



Efficient Content-Based Video Retrieval in HEVC Standard Using Auto-Correloblock: A Novel Approach

Yaghoub Saberi ^a, Mohammadreza Ramezanpour ^{b,*}, Shervan Fekri Ershad ^a,
Behrang Baraktin ^a

^a Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

^b Department of Computer Engineering, Mobarakeh Branch, Islamic Azad University, Mobarakeh, Iran

Received 05 July 2023, Accepted 22 October 2023

Abstract

This paper proposes a new method for content-based video retrieval in the HEVC standard, which is becoming increasingly popular for video compression. Retrieving compressed videos can be time-consuming due to the need for decompression, but the proposed method utilizes the features of the HEVC standard in compressed mode and introduces a new concept called Auto Correloblock to enable retrieval without full decompression. The method uses the histogram of prediction mode within the standard HEVC frame after normalization, as well as the value and spatial distance of the blocks, to retrieve videos. The simulation results demonstrate the high efficiency of the proposed method, with an average recall of 96.27% and an average precision of 77.34% for 50 search operations. This approach outperforms similar methods and has potential applications in various fields that use the HEVC standard. Overall, this paper presents a promising solution to the challenge of content-based video retrieval in the HEVC standard, which can save time and improve efficiency in various applications.

Keywords: Content-based video retrieval, HEVC standard, PU size, Auto-correloblock.

1. Introduction

Videos have the ability to show the details of an event better and more accurately than the image of an event. For this reason, it is considered in many cultural, educational, economic, medical, advertising, industrial, architectural, tourism, military and law enforcement applications. Due to the high volume of videos, it is necessary to compress them using different standards so that their volume can be reduced to an acceptable size and have better storage capabilities. The High-Efficiency Video Coding (HEVC) standard is the latest video compression standard. This standard is defined by ITU-T and ISO/IEC where the final videos are of high quality and small size. Considering that the video database is increasing rapidly, one of the main challenges for users is to search and retrieve the required videos from a large number of videos. Various video retrieval methods are used as a tool to search and find various videos in video databases and the web. Also, video retrieval methods can be used as a tool to analyse different videos and check their content. The

main challenge to retrieve compressed videos is decompressing them and converting them to pixel mode, which takes time. If it is possible to retrieve the video without full compression, the retrieval speed will increase and thus the efficiency of the retrieval system will be improved. One of the major challenges we encounter is the time-consuming recovery of compressed videos, which can lead to additional overhead. In this paper, our objective is to address this issue by solving the time-consuming problem that arises from compressed videos in the compressed domain.

2. Literature Review

This section begins with a brief overview of various Content-Based Video Retrieval (CBVR) methods. It is worth noting that there has been no research conducted on CBVR using the HEVC standard beyond 2017. Hence, the papers published prior to 2017 have been considered for review in detail.

*Corresponding Author. Email: ramezanpour@mau.ac.ir

CBVR methods include segmentation, feature extraction, dimensionality reduction, and machine learning [1].

Segmentation is a crucial step in CBVR, as it helps to separate different parts of the video. Object detection methods, such as convolutional neural networks, are commonly used for this purpose. The video is divided into different parts, and features related to each part are extracted. These features are then used to search for similar videos. Other methods, such as motion analysis and image processing techniques like color analysis, shape analysis, and texture analysis, can also be used for segmentation in CBVR. Jain et al [2] proposed a semantic segmentation model for frame summarization of videos, which combines different machine learning algorithms. Videos are queried using KD tree. In another study, Raviprakash and his colleagues [3] proposed a method that divides the video into different views and selects key frames to determine the boundaries of the shot. This approach provides a more efficient video retrieval.

Feature extraction is a crucial step in CBVR, as it helps to identify and extract relevant features from video content. These features can include color, texture, shape, movement, sound, and other visual and auditory features of video frames. The goal of feature extraction is to reduce the dimensionality of the video and identify the most important aspects of the video content. Convolutional Neural Networks (CNN) are commonly used for feature extraction in CBVR because they are effective in identifying complex patterns and features in images and videos. Other techniques, such as GIST [4] and SIFT [5], can also be used for feature extraction. Once the features are identified, they can be used to calculate similarity measures between videos, such as Euclidean distance, correlation coefficient, or cosine similarity. Kumar and Seetharaman [6] extracted various features using deep learning techniques and compared them with other existing feature extraction techniques, such as histogram, gradient-oriented methods (HOG), local binary patterns (LBP), and convolution neural network (CNN) methods. They showed that their proposed system produces better results for the same query compared to existing techniques based on Recall and Precision in video retrieval. Chivadshetti et al. [7] proposed a system that performs video retrieval in three different stages. In the first stage, video segmentation and extraction of meaningful key frames are done. In the second step, OCR, HOG, and ASR algorithms are applied to the keyframe to extract the textual keyword. In the third stage, Color, Texture, and Edge features are

extracted. Finally, the search similarity measurement is performed on the extracted features that are stored in the database, and the output is provided. Zhang et al. [8] presented a content-based video retrieval method that improves both video compression and retrieval. They extracted features of keyframes, which are part of the structure of compressed files, and defined key objects that contain moving regions of video that move in successive frames. By developing these key objects, they were able to retrieve similar videos that are identical and can be searched in content-based video retrieval.

Dimensionality reduction is an important technique used in content-based video retrieval to reduce the number of features used to display a video. This helps to improve the efficiency and speed of the retrieval process and reduce storage requirements for the video database. Several techniques are used for dimensionality reduction, including principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed random neighbor embedding (t-SNE). These techniques can be used to reduce the number of features used to display a video while preserving the most important information about the video content. Pacharaney et al. [9] suggested that by extracting suitable features and reducing the dimensions in both stages of the video search and retrieval system, the efficiency and accuracy of the retrieval process can be improved. They used principal component analysis to transform the original high-dimensional data into a new low-dimensional coordinate system, then used sparse representation before applying similarity matching to quickly and accurately search and retrieve videos. Dhulavvagol et al. [10] proposed using different techniques for feature extraction and similarity calculation to retrieve related videos, such as shot boundary detection based on histogram, PCA Shift, Gist, and SURF (High-Speed Features). They used a second-degree equation for feature extraction and similarity calculation, which reduces the dimensionality and results in better performance of content-based video retrieval. Their results show that the proposed method provides better results compared to other techniques.

Machine learning methods are a powerful tool for developing predictive models that can identify and retrieve similar videos. These algorithms can be trained on large datasets of videos that include content and descriptive features associated with videos. By analyzing these features, machine learning algorithms can identify patterns and relationships between them, allowing them to predict similarities between videos.

This approach can be used to quickly and accurately search and retrieve videos based on their content and other descriptive features. Overall, machine learning methods offer a powerful and efficient way to develop predictive models for content-based video retrieval, making it easier to find and organize videos based on their content and other relevant features. Sharma et al. [11] presented an object classification technique using a machine learning approach that uses Haar-like features to train the classifier. Feature computation is performed using integrated image representation and they trained the classifier offline using stepwise incremental modelling and multiclass exponential loss function (SAMME). The experimental results shown that the proposed method can accurately classify the objects in the video into the corresponding classes. Kumar et al. [12] proposed a content-based video retrieval framework using spatiotemporal intensive features learned by machine learning. For effective hash code learning, the proposed framework is taught in two steps. The first stage learns the dynamics of the video and in the second stage it learns the compressed code using the time changes of the video learned from the first stage. Their results show that the proposed approach is able to improve the performance compared to existing methods.

Also, during recent years, many studies and research works on CBVR have been presented with different approaches and techniques to improve the accuracy and speed of video retrieval. One approach is to extract and use multiple features to represent the video. For example, a combination of color, texture, shape, and

motion features can be used to display a video. Chen et al. [13] optimized and parallelized a series of typical visual feature extraction applications in CBVR. They performed a detailed performance analysis of these parallel programs on a quad-core, dual-socket system to identify bottlenecks and provide better real-time performance. Miao et al. [14], extracted features from a CBVR system with 8 and 16 cores parallel processing, performed a fast video retrieval that had better results than similar methods. Another approach is to use deep learning techniques, such as Convolutional Neural Networks (CNN), to extract relevant features from video frames [15-17]. CNNs are particularly effective in identifying complex patterns and features in image and video data, which can significantly improve the accuracy of CBVR.

This paper is organized as follows. Section 2 is the background of content-based video retrieval and provides explanations about the HEVC standard and Auto-correloblock. In section 3, explanations about the proposed method and different stages of implementation proposed CBVR are provided. Section 4 describes the parameter settings of the proposed algorithm and database specifications, then the simulation results are analyzed, and Section 5 describes the conclusions and suggestions for future work. Table 1 presents a comparison of the achievements, advantages, and disadvantages of similar articles to the current research. This table provides a useful overview of the strengths and weaknesses of existing research in the field, highlighting areas where the current study can make a valuable contribution.

Table1
Comparison of similar paper: Achievements, Advantages, and Disadvantages

Ref.	Method	Achievements	Advantages	Disadvantages
[3]	Segmentation	This paper proposes a novel method for shot segmentation in videos using visual features such as color, texture, and motion. The proposed approach is effective in dividing a video into smaller, meaningful segments that can be analyzed and searched efficiently.	high accuracy in shot segmentation	high computational complexity
[6]	Feature extraction	This paper, presents a method for content-based video retrieval using deep learning techniques. They utilize a modified version of the VGG_16 deep learning model to extract features from videos. They claimed that their proposed approach improves the accuracy and effectiveness of video retrieval by leveraging the power of deep learning.	The use of deep learning feature extraction by modified VGG_16 allows for more accurate and efficient video retrieval compared to traditional methods.	The dataset used in the experiments is relatively small, which may limit the generalizability of the results to larger and more diverse datasets.
[7]	Feature extraction	This paper presents an approach that combines feature extraction techniques with personalized result ranking in video retrieval.	consideration of personalized preferences in result ranking can enhance user satisfaction and provide more tailored video recommendations.	The absence of specific information about the feature extraction techniques used in the research hinders a thorough assessment of their effectiveness and suitability.
[8]	Feature extraction	This paper uses machine learning algorithms to analyze the content of videos and extract features that can be used to search for specific content within the videos. This allows	proposing a unified solution for both content-based video retrieval and compression	High computational complexity

		users to quickly and easily find the content they are looking for without having to manually search through large amounts of video data.		
[9]	Dimensionality reduction	The paper presents a novel method for dimensionality reduction that combines the advantages of both global and local feature descriptors to achieve fast and accurate video retrieval. The proposed method uses a combination of PCA and LDA techniques to reduce the dimensionality of the feature space, which helps to speed up the search process while maintaining high retrieval accuracy.	The combination of PCA and LDA techniques can effectively reduce the dimensionality of the feature space, which can help to speed up the search process.	The use of PCA and LDA techniques to reduce the dimensionality of the feature space may result in some loss of information, which could potentially lead to a decrease in retrieval accuracy.
[10]	Dimensionality reduction	The paper introduces an adaptive and dynamic approach that can adjust the dimensionality reduction process based on the characteristics of the video dataset being analyzed.	improving the retrieval performance and reducing the computational cost.	High computational complexity
[11]	Machine learning	this paper presents an approach for video object classification, which involves using offline feature extraction and machine learning algorithms for improved accuracy.	Extracting of offline feature can reduce the computational load during real-time video object classification.	The offline feature extraction process may require a large amount of storage space to store the extracted features.
[12]	Machine learning	this paper proposed an approach to learning compact spatio-temporal features for efficient content-based video retrieval	high retrieval accuracy while maintaining low computational	High computational complexity

3. Background

CBVR is an important field in video processing that seeks to find videos similar to a specific video. In general, video retrieval methods are divided into non-compressed and compressed domains. In the non-compressed domain, the retrieval operation is performed on the feature vector extracted from the pixels of the video frame, such as histogram, edge, and texture, while in the compressed domain, retrieval is performed using the feature vector, such as block size (PU size), prediction modes, motion vectors and residual coefficients are performed. This paper is presented in the compressed domain where the video retrieval operation is implemented using the HEVC standard and a concept called Auto-correloblock. For a better understanding of the readers, explanations about the HEVC standard and Auto-correloblock are provided.

Currently, the HEVC compression standard is used as the best way to compress videos. Among the reasons for the success and popularity of this standard can be mentioned its high quality in low bit rate and wide support of broadcasting devices. The purpose of video compression is to provide higher quality video content while using less bandwidth. HEVC owes its high compression largely to its intra and inter prediction. Intra prediction is a method to reduce the amount of data required to display a video frame by predicting the pixel values in a frame based on adjacent the pixel values. Intra prediction in HEVC uses different modes to predict pixel values in a block. These modes include

angular, DC and planar modes. Angular mode predicts the values of pixels in a block using a directional prediction based on the values of pixels in adjacent blocks. DC mode predicts the values of pixels in the block by using the average value of the pixels in the block. Planar mode predicts pixel values in a block by using a linear function of pixel values in adjacent blocks [18, 19].

In the HEVC standard, to reduce the computational complexity and improve the video quality, the video frame is divided into parts called CTU (Coding Tree Unit). Each CTU contains one or more blocks called PUs, which can have different sizes. PU in HEVC starts with size 64×64 by default and is divided into smaller blocks with sizes 32×32, 16×16, 8×8 and 4×4 as a quad-tree. This blocking in HEVC makes it possible to improve image quality and reduce the size of video files. Fig. 1 shows the CTU partition structure in 5 depths from 64×64. Dividing large blocks into smaller blocks in HEVC is done to improve image quality, reduce computational complexity, and reduce video file size. Larger block sizes can cause large differences in image quality and in some cases may increase image noise. By dividing large blocks into smaller blocks, the differences in image quality can be reduced and the image reconstructed more accurately. Also, in HEVC, as the number of smaller blocks increases, the number of predictions required for each block increases, which increases the computational complexity. Fig. 2 shows how to divide a CTU with 64×64 dimensions into 32×32 to 4×4 blocks.

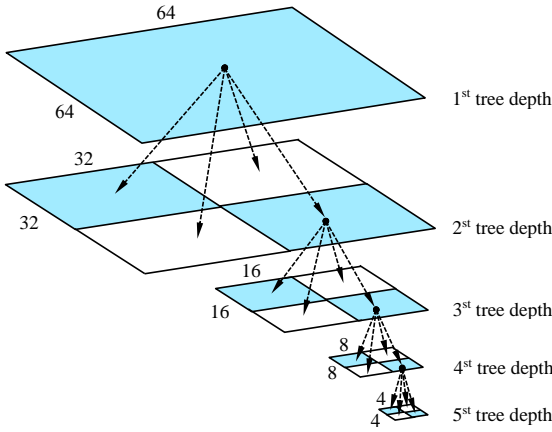


Fig. 1. Partitioning structure of a CTU from 64x64 to 4x4 in 5-depth quad tree [20]

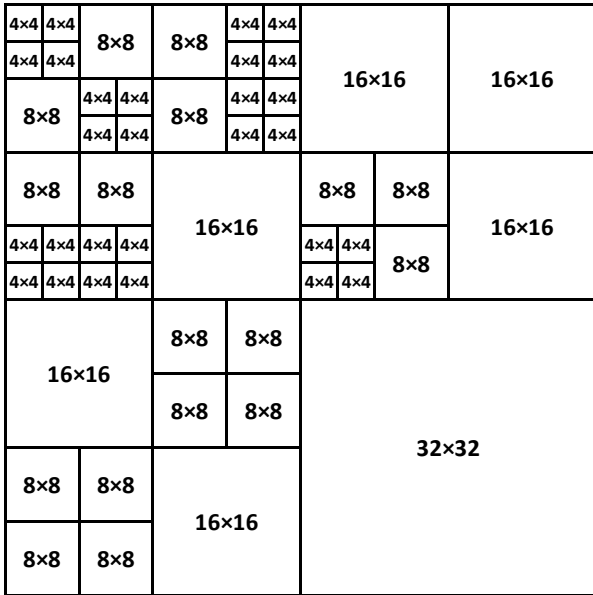


Fig. 2. How to divide a CTU with dimensions 64x64 into 32x32, 16x16, 8x8 and 4x4 blocks

In the proposed method for video retrieval, a new concept called Auto-correloblock is used in the frames that are predicted within the frame. Auto-correloblock is a feature extraction technique that can be used in content-based video retrieval (CBVR) that calculates the distribution of block size values in a video frame with respect to their spatial relationships using (1).

$$Auto\text{-}correloblock_{nb}^{(k)}(I) \square \Pr_{b_i \in I_L, b_j \in I_L} [b_i \in I_L \square |b_i - b_j| = k] \quad (1)$$

In this relationship, K is the distance between two blocks b_i and b_j . To extract the Auto-correloblock

histogram, each image is quantized into 5 types of blocks, which have values from 1 to 5, and the distance set is defined as $K = \{1, 2, 3, 4, 5\}$.

Auto-correloblock calculates the spatial correlation between block size values in a frame of video by calculating the probability of finding a block with a certain value at a certain distance from another block with the same block size value. Fig. 3 shows how to calculate the spatial correlation of the Auto-correloblock technique. This information can then be used as an image feature vector. The Auto-correloblock feature extraction technique is useful in CBVR because it captures the spatial distribution of block size values in a video frame that closely resembles the texture of similar frames, which can be important for identifying similar images. It can also be used together with other feature extraction methods, such as color histograms, to improve the accuracy and speed of image retrieval.

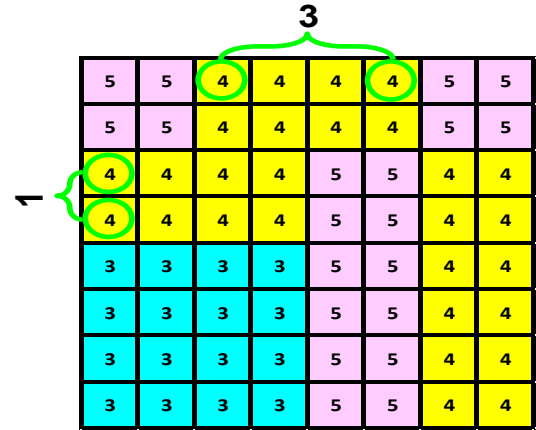


Fig. 3. Auto-correloblock spatial distance in a 32x32 block

4. Proposed Method

Video frames contain important information from the video, whose features can be extracted and used in video retrieval. To retrieve the video in the non-compressed domain, the video must first be converted from the compressed domain to the pixel domain so that its feature vector can be extracted and the retrieval operation can be performed. This imposes relatively high processing time and overhead on the retrieval system. Considering that the videos are large and the videos need to be stored in compressed form, if video retrieval can be done from compressed frames, the retrieval time will be reduced. Extracting video features in the compressed domain from video frames is more complicated than the non-compressed domain due to the compression process. In HEVC compressed domain, the values of PU sizes, Prediction Modes, motion vectors and residual coefficients that are

available in the compressed domain are used for retrieval.

PU sizes are one of the important factors in video coding. To reduce the size of video data, frames are divided into smaller, more compressible units, called PUs. In HEVC, PU size can be one of 64×64, 32×32, 16×16, 8×8 and 4×4. Choosing the right PU size in HEVC can help reduce computational complexity and increase video quality. In general, a smaller PU size improves the image quality in more complex parts, but may increase the computational complexity. A larger PU size reduces the computational complexity but may cause noise in smaller parts of the video. Therefore, choosing the appropriate PU size in HEVC is very important because of its significant impact on video quality and computational complexity. HEVC selects the best PU size based on rate-distortion optimization process.

In this paper, a new method for content-based video retrieval in the compressed domain using PU sizes and Auto-correloblock technique is presented. The proposed video retrieval method is performed by using the spatial features of the videos encoded with the HEVC standard. For this reason, video retrieval in the proposed method does not require video decompression and has a much lower computational load than the non-compressed domain. On the other hand, the structure of HEVC blocking is such that in I-frame, regions with smooth texture (more smooth) are divided into larger blocks (64×64/32×32) and regions with finer texture and higher complexity are divided into smaller blocks (8×8/4×4) are divided. Therefore, it can be concluded that the size of the blocks in HEVC represents the texture of the video frame and can be used for video retrieval. Further details on the HEVC standard blocking method can be found in [21]. In the following, explanations are given details of the proposed video retrieval method.

The proposed video retrieval method includes five steps.

Step 1: First, the database videos are compressed with the HEVC standard and stored in the video database.

Step 2: PU size normalization is performed on I-frames of each video. In the HEVC standard, we have 5 types of PU size, which are 64×64, 32×32, 16×16, 8×8 and 4×4. For PU size normalization, first the size of I-frame PUs is extracted from HEVC videos, then the size of 64×64, 32×32, 16×16, and 8×8 PUs is multiplied by 256, 64, 16, and 4, respectively. to break all PUs larger than 4×4 into 4×4 PUs. Then, instead of 64×64, 32×32, 16×16, 8×8, and 4×4 PUs, the values 1, 2, 3, 4, and 5 are recorded in the corresponding 4×4 PUs (Values 1

to 5 are the depth of the quad-tree of PUs). It is noted that the PU size normalization operation is performed in order to increase the accuracy of retrieval and remove the effect of resolution. Fig. 4 (a) shows an example of CTU before PU size normalization. The values written in each cell represent PU sizes. Fig. 4 (b) shows the same CTU after PU size normalization. The values recorded in each block indicate the normalized PU sizes, which have values between 1 and 5.

Step 3: After the PU size normalization, the I-frames of each video are processed using the Auto-correloblock technique and their feature vector which includes the same PU size values and the spatial distance of each of the normalized PU sizes in the I-frame.

In the Auto-correloblock technique, the value of K represents the spatial distance between two PUs of the same size, and its value is determined by the user. Auto-correloblock technique, upon receiving the K value, counts all the same PU sizes with a distance from 1 to K. For example, if the value of K = 4, the Auto-correloblock technique calculates all the same PU sizes with the distance of 4 and shows them as 5 bins. Histogram bins in a group, includes: bin 1, frequency of PUs of depth 1, bin 2, frequency of PUs of depth 2, bin 3, frequency of PUs of depth 3, bin 4, frequency of PUs of depth 4 and bin 5 will be the frequency of PUs of depth 5 in I-frame. Fig. 5 shows an example of the of Auto-collreloblock histogram.

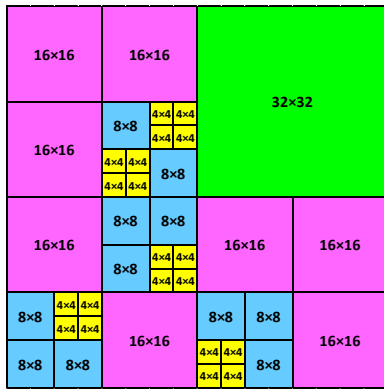
Step 4: PU size normalization of query video is performed on I-frames of the query video, as in the second step.

Step 5: I-frames of the query video, like the third step after the PU size normalization operation, are processed using the Auto-correloblock technique and their feature vector is extracted. Then the features of the query video are compared with the features in the feature database using the Manhattan similarity according to (2):

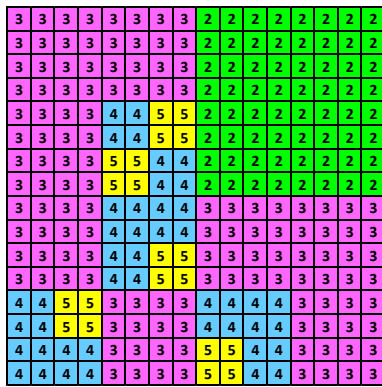
$$D_{man}(Q, T) = \sum_{i=0}^{N-1} |Q_i - T_i| \quad (2)$$

Where $Q = \{Q_0, Q_1, \dots, Q_{N-1}\}$ and $T = \{T_0, T_1, \dots, T_{N-1}\}$ are the query and target feature vectors, respectively, and $D_{man}(Q, T)$ is the Manhattan distance.

Fig. 6 shows the retrieval flowchart of the proposed method. The parts shown with dashed lines are related to the proposed method, which includes Auto-coeloblock PU sizes of database videos and query video.



(a)



(b)

Fig. 4. An example of PU size normalization operation in a CTU

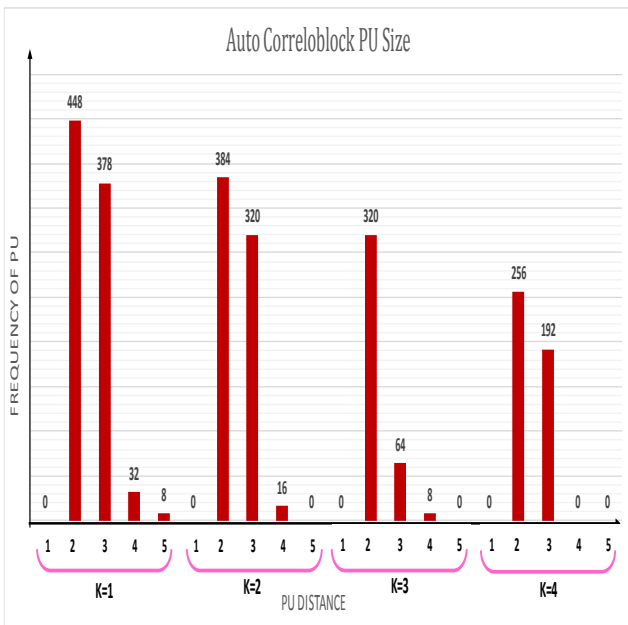


Fig. 5. An example of Auto Collreloblock histogram with a distance value of K = 4 for a CTU.

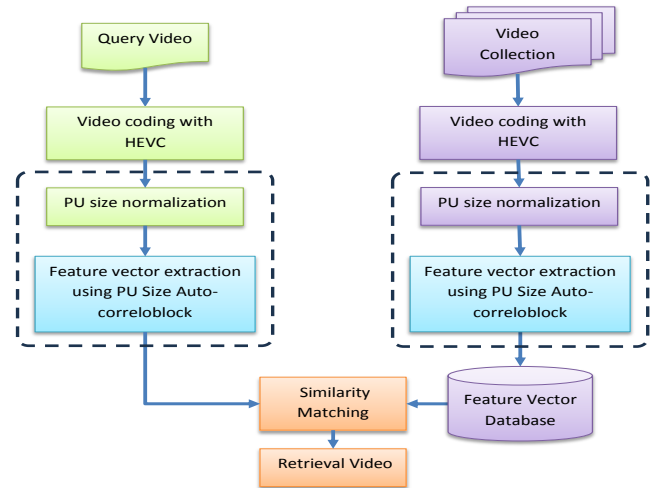


Fig. 6. Flowchart of proposed video retrieval method.

5. Simulation Results

To evaluate the proposed method, its performance was compared with the results of the methods presented in papers [22] and [23] on the same database. In the experiments, Precision and Recall criteria have been used to evaluate the performance of the proposed method. In CBVR, Precision means the number of retrieved videos that contain the desired content, divided by the total number of videos retrieved by the system. In other words, Precision indicates how much of the videos retrieved by the retrieval system match the user's query video. Recall means the total number of retrieved videos that contain the desired content, divided by the total number of videos in the database. In order to evaluate the proposed method, the UCF50 video dataset is used, which is one of the largest datasets in the field of video retrieval and includes 50 categories of different real videos. These videos are taken from different sports such as basketball, boxing, walking, swimming, football, volleyball, etc. Each category contains 1 to 13 videos and the total dataset contains 687 sports videos. The purpose of creating this dataset is to recognize sports videos for applications such as video retrieval, video classification, sports movement recognition, and player state recognition. Fig. 7 shows some videos of the UCF50 video dataset. The main challenge in the UCF50 video dataset for video retrieval is the large differences in lighting conditions, size and scale, viewing angle, and motion speed in different videos. Also, some videos were recorded in unfavorable conditions such as low light, camera shake, and obstructions in the foreground or background.

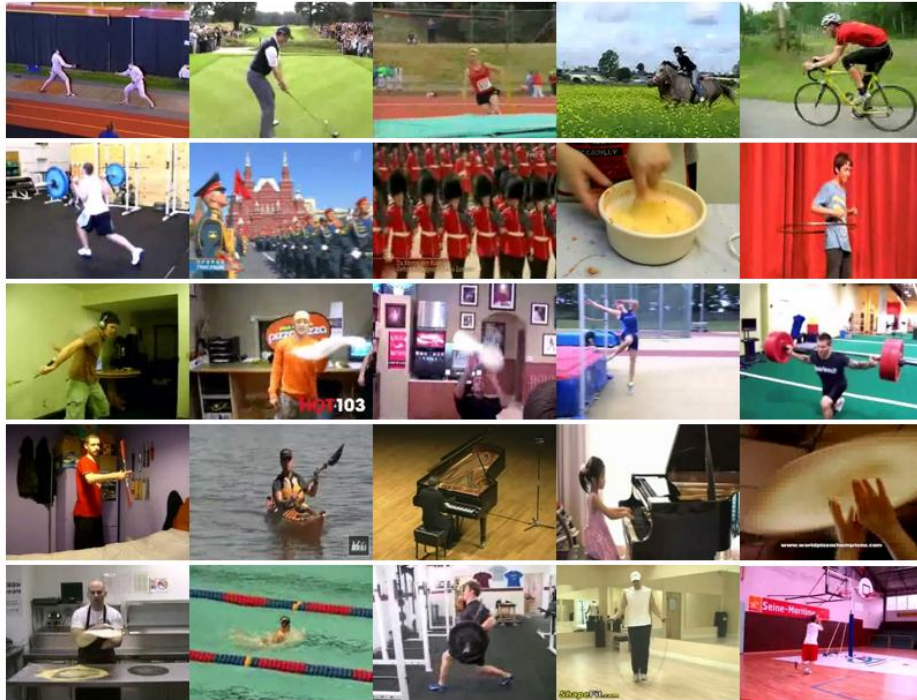


Fig. 7: A number of videos from different videos of the UCF50 video dataset

In order to compare the results of the experiments, the proposed method was implemented and executed with method [22] and method [23] in the UCF50 dataset with query videos and the same test settings. The test parameters set 30 frames from the beginning of each video were encoded using HEVC reference software (HM-16.25) with IBBPBBP structure, GOP with P/B ratio equal to 18/9. The features of the query video were compared with the features in the feature database using the Manhattan similarity criterion according to (2) and the most similar features to the query video were searched and sorted in descending order Fig. 8. Comparing the results of methods [22], [23] and the proposed method in Fig. 9, show that the proposed method provides better retrieval accuracy in most cases compared to methods [22] and [23], and the retrieval videos match the query video. This superiority can be due to the use of the Auto-correloblock technique and the appropriate extracted features in the compressed

domain and the use of a better similarity criterion for video retrieval.

In the proposed method, the average Recall is 96.27% and the average Precision is 77.34% for the number of 50 search operations, which shows that this method has high accuracy in finding related videos. Therefore, it can be said that the proposed method is effective and acceptable in video retrieval.

Fig. 9 shows the Precision and Recall curves for the proposed method and the methods [22] and [23] for testing video retrieval. In videos where the subjects have a high movement speed (Basketball), the proposed method performs better in most cases, and in videos where the subjects have an average movement speed (Military Parade) and low (Playing Violin), the proposed method is close to the method [23]. But on average, the performance of the proposed method is better than the other two methods.

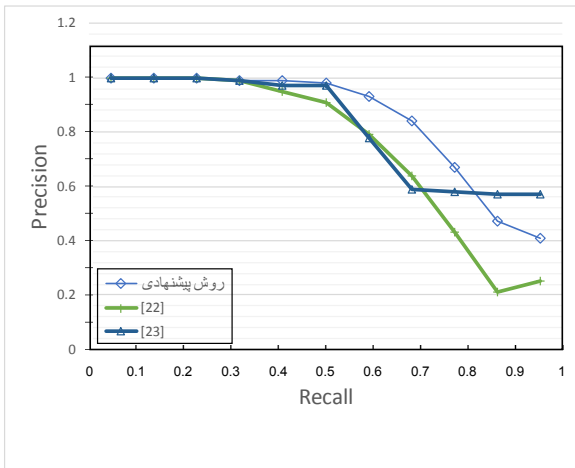
Method	Query video	Retrieved videos
[22]		
		
[23]		
		
proposed method		
		

Fig. 8. Video retrieval results with different methods using the UCF50 dataset.

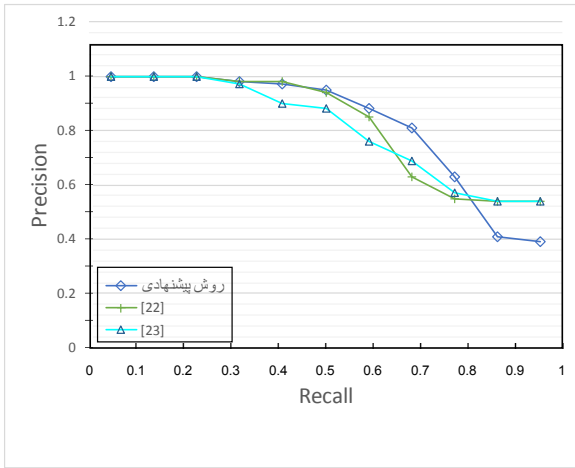
6. Conclusion

In this paper, a new method for video retrieval in compressed domain is presented. This method is implemented based on feature extraction from video frames in compressed domain, using the Auto-correloblock technique that calculates both the value and the spatial distance of blocks of the same size. By using this method, the time and cost of the retrieval process is reduced effectively. In the proposed method, first the videos are coded with the HECV standard, and then their feature vectors are extracted using the Auto-correloblock technique. Various evaluations for video retrieval using the proposed method have been performed and compared with

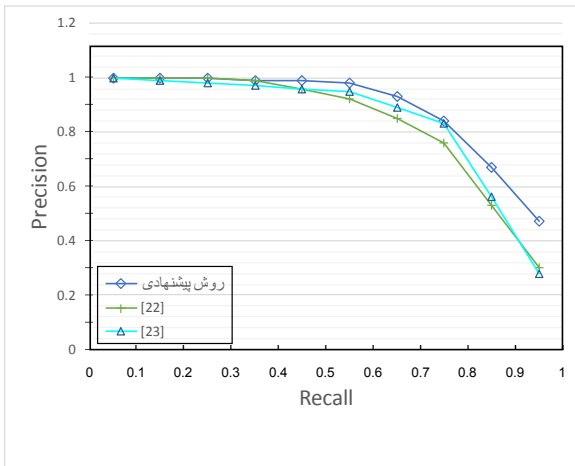
other methods. The simulation results show that in the proposed method, the average Recall value is 96.27% and the average Precision is 77.34%, which has better performance in both Recall and Precision compared to other methods. Therefore, the proposed method can be effectively used in the video retrieval process in the compressed domain. As a suggestion for future work, the method proposed in this paper can be combined with other methods such as color and texture for CBVR. It is also possible to combine the proposed method with deep neural networks to extract features of videos and use machine learning algorithms to search for similar videos.



a) Basketball



b) Military parads



c) Playing Violin

Fig. 9. Precision and Recall chart for video retrieval in methods [22], [23] and proposed method for videos a) Basketball b)

Reference

- [1] N. Spolaôr, H. D. Lee, W. S. R. Takaki, L. A. Ensina, C. S. R. Coy, and F. C. Wu, "A systematic review on content-based video retrieval," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103557, 2020.
- [2] R. Jain, P. Jain, T. Kumar, and G. Dhiman, "Real time video summarizing using image semantic segmentation for CBVR," *Journal of Real-Time Image Processing*, vol. 18, pp. 1827-1836, 2021.
- [3] N. Raviprakash, M. Suresh, A. Rathis, A. Yadav, D. Devarla, and G. Nagaraj, "Shot Segmentation for Content Based Video Retrieval," *email: drssridhar@yahoo.com*, p. 74, 2015.
- [4] R. Desai and B. Sonawane, "Gist, hog, and dwt-based content-based image retrieval for facial images," in *Proceedings of the International Conference on Data Engineering and Communication Technology: ICDECT 2016, Volume 1*, 2017: Springer, pp. 297-307.
- [5] S. Gupta, N. Gupta, and S. Kumar, "Evaluation of object based video retrieval using SIFT," *Int. J. Soft Comput. Eng.*, vol. 1, pp. 1-6, 2011.
- [6] B. S. Kumar and K. Seetharaman, "Content based video retrieval using deep learning feature extraction by modified VGG_16," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 9, pp. 4235-4247, 2022.
- [7] P. Chivadshetti, K. Sadafale, and K. Thakare, "Content based video retrieval using integrated feature extraction and personalization of results," in *2015 International Conference on Information Processing (ICIP)*, 2015: IEEE, pp. 170-175.
- [8] H. Zhang, J. Y. Wang, and Y. Altunbasak, "Content-based video retrieval and compression: A unified solution," in *Proceedings of International Conference on Image Processing*, 1997, vol. 1: IEEE, pp. 13-16.
- [9] U. S. Pacharaney, P. S. Salankar, and S. Mandalapu, "Dimensionality reduction for fast and accurate video search and retrieval in a large scale database," in *2013 Nirma University International Conference on Engineering (NUiCONE)*, 2013: IEEE, pp. 1-9.
- [10] P. M. Dhulavvagol, S. Totad, A. S. Meti, and V. Shashidhara, "An adaptive and dynamic dimensionality reduction method for efficient retrieval of videos," in *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21-22, 2018, Revised Selected Papers, Part I 2*, 2019: Springer, pp. 624-639.
- [11] C. M. Sharma, A. K. S. Kushwaha, R. Roshan, R. Porwal, and A. Khare, "Intelligent Video Object Classification Scheme using Offline Feature Extraction and Machine Learning based

- Approach," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 1, p. 247, 2012.
- [12] V. Kumar, V. Tripathi, and B. Pant, "Learning compact spatio-temporal features for fast content based video retrieval," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 2, pp. 2402-2409, 2019.
- [13] Y. Chen, E. Li, J. Li, and Y. Zhang, "Accelerating Video Feature Extractions in CBVIR on Multi-core Systems," *Intel Technology Journal*, vol. 11, no. 4, 2007.
- [14] Q. Miao, Y. Chen, J. Li, Q. Zhang, Y. Zhang, and G. Chen, "Parallelization and optimization of a CBVIR system on multi-core architectures," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, 2009: IEEE, pp. 1-8.
- [15] N. Spolaôr, H. D. Lee, N. A. Maciejewski, L. A. Ensina, W. SR, and C. S. Takaki12, "Deep learning approaches in content-based video indexing and retrieval."
- [16] C. Kehl and A. L. Varbanescu, "Towards Distributed, Semi-Automatic Content-Based Visual Information Retrieval (CBVIR) of Massive Media Archives," *Advances in neural information processing systems*, pp. 1097-1105, 2012.
- [17] D. Zafirova, "Shot boundary detection: a fundamental base for automatic video analysis," Wien, 2020.
- [18] Y. Saberi, M. Ramezanpour, and R. Khorsand, "An efficient data hiding method using the intra prediction modes in HEVC," *Multimedia Tools and Applications*, vol. 79, pp. 33279-33302, 2020.
- [19] Y. Saberi, M. Ramezanpour, and R. Khorsand, "Information Hiding Based on Matrix Embedding and Motion Vector in the HEVC Standard," *Journal of Advanced Signal Processing*, vol. 3, no. 2, pp. 213-225, 2019.
- [20] G. Correa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Complexity scalability for real-time HEVC encoders," *Journal of Real-Time Image Processing*, vol. 12, pp. 107-122, 2016.
- [21] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," in *Integrated circuit and systems, algorithms and architectures*, vol. 39: Springer, 2014, p. 40.
- [22] F. Akrami and F. Zargari, "An efficient compressed domain video indexing method," *Multimedia tools and applications*, vol. 72, pp. 705-721, 2014.
- [23] F. Rahmani and F. Zargari, "Compressed domain visual information retrieval based on I-frames in HEVC," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 7283-7300, 2017.