

RESEARCH

Open Access



# Video block and FABEMD features for an effective and fast method of reporting near-duplicate and mirroring videos

Abderrahmane Adoui El Ouadrhiri\* , Said Jai-Andalousi and Ouail Ouchetto

\*Correspondence:  
a.adouielouadrhiri-  
etu@etude.univcasa.ma  
LR2I Laboratory, FSAC,  
Hassan II University  
of Casablanca, B.P. 5366,  
Maarif 20100, Casablanca,  
Morocco

## Abstract

Near-duplicate video content has taken the large storage space in the age of big data. Without respecting the copyright ethic, social media users mirror, resize, and/or hide certain online video content and re-upload it as new data. This research aims to avoid the complex and high-dimensional matching and present an efficient approach for detecting near-duplicate videos, this detection is based on feature extraction using visual, motion, and high-level features. Fast and adaptive bidimensional empirical mode decomposition is used to preserve the relevant data to the furthest extent possible during the low/high-frequency transition and vice-versa. In addition, for a generic model, the invariant moments are added to the aforementioned features in order to reinforce them against different video transformations such as rotating and scaling. Furthermore, the video frames are divided into blocks with a fixed number of features, this set of features is represented by a signature, where its mean and standard deviation represents a single video map allowing easy similarity computation. The F1-score and accuracy are used to evaluate the results of this study; the relevant results are ranked by  $Top_1$  for the best result, and the five top-ranked results are presented by  $Top_5$ . Further, our result of  $Top_1$  reached over 80% on F1-score, with a difference of  $\pm 4\%$  from the  $Top_5$  results, and it is over 90% on Accuracy using different datasets, such as UCF11, UCF50, and HMDB51.

**Keywords:** Near-duplicate videos, Mirroring videos, Video signature, Block features, Visual and motion data

## Introduction

Big data has ushered in a new era of data. Despite significant progress in recent years, due to rapidly rising search needs, Content-Based Video Retrieval (CBVR) remains a compelling research topic on the Internet era of big data. Thus, the massive number of videos that are circulating on social media platforms (e.g., YouTube, Facebook, Instagram, etc.) poses significant hurdles for analyzing and retrieving data. As an example, approximately 500 hours of video are uploaded to YouTube every minute [1].

Therefore, in this age of information explosion and resource sharing, CBVR has received substantial attention and suggests a variety of content similarity patterns [2] as a representation of videos via global and/or local features (signature) [3] for compressed

video formats like Moving Picture Experts Group (MPEG) (MPEG-4 AVC/H.264). The local feature is invariant to partial occlusion, background, and viewpoint variants. The global feature processes all pixels that can be potentially used to compute the descriptor, including visual and motion geometric structures. In addition, the frame decomposition strategy for exploring the deep data of visual aspects is not applicable in all conditions, and certain techniques lose data via video manipulation, like low to high-frequency transformation.

To find the most relevant videos from a data set for a query video, CBVR uses similarity relationships among videos. Thus, video retrieval requires a schema that can identify near-duplicate videos and index the video content to avoid fraud, protect the copyrights of content owners, and alleviate storage and high-dimensional computing issues.

Near-duplicate video data assumes different forms, such as mirroring videos, hiding certain content, and changing the size or dimension of the video. Video processing can extract hundreds to thousands of features for all video frames based on visual characteristics, such as color and texture, which are insufficient to detect similarities. At the same time, it takes a long time and a large number of resources.

On the other side, when the size or illumination conditions change, the intrinsic feature alters rapidly in the local area due to nonlinear distortion caused by inconsistent sizing, illumination intensity, and motion direction. The inherent structure can then be considered by examining these high-frequency patterns. Fast and Adaptive Bidimensional Empirical Mode Decomposition (FABEMD) [4] is a Bidimensional Empirical Mode Decomposition (BEMD) [5] method that can decompose an image into multilayer high-frequency images that represent detail features and low-frequency images that represent analogy features without requiring time-consuming plane interpolation computation.

Feature extraction and computational efficiency are the two primary keys of video retrieval systems. The first focuses on extracting robust video features; the second aims to reduce the time-consuming retrieval and video-matching process. This trend is accompanied by a strong focus on retrieval accuracy, with less attention to matching speed. Thus, a full reduction of data is required to comprehensively represent the video content.

Recently, Convolutional Neural Networks (CNN) have become popular in a variety of fields, such as image/video classification [6], retrieval [7], matching [8] and action recognition [9].

CNN-based video retrieval algorithms, unlike standard approaches, can learn the feature representation from a sequence of frames (frame by frame) and determine the similarity relationship among videos. On the other hand, videos constitute low-level visual content in each frame and high-level temporal consistency among frames. Consequently, CNN algorithm cannot function effectively due to a lack of modeling of the temporal coherence among subsequent video frames [10]. The two-stream CNN model [11] uses two inputs, first by handling the salient content of the Red, Green and Blue colors (RGB) frames and second by handling the salient content of the optical flow. The outputs of the two training streams are then combined to produce an outcome. In addition, instead of using all the spatial or temporal features of data, the concept underlying the use of salient content is to use specific regions of a video frame.

It is commonly known that deep learning applications satisfactorily outperform other methods. However, the computation time of deep learning algorithms grows exponentially with the number of video data sets and features [10], but it does not provide any information regarding the core characteristics employed. In general, deep learning-based algorithms offer good retrieval accuracy for real-time video retrieval, but their high computing cost limits their use in video retrieval and other related applications.

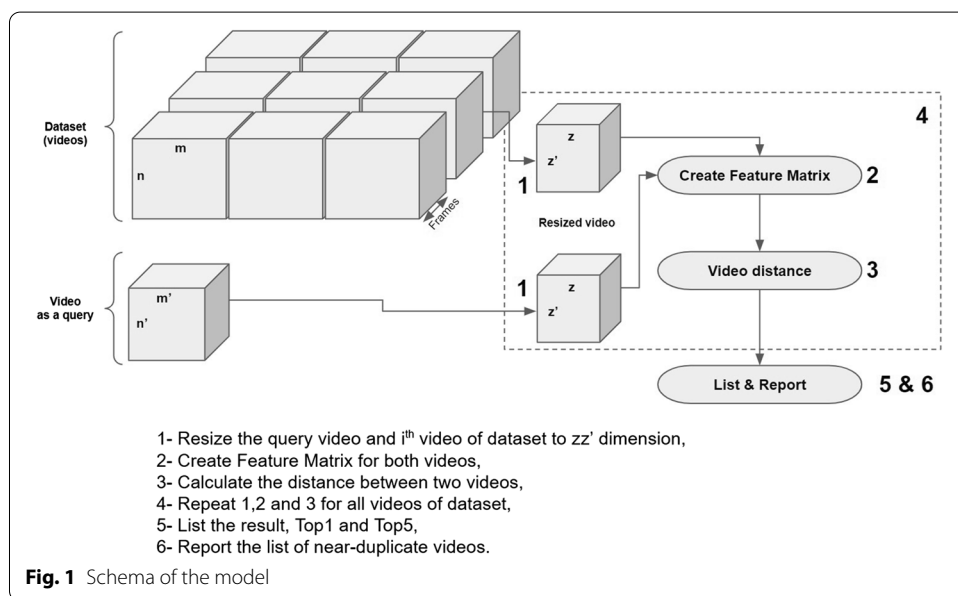
In this regard, this research aims to create a video signature that can be used to identify near-duplicate videos using a combination of local and global features (based on visual, motion data, high-frequency frame components, and invariant moments) by using relevant data to the furthest extent possible. This data must represent all video frames with a fixed number of features to make similarity computing rather fast, even if the video sizes differ. For this reason, the global video signature is the union of video blocks.

As mentioned earlier, the near-duplicate video takes on different forms. Thus, rather than retrieval and matching the hiding content and mirrored content, this work includes the scale, rotation and translation transformations. Whereas, comparing the same frame scale with different blocks after resizing the video is indicated in this research.

Consequently, we propose the following two-step system (Fig. 1) for all videos to investigate the performance of different transformations, such as resizing and mirroring videos based on the similarity detection metric, where the feature extraction assumes different frame/block dimensions.

(1) Feature extraction as a video signature based on:

1. Visual feature based on the luminance (Y) frame component,
2. High-frequency frame component based on FABEMD
3. Color motion features based on motion vectors.
4. Generation of invariant moments with regard to translation, scaling, and rotation.



*(II) Measurement similarity based on Minkowski ( $L_2$ ) distance:*

Since we adopted a fixed number of features, the  $L_2$  distance will be the fastest one used in this model.

Three video data sets are utilized to implement this system: UCF11 [12], UCF50 [13], and HMDB51 [14]. The experiments and tests reveal that they can yield remarkable performance.

The remainder of this paper is organized in the following manner: We summarize related works in "Related work" Section. We describe the proposed model in "Proposed model" Section. We present the experiments and results in "Statistical features" Section. Thereafter, in "Visual signature" Section, we conclude the paper.

### **Related work**

Image representation is characterized by local and global features. The extracted features must generalize several variations to guarantee a relevant classification. The local feature is invariant to partial occlusion, background, and viewpoint variants. The global feature processes all pixels that can be potentially used to compute the descriptor, including visual and motion geometric structures [15].

Izumi Ito [16] utilized global features to extract the relevant information from the salient portions of frames using seam carving without an assigned area that is not relevant in the frame. Their technique consists of using the scalar gradient-based local features in two layers as gradient-based global features. To prevent the distortion of salient objects when comparing and matching the relevant data, the block size utilized for each layer is  $2 \times 2$  and  $4 \times 4$ . Leila Kabbai et al. [3] attempted to combine local and global features; their main goal was to extract the most relevant information. Thus, they used Local Binary Pattern (LBP) and Local Ternary Pattern (LTP) to present a new pattern called Elliptical Local Pattern (ELP) as a local feature and Color Wavelet Transform (CWT) as a global feature. The concatenation of these two features generated the Concatenation of Local and Global Color (CLGC) histogram. These methods helped to create a pattern to compute the similarity between images or video frames.

In addition, Saddam Bekhet and Amr Ahmed [17] presented a visual similarity framework using a graph-based signature that represents the motion information of block movement across the shot. They have used the dominant color graph profile based on just I-frames and blocks of  $6 \times 6$  and  $7 \times 7$ , and they have evaluated the model by different similarity measures [18], such as Manhattan ( $L_1$ ), Euclidean ( $L_2$ ), Cosine angle, Histogram intersection, Chi-square, Bhattacharyya, Chebyshev, and Earth Mover's Distance.

On the other hand, numerous existing gradient-based feature extractors including the Histogram of Oriented Gradients (HOG), the Histogram of Optical Flow (HOF), the Motion Boundary Histograms (MBH), and the Histogram of Motion Gradients (HMG), generate histograms to represent various actions in the spatio-temporal domain of a video. However, these methods require that the number of bins used to aggregate data be determined in advance. Zuo et al. [19] demonstrated two advantages of fuzzifying these feature extractors: (i) accurate representation of the bin boundaries; and (ii) each pixel may be controlled flexibly for different fuzziness parameters. Thus, the

fuzzy descriptor and their combination may outperform alternative techniques, while the authors of [20] attempt to combine different dense, HOG, HOF, MBH over spatial pyramids using Fisher vectors for quantization and Support Vector Machine (SVM). In contrast, [21] extracted features with differential filters with multiple time scales and the SVM as a linear kernel. Moreover, although [22] achieves state-of-the-art accuracy, it comes with high computational requirements. According to [17] optical flow calculation takes up 61% of Dense-Trajectories (DTs), while combining dense flow information into histogram descriptors takes up 36% of the time.

In big data, time series distance measures are required in order to choose the suitable distance metric. The distance measures on time series are divided into two classes: the first includes Euclidean, dynamic time warping, and—for clustering—k-means and hierarchical clustering, and the second uses input/output data to create a model-based on norm distance  $H_2$ ,  $H_\infty$  to determine which systems are similar. On the other hand, the increasing size of data sets makes it impossible to store and process all the data at once, thereby presenting an approach that splits the attributes into two parts based on a Fast Incremental Model Tree with Drift Detection (FIMT-DD) to evaluate the performance of data streaming [23].

Using a real-time distributed cluster, Uddin [24] and Xu [25] used the adaptive local motion descriptor with Spark, while Saoudi and Jai-Andaloussi [26] technique applies Storm with a keyframe extraction algorithm to create a compact video representation based on the Bounded Coordinate System (BCS) that integrates all relevant content information of the video. [26] demonstrates that the selected keyframes are meaningful and adequately represent the video content, and their results suggest that the reduced processing time can be accomplished with a real-time distributed approach, but the shortcoming of this approach is that it is not applied for diverse transformations such as translation and rotation.

Phan et al. [27] used machine learning to examine relationships between local distributions of gradients of consecutive image sequences and to characterize information that changes orientation. Their descriptor provides two key contributions: (i) Post-processing of the principal component analysis followed by vector coding of locally aggregated descriptors to decorrelate and reduce the motion dimension of the Motion of Oriented Magnitude Patterns (MOMP) descriptors; (ii) Incorporating feature selection (i.e. statistical dependence, mutual information, and minimal redundancy) into the algorithm to determine best features using support vector machine (SVM) techniques.

Furthermore, the progress of deep learning leads to the application of a CNN in CBVR [28]. However, the increased usage and good results obtained by CNN-based techniques does not imply that the associated disadvantages can be ignored, which restrict their use in the targeted generic visual similarity problem:

1. The performance of CNN-based approaches is dependent on the size of the training data. Thus, it is impossible to develop the ideal CNN model using all available data. Furthermore, CNN models are typically constructed by training on high-dimensionality feature vectors, which is a time-consuming operation in terms of extracting/

- quantizing these features and train the CNN model. The dilemma is exacerbated by the recent trend of deep learning, which can take up to a month to complete [29].
2. CNN-based algorithms are well suited to domain-specific applications (e.g., action recognition [30]) since different videos are likely to produce identical feature vectors and be classified similarly. Consequently, CNN is less suited to solving the general visual similarity problem based on more than just action similarity.
  3. Instead of giving a unique fixed signature for each video, CNN-based techniques rely on training one or more models for the entire data set. These models cannot be used to identify the degree of similarity or dissimilarity between two videos because they can only determine whether they belong to the same category.
  4. By merging all collected video descriptors into a single vector, CNN training process may neglect important temporal information in videos [31].

Thus, [32] focused on two-stream CNN; their model captured spatio-temporal properties that are based on spatial and dense optical flow data and they are merged using (SVM). This approach needs good performance, like multiple Graphics Processing Unit (GPUs), to generate the model.

On the other side, Zheng et al. [33] make use of spatial-temporal data acquired by heterogeneous deep CNNs, Visual Geometry Group (VGG), and 3D Convolutional Networks (C3-D). Random projections reduce the dimension of high-dimensional video representations to a set of low-dimensional subspaces from which a prediction ensemble of classifiers is learned. The classification results are then further encoded using a novel Rectified Linear Encoding (RLE) layer, followed by a fully connected layer that combines all of the classifiers to produce the final classification results, which demonstrate exceptional efficacy for a variety of video classification tasks.

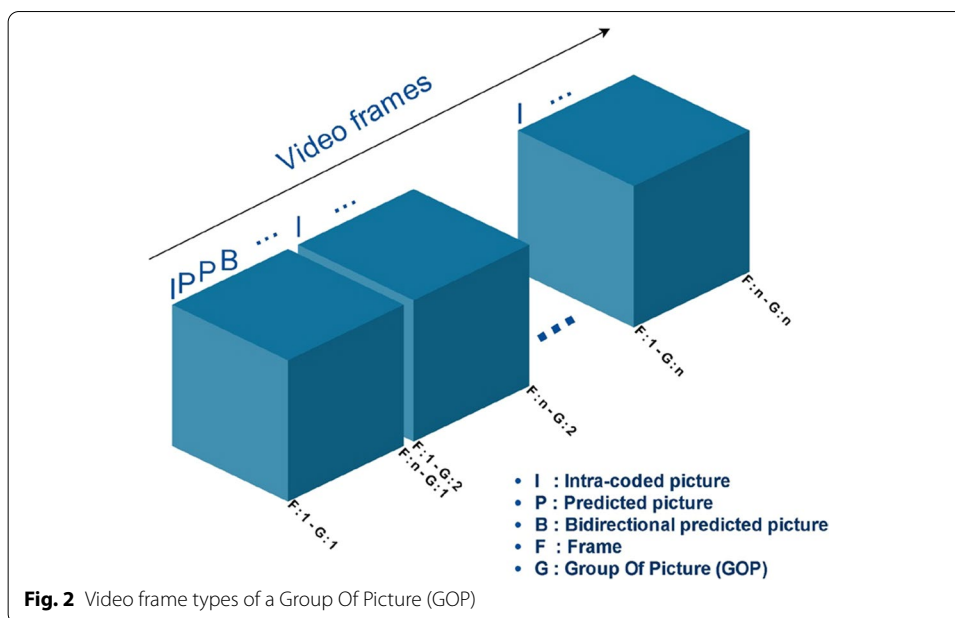
Therefore, we propose our model to identify near-duplicate video content based on the aforementioned research gaps, including different content transformations, such as mirroring, scaling, and rotating data.

### Proposed model

In general, a video is a sequence of frames with specific dimensions. Different methods, such as the Discrete Cosine Transform (DCT) [34], are used to compress video data. Thus, the video features are based on the data of the frame type (I-, P-, and B-frames) and the frame rate of each Group Of Pictures (GOP) (Fig. 2), from which the dominant color and intensity can be extracted. Moreover, the concept behind the use of FABEMD is to characterize the multiple statistical frame data in a few parameters while achieving a decomposition layer quality superior to any other decomposition technique (e.g., DCT). BIMFs and residues are intrinsic components of FABEMD and provide better performance in terms of perceptual invisibility and watermark robustness.

In video compression, three pictures (or frames) are used: I-, P-, and B-frames.

- I-frame (intra-coded picture): A complete image, such as a JPG or BMP image file.
- P-frame (predicted picture): Only the changes in the previous frame are stored. For example, only the car's movements must be encoded in a scene where the car moves



across a static background. The encoder saves space by not storing the unchanging background pixels in the P-frame. P-frames are also called delta frames.

- B-frame (bidirectional predicted picture) saves space by specifying its information using differences between the current, prior, and subsequent frames.
- Inter frames are also known as P- and B-frames. A GOP refers to the arrangement of the I-, P-, and B-frames.

In addition, the frame rate refers to the number of frames per second (fps), with PAL (Europe, Asia, Australia, etc.) and SECAM (France, Russia, African regions, etc.) standards specifying 25 fps and NTSC (USA, Canada, Japan, etc.) standards specifying 29.97 fps, respectively [35].

Moreover, no information is lost when frames are decomposed from high- to low-frequency components using the FABEMD, while the original frame is a replica of the BIMF [36, 37] frames. Furthermore, each frame that follows the Generalized Gaussian Distribution (GGD) model can be represented by a small number of appropriate parameters rather than using all statistical data, which simplifies comparison.

Additionally, it is necessary to include the concept of transformation (rotation, scaling, translation, etc.) in the near-duplicate model to be generic. Thus, Feature Extraction is enhanced with the Invariant Moment.

This subsection proposes two video signatures that reflect the feature extraction for all GOPs of video data (Figs. 3 and 4) by frame-block (part of video frames) and then by video-block. Thus, the first feature extraction collects statistical information from video shots (visual content), and the motion information (frame changes), while the

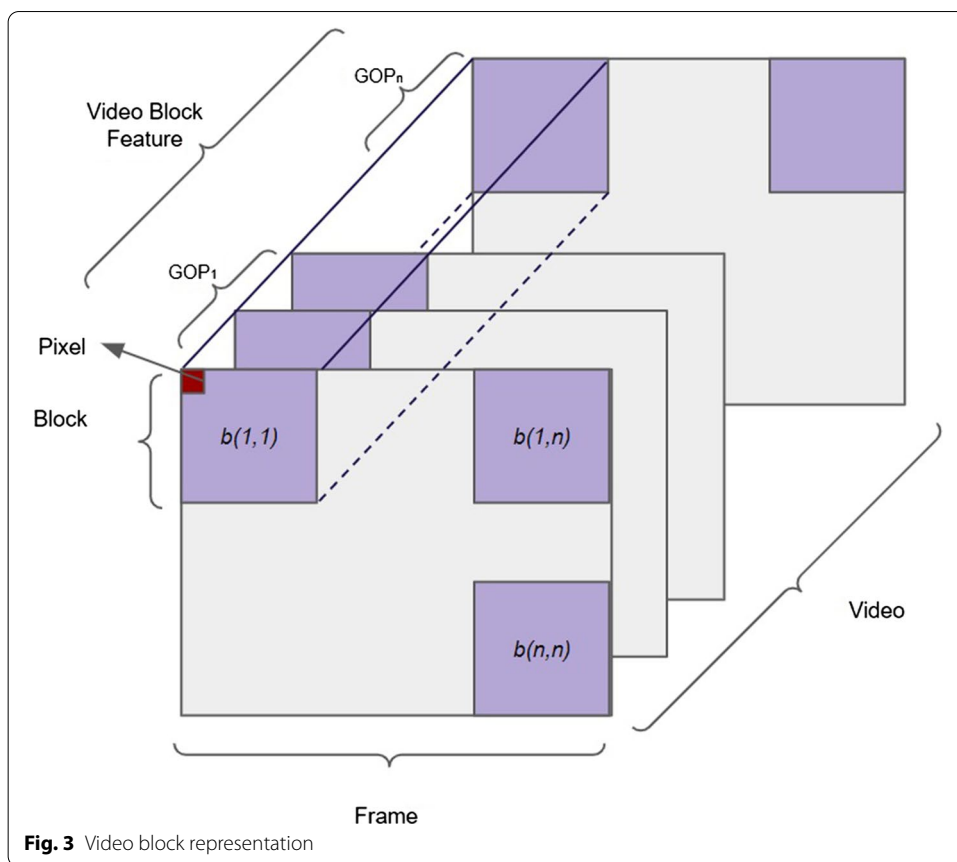


Fig. 3 Video block representation

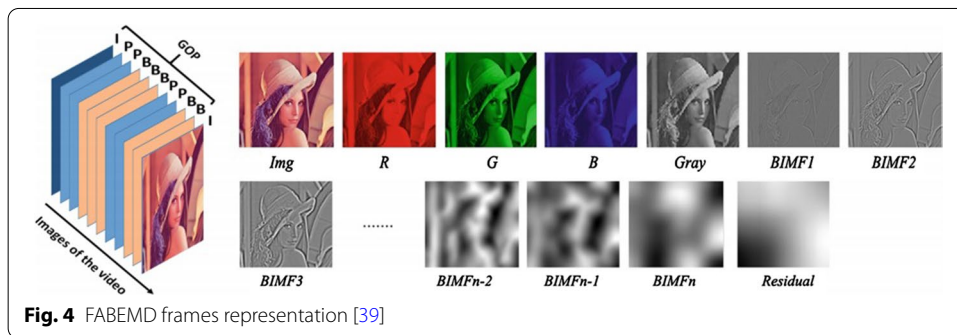


Fig. 4 FABEMD frames representation [39]

second captures the Invariant Moment of frames; thereafter, the two signatures are combined to create a video signature.

**Statistical features**

**Visual signature**

The color, texture, shape structures, and temporal order of video shots influence human perception [17]. Because of the arrangement of the frame’s elements, most frames do not generally contain valuable details of the overall content. Color function is essential in frames; it is evident that it is not an ideal element when there is noise, shadow, or illumination’s change [15], but it is helpful for a fixed camera to detect and extract objects through



foreground detection. On the other hand, a video shot with mobile cameras does not always follow the same pattern; it can be sluggish at times and fast at others, which affects the color changes of visual data proportionally in the P- and B-frames and, thus, changes in the I-frame. Consequently, the dominant color of the frame does not always represent the reality of extracted data compared to using statistical models, particularly in slow videos. Thus, the dominant color of the frame often ignores data in the context of the frame.

We divide each frame into blocks for an equitable extracting feature, specifically into a fixed block dimension  $n \times n$ , where the predominant color represents each block ( $b$ ) on its dimensions. Since it is possible to reverse a video or mask parts of it, each feature block ( $f_b$ ) represents the average dominant color between vertical symmetrical blocks (Eq. (1)).

$$f_{b(i,j)} = \left[ \frac{\max(b_{(i,j)}) + \max(b_{(n-i+1,j)})}{2} \right]_{1 \leq i,j \leq n} \tag{1}$$

We use k-means clustering to define the dominant color used Eq. (1), and since the color frame has triplet values, we concatenate them to obtain a single value. Thus, we are interested in the luminance ( $Y = 0.299R + 0.587G + 0.114B$ ), as human eyes are more sensitive to luminance changes than chrominance.

Usually, the number of frames in every video is different, which implies high-dimensional characteristics; the proposed approach is to standardize the video signature even though the videos contain different numbers of frames. Thus, the focus is on blocks per GOP rather than frame functionality. Therefore, the visual signature of videos is presented in the following manner Eq. (2):

$$S_1 = \left\langle \mu(f_{b(i,j)}^k)_g, \sigma(f_{b(i,j)}^k)_g \right\rangle_{1 \leq k \leq m; 1 \leq i,j \leq n; 1 \leq g \leq z} \tag{2}$$

where  $k$  is the position of the video frame,  $m$  is the total number of frames,  $g$  is the position of the GOP, and  $z$  is the total number of video GOPs. The mean ( $\mu$ ) and standard deviation ( $\sigma$ ) represent the average and the variation of dominant color along the length of GOPs.

**FABEMD**

The running maxima and minima of the data are obtained using spatial domain sliding order-statistics filters, namely: MAX and MIN filters. Smoothing the running maxima and minima yields the ideal upper and lower envelopes, respectively. The size of the order-statistics filters is determined by the information provided by the maxima and minima maps.

Let  $S_{(m,n)}$  stand for the original image,  $I_{i(m,n)}$  for the  $i$ th BIMF, and  $R_{(m,n)}$  for the residue.  $I_{i(m,n)}$  is obtained from its source image  $J_{i(m,n)}$  in the decomposition process, where  $J_{i(m,n)} = J_{i-1(m,n)} - I_{i-1(m,n)}$  and  $J_{1(m,n)} = S_{(m,n)}$ . The FABEMD method details are explained in [38].

GGD (Eq. (3)) can model BIMF and find the appropriate scale and shape parameters ( $\alpha, \beta$ )[39].

$$P(x, \alpha, \beta) = \frac{\beta}{2\alpha \Gamma(\frac{1}{\beta})} e - \left( \frac{|x|}{\alpha} \right) \beta \tag{3}$$

Varanasi and Aazhang [40] showed that a maximum probability estimator of  $GGD(\hat{\alpha}, \hat{\beta})$  could be used to find the unique solution of  $(\alpha, \beta)$ (Fig. 5).

$$\begin{cases} \gamma_1 = \frac{\psi(\frac{1}{\hat{\beta}})}{\hat{\beta}} - \frac{\sum_{i=1}^L x_i^{\hat{\beta}} \log(|x_i|)}{\sum_{i=1}^L |x_i|^{\hat{\beta}}} \\ \gamma_2 = \frac{\log(\frac{\hat{\beta}}{L} \times \sum_{i=1}^L |x_i|^{\hat{\beta}})}{\hat{\beta}} \end{cases} \tag{4}$$

where L is the total of frame’s blocks, and the digamma function is  $\psi(t) = \frac{\Gamma'(t)}{\Gamma(t)}$ , gamma function is  $\Gamma(x) = \int_0^\alpha e^{-t} t^{x-1} dt$ , and  $x > 0$ , while  $x_i$  denotes the values in the histogram of the frequency distribution.

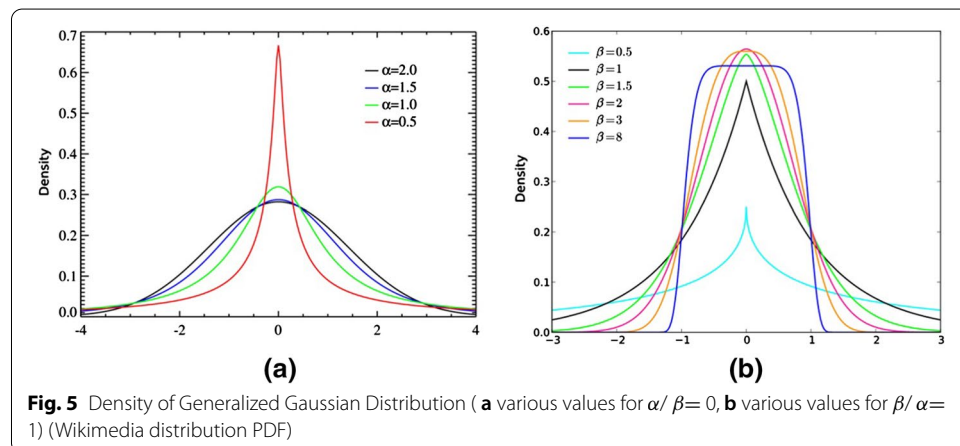
$$\begin{cases} \hat{\alpha} = (\frac{\hat{\beta}}{L} \sum_{i=1}^L |x_i|)^{\hat{\beta}} \\ 1 + \gamma_1 + \gamma_2 = 0 \end{cases} \tag{5}$$

At this level, we calculate  $(\hat{\alpha}, \hat{\beta})$  of  $BIMF_1$  for each block using the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) that represent the average and the variation of BIMF along the GOPs.

$$S_2 = \left\langle \mu(\hat{\alpha}_{b(i,j)}^k)_g, \sigma(\hat{\alpha}_{b(i,j)}^k)_g, \mu(\hat{\beta}_{b(i,j)}^k)_g, \sigma(\hat{\beta}_{b(i,j)}^k)_g \right\rangle_{1 \leq k \leq m; 1 \leq i, j \leq n; 1 \leq g \leq z} \tag{6}$$

**Motion signature**

In a GOP, the difference between I-frame and P-frame and/or B-frame refers to motion vector data[41], which is extracted from the compressed (MPEG AVC/H.264 version) video stream. Motion vectors can take different directions from 0 to 360 degrees ( $\Omega(\mu)$ ), while  $\Omega_\pi(\mu)$  is calculated in radians, thus they could be useful to form the motion features. The direction of motion vector  $\mu(x, y)$  is calculated by Eq. (7).



$$\Omega_{\pi}(\mu) = \begin{cases} \arccos \frac{x}{|\mu|}, & y \geq 0 \\ 2\pi - \arccos \frac{x}{|\mu|}, & y < 0 \end{cases} \tag{7}$$

where  $(x, y)$  are the coordinates of  $\mu$  and  $|\mu|$  is the magnitude.

In addition, to reduce the time complexity of processing, we adopt intervals of  $30^\circ$  per direction class ( $D_C$ ). Thus, each class of 12 classes ( $d=12$ ), in Eq. (8), indicates a specific direction (angle of motion), and the 13<sup>th</sup> represents no-motion (Fig. 6).

$$D_C(\Omega_{\pi}) = \begin{cases} 0, & \mu = (0, 0) \\ 1 + \left( \left[ \frac{\Omega_{\pi}(\mu) \times (d) + \pi}{2\pi} \right] \text{mod}[d] \right), & \text{otherwise} \end{cases} \tag{8}$$

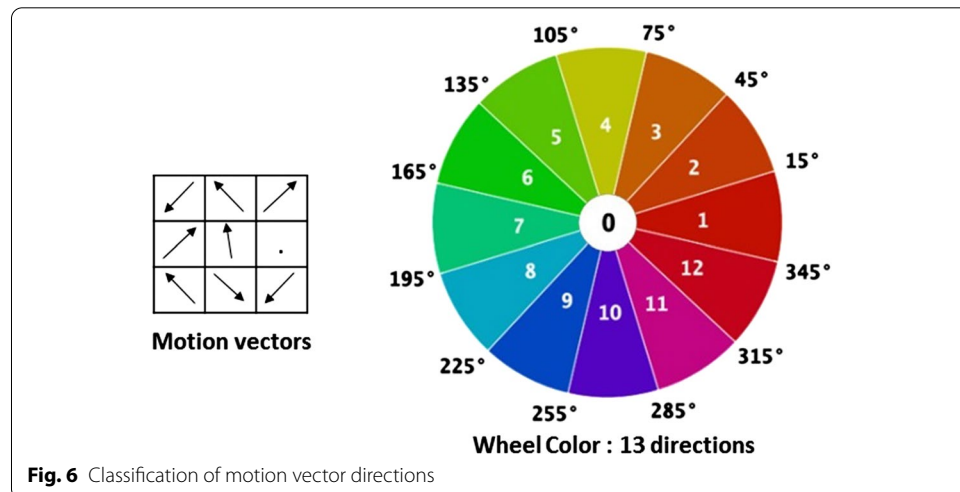
For each block, we focus on the dominant direction ( $D^*_C$ ) that represents the dominant class of motion, but to differentiate one content from another of neighboring classes, the intensity ( $I$ ) of the dominant direction is added to motion signature (Eq. (10)) via Eq. (9), where  $N$  is the number of  $D^*$  vectors.

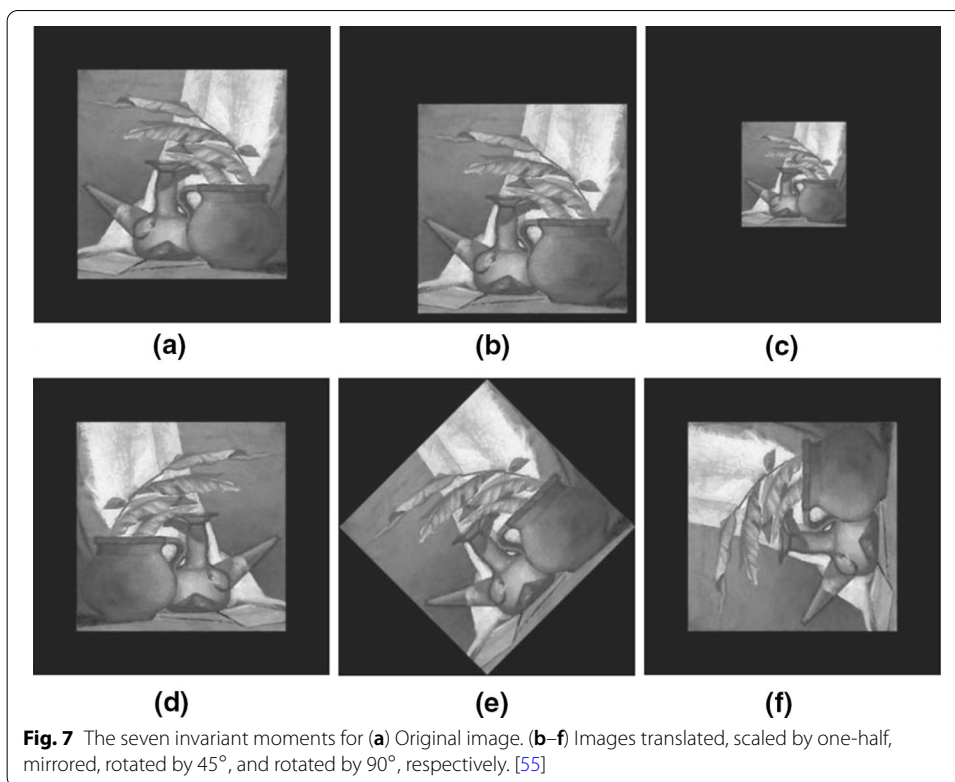
$$I_{D^*_{Class}} = \frac{1}{N} \sum_{i=1}^N |\mu|_{D^*_{Class}} \tag{9}$$

$$S_3 = \left\langle \mu(D^k_{C_{b(i,j)}})_g, \sigma(D^k_{C_{b(i,j)}})_g, \mu(I^{b(i,j)^k}_{D^*_C})_g, \sigma(I^{b(i,j)^k}_{D^*_C})_g \right\rangle_{1 \leq k \leq m; 1 \leq i, j \leq n; 1 \leq g \leq z} \tag{10}$$

**Invariant moment of video frame**

Image moment is well-known in image analysis and matching for its capacity to extract features with respect to specified transformation classes. The most fundamental frame transformations are rotation, scaling, and translation. Keep in mind that the only moments that are invariant are the central moments, which are accurate only in the continuous domain. Scaling and rotation are imprecisely defined and irreversible in





**Table 1** Moment invariants of the images in Fig. 7

Moment Invariant	Original Image	Translated	Half Size	Mirrored	Rotated 45°	Rotated 90°
$\phi_1$	2.8662	2.8662	2.8664	2.8662	2.8661	2.8662
$\phi_2$	7.1265	7.1265	7.1257	7.1265	7.1266	7.1265
$\phi_3$	10.4109	10.4109	10.4047	10.4109	10.4115	10.4109
$\phi_4$	10.3742	10.3742	10.3719	10.3742	10.3742	10.3742
$\phi_5$	21.3674	21.3674	21.3924	21.3674	21.3663	21.3674
$\phi_6$	13.9417	13.9417	13.9383	13.9417	13.9417	13.9417
$\phi_7$	-20.7809	-20.7809	-20.7724	20.7809	-20.7813	-20.7809

a discrete domain; hence, the image’s description is approximate. As a result, moment invariants are especially advantageous for recognizing and matching invariant patterns (Table 1).

The 2-D moment of order  $(p + q)$  of  $M \times N$  digital frame  $f(x, y)$  is defined as :

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \tag{11}$$

where  $p = 0, 1, 2, \dots$  and  $q = 0, 1, 2, \dots$  are integers. In practice, the image is summarized with functions of a few lower order moments.

The corresponding central moments  $\mu_{pq}$  are invariant with respect to translations,  $\mu_{pq}$  are defined as:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{12}$$

where  $\{\bar{x}, \bar{y}\} = \left\{ \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right\}$  represents the centroid or the center of gravity of the frame.

The normalized central moments of order  $(p + q) \geq 2$ , denoted  $\eta_{pq}$ , are invariant with respect to both translation and scale,  $\eta_{pq}$  are defined as follows, where  $\gamma = \left( \frac{p+q}{2} + 1 \right)$ .

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \tag{13}$$

Hu [42] presents invariant moments with respect to translation, scale, and rotation, which are Eq. (14) through Eq. (20). Other authors, such as Flusser and Suk [43], have extended Hu’s theory for N-rotationally symmetric shapes case by developing new invariant, while Mamistvalov’s proof [44] generalizes moment invariants to  $n$  dimensions. Nevertheless, these seven invariant moments continue to be relevant for  $p + q = 2, 3, 4, \dots etc.$

$$\phi_1 = \eta_{20} + \eta_{02} \tag{14}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{15}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{16}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{17}$$

$$\phi_5 = \theta_1 + \theta_2; \begin{cases} \theta_1 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ \theta_2 = (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{cases} \tag{18}$$

$$\phi_6 = \theta_1 + \theta_2; \begin{cases} \theta_1 = (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \theta_2 = 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{cases} \tag{19}$$

$$\phi_7 = \theta_1 + \theta_2; \begin{cases} \theta_1 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ \theta_2 = (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{cases} \tag{20}$$

The  $\phi_1$  is analogous to the moment of inertia around the image’s centroid, where the pixel intensities correspond to physical density. The  $\phi_1$  to  $\phi_6$  are reflection symmetric, which means they remain unchanged when the frame is mirrored, and  $\phi_7$  is anti-symmetric to reflection (changing sign when reflected), and allowing identifying mirror content of similar frames.

In addition, a similarity of motion blocks can be estimated by Eq. (21), where the direction of motion vector can be provided by Eq. (22) or by *Zernike moments*[45], which are based on all seven invariant *Hu moments* (Eq. (14)—Eq. (20)). Thus, if  $M(B_c, B_f)$  value is less, then the blocks in two neighbor frames are more similar.

$$M(B_c, B_f) = \sum_{i=1}^7 \left| (\text{sign}(\phi_i^{B_c}) \ln(\phi_i^{B_c})) - (\text{sign}(\phi_i^{B_f}) \ln(\phi_i^{B_f})) \right| \quad (21)$$

where  $B_c(x, y)$  is a block in a current frame  $I_t$ , and  $B_f(x, y)$  represent the nine blocks in a following frame  $I_{t+1}$  (8 neighbor blocks and 1 central block).

$$\text{motion\_vector}_{Hu} = \min_{b \in \{0,1,\dots,8\}} (M_b(B_c, B_f)) \quad (22)$$

Another key trend related to invariant moments is the development of efficient algorithms, which requires a high computational speed. However, the invariant moments of Hu based on blocks suffer from limited recognition power and the major weakness comes from the non-possibility of block generalization. Thus, by utilizing the mean and standard deviation of the Hu moments within the luminance  $Y$  frame, processing time is reduced. Given that the motion direction is already determined by Eq. (8), adding another moment based on the blocks is needless due to the increased time-consuming and limited recognition, except when analyzing the noise of video sequences [46].

Finally, Eq. (23) represents the concatenation of all invariant moment values, where  $n$  denotes the dimension of the frame moments.

$$S_4 = \bigcup_{i=1}^n \left\langle \mu(\phi_i^k)_g, \sigma(\phi_i^k)_g \right\rangle_{1 \leq k \leq m; 1 \leq g \leq z} \quad (23)$$

### Video signature

Since videos could include a longer scene, the average of GOP statistical features and invariant moments represents the mean of video block features. Thus, we calculate the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) for each GOP ( $g$ ) using a two-level signature. The first video signature ( $S^*$ ) is a representation of statistical features that takes mirroring and hidden content into account. The concatenation of visual and motion signatures is defined as a global feature via Eq. (24), while the second video signature ( $S^{**}$ ) adds to  $S^*$  the invariant moments ( $S_4$ ) to support the translation, scaling, and rotation transformations.

$$S^* = \bigcup_{i=1}^3 S_i \quad (24)$$

Therefore, for each GOP,  $S^*$  and  $S_4$  are normalized using Eq. (25) to show significant values between 0 and 1. Afterwards, for all GOPs,  $F_{V_b}(\mu, \sigma)$  denotes the video feature blocks for all  $S^*$  items ( $f_b^g, \hat{\alpha}_b^g, \hat{\beta}_b^g, D_{C_b}^g, I_{D^*C_b}^g$ ), and  $\phi(\mu, \sigma)$  represents the video invariant moments for all  $S_4$  items ( $\phi_1^g, \phi_2^g, \phi_3^g, \phi_4^g, \phi_5^g, \phi_6^g, \phi_7^g$ ).

$$X_i = \frac{X_i - \min(X)}{\max(X) - \min(X)} \quad (25)$$

Finally, the generic  $S^*$  of video  $i$  is denoted by the matrix signature of video features :

$$S^*_{\text{video}_i} = \begin{bmatrix} F_{V_b(1,1)} & F_{V_b(1,2)} & \dots & \dots & \dots \\ F_{V_b(2,1)} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & F_{V_b(n-1,n)} \\ \dots & \dots & \dots & F_{V_b(n,n-1)} & F_{V_b(n,n)} \end{bmatrix} \quad (26)$$

and the generic  $S^{**}$  is:

$$S^{**} = (S_{video_i}^*, S_{4video_i}) \tag{27}$$

**Measure function**

The measurement distance is important to calculate the similarity between two entities ( $d(E_1, E_2)$ ). Many metrics are considered to compute video similarity such as Manhattan [47], Euclidean[48], Cosine angle [49], Chi-square [50], and Earth Mover’s Distance [51]. Since we have linked the comparison of two videos by blocks and none by video frames, we will have the same number of video features to compare. Thus, for  $S^*$ , our choice Eq. (28) is based on [18], and the optimal metric used in this study is Minkowski ( $L_2$ ), where  $z$  is  $n^{th}$  feature.

$$d_{S^*}(E_i, E_j) = \sqrt{\sum_{z=1}^n (E_{(i,z)} - E_{(j,z)})^2} \tag{28}$$

For  $S^{**}$ , the distance is:

$$d_{S^{**}}(E_i, E_j) = \frac{d_{S^*} + \sqrt{\sum_{l=1}^m (S_{4(i,l)}^\phi - S_{4(j,l)}^\phi)^2}}{2} \tag{29}$$

where  $m$  denotes the dimensions of the invariant moment.

**Experiment results and discussion**

Our implementation is based on Python-3 and is worked on a workstation with a 3.1 GHz processor, 16 GB of RAM, and 2GB NVIDIA VRAM. The datasets used in this study are UCF11[12], UCF50 [13], and HMDB51[14]. For the first approach ( $S^*$ ), the video frames and blocks are taken with different sizes from 120 to 600 pixels and from 10 to 30 pixels, respectively. The results are evaluated in terms of F1-score and accuracy; the relevant results are ranked by  $Top_1$  for the best result, and the five top-ranked results are presented by  $Top_5$ .

The optimal size and block frame features of  $S^*$  are evaluated using the second approach  $S^{**}$  that reinforce the features against various transformations. Finally, the two proposed approaches ( $S^*$  and  $S^{**}$ ) are compared to the state-of-the-art methodologies.

**Implementation**

**Datasets**

UCF11 and UCF50 are an action recognition datasets with 11 and 50 action categories, consisting of 1600 and 6681 realistic videos, respectively, taken from Youtube.

UCF50 dataset is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. For each of the 50 different categories, the videos are grouped into 25 groups, where each group consists of more than 4 action clips. The video clips in the same group may share some common features, such as the same person, similar background, similar viewpoints, etc.

HMDB51 is an action recognition dataset, collected from various sources, which are mostly from movies, and a small proportion from public databases such as YouTube. The dataset contains 6766 clips divided into 51 action categories, each containing a minimum of 100 clips.

### Measurement performance

In this paper, we used Precision (Eq. (30)), Recall (Eq. (31)), F1-score (Eq. (33)), Average Precision (AP) (Eq. (34)), and Average Precision (mAP) (Eq. (35)), where precision (P) means the percentage of the results that are relevant, and recall (R) refers to the percentage of the total relevant results correctly classified by the proposed model. Moreover, we preferred to add F1-Score to our work, which combines precision and recall as a harmonic mean. The reason for adding F1 was that accuracy (Eq. (32)) can be a misleading metrics for unbalanced datasets. AP and mAP are used when we utilize more than one value as a matched result for each video of each class in the video dataset.

$$Precision = \frac{tp}{tp + fp} \quad (30)$$

$$Recall = \frac{tp}{tp + fn} \quad (31)$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (32)$$

$$F1_{Score} = 2 * \left( \frac{Precision * Recall}{Precision + Recall} \right) \quad (33)$$

where  $tp$ ,  $tn$ ,  $fp$ ,  $fn$  are true positive, true negative, false positive, false negative, respectively.

$$AP@n = \frac{1}{GTP} \sum_k^n P@k \times rel@k \quad (34)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (35)$$

Average Precision (AP) is a better measurement of a model in its ability on the sorting results of the query, where Ground Truth Positives (GTP) refers to the total number of ground truth positives which means the matched values obtained compared to the dataset,  $n$  refers to the total number of videos that are interested in  $P@k$  that refers to the  $precision@k$  and  $rel@k$  is a relevance function. The relevance function is an indicator function which equals 1 if the video at rank  $k$  is relevant and equals 0 otherwise, and the mean Average Precision (mAP) is the average of AP for all classes in the dataset.



**Video frame and block dimensions**

We resize videos to the same dimensions (Fig. 8) to compare them, and we use different dimensions to display which one is best suited for a large-scale dataset.

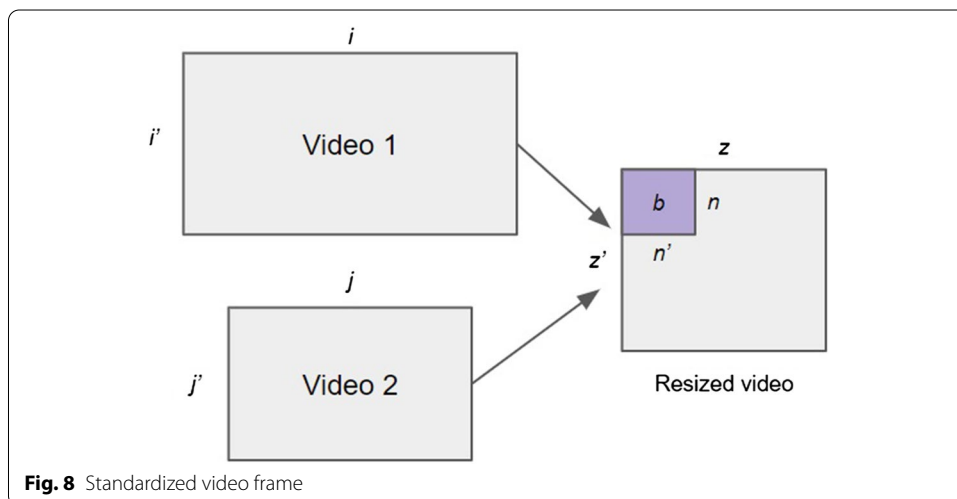
We compared one random frame from each video with others in the video dataset to pick the most important measurements. The relevant results of *Top*<sub>1</sub>, *Top*<sub>5</sub>, and *Top*<sub>10</sub> are shown in Fig. 9 for blocks and in Fig. 10 for frames. To maximize the correct retrieved matches, we added weight to the top ranks (Eq. (36));  $\alpha$ ,  $\beta$ , and  $\gamma$ , where  $\alpha > \beta > \gamma$  and are 1, 0.8, 0.2, respectively[52].

We chose  $10 \times 10$ ,  $15 \times 15$ ,  $20 \times 20$ ,  $24 \times 24$ ,  $30 \times 30$ , and  $60 \times 60$  blocks to compare and verify which of them could present the appropriate pre-processing results for all frames. Furthermore, we used the same approach to select frame sizes of  $120 \times 120$ ,  $240 \times 240$ ,  $360 \times 360$ ,  $480 \times 480$ ,  $600 \times 600$ , ...,  $1080 \times 1080$ , first because users have attempted to downsize rather than upsize video, and second because increasing frame size increases computational cost.

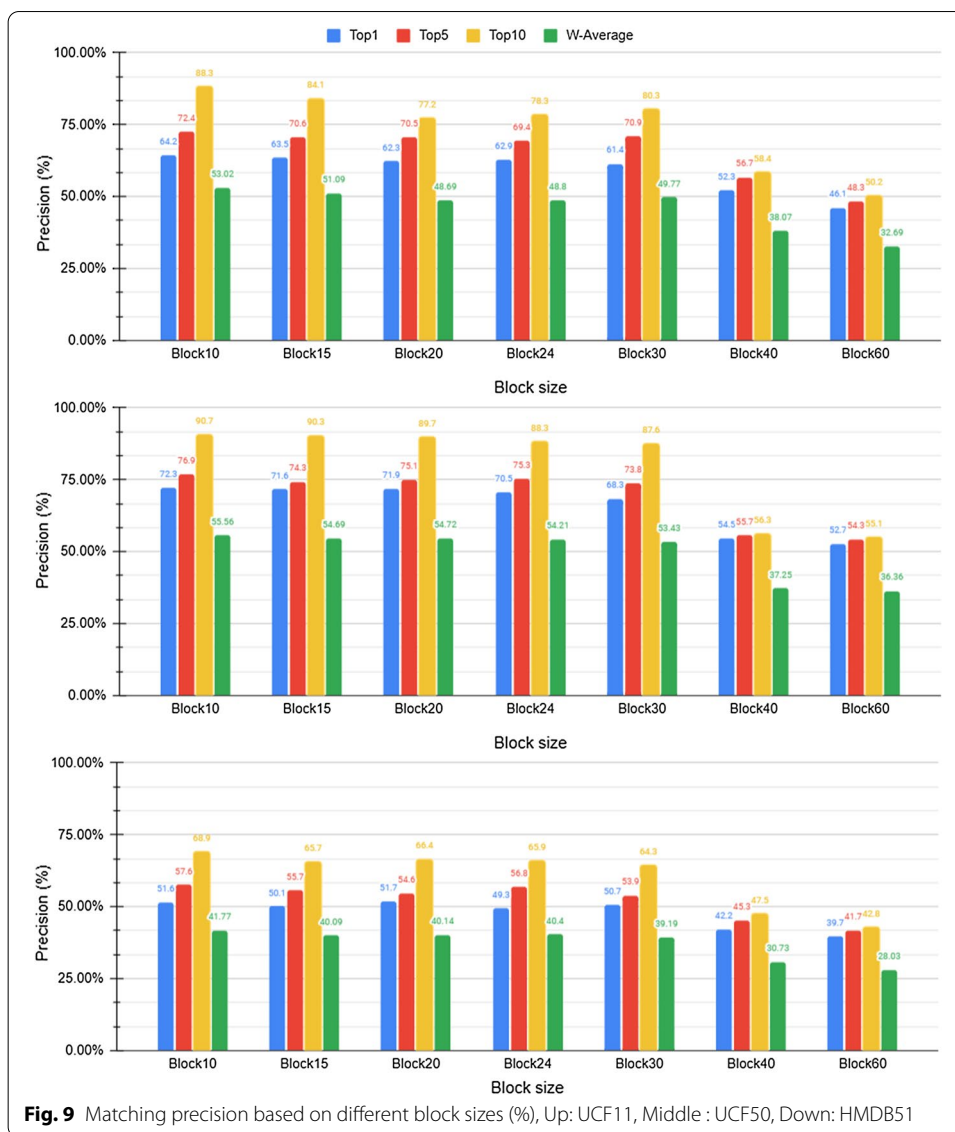
$$W - Average = \frac{\alpha \times P_{Top10} + \beta \times P_{Top5} + \gamma \times P_{Top1}}{3} \tag{36}$$

Based on Fig. 9, the small blocks are more efficient than the large ones, the matching percentage of precision decreases as Block60 is used, and all findings from *Top*<sub>1</sub>, *Top*<sub>5</sub>, and *Top*<sub>10</sub> become nearly approximate compared to findings of Block10. Block10 has a matching percentage between 60% and 90% for the two datasets UCF11 and UCF50 and between 50% and 70% for HMDB51, with an average precision of 55% for the first two datasets and 41% for the third. Since the precision rate of different Tops is close to 0 when employing Block10 through Block60, the *Top*<sub>1</sub> is more convenient in terms of computing time than using *Top*<sub>5</sub> or *Top*<sub>10</sub>. On the other hand, the precision rate is outstanding between distinct Tops when Block10 is used. As a result, it is recommended to use Block10 rather than other blocks.

According to Fig. 10, the precision rate between the *Top*<sub>1</sub>, *Top*<sub>5</sub>, and *Top*<sub>10</sub> results is minimal between  $600 \times 600$  and  $1080 \times 1080$  frames, and more precisely on UCF50



**Fig. 8** Standardized video frame



and HMDB51 datasets. While the average precision is almost symmetrical towards the  $600 \times 600$  frame size, the focus will be on the frame sizes smaller than  $600 \times 600$  in order to achieve high precision with minimal time-consuming. Additionally, it appears as though the close frame sizes of  $600 \times 600$  are more pertinent.

**Proposed approach performance**

**S\* performance and computational cost**

The Table 2 shows  $F_1$ -score for all UCF11 (11 classes) classes, which were divided into different dimensions and blocks. It is clear that in Table 2, the  $Top_1$  range of  $F_1$  is between 70% and 90%. The combination frame/block  $120 \times 120/30 \times 30$  shows the lowest percentage and  $600 \times 600/10 \times 10$  shows the highest percentage, while the results show the same pattern in Table 3 and Table 4, with scores ranging from 75% to 82% for UCF50 (50 classes), and 57% and 66% for HMDB51 (51 classes).

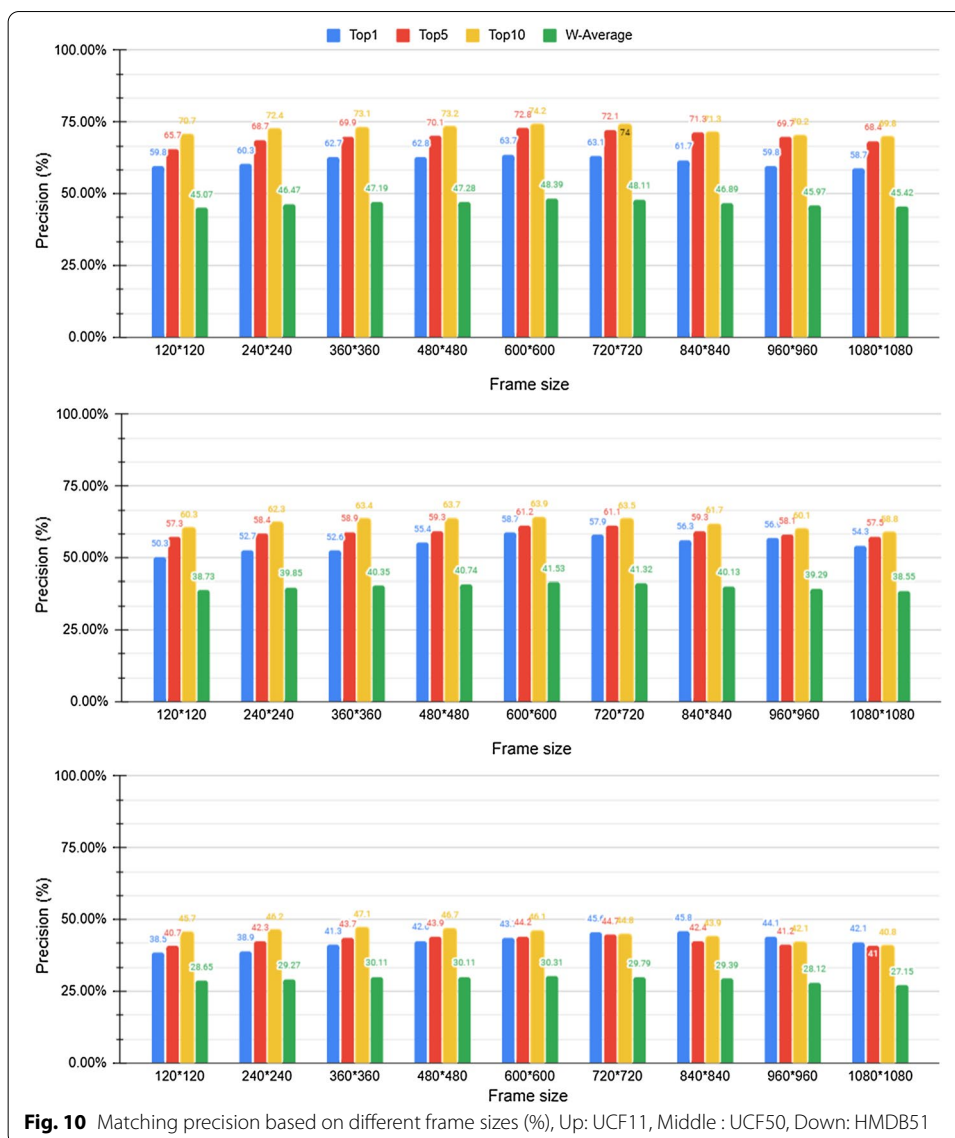


Fig. 10 Matching precision based on different frame sizes (%), Up: UCF11, Middle : UCF50, Down: HMDB51

Table 2 F1-Score performance of S\* using UCF11 dataset

Block/Frame	120*120	240*240	360*360	480*480	600*600
10*10	0.831	0.839	0.854	0.872	0.896
15*15	0.776	0.784	0.814	0.842	0.865
20*20	0.767	0.774	0.786	0.803	0.843
24*24	0.732	0.743	0.759	0.789	0.813
30*30	0.701	0.718	0.736	0.766	0.781

In general, UCF11 and UCF50 reflect a high degree of visual and motion data similarity between videos in the same category (class), but not so with HMDB51, where motion data is much more similar than visual content. As a result, if the visual content is also as

**Table 3** F1-Score performance of  $S^*$  using UCF50 dataset

Block/Frame	120*120	240*240	360*360	480*480	600*600
10*10	0.778	0.786	0.794	0.807	0.823
15*15	0.772	0.781	0.784	0.788	0.794
20*20	0.768	0.775	0.780	0.782	0.786
24*24	0.761	0.769	0.774	0.776	0.781
30*30	0.753	0.762	0.771	0.769	0.776

**Table 4** F1-Score performance of  $S^*$  using HMDB51 dataset

Block/Frame	120*120	240*240	360*360	480*480	600*600
10*10	0.618	0.627	0.639	0.647	0.658
15*15	0.602	0.609	0.617	0.630	0.642
20*20	0.593	0.598	0.609	0.615	0.635
24*24	0.582	0.586	0.591	0.602	0.622
30*30	0.576	0.579	0.583	0.593	0.613

similar as possible, two videos with near-motion content may be considered the nearest to each other.

On the other side, the average in terms of frame dimensions of the results per dataset (UCF11, UCF50, and HMDB51, respectively) for block  $10 \times 10$  is relevant  $\simeq 86\% \pm 3\%$ ,  $\simeq 80\% \pm 2.5\%$  and  $\simeq 63.5\% \pm 2.5\%$ , compared to  $30 \times 30$  that is less interesting,  $\simeq 74\% \pm 4\%$ ,  $\simeq 76\% \pm 1.5\%$  and  $\simeq 59\% \pm 2\%$ .

In addition, the Figs. 11, 12 and 13 display the accuracy of the cross matrix for the video classes based on the UCF11, UCF50, and HMDB51, respectively. Thus, we found that the majority of the comparison results for the same video class are between 90% and 100%, whereas the different classes range between 0 and 80%. Moreover, on UCF50 and HMDB51, which contain thousands of videos and various classes in comparison to UCF11, the difference between  $120 \times 120$  and  $600 \times 600$  results is just about 5% on average.

Therefore, the  $10 \times 10$  block with  $600 \times 600$  frame size present relevant results unless we are interested in the computation time that varies from 1.30s (for  $120 \times 120$ ) to 20.30s (for  $600 \times 600$ ) depending on the block and/or the frame dimensions used.

The average of computational time is shown in Table 5, and in this study, we need approximately 20 seconds or fewer to present similar content from a query video. Concerning time computing, we note that the difference is important between 01.32s and 20.34s, which highlights our need for a fast calculation manner and presentation of the results. For example, if we take the  $10 \times 10$  blocks into consideration, we notice that the difference is 19s between  $120 \times 120$  and  $600 \times 600$ , and since the  $F_1$  precision interval on UCF11, UCF50, and HMDB51, respectively, is 6%, 6%, 4%, it is possible to use  $120 \times 120$  as the basic step in similarity computing to consume only 2.32s and utilize  $600 \times 600$  in certain cases where the high precision verification is needed. In addition, we also state that our process is sequential, thus, time computing depends on the size of the video and

**Table 5** Average processing time (APT) of  $S^*$  (by second)

Block/Frame	120*120	240*240	360*360	480*480	600*600
10*10	02.32	07.50	10.50	14.48	20.34
15*15	02.24	05.31	08.54	11.82	16.85
20*20	01.95	03.88	06.57	09.66	12.37
24*24	01.64	02.98	04.85	07.89	08.84
30*30	01.32	02.30	03.18	04.30	05.43

the dataset used, while parallel processing could completely reduce the processing time [53, 54].

#### ***$S^*$ performance for both masked and mirrored content***

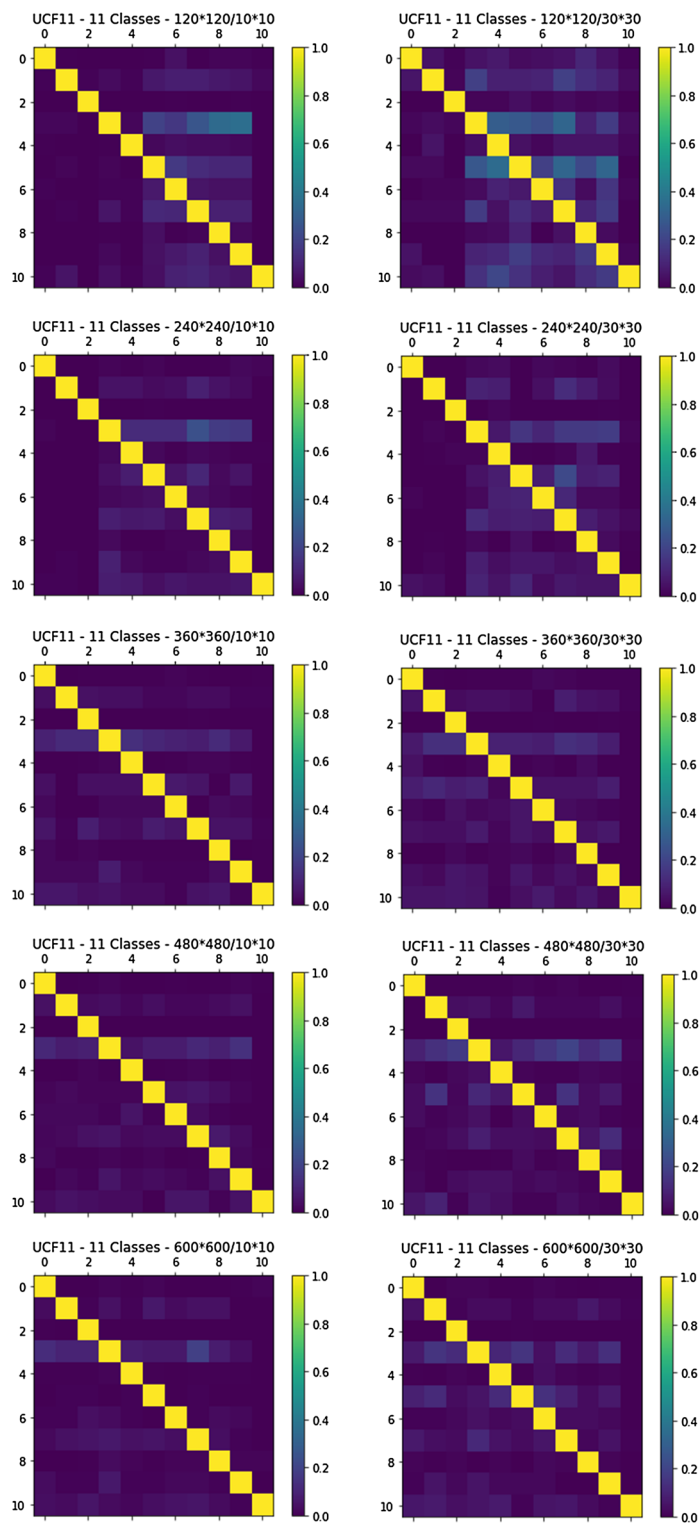
Our methodology is a little different compared to related work, but kept the same main objective, as we have added mirroring and masking video content forms, the accuracy metric comparison using UCF11 without video modification confirms that our method overcomes [18], our accuracy is 91.7% against 82.9%. On the other hand, the changed video experiments presented also good results. For UCF50 and HMDB51, we conducted three tests: the first was with out any video transformation; the second was with 1/4 of the videos unchanged, 1/4 of the videos are masked, randomly hiding part of the video frames that do not exceed 20% of the total content, 1/4 of the videos mirrored, and 1/4 of the videos are both masked and mirrored. For the third test, we kept 1/4 of the videos unchanged, 1/4 of the videos are masked, randomly hiding part of the video frames that do not exceed 20% of the total content, 1/4 of the videos are transformed : mirrored, scaled, or rotated, and 1/4 of the videos are bi-transformed : masked and mirrored, mirrored and scaled, scaled and rotated.

The results of Table 6 revealed an average score of  $\pm 5\%$ ,  $\pm 4\%$ , for the comparison between  $120 \times 120$  and  $600 \times 600$  frame size, and between  $Top_1$  and  $Top_5$ , respectively. In short, when using  $Top_1$  data, inter/intra classes can have an impact on outcomes, while  $Top_5$  data provide additional details and demonstrate good performance through AP, which can override  $Top_1$  by  $4\% \pm 1.5\%$  using mAP. As a result, the average of the five related results (Table 6) is more important than one minimum distance. Besides the relevant results of  $S^*$  of test 1 and test 2, they are irrelevant when additional transformations such as translation, scaling, and rotation are applied.

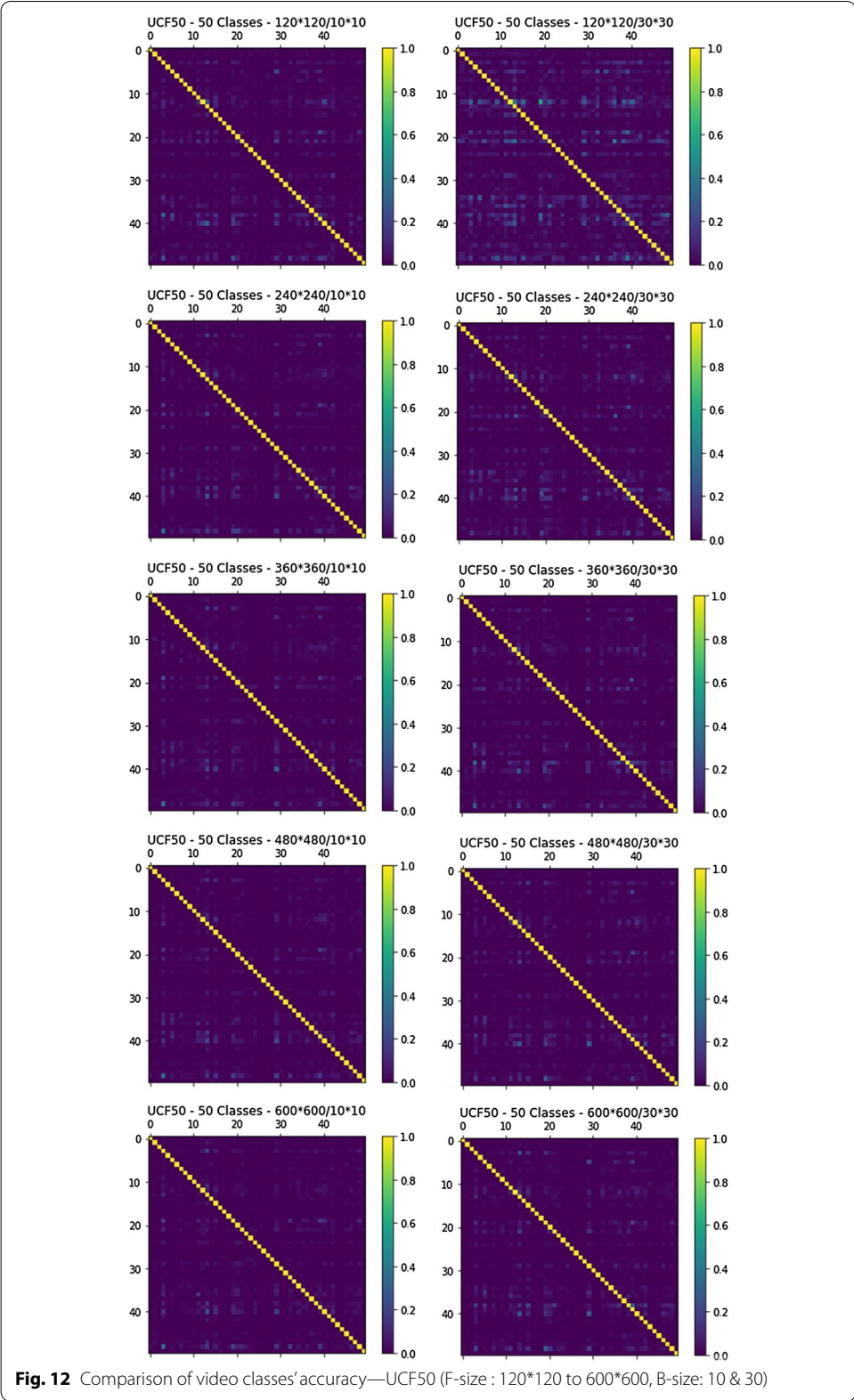
#### ***$S^{**}$ performance and discussion***

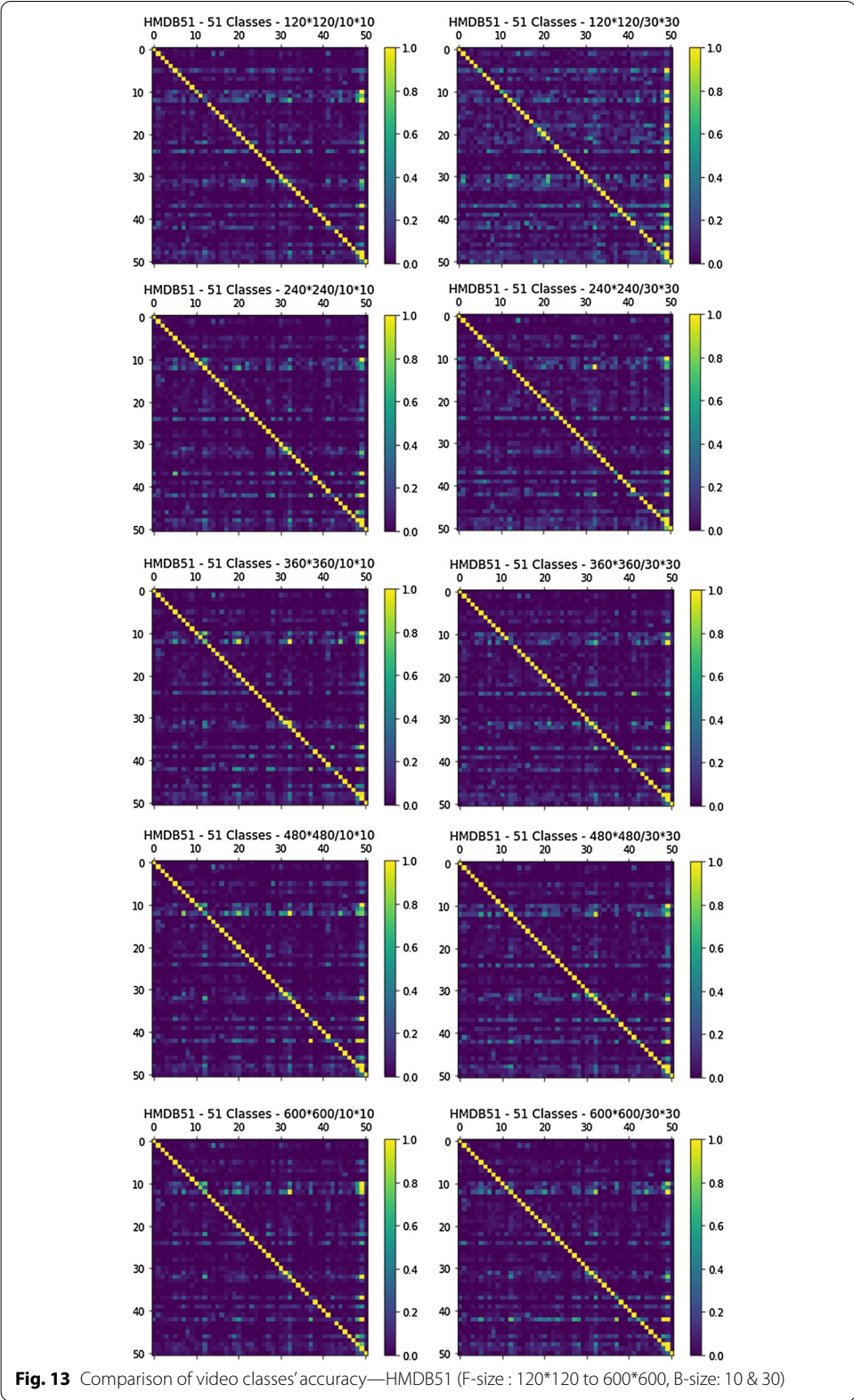
Based on Table 7, the results denote that  $S^{**}$  supports video transformations due to the presence of invariant moments, unlike  $S^*$ , which is only suitable for hiding and mirroring content. Thus, it emerges from Table 7 that the results of  $S^{**}$  are more relevant, specially for  $Top_5$ , than those of  $S^*$ , and they can be used to compare with the state-of-the-art findings.

As the results are not significantly different between  $Top_1$  and  $Top_5$ , we continue the tests using only  $Top_1$ , and obviously,  $Top_5$  is utilized to index videos in order to provide the five top-ranked videos that are most closely related to the requested video.



**Fig. 11** Comparison of video classes' accuracy—UCF11 (F-size : 120\*120 to 600\*600, B-size: 10 & 30)





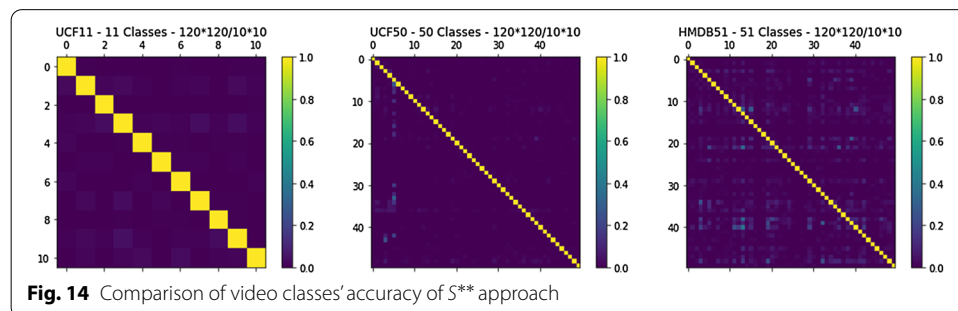


**Table 6** F1-Score performance of  $S^*$ , unchanged videos ( $F^-$  test 1), modified videos ( $F^+$  test 2), translated, rotated, scaled videos ( $F^\circ$  test 3) using  $10^*10$  block

Dataset	20*120	600*600
UCF11	$F_1^- - Top1 = 0.831$	$F_1^- - Top1 = 0.885$
	$F_1^- - Top5 = 0.863$	$F_1^- - Top5 = 0.917$
	$F_1^+ - Top1 = 0.843$	$F_1^+ - Top1 = 0.897$
	$F_1^+ - Top5 = 0.892$	$F_1^+ - Top5 = 0.934$
	$F_1^\circ - Top1 = 0.654$	$F_1^\circ - Top1 = 0.678$
UCF50	$F_1^- - Top1 = 0.778$	$F_1^- - Top1 = 0.823$
	$F_1^- - Top5 = 0.813$	$F_1^- - Top5 = 0.859$
	$F_1^+ - Top1 = 0.834$	$F_1^+ - Top1 = 0.834$
	$F_1^+ - Top5 = 0.862$	$F_1^+ - Top5 = 0.862$
	$F_1^\circ - Top1 = 0.586$	$F_1^\circ - Top1 = 0.637$
HMDB51	$F_1^- - Top1 = 0.618$	$F_1^- - Top1 = 0.658$
	$F_1^- - Top5 = 0.679$	$F_1^- - Top5 = 0.694$
	$F_1^+ - Top1 = 0.652$	$F_1^+ - Top1 = 0.687$
	$F_1^+ - Top5 = 0.693$	$F_1^+ - Top5 = 0.724$
	$F_1^\circ - Top1 = 0.834$	$F_1^\circ - Top1 = 0.591$
	$F_1^\circ - Top5 = 0.862$	$F_1^\circ - Top5 = 0.637$

**Table 7** F1-Score comparison of  $S^*$  and  $S^{**}$

Dataset	120*120	600*600
UCF11	$S^*_{F_1^\circ} - Top1 = 0.654$	$S^*_{F_1^\circ} - Top1 = 0.678$
	$S^*_{F_1^\circ} - Top5 = 0.669$	$S^*_{F_1^\circ} - Top5 = 0.685$
	$S^{**}_{F_1^\circ} - Top1 = 0.878$	$S^{**}_{F_1^\circ} - Top1 = 0.912$
	$S^{**}_{F_1^\circ} - Top5 = 0.907$	$S^{**}_{F_1^\circ} - Top5 = 0.927$
UCF50	$S^*_{F_1^\circ} - Top1 = 0.586$	$S^*_{F_1^\circ} - Top1 = 0.637$
	$S^*_{F_1^\circ} - Top5 = 0.627$	$S^*_{F_1^\circ} - Top5 = 0.681$
	$S^{**}_{F_1^\circ} - Top1 = 0.857$	$S^{**}_{F_1^\circ} - Top1 = 0.905$
	$S^{**}_{F_1^\circ} - Top5 = 0.897$	$S^{**}_{F_1^\circ} - Top5 = 0.924$
HMDB51	$S^*_{F_1^\circ} - Top1 = 0.569$	$S^*_{F_1^\circ} - Top1 = 0.591$
	$S^*_{F_1^\circ} - Top5 = 0.614$	$S^*_{F_1^\circ} - Top5 = 0.637$
	$S^{**}_{F_1^\circ} - Top1 = 0.839$	$S^{**}_{F_1^\circ} - Top1 = 0.876$
	$S^{**}_{F_1^\circ} - Top5 = 0.873$	$S^{**}_{F_1^\circ} - Top5 = 0.904$



Numerous studies present their findings based on accuracy; keep in mind that our final system is built based on Block10 and frame size of  $120 \times 120$ , thus Fig. 14 shows more relevant matching results than those of  $S^*$ .

In addition, the following Table 8 compares our findings to those of the state-of-the-art; take note that  $S^*$  does not support different transformations. Thus,  $S^{**}$ , which supporting scaling, rotation mirroring, and translation, presents competitive results that achieves a 90% of an accuracy rate compared to other approaches.

Phan et al. [27] approach achieved an accuracy rate of 31.5% on HMDB51 by reducing the feature dimensions using PCA and SVM. While Zheng et al. [33] employed Deep Ensemble Machine method that is based on VGG and C3-D using a novel RLE layer, to achieve 64.9%.

In addition, Fuzzy descriptors of Zuo et al. [19] achieves 90.3% on UCF50 but it does not support video transformations, like Saoudi and Jai-Andalousi [26] approach, which only presents 79.9% based on BCS and distributed clusters. On the other side, Uijlings et al. [20] attempt to combine different densities of HOG, HOF, MBH using SVM and Fisher vectors; for quantization. However, this combination achieved an accuracy of 81.8% on UCF50, but it leads to a slow processing.

Saddam Bekhet and Amr Ahmed [17] presented a visual similarity framework using a graph-based signature, which represents the motion information by block across the shot. The results exceed 90% on UCF11 and UCF50 and it is greater than 50% on HMDB51, but this approach does not support the rotation. Although Lan et al. [22] method achieved an accuracy of 94.4% on UCF50 and 65% on HMDB51, it comes with high computational requirements. Finally, Simonyan et al. [32] focused on two-stream CNN, but their model and results are not very interesting, only 59.4% based on HMDB51.

According to our model, test, and analysis using one node and 4 threads, our approach  $S^{**}$  outperforms the state-of-the-art models in terms of accuracy. The average time-processing is not more than 4.2s for feature extraction and 6s at maximum for comparing near-duplicate content of 6000 videos. Thus, despite our approach takes twice as long as [26], it saves 90% of the resources (one node against 10 nodes

**Table 8** Comparison of our results to the state-of-the-art in terms of accuracy (%)

Approach	UCF11	UCF50	HMDB51
Two-Stream CNN [32]	–	–	59.4
Dense HOG/HOF/MBH [20]	–	81.8	–
MIFS [22]	–	94.4	65
Motion Dense Sampling [21]	87.5	–	–
DCGP [17]	90.8	90.6	50.8
PCA, VLAD, SVM [27]	–	–	31.5
Deep Ensemble Machine [33]	–	–	64.9
Fuzzy Descriptors [19]	–	90.3	–
Real-time BCS matching [26]	–	79.9	72.4
Proposed approach $S^*$ (Test 1)	89.7	91.2	64.1
Proposed approach $S^*$ (Test 2)	92.3	93.4	68.6
Proposed approach $S^*$ (Test 3)	67.4	65.6	63.7
Proposed approach $S^{**}$	93.4	90.1	84.3

**Table 9** Average processing time (APT) of  $S^{**}$  compared to the-state-of-the-art

Approach	APT (by second)—UCF50
Udding et al. [24]	$\simeq 68$ s
Xu et al. [25]	$\simeq 33$ s
Saoudi et al. [26](10 nodes $\times$ 4 threads)	$\simeq 05$ s
Proposed approach $S^{**}$	$\simeq 10$ s

(Table 9)). However, the parallel approach could help to economize more processing time.

Finally, the proposed approach, which is based on dominant color, motion signatures, and invariant moments for all GOP blocks of the video using different frame/block dimensions, is fast and applicable to find near-duplicate videos even if they are mirrored, rotated, scaled, translated or if parts of the content are hidden (no more than 20%), where the  $F_1$  exceeds 60%, 70%, and 80% for HMDB51, UCF50, and UCF11, respectively and the accuracy exceeds 90% specially on UCF datasets.

## Conclusions

This paper presents a robust and fast CBVR approach to detecting similar videos even if users hide content parts and mirror, resize, rotate or translate videos. Different dimensions of video frames and blocks are used to extract the relevant visual and motion features. Thus, the video is represented by visual and motion signatures with a fixed number of functionalities. The evaluation of  $Top_1$  results overcomes other accuracy-based work. The F1-score and its average are used to compare the results pertaining to different frame/block dimensions for  $Top_1$  and  $Top_5$ . Thus, the results reached over 80% for the F1-score and over 90% for accuracy using realistic video datasets.

We plan to extend our research in the future to test different video transformations based on deep learning in order to identify another means to strengthen this study.

## Abbreviations

AP: Average Precision; BCS: Bounded Coordinate System; BEMD: Bidimensional Empirical Mode Decomposition; BIMFs: Bidimensional Intrinsic Mode Functions; C3-D: 3D Convolutional Networks; CBVR: Content-Based Video Retrieval; CLGC: Concatenation of Local and Global Color; CNN: Convolutional Neural Networks; CWT: Color Wavelet Transform; DCT: Discrete Cosine Transform; DTs: Dense-Trajectories; ELP: Elliptical Local Pattern; FABEMD: Fast and Adaptive Bidimensional Empirical Mode Decomposition; FIMT-DD: Fast Incremental Model Tree with Drift Detection; GGD: Generalized Gaussian Distribution; GOP: Group Of Pictures; GPUs: Graphics Processing Unit; GTP: Ground Truth Positives; HMG: Histogram of Motion Gradients; HOG: Histogram of Oriented Gradients; HOF: Histogram of Optical Flow; LBP: Local Binary Pattern; LTP: Local Ternary Pattern; mAP: mean Average Precision; MOMP: Motion of Oriented Magnitude Patterns; MBH: Motion Boundary Histograms; MPEG: Moving Picture Experts Group; RGB: Red, Green and Blue colors; RLE: Rectified Linear Encoding; SVM: Support Vector Machine; VGG: Visual Geometry Group.

## Acknowledgements

Not applicable.

## Authors' information

Abderrahmane Adoui El Ouadrhiri received his B.Sc. degree in mathematics and computer science from Hassan II University of Casablanca, in 2008 and M.Sc. degree in computer networking systems from Hassan I University, Settat, in 2010. He is a Ph.D. student with LR2I Lab., FSAC, Hassan II University of Casablanca, Morocco. His research interests include image and video processing, information extraction systems, machine learning and computer vision.

Said Jai Andaloussi was born in Fez (Morocco) on October 21, 1981. In 2010, he received a Ph.D. degree in computer science jointly supervised by the faculty of science - University of Sidi Mohamed Ben-Abdillah (Fez - Morocco) and Telecom Bretagne (Brest - France). He is a Professor of computer science in mathematics and computer science department,

Faculty of Sciences Ain Chock (FSAC), Hassan II University of Casablanca. He is the author of more than 40 papers. His current research interests include medical information processing, information modeling and analysis in medical images.

Ouail Ouchetto received M.Sc. degree (DEA) in applied Mathematics from Pierre, Marie Curie University (Paris VI), Paris, France, M.Sc. degree in informatics from Télécom Bretagne School of Engineers, Brest, France, and Ph.D. degree in modeling and engineering science from Paris-Sud XI University, Orsay, France, in 2006. He was a Postdoctoral Researcher with University Blaise Pascal, Clermont-Ferrand, France, and an Assistant Professor (ATER) at University of Paris-Sud. In September 2010, he joined Hassan II University, Casablanca, Morocco, as a Professor of Mathematics and Computer Science. His research interests include scientific computing, numerical computation, and computer science.

#### Authors' contributions

AAEO designed and implemented the proposed model, SJ-A supervised the methodology and tests, OO supervised the mathematical model. All authors read and approved the final manuscript.

#### Funding

Not applicable.

#### Availability of data and materials

Not applicable.

#### Declarations

##### Ethics approval and consent to participate

The authors accept the Journal of Big Data's ethics approval and agree to share their work for scientific advancement.

##### Consent for publication

The authors hereby consent to the publication of the work in the Journal of Big Data.

##### Competing interests

The authors declare they have no actual or potential conflicts of interest.

Received: 4 May 2021 Accepted: 8 October 2021

Published online: 23 October 2021

#### References

1. YouTube, YouTube Statistics 2021. <https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/>.
2. Aafaq N, Mian A, Liu W, Gilani SZ, Shah M. Video description: a survey of methods, datasets, and evaluation metrics. *ACM Comput Surv.* 2019;52(6):1–37.
3. Kabbai L, Abdellaoui M, Douik A. Image classification by combining local and global features. *Vis Comput.* 2019;35(5):679–93.
4. Bhuiyan SM, Adhami RR, Khan JF. Fast and adaptive bidimensional empirical mode decomposition using order-statistics filter based envelope estimation. *EURASIP J Adv Signal Process.* 2008;2008:1–18.
5. Palkar PM, Udipi VR, Patil SA. A review on bidimensional empirical mode decomposition: A novel strategy for image decomposition, In: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDs), 2017; pp. 1098–1100.
6. Banerji S, Verma A, Liu C. LBP and color descriptors for image classification. Berlin: Springer; 2012.
7. Yu J, Qin Z, Wan T, Zhang X. Feature integration analysis of bag-of-features model for image retrieval. *Neurocomputing.* 2013;120:355–64.
8. Hu X, Ding Y. Image matching with an improved descriptor based on SIFT, Seventh International Conference on Electronics and Information Engineering. 2017;10322: 103221.
9. Chéron G, Laptev I, Schmid C. P-cnn: Pose-based cnn features for action recognition, In: Proceedings of the IEEE international conference on computer vision, 2015; pp. 3218–3226.
10. Feng Y, Zhou P, Xu J, Ji S, Wu D. Video big data retrieval over media cloud: a context-aware online learning approach. *IEEE Trans Multimedia.* 2018;21(7):1762–77.
11. Ye H, Wu Z, Zhao R-W, Wang X, Jiang Y-G, Xue X. Evaluating two-stream CNN for video classification, In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015; pp. 435–442.
12. UCF11, YouTube Action Data Set 2011. [https://www.crcv.ucf.edu/data/UCF\\_YouTube\\_Action.php](https://www.crcv.ucf.edu/data/UCF_YouTube_Action.php).
13. UCF50, YouTube Action Recognition Data Set 2012. <https://www.crcv.ucf.edu/data/UCF50.php>.
14. HMDB51, Large Video Database for Human Motion Recognition 2011. <https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/>.
15. Jegham I, Khalifa AB, Alouani I, Mahjoub MA. Vision-based human action recognition: an overview and real world challenges. *Forensic Sci Int.* 2020;32:200901.
16. Ito I. Gradient-based global features for seam carving. *EURASIP J Image Video Process.* 2016;2016(1):1–9.
17. Bekhet S, Ahmed A. An integrated signature-based framework for efficient visual similarity detection and measurement in video shots. *ACM Trans Inf Syst.* 2018;36(4):1–38.
18. Bekhet S, Ahmed A. Evaluation of similarity measures for video retrieval. *Multimed Tools Appl.* 2020;79(9):6265–78.
19. Zuo Z, Yang L, Liu Y, Chao F, Song R, Qu Y. Histogram of fuzzy local spatio-temporal descriptors for video action recognition. *IEEE Trans Industr Inform.* 2019;16(6):4059–67.

20. Uijlings J, Duta IC, Sangineto E, Sebe N. Video classification with densely extracted hog/hof/mbh features: an evaluation of the accuracy/computational efficiency trade-off. *Int J Multimed Inf Retr*. 2015;4(1):33–44.
21. Aihara K, Aoki T. Motion dense sampling and component clustering for action recognition. *Multimed Tools Appl*. 2015;74(16):6303–21.
22. Lan Z, Lin M, Li X, Hauptmann AG, Raj B. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015; pp. 204–212.
23. Wibisono A, Mursanto P, Adibah J, Bayu WD, Rizki MI, Hasani LM, Ahli VF. Distance variable improvement of time-series big data stream evaluation. *J Big Data*. 2020;7(1):1–13.
24. Uddin MA, Joolee JB, Alam A, Lee Y-K. Human action recognition using adaptive local motion descriptor in spark. *IEEE Access*. 2017;5:21157–67.
25. Xu W, Uddin MA, Dolgorsuren B, Akhond MR, Khan KU, Hossain MI, Lee Y-K. Similarity estimation for large-scale human action video data on spark. *Appl Sci*. 2018;8(5):778.
26. Saoudi EM, Jai-Andaloussi S. A distributed content-based video retrieval system for large datasets. *J Big Data*. 2021;8(1):1–26.
27. Phan H-H, Vu N-S, Nguyen V-L, Quoy M. Action recognition based on motion of oriented magnitude patterns and feature selection. *IET Comput Vis*. 2018;12(5):735–43.
28. Iqbal S, Qureshi AN, Lodhi AM. Content based video retrieval using convolutional neural network. In: *Proceedings of SAI Intelligent Systems Conference*, 2018; pp. 170–186.
29. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L. Large-scale video classification with convolutional neural networks. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014; pp. 1725–1732.
30. Idrees H, Zamir AR, Jiang Y-G, Gorban A, Laptev I, Sukthankar R, Shah M. The THUMOS challenge on action recognition for videos “in the wild.” *Comput Vis Image Underst*. 2017;155:1–23.
31. Baldomero-Naranjo M, Martínez-Merino LI, Rodríguez-Chía AM. A robust SVM-based approach with feature selection and outliers detection for classification problems. *Expert Syst Appl*. 2021;178:115017.
32. Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos; 2014. arXiv preprint [arXiv:1406.2199](https://arxiv.org/abs/1406.2199). Accessed 17 Mar 2021.
33. Zheng J, Cao X, Zhang B, Zhen X, Su X. Deep ensemble machine for video classification. *EEE Trans Neural Netw Learn Syst*. 2018;30(2):553–65.
34. Ahmed N, Natarajan T, Rao KR. Discrete cosine transform. *IEEE Trans Comput*. 1974;100(1):90–3.
35. Television, Television Standards—formats and techniques 2021. [https://en.wikipedia.org/wiki/Broadcast\\_television\\_systems](https://en.wikipedia.org/wiki/Broadcast_television_systems).
36. Ye H, Qu X, Liu S, Li G. Hybrid sampling method for autoregressive classification trees under density-weighted curvature distance. *Enterp Inf Syst*. 2021;15(5):749–68.
37. Nunes JC, Guyot S, Deléchelle E. Texture analysis based on local analysis of the bidimensional empirical mode decomposition. *Mach Vis Appl*. 2005;16(3):177–88.
38. Mahraz MA, Riffi J, Tairi H. Motion estimation using the fast and adaptive bidimensional empirical mode decomposition. *J Real Time Image Process*. 2014;9(3):491–501.
39. Ouadrhiri AAE, Andaloussi SJ, Saoudi EM, Ouchetto O, Sekkaki A. Similarity performance of keyframes extraction on bounded content of motion histogram. In: *International Conference on Big Data, Cloud and Applications*, 2018; pp. 475–486.
40. Varanasi MK, Aazhang B. Parametric generalized Gaussian density estimation. *J Acoust Soc Am*. 1989;86(4):1404–15.
41. Ding JR, Yang JF. Adaptive group-of-pictures and scene change detection methods based on existing H. 264 advanced video coding information. *IET Image Process*. 2008;2(2):85–94.
42. Hu M-K. Visual pattern recognition by moment invariants. *IEEE Trans Inf Theory*. 1962;8(2):179–87.
43. Flusser J, Suk T. Rotation moment invariants for recognition of symmetric objects. *IEEE Trans Image Process*. 2006;15(12):3784–90.
44. Mamistvalov AG. N-dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids. *IEEE Trans Pattern Anal Mach Intell*. 1998;20(8):819–31.
45. Hosny KM. Fast computation of accurate Zernike moments. *J Real Time Image Process*. 2008;3(1):97–107.
46. Favorskaya M, Pyankov D, Popov A. Motion estimations based on invariant moments for frames interpolation in stereovision. *Procedia Comput Sci*. 2013;22:1102–11.
47. Krause EF. *Taxicab geometry: an adventure in non-Euclidean geometry*. North Chelmsford: Courier Corporation; 1986.
48. Patel SP, Upadhyay SH. Euclidean distance based feature ranking and subset selection for bearing fault diagnosis. *Expert Syst Appl*. 2020;154:113400.
49. Lowe DG. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis*. 2004;60(2):91–110.
50. Chardy P, Glemarec M, Laurec A. Application of inertia methods to benthic marine ecology: practical implications of the basic options. *Estuar Coast Marine Sci*. 1976;4(2):179–205.
51. Rubner Y, Tomasi C, Guibas LJ. The earth mover’s distance as a metric for image retrieval. *Int J Comput Vis*. 2000;40(2):99–121.
52. Bekhet S, Ahmed A. Compact signature-based compressed video matching using dominant color profiles (dcp). In: *2014 22nd International Conference on Pattern Recognition*, 2014; pp. 3933–3938.
53. Saoudi EM, Ouadrhiri AAE, Andaloussi SJ, Warrak OE, Sekkaki A. Content based video retrieval by using distributed real-time system based on storm. *IJERTCS*. 2019;10(4):60–80.
54. Thamsen L, Beilharz J, Tran VT, Nedelkoski S, Mary Kao O. Hugo, and Hugo\*: learning to schedule distributed data-parallel processing jobs on shared clusters. *Concurr Comput*. 2020;33:e5823.
55. Gonzalez RC, Woods RE. *Digital Image Processing*, Publisher: Pearson; 4th edition (March 20, 2017), ISBN-13: 978-0133356724.

## Publisher’s Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.