ISSN: 0067-2904

Video Abstraction Method Using Color Moment and Density-Based Clustering Algorithm

Eman Hato[1]*, Matheel E. Abdulmunem[2], Zeyad FarooqLutfi[1]
[1]*Computer Science Department, College of Sciences, Mustansiriyah University, Baghdad, Iraq.*
[2]*Computer Science Department, University of Technology-Iraq, Baghdad, Iraq.*

Received: 27/11/2024 Accepted: 17/3/2025 Published: 30/3/2026

Abstract

The video summarization abstracts the essential information of video content. This paper proposed a new video abstraction method to extract meaningful video frames. The proposed method encompasses the following stages: the extracted frames are converted to grayscale images. The quality of the frames is assessed using the NIQE method. The feature vector for each frame is obtained using the kurtosis moment, and the difference between two consecutive feature vectors is calculated. The DBSCAN algorithm is applied to classify these difference values, recording any temporal transitions when a difference value is identified as an outlier. Finally, the frame with the highest NIQE value in each segment is compiled into the video abstraction. The results demonstrated excellent performance of the proposed method, which achieved 100% Accuracy and F- Score. The average time of the abstraction videos is 2.962 seconds. Various factors were analyzed for their impact on the method, revealing that using Euclid's metric and setting epsilon to 15 yielded the best results for DBSCAN-based temporal segmentation.

Keywords: Color Moment, DBSCAN, NIQE, Temporal Segmentation, Video Abstraction.

طريقة تلخيص الفيديو باستخدام المعلومات اللونية و خوارزمية كثافة التجميع

ايمان هاتو[1]*, مثيل عبد المنعم عماد الدين[2], زياد فاروق لطفي[1]
[1]قسم علوم الحاسوب، كلية العلوم، الجامعة المستنصرية، بغداد، العراق.
[2]قسم علوم الحاسوب، الجامعة التكنولوجية–العراق، بغداد، العراق.

الخلاصة

ان عملية تلخيص الفيديو هي استخراج المعلومات المهمة لمحتويات الفيديو. في هذا البحث تم تقديم طريقة جديدة لتلخيص الفديو والتي تهدف الى استخراج الصور ذات المعنى المهم من الفيديو. تتكون الطريقة المقترحة لتلخيص الفديو من المراحل التالية: بدايةً، يتم تحويل الصورالمستخرجة من الفيديو إلى صور رمادية اللون، وبعدها يتم حساب مقياس جودة الصورة لكل صورة. يتم استخراج متجه الميزات للصور المستخرجة باستخدام المعلومات اللونية ويتم حساب الفرق بين كل متجهين تتاليين من ضمن متجه الميزات. يتم بعد ذلك استخدام خوارزمية كثافة التجميع لتصنيف قيم الاختلافات التي تم حسابها في الخطوة السابقة، حيث يتم

*Email: emanhato@uomustansiriyah.edu.iq

تثبيت التجزئة الزمنية لتحديد اجزاء الفيديو كلما تم تصنيف قيمة الفروقات على انها قيمة متطرفة. يتم اختيار الصورة التي تمتلك اعلى قيمة وفق مقياس جودة الصورة من كل جزء من الفيديو وتجميعها في الفيديو الملخص. أظهرت النتائج الأداء الممتاز للطريقة المقترحة، والتي حققت 100٪ وفقًا لمقياس الدقة ومقياس F . كما ان معدل الوقت لمقاطع الفديو الملخصة هو 2.962 ثانية. تمت مقارنة تأثير العديد من عوامل خوارزمية كثافة التجميع على الطريقة المقترحة وأظهرت النتائج أن مقياس اقليديس و epsilon عند القيمة 15 كانا العوامل الافضل لتحقيق نتائج التجزئة الزمنية عالية الدقة.

1. Introduction

There has been a significant increase in video uploading and sharing, allowing virtually anyone to become a provider of online video content. However, watching videos can be time-consuming due to redundant information, requiring viewers to invest time in analyzing and extracting relevant details. This creates challenges in video searching, management, and retrieval [1,2]. To address these challenges, video summarization comes into play. Video summarization automatically extracts important and useful scenes from a larger video, offering users the most interesting and relevant segments. Essentially, it is creating a shorter version of a lengthy video. An effective summary should allow viewers to gather maximum information in less time without losing the essential content details [3,4].

The result of video summarization may vary depending on the application domain, so many techniques have been developed to produce a summary video by highlighting important video scenes, such as key events, scenes, or objects in the video. Video summarization is used in various applications based on video processing such as searching, indexing, classifying, and locating video objects [5, 6]. Video summarization techniques can be categorized into two main categories which are dynamic and static video summarization [7,8].

Dynamic video summarization, often called video skimming, consists of three main processes: video segmentation, importance score prediction, and segment selection. First, the video is divided into segments, known as skim units, which are processed independently. These segments focus on the most important information within the video. Next, the importance score for each segment is calculated. This score is determined using all segments or only the keyframes retrieved from those segments. Finally, unnecessary segments are eliminated to shorten the total duration of the original video by selecting only those segments with the highest importance scores [9].

In static video summarization, or video abstraction, the process begins by dividing the video into individual frames. Visual features are then extracted from each frame using various feature extraction methods. Unnecessary frames are removed, and the remaining relevant frames are grouped into clusters using unsupervised or supervised machine learning techniques. Ultimately, the keyframes selected from these clusters form the static summary of the video [10].

Temporal video segmentation is a widely used technique in video abstraction. Its primary goal is to divide a video into its fundamental units, called shots, by detecting transitions and identifying the boundaries between successive shots. Keyframes are extracted from these shots to create a video summary with the most relevant and informative content [11]. Despite advancements in video summarization methods, several challenges remain. One major issue is that these methods often struggle to cover video content effectively and accurately, as they primarily rely on comparing visual features against fixed thresholds. Additionally, many video abstraction techniques are computationally intensive, resulting in longer processing times [12].

This paper presents a new method for video summarization that focuses on accurately selecting video content while minimizing execution time. The structure of the paper is as

follows: Section 2 reviews relevant related work. Section 3 explains the techniques employed, followed by a detailed discussion of the proposed method in Section 4. Experimental results are presented in Section 5. Finally, Section 6 provides concluding remarks and outlines directions for future research.

2. Literature Review

Video content summarization is commonly utilized in various applications. In the field of video abstraction research, key frames are identified by randomly or uniformly sampling frames at specific time intervals. Effective key frame extraction algorithms must select frames representing the entire video content while ensuring that no important information is missed. To achieve this, many methods have been proposed for summarizing videos by extracting informative segments related to the topic through a combination of segmentation processes and frame feature analysis.

Khan et al. [13] proposed a video summarization method that leverages deep features. In this approach, features are extracted from video frames using a Convolutional Neural Network (CNN) to segment the video into shots. The memorability of each frame within a shot is then predicted, and the frame with the highest entropy and memorability is selected to create the video summary. The method achieved an F-Score of 0.79. In another study by Hana et al. [14], keyframes were extracted to provide a video summary of the most informative frames. Initially, a set of candidate frames is selected using a window-splitting rule. Interest points are identified from this candidate set, and the frequency values between each pair of frames are calculated and represented using a repeatability-directed graph. Keyframe selection is conducted through graph clustering, resulting in an F-Score of 0.723 for the keyframes generated.

In [15] Shruti J. and Mahmood J. presented a video summarization method based on keyframe extraction. Low-level features namely image histograms, SIFT, and image features are extracted from CNN. Then K-means was used to partition the features obtained from the frames into different clusters so the sum of squared differences within the cluster is minimal. Keyframes were selected from each cluster. Gaussian clustering was applied to classify all the extracted key frames into interesting and uninteresting frames. The cluster with interesting frames generated the video summary. The result of CNN features and Gaussian clustering was 0.212 in terms of the F-score for different videos used from the SumMe dataset.

A key frame extraction algorithm based on deep prior information and fusion of multiple features was introduced in [16] by QI Z. et al. The method is divided into a feature extraction module and an importance map prediction. The feature extraction module uses modified VGG16, and the importance map prediction is performed by the nearest neighbor classifier in the feature space. Finally, according to the importance of the moving pedestrian target, the frame that can best represent the moving pedestrian target is extracted as the keyframe. The average accuracy of the method was 0.95 in terms of precision measure.

In [17] Buyun L. et al. presented a video summarization method that extracts SURF features from video sequences and matches features between adjacent frames. The boundaries of shots are detected by calculating the similarity of adjacent frames with the help of double thresholds. Then, the color histogram of frames within the shot is clustered and the frame closest to the center of the cluster is selected as the keyframe. The accuracy scores for detecting of the boundaries of shots were 97.22 and 93.33 on the recall and precision measures, respectively. The method proposed by H.M. Nandini et al. in [18] detected shots by extracting texture features based on the Local Binary Pattern (LBP) method. The shots were detected using Euclidean distance and adaptive threshold. In the keyframe extraction stage, the magnitude gradient using the Sobel factor was extracted from each frame of the

segmented shot. Then, the coefficient of variation was calculated for each frame and the frame with the highest value was selected as the key frame. The results show that the proposed method has an average F-score of 98.15.

Hafez B. U. Alvi [19] presented a deep learning-based video summarization framework. The proposed framework summarizes videos according to the object of interest, e.g., person, mobile phone, airplane, and car. Initially, objects in the video are located and detected using the You Look Only Once (YOLOv3) detector. Then, the video is summarized by taking the frames containing the detected objects. The overall accuracy achieved by the proposed summarization framework was 99.6.

HAO T. et al. presented a key frame extraction method using Density Peak Pooling (DPC) and CNNs [20]. First, deep features are extracted from the video frame and mapped into a high-dimensional feature space. Next, the video is divided into several segments, and the key frame in each segment is identified using temporal segment density peak pooling (TSDPC). Finally, all the key frames extracted from each segment are combined to form the final video summary that replaces the original video. The method achieved an accuracy of 81.44% on the HMDB51 dataset and 98.45% on the UCF dataset. In [21] Yunzuo Z. et al. proposed a key frame extraction method for lecture videos. The method first extracts the spatio-temporal slices of the subtitle area in the video sequence to generate the spatio-temporal subtitle of the video. Then, the spatio-temporal subtitle of the video is processed in a binary process, and the projection method is used to generate the pixel accumulation curve of the spatio-temporal subtitle. Finally, the keyframe is extracted by combining the edge detection of the curve and the subtitle presence threshold. The test results showed that the average F-score value was 0.97, the average recall value was 0.98, and the average precision value was 0.97.

Extracting representative keyframes from videos is very important in video summarization because it greatly reduces computational time. Despite the recent progress made, video summarization is still an open problem, as existing methods have not balanced performance and efficiency simultaneously. To address this problem, this paper presents a new method for video summarization, which aims to summarize of the video to avoid redundancy in the content while preserving the important information in it and with the least.

3. Background Theory

3.1. Color Moment

Color moments is a popular color feature representation approach in image processing because it is simple and effective. The color information moments can fully describe the image's color distribution. The order moment of the image is defined by Equation 1 [22]:

$$m_{pq} = \sum_{x=1}^{N} \sum_{y=1}^{M} x^p \, y^q f(x,y) \quad p,q = 0,1,2,\ldots. \tag{1}$$

Where f (x, y) is an image, N is image width, M is image height, and p + q is moment order. The central moment is calculated by Equation 2 [22]:

$$\mu_{pq} = \sum_{x=1}^{N} \sum_{y=1}^{M} (x-\bar{x})^p \, (y-\bar{y})^q f(x,y) \quad p,q = 0,1,2,\ldots. \tag{2}$$

Were $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$, which are the gravity center coordinates of the image.

Skewness measures the tendency of a distribution to exhibit outliers. Distributions more prone to outliers than a normal distribution have a skewness greater than 3, while those less prone have a skewness less than 3. A normal distribution has a skewness of 0. The skewness value describes the heaviness of the distribution. The definition of skewness is defined as in Equation 3: [23]

$$\mathrm{k} = \frac{\mathrm{E}(\mathrm{x}-\mu)^4}{\sigma^2} \tag{3}$$

where μ is the x mean, σ is the x standard deviation, and E $(x - \mu)$ represents the expected value of the quantity $(x - \mu)$. [22,23].

3.2 Density-Based Clustering

One of the most important techniques used in various fields such as text mining, image segmentation, and video processing is clustering. The clustering technique divides the data points into separate groups to minimize interclass similarity and maximize intraclass similarity. Density-Based Spatial Clustering for Applications with Noise (DBSCAN) is a density-based clustering algorithm designed to detect clusters and noise in data. The algorithm identifies two important parameters: epsilon (neighborhood number around a data point) and minpts (minimum neighbors' number within the epsilon radius). DBSCAN randomly picks a point in the data. If there are at least minpts points within the epsilon radius of the point, the algorithm considers all these points to be part of the same cluster. Otherwise, the point is considered a noise point. This is repeated until all points in the data have been labeled [24]. DBSCAN has several strengths that make it a popular clustering algorithm for many types of datasets and clustering tasks. It offers the ability to handle outliers, meaning it can effectively identify and discard noise points that do not belong to any cluster. It discovers clusters of different shapes and does not need to know the clusters' number of datasets in advance. In addition, its simplistic approach has helped it become widely applicable in many fields of science [25].

3.3. Natural Image Quality Evaluator (NIQE)

Natural image quality evaluation (NIQE) is a non-reference image quality evaluation method created by calculating deviations from statistical regularities observed in images without knowledge of human opinions about them or prior distortions. NIQE expresses the quality of a distorted image as the gap between the model statistics and those of the distorted image. NIQE is useful in training human judgments on known distorted images [26].

4. The Proposed Method

Video content summarization is widely utilized in various applications to enhance the user experience. In video abstraction research, keyframes are identified randomly or uniformly sampling video frames at specific intervals. For an effective key frame extraction algorithm, the extracted key frames must adequately represent the entire video content without omitting important information. Consequently, many methods have been proposed to summarize videos by analyzing frame features and segmenting the content based on its topic.

The rapid increase in the number of videos has made searching and retrieving content a tedious and time-consuming task. Video abstraction is a popular solution to this issue by creating a condensed version of the original video. This paper introduces a new video abstraction method that generates shorter video versions. The proposed method is based on temporal segmentation, which serves as the foundation for abstraction techniques and significantly influences the overall quality of the resulting abstract video.

Many methods for detecting temporal transitions rely on comparing consecutive frames based on the visual features of the video. First, features are extracted from each frame. Then, similarity or dissimilarity measures are calculated and compared against a specified threshold to distinguish temporal transitions. However, predefined thresholds often prove ineffective because video content can change significantly, making it impossible to find a universal threshold that works for all types of videos. A key challenge for these methods is identifying

suitable features and establishing a threshold that adapts to video changes to identify transitions accurately.

Hence, the proposed method aims to achieve high detection accuracy and maintain low computational costs without using any thresholds. The proposed method formulates the video abstraction problem as a clustering problem using a density-based clustering algorithm. The general structure of the proposed method is shown in Figure 1 and the following steps are devoted to explaining the implementation of the proposed video abstraction method, along with a detailed description of each step:

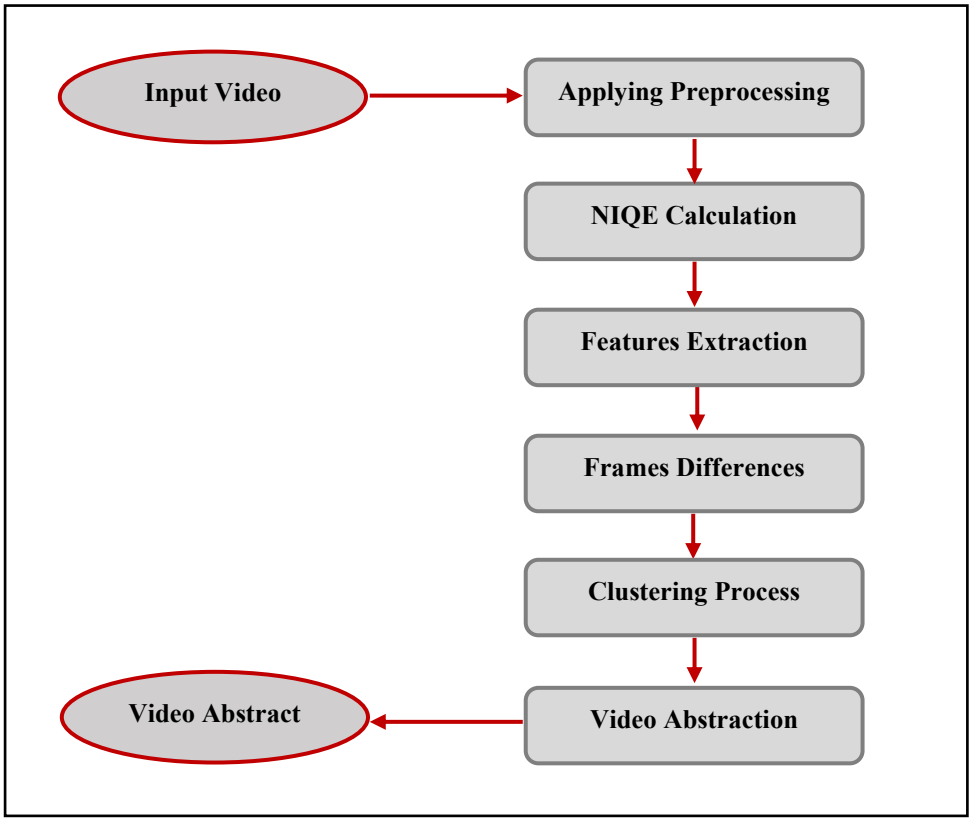

Figure 1: The proposed method

Step1: Applying Preprocessing

Frames are extracted from the input video and resized to 256x256 to reduce execution time. The extracted frames are converted to grayscale images. It is worth noting that changing the frame color will not change the moment features.

Step2: NIQE Calculation

The quality of each frame is evaluated using the NIQE method. After preprocessing the input video, the NIQE value for each frame is calculated and stored in the F-Quality matrix. This matrix has dimensions of FN x 1, where FN represents the total number of frames in the input video.

Step3: Features Extraction

The F-Moment matrix, which has dimensions FN x 16, is initialized to store the features of each frame. Each frame in the input video is divided into 16 blocks of size 4x4. The kurtosis moment of each block in the frame is calculated, and these values are saved in a single row of the F-Moment matrix. This process is repeated for all frames, resulting in a feature matrix for the entire video, as illustrated in Figure 2.

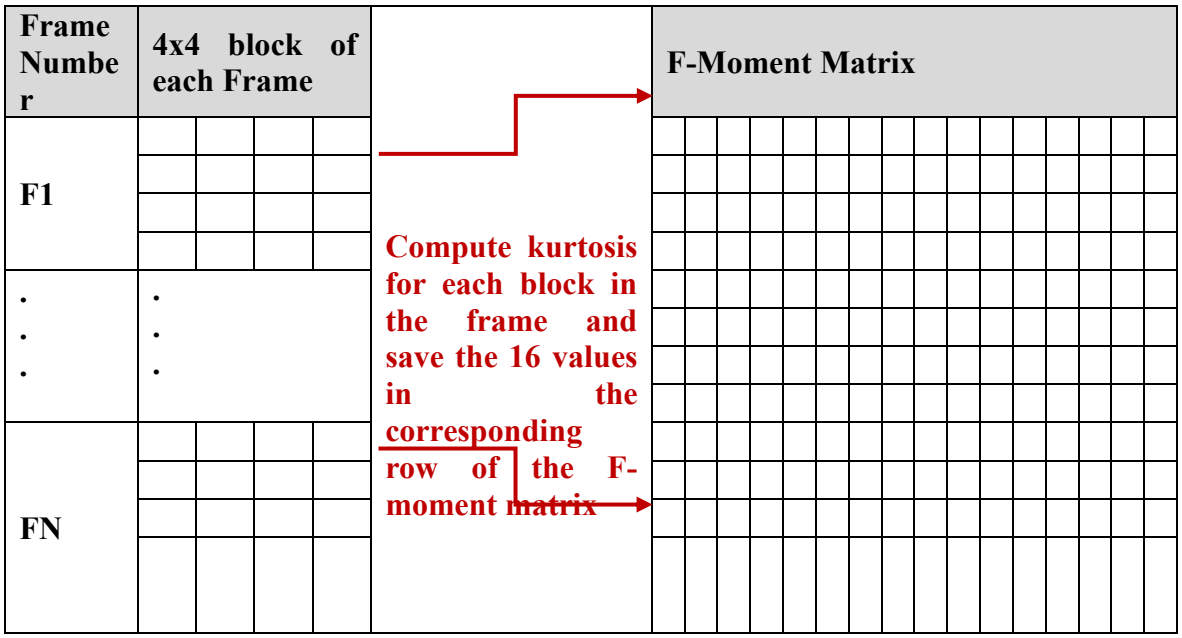

Figure 2: The features matrix extraction.

Step4: Frames Differences

The difference between two consecutive rows of the F-Moment matrix is computed using the Mean Absolute Difference (MAD) measure. This involves calculating the difference between features extracted from the frames stored as rows in the F-Moment matrix. The resulting difference values are then organized in an F-Difference matrix with dimensions FNx1.

Step5: Applying Clustering Process

In this step, the DBSCAN algorithm is utilized to cluster the values of the F-Difference matrix. DBSCAN identifies clusters of closely related values with many nearby neighbors and flags outliers isolated in low-density regions, meaning their nearest neighbors are quite distant. The proposed method detects temporal transitions by examining the differences between frames using the DBSCAN algorithm. As is well understood, the last frame of a temporal segment is markedly different from the first frame of the subsequent segment, resulting in a significant difference measure between these two segments.

In contrast, the differences between frames within the same segment are typically small. The DBSCAN algorithm groups values with minor differences into clusters while identifying values with substantial differences as outliers; these outliers signify temporal transitions in the video. Whenever a difference value in the F-Difference matrix is flagged as an outlier by the DBSCAN algorithm, the temporal transition is recorded. A graphical representation of the different sequences, including transitions and normal scenes, is displayed in Figure 3.

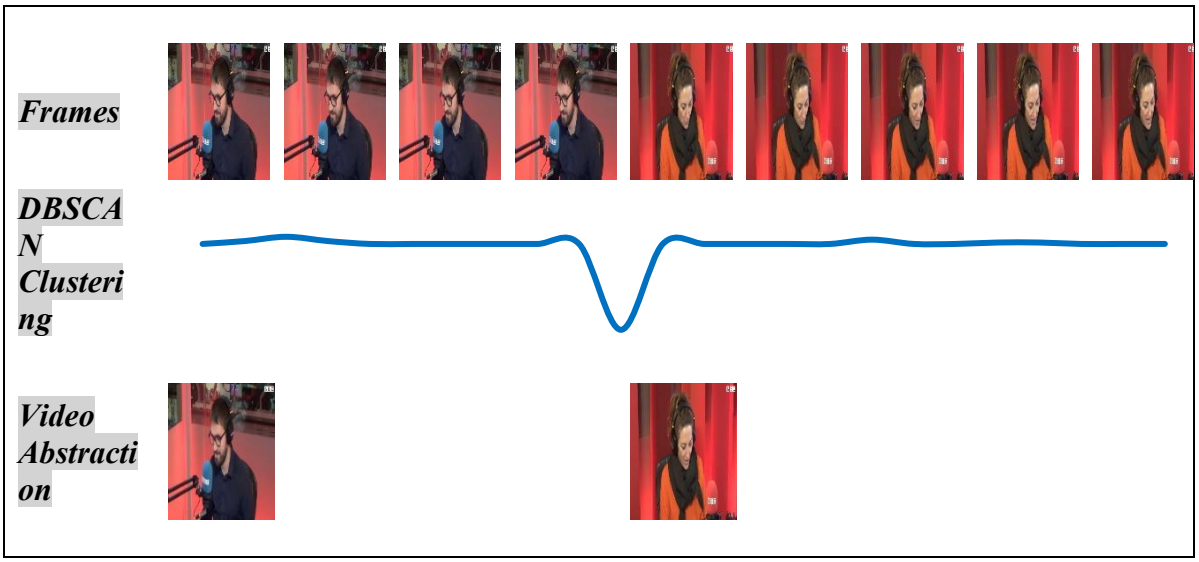

Figure 3: DBSCAN groups small difference values into a cluster and considers large difference values as outliers.

Step6: Video Abstraction

Each segment's keyframe is chosen based on calculations made earlier in the F-Quality matrix, where the NIQE value for each frame is stored. The frame with the highest NIQE value within the segment is selected. These selected keyframes are then compiled and saved into a separate video clip, which serves as a representation of the video abstraction.

5. Experimental Results

Fifteen videos were randomly selected from the BBC YouTube archive to evaluate the performance of the proposed method. The ground truth for the videos was manually annotated (video annotation is the process of adding labels to videos to make it easier for the algorithm to identify the beginning and end of video scenes). Table 1 provides details about the video characteristics.

Accuracy and F-Score [27, 28] were used to evaluate the effectiveness of video segment detection. The proposed method was implemented in MATLAB 2022b on a system equipped with an Intel(R) Core i7 processor running at 2.27 GHz, 12 GB of RAM, and a 64-bit version of Microsoft Windows 8 Ultimate. Evaluation tests were conducted to assess the proposed method's ability to detect temporal segmentation transitions, with the results presented in Table 2. It is worth noting that the DBSCAN algorithm adopted an epsilon value of 15, a minpts value of 5, and Euclidean as the distance measure. The proposed method appears to be robust to frame color and background changes due to the high values of Accuracy and F-score proves the high-level performance of the proposed method.

Table 1: The characteristics of videos

Files	Number of Frames	Number of Transitions	Online Link
V01	7675	31	BBC archive https://www.bbc.co.uk
V02	23300	61	
V03	14500	68	
V04	12025	72	
V05	14475	76	
V06	15175	81	
V07	14925	86	
V08	18250	102	
V09	11300	53	
V10	11500	67	
V11	15275	70	
V12	14200	85	
V13	13700	82	
V14	16250	83	
V15	15250	94	

Table 2: The transition detection results of temporal segmentation

Videos	Accuracy	F- Score
V01	1	1
V02	1	1
V03	1	1
V04	1	1
V05	1	1
V06	1	1
V07	1	1
V08	1	1
V09	1	1
V10	1	1
V11	1	1
V12	1	1
V13	1	1
V14	1	1
V15	1	1
Average	**1**	**1**

The proposed method utilizes kurtosis moments as feature vectors, which are resilient to geometric transformations of frames and provide comprehensive information about the frequency distribution of the data. Additionally, the DBSCAN algorithm effectively identifies and removes outlier values, making it a valuable tool for detecting transformations. To evaluate the performance accuracy of kurtosis moments for feature extraction, they were compared to Standard Deviation (STD). The results are presented in Table 3. Figures 4 and 5 illustrate the comparison between the detection accuracy of kurtosis moments and STD, with kurtosis moments demonstrating superior performance.

Table 3: The transition detection results of temporal segmentation using STD

Files	Accuracy	F- Score
V01	1	1
V02	1	1
V03	1	1
V04	0.208	0.344
V05	1	1
V06	0.185	0.312
V07	1	1
V08	0.176	0.3
V09	1	1
V10	0.85	0.919
V11	1	1
V12	0.176	0.3
V13	0.182	0.309
V14	1	1
V15	0.893	0.944
Average	**0.711**	**0.762**

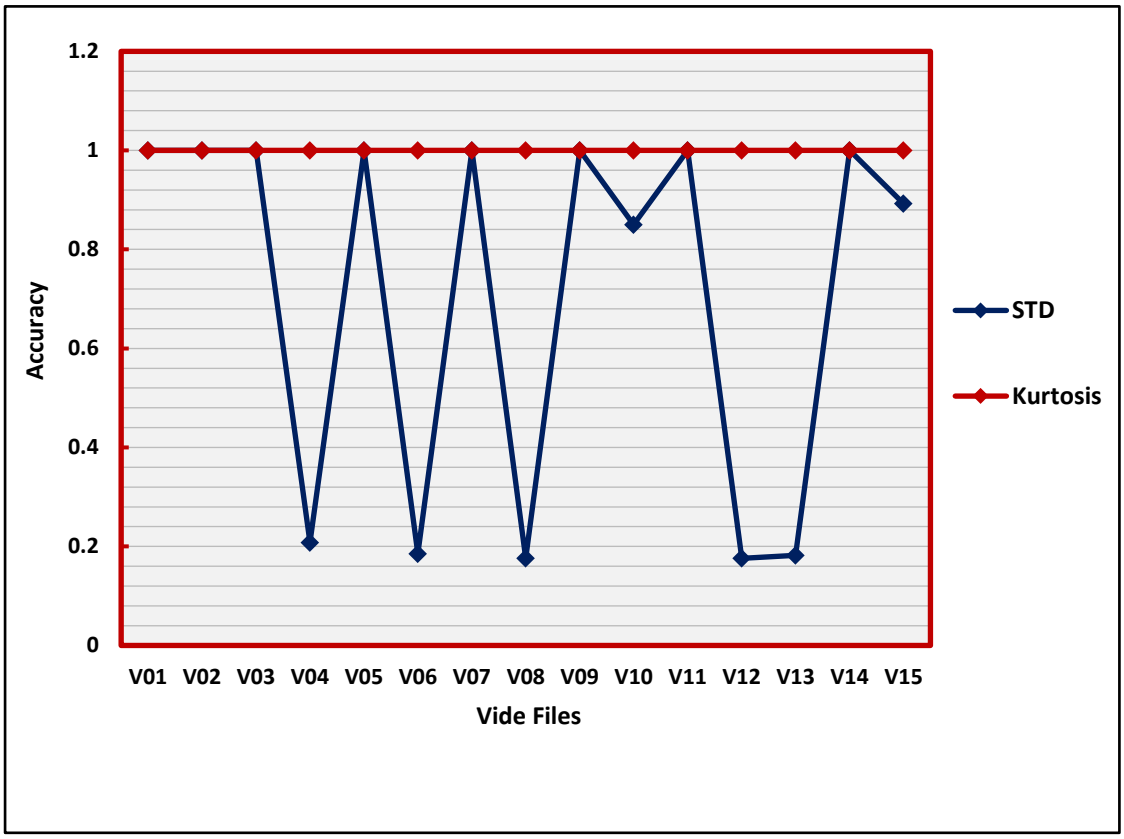

Figure 4: The Accuracy detection of kurtosis and STD moments**.**

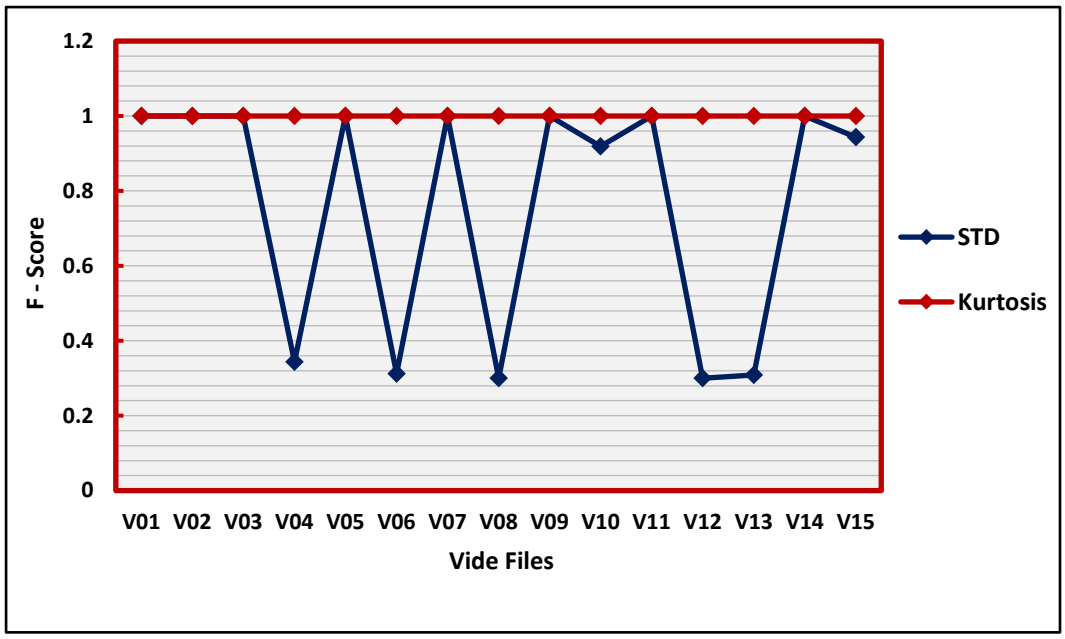

Figure 5: The F- Score detection of kurtosis and STD

The robustness of the DBSCAN algorithm contributed to achieving a high accuracy rate for the proposed method. Several factors influence the performance of this algorithm, including the epsilon value and the distance metric used. The epsilon neighborhood of a point is defined as a numerical measure that specifies the radius for searching neighbors around that point. A core point in a cluster should have at least a minimum number of neighbors within its epsilon neighborhood. Therefore, choosing the epsilon value is crucial. Table 4 illustrates the changes in epsilon values to assess their impact. Additionally, Figure 6 visualizes the differences in accuracy corresponding to the various epsilon values (The epsilon value of 15, 20, 25, 30).

Table 4: The effect of different epsilon values used by DBSCAN

Files	Epsilon=20		Epsilon=25		Epsilon=30	
	Accuracy	F- Score	Accuracy	F- Score	Accuracy	F- Score
V01	1	1	0.064	0.121	0	0
V02	0.063	0.118	0.016	0.032	0	0
V03	0.161	0.278	0.073	0.137	0	0
V04	1	1	0.277	0.435	0.068	0.129
V05	0.157	0.272	0.026	0.051	0	0
V06	0.012	0.024	0	0	0	0
V07	0.193	0.323	0.113	0.204	0.034	0.065
V08	0.009	0.019	0	0	0	0
V09	0.056	0.107	0	0	0	0
V10	1	1	0.735	0.847	0.294	0.454
V11	0.071	0.133	0	0	0	0
V12	0.141	0.247	0.047	0.089	0	0
V13	0.012	0.024	0	0	0	0
V14	0.168	0.288	0.036	0.068	0	0
V15	0.202	0.336	0.074	0.138	0.01	0.021
Average	**0.283**	**0.3446**	**0.0974**	**0.141**	**0.027**	**0.0446**

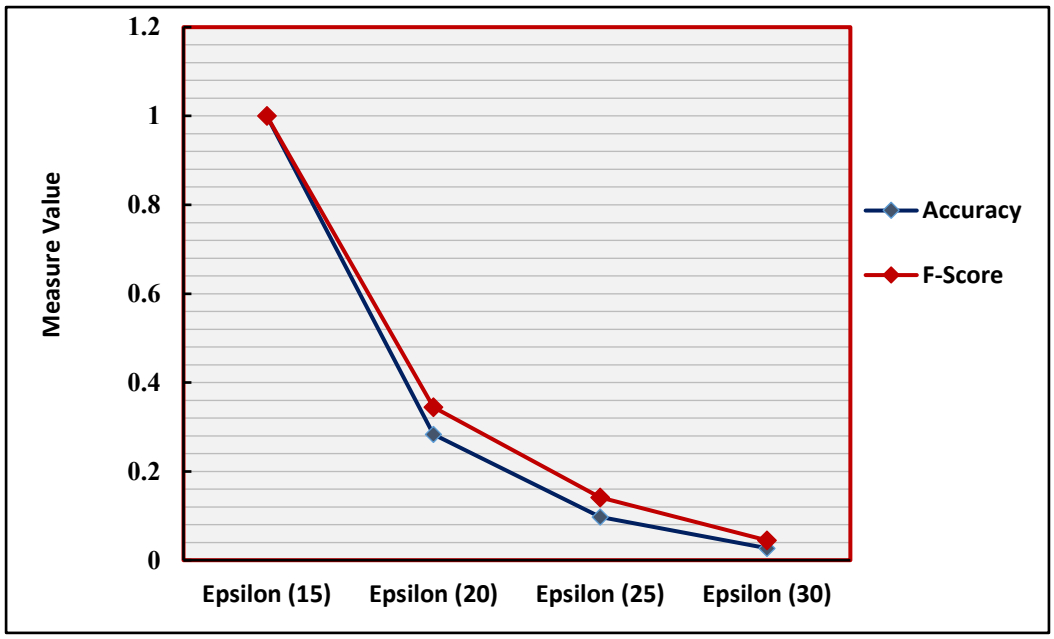

Figure 6: Comparison of the effect of different epsilon values used by DBSCAN.

Upon examining Figure 6, which illustrates the effects of various epsilon values, it is evident that the highest detection accuracy was attained with an epsilon value of 15. Increasing the epsilon further led to a significant decline in detection accuracy. Various distance metrics were tested to identify the most effective one for achieving high accuracy with the DBSCAN algorithm. The metrics evaluated included Euclidean, Correlation, Hamming, and Spearman. Experimental tests have demonstrated that the Euclidean metric provides superior accuracy compared to the other metrics, as confirmed by the results shown in Table 5 and Figure 7.

Table 5: The effect of different distance metrics used by DBSCAN

Files	Correlation		Hamming		Spearman	
	Accuracy	F- Score	Accuracy	F- Score	Accuracy	F- Score
V01	0.04	0.077	0.096	0.176	0.483	0.652
V02	0.02	0.04	0.081	0.151	0.262	0.415
V03	0.133	0.235	0.117	0.21	0.368	0.537
V04	0.36	0.558	0.027	0.054	0.194	0.324
V05	0.15	0.261	0.157	0.272	0.526	0.689
V06	0.015	0.03	0.012	0.024	0.283	0.442
V07	0.43	0.601	0.279	0.436	0.348	0.517
V08	0.34	0.507	0.245	0.393	0.412	0.583
V09	0.407	0.579	0.32	0.485	0.188	0.318
V10	0.268	0.422	0.253	0.405	0.179	0.304
V11	0.175	0.297	0.271	0.427	0.485	0.654
V12	0.141	0.248	0.211	0.349	0.506	0.672
V13	0.102	0.186	0.109	0.198	0.292	0.453
V14	0.184	0.311	0.361	0.531	0.397	0.568
V15	0.125	0.223	0.042	0.082	0.512	0.675
Average	**0.193**	**0.305**	**0.172**	**0.279**	**0.362**	**0.52**

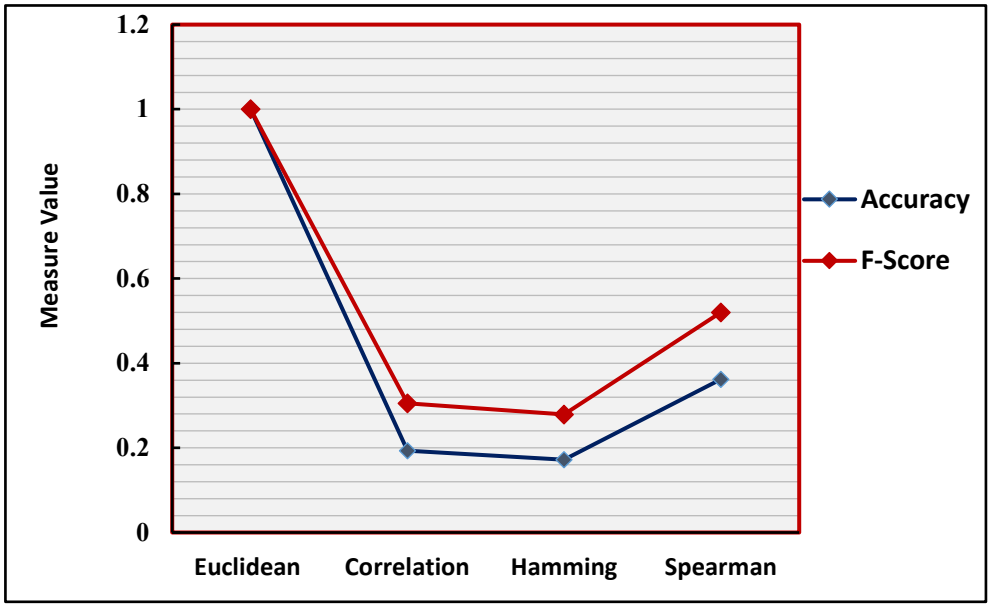

Figure 7: Comparison of the effect of different distance metrics used by DBSCAN.

This analysis shows that the correlation metric is more accurate than the Hamming and Spearman metrics when used by the DBSCAN algorithm in the detection process. While the Hamming and Spearman measures yield similar results, the correlation metric stands out. The summary generated is typically comprised of a set of representative frames, known as keyframes. Keyframe extraction is a crucial technology in video abstraction, as it reduces the computational redundancy between adjacent frames. It is important that these frames are maintained in their original order, and the duration of the summary should be significantly shorter than that of the original video.

The judgment is based on comparing the time duration of the summarized video and the original video. The results are presented in Table 6, where the duration of both the original and summarized videos is expressed in seconds for various video files.

Table 6.: The comparison of the original and summary video time duration

Videos	Original Videos		Abstraction Videos	
	Frames Number	Time Duration in Sec.	Frames Number	Time Duration in Sec.
V01	7675	307	31	1.24
V02	23300	932	61	2.44
V03	14500	580	68	2.72
V04	12025	481	72	2.88
V05	14475	579	76	3.04
V06	15175	607	81	3.24
V07	14925	597	86	3.44
V08	18250	730	102	4.08
V09	11300	452	53	2.12
V10	11500	460	67	2.68
V11	15275	611	70	2.8
V12	14200	568	85	3.4
V13	13700	548	82	3.28
V14	16250	650	83	3.32
V15	15250	610	94	3.76
Average	-	**580.8**	-	**2.962**

The average time of the original videos is 580.8 and the average time of the abstraction videos is 2.962, so the abstraction videos have a much shorter duration than the original videos.

Conclusions

Video summarization is an effective solution for managing large volumes of video content. It aims to extract the most relevant frames from a given video to create a shorter, more informative version while preserving the main content. This paper presents a novel method for video abstraction that utilizes the DBSCAN clustering technique to capture the temporal variation in the video, which is crucial for effective video abstraction and significantly affects its overall quality. In this approach, the frame with the highest quality, as determined by the NIQE method, is selected from each video to be included in the video abstraction. This process results in the creation of abstract videos with commendable quality. Results demonstrate the effectiveness of the proposed method, achieving excellent accuracy as measured by the Accuracy and F-score metrics. The video abstracts contain high-quality content while remaining concise, allowing users to browse the videos efficiently without wasting time.

In future work, the authors aim to evaluate the importance of the score of each video segment based on user preferences and subsequently aggregate these scores into a comprehensive video summary.

Acknowledgments

The authors would like to thank Mustansiriyah University (www. uomustansiriyah.edu.iq) Baghdad-Iraq for its support in the present work.

Disclosure and conflict of interest

“Conflict of Interest: The authors declare that they have no conflicts of interest.”

References

[1] R. S. Kızıltepe, J. Q. Gan, and J. J. Escobar, "A novel keyframe extraction method for video classification using deep neural networks," *Neural Computing and Applications,* vol. 35, no. 34, pp.24513-24524, 2023. doi.org/10.1007/s00521-021-06322-x.

[2] E. Hato, "Temporal video segmentation using optical flow estimation," *Iraqi Journal of Science,* vol.62, no.11, pp. 4181-4194, 2021.

[3] X. Wang, Y. Li, H. Wang, L. Huang, and S. Ding, "A Video Summarization Model Based on Deep Reinforcement Learning with Long-Term Dependency," *Sensors*, vol. 22, no. 19, pp. 1-21, 2022.

[4] J. Wu, S. Zhong, and Y. Liu, "Dynamic graph convolutional network for multi-video summarization." *Pattern Recognition*, vol.107, no. 107382, pp. 1-13, 2020.

[5] S. M. U., and B. C. Kovoor, "Towards genre-specific frameworks for video summarisation: A survey," *Journal of Visual Communication and Image Representation*, vol. 62, pp.340-358, 2019.

[6] W. Zhu, J. Lu, J. Li, and J. Zhou, "Dsnet: A flexible detect-to-summarize network for video summarization," *IEEE Transactions on Image Processing*, vol.30, pp. 948-962, 2021.

[7] A. S. Murugan, K. S. Devi, A. Sivaranjani, and P. Srinivasan, "A study on various methods used for video summarization and moving object detection for video surveillance applications," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23273-23290, 2018.

[8] P. Saini, K. Kumar, S. Kashid, A.Saini, and A. Negi, "Video summarization using deep learning techniques: a detailed analysis and investigation," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12347-12385, 2023.

[9] G. El-Nagar, A. El-Sawy, and M. Rashad, "A deep audio-visual model for efficient dynamic video summarization," *Journal of Visual Communication and Image Representation,* vol. 100, no. 8, pp.1-9, 2024.

[10] V. Tiwari, and C. Bhatnagar, "A survey of recent work on video summarization: approaches and techniques," *Multimedia Tools and Applications*, vol.80, no. 18, pp. 27187-27221, 2021.

[11] A.A. Baniya, T. Lee, P. Eklund, and S. Aryal, "Frame Selection Using Spatiotemporal Dynamics and Key Features as Input Pre-processing for Video Super-Resolution Models," *SN Computer Science*, vol. 5, no. 3, pp. 323- 338, 2024.

[12] C. E. Matthews, L. I. Kuncheva, and P. Yousefi, "Classification and comparison of on-line video summarisation methods," *Machine Vision and Applications*, vol. 30, no. 3, pp. 507-518, 2019.

[13] K. Muhammad, T. Hussain, and S. W. Baik, "Efficient CNN based summarization of surveillance videos for resource-constrained devices," *Pattern Recognition Letters*, vol. 130, pp. 370-375, 2020.

[14] H. Gharbi, S. Bahroun, and E. Zagrouba, "Key frame extraction for video summarization using local description and repeatability graph clustering," *Signal, Image and Video Processing*, vol. 13, pp. 507-515, 2019, doi.org/10.1007/s11760-018-1376-8.

[15] S. Jadon, and M. Jasim, "Unsupervised video summarization framework using keyframe extraction and video skimming," In *IEEE 5th International Conference on computing communication and automation (ICCCA)*, IEEE, pp. 140-145, 2020.

[16] Q. Zhong, Y. Zhang, J. Zhang, K. Shi, Y. Yu, and C. Liu. "Key frame extraction algorithm of motion video based on priori," *IEEE Access,* vol. 8, pp. 174424-174436, 2020.

[17] B. Liang, N. Li, Z. He, Z. Wang, Y. Fu, and T. Lu, "News video summarization combining surf and color histogram features, " *Entropy*, vol. 23, no. 8, p.1-14, 2021.

[18] H. M. Nandini, H. K. Chethan, B.S. Rashmi, "Shot based keyframe extraction using edge-LBP approach, "*Journal of King Saud University-Computer and Information Sciences,* vol. 34, no. 7, pp. 4537-4545, 2022.

[19] H. B. Ul Haq, M. Asif, M. B. Ahmad, R. Ashraf, and T. Mahmood, "An effective video summarization framework based on the object of interest using deep learning," *Mathematical Problems in Engineering,* vol. 2022, no. 1, pp. 1-25, 2022.

[20] H. Tang, L. Ding, S. Wu, B. Ren, N. Sebe, and P. Rota, "Deep unsupervised key frame extraction for efficient video classification," *ACM Transactions on Multimedia Computing, Communications and Applications,* vol. 19, no. 3, pp. 1-17, 2023.

[21] Y. Zhang, Y. Li, Z. Cai, X. Wang, J. Zhang, and S. Lam, "Key frame extraction method for lecture videos based on spatio-temporal subtitles," *Multimedia Tools and Applications,* vol. 83, no. 2, pp.5437-5450, 2024.

[22] E. Hato, "Extracting Descriptive Frames from Informational Videos," *Iraqi Journal of Science,* vol. 64, no. 8, pp. 4260-4277, 2023.

[23] N. Varish, "A modified similarity measurement for image retrieval scheme using fusion of color, texture and shape moments," *Multimedia Tools and Applications,* vol. 81, no. 15, pp. 20373-20405, 2022.

[24] V. Mehta, S. Bawa, and J. Singh, "Analytical review of clustering techniques and proximity measures," *Artificial Intelligence Review,* vol. 53, pp.5995-6023, 2020, doi.org/10.1007/s10462-020-09840-7.

[25] A. A. Bushra, and G. Yi, "Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms," *IEEE Access*, vol. 9, pp. 87918-87935. 2021.

[26] J.J. M. Escobar, O. M. Matamoros, I. L. Reyes, R. T. Padilla, and L. C. Hernández, "Defining a no-reference image quality assessment by means of the self-affine analysis," *Multimedia Tools and Applications,* vol. 80, no. 9, pp. 14305-14320, 2021.

[27] W.J. Hadi, A. S. Ajrash, S. M. Salman, and M.T. Ibrahim, "Densenet Model for Binary Glaucoma Classification Performance Assessment with Texture Feature," *Baghdad Science Journal*, 2024, doi.org/10.21123/bsj.2024.9857.

[28] A.H. Sadhin, S. Z. M. Hashim, H. Samma, and N. Khamis, "YOLO: A Competitive Analysis of Modern Object Detection Algorithms for Road Defects Detection Using Drone Images," *Baghdad Science Journal*, vol. 21, no. 6, pp. 2167-2167, 2024, doi.org/10.21123/bsj.2023.9027.