[47] Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., Kompatsiaris, Y. (2017). Near-duplicate video retrieval by aggregating intermediate CNN layers. In International Conference on Multimedia Modeling, 10132: 251-263. https://doi.org/10.1007/978-3-319-51811-4_21

[48] Dong, J., Li, X., Snoek, C.G. (2018). Predicting visual features from text for image and video caption retrieval. IEEE Transactions on Multimedia, 20(12): 3377-3388. https://doi.org/10.1109/TMM.2018.2832602

[49] Liu, Y., Sui, A. (2018). Research on feature dimensionality reduction in content based public cultural video retrieval. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), pp. 718-722. https://doi.org/10.1109/ICIS.2018.8466379

[50] Song, J., Zhang, H., Li, X., Gao, L., Wang, M., Hong, R. (2018). Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Transactions on Image Processing, 27(7): 3210-3221. https://doi.org/10.1109/TIP.2018.2814344

[51] Wu, G., Han, J., Guo, Y., Liu, L., Ding, G., Ni, Q., Shao, L. (2018). Unsupervised deep video hashing via balanced code for large-scale video retrieval. IEEE Transactions on Image Processing, 28(4): 1993-2007. https://doi.org/10.1109/TIP.2018.2882155

[52] Nie, X., Jing, W., Cui, C., Zhang, J., Zhu, L., Yin, Y. (2019). Joint multi-view hashing for large-scale near-

duplicate video retrieval. IEEE Transactions on Knowledge and Data Engineering, 32(10): 1951-1965. https://doi.org/10.1109/TKDE.2019.2913383

[53] Swapna, D., Vickram, P., Krishna, A.S., Srinivas, V.S. (2016). A survey on local patterns for signature verification. In 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 0700-0704. https://doi.org/10.1109/ICCSP.2016.7754234

[54] Ejaz, N., Mehmood, I., Baik, S.W. (2013). Efficient visual attention based framework for extracting key frames from videos. Signal Processing: Image Communication, 28(1): 34-44. https://doi.org/10.1016/j.image.2012.10.002

[55] Srikrishna, A., Reddy, B.E., Pompapathi, M. (2016). Pixon based image denoising scheme by preserving exact edge locations. Journal of the Institution of Engineers (India): Series B, 97(3): 395-403. https://doi.org/10.1007/s40031-014-0178-9

[56] Li, H.C., Celik, T., Longbotham, N., Emery, W.J. (2015). Gabor feature based unsupervised change detection of multitemporal SAR images based on two-level clustering. IEEE Geoscience and Remote Sensing Letters, 12(12): 2458-2462. https://doi.org/10.1109/LGRS.2015.2484220

[57] Liu, C., Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Transactions on Image Processing, 11(4): 467-476. https://doi.org/10.1109/TIP.2002.999679

[58] Liu, C.L., Koga, M., Fujisawa, H. (2005). Gabor feature extraction for character recognition: comparison with gradient feature. In Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pp. 121-125. https://doi.org/10.1109/ICDAR.2005.119

[59] Downloaded from https://www.youtube.com/watch?v=TQSaPsKHPqs, accessed on 11 November 2019.

[60] Downloaded from https://www.youtube.com/watch?v=Jc-LEG0T_4c, accessed on 11 November 2019.

[61] de Carvalho, F.D.A., de Melo, F.M., Lechevallier, Y. (2015). A multi-view relational fuzzy c-medoid vectors clustering algorithm. Neurocomputing, 163: 115-123. https://doi.org/10.1016/j.neucom.2014.11.083

[62] Anderson, T.W. (2003). An Introduction to Multivariate Statistical Analysis. 3rd ed., John Wiley & Sons, Inc., 2003.

[63] Johnson, R.A., Wichern, D.W. (2013). Applied Multivariate Statistical Analysis, Sixth Edition, PHI Learning Pvt. Ltd., New Delhi.

[64] Dong, Y., Li, J. (2018). Video retrieval based on deep convolutional neural network. In Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing, pp. 12-16. https://doi.org/10.1145/3220162.3220168

[65] Dai, C.Q., Lv, Y.L., Long, Y.X., Sui, H.T. (2018). A novel image enhancement technique for tunnel leakage image detection. Traitement du Signal, 35(3-4): 209-222. https://doi.org/10.3166/TS.35.209-222