Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01417-z

ORIGINAL ARTICLE

Keyframe recommendation based on feature intercross and fusion

Guanci Yang[1,3] · Zonglin He[1] · Zhidong Su[2] · Yang Li[1] · Bingqi Hu[3]

Received: 7 September 2023 / Accepted: 9 March 2024

Abstract
Keyframe extraction can effectively help users quickly understand video content. Generally, keyframes should be representative of the video content and simultaneously be diverse to reduce redundancy. Aiming to find the features of frames and filter out representative frames of the video, we propose a method of keyframe recommendation based on feature intercross and fusion (KFRFIF). The method is inspired by the implied relations between keyframe-extraction problem and recommendation problem. First, we investigate the application of a recommendation framework to the keyframe extraction problem. Second, the architecture of the proposed KFRFIF is put forward. Then, an algorithm for extracting intra-frame image features based on the combination of multiple image descriptors is proposed. An algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods is designed. Moreover, A recommendation model based on feature intercross and fusion is put forward. An ablation study is further performed to verify the effectiveness of the submodule. Ultimately, the experimental results on four datasets with five outstanding approaches indicate the superior performance of our approach.

Keywords Keyframe recommendation · Feature extraction · Feature fusion · Video summarization

Introduction

The proliferation of mobile devices have led to an explosion of video data [1, 2]. For video service companies, making video data easier to retrieve [3], preview, and manage is an urgent need [4]. Users want a user-friendly way to quickly understand a large number of long videos before viewing them [5]. Keyframe extraction is extremely relevant as an effective method for processing large amounts of video data [6]. Keyframe extraction can also be used as a preprocessing step in video-analysis applications to solve the problem of processing a large number of video frames [7, 8]. For the video summarization task, the extraction of keyframes is its foundation, and whether the keyframes are reasonable or not directly determines the quality of the video summarization.

Given the potential applications of keyframe extraction, many effective keyframe-extraction methods have been proposed. We believe that keyframe extraction techniques can be divided into two distinct phases with respect to whether or not deep learning methods are introduced. The extraction of keyframes in the first stage relies on manually constructed video features [9–31]. Bommisetty et al. [9] determined the shot transition boundaries by estimating the similarity of gradient size features between consecutive frames, and then selecting the frame with the largest mean and standard deviation as the keyframe for that shot, which resulted in an adaptive thresholding keyframe selection algorithm [10]. Bommisetty et al. [11] investigated the extraction of keyframes using Pearson Correlation Coefficient (PCC) and Color Moments (CMs), which has the advantage of using a combination of linear transformation invariant features of

✉ Guanci Yang
gcyang@163.com

Zonglin He
DKtys@outlook.com

Zhidong Su
zhidong.su@okstate.edu

Yang Li
liyanggzu@163.com

Bingqi Hu
bqhu@gzu.edu.cn

1 Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University, Guiyang 550025, China

2 School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74074, USA

3 State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China

Published online: 15 April 2024

PCC and scale and rotation invariant features of CM. The technique of detecting lens boundaries using the combined feature set (PCC and CM) provides a new idea for efficient segmentation of lenses; the method, however, still suffers from the problem of difficulty in dealing with high motion intensity lenses. Sun et al. [12] proposed a new method for keyframe extraction by affine-propagation clustering algorithm, achieving good evaluation results of the sequence reconstruction of keyframes. Starting from the perspective that single image descriptors (color, texture, etc.) are not always effective in extracting keyframes, Ioannidis et al. [13] proposed a weighted fusion method of multiple descriptors, i.e., automatically estimating the weight of each descriptor, then constructing correlation matrices between each descriptor and a particular video shot, and using them as inputs to a spectral clustering algorithm to finally obtain keyframes. They obtain keyframes with good robustness in the presence of rich video content. The shortcoming of this method is that it is difficult to set the number and center of clusters in advance. Represented by the work of Mei et al. [14], filtering keyframes with the sparse constraint L-0 criterion and constructing a linear combination of features of keyframes as a sparse dictionary provides a new idea for good reconstruction of all video frames. Although this type of method can extract keyframes better, the dictionary takes up more memory, while learning the dictionary requires high computational overhead and the generalization performance of the dictionary is weak. With the improvement of the accuracy of motion capture data, two important properties of motion data, namely sparsity and Riemannian flow structure, have been identified. Accordingly, Xia et al. [15] proposed a joint kernel sparse representation model to analyze the sparsity and Riemannian flow structure features, and constructed a method based on a reconstruction-error optimization algorithm to extract the optimal keyframes from the initial keyframes (the local maxima of the constructed pose saliency curves) [16]. Banerjee et al. [31] introduced a novel deep spatiotemporal feature extraction method, leveraging particle swarm optimization, to enhance the efficiency of video retrieval. In summary, we can see that this stage of the keyframe approach allows the authors to clearly select a certain dimension of video features as a screening criterion, while the computational acquisition of features is transparent.

In the second phase after the introduction of deep learning methods, the method of manually constructing features is gradually replaced by the method of extracting features from the backbone network, while at the same time the process of acquiring features first and then filtering keyframes is replaced by an end-to-end process. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can efficiently process both spatial and temporal information. To improve the performance of the identified architectures to efficiently combine CNNs and RNNs, Kiziltepe et al. [17] proposed a novel action-template based keyframe extraction method, which identifies the informative regions of each frame and selects the keyframes based on the similarity between these regions. Mahas Seni et al. [18] first applied Generative Adversarial Networks (GANs) to keyframe detection in video, which used CNNs to extract the features of each frame, and then encoded the features through LSTMs. Kar et al. [19] used a two-stream network containing spatial and temporal nets, combining CNNS with MIL framework to detect high scoring keyframes in the video for action recognition. To solve the keyframe extraction problem using features extracted by CNNs, Muhammad et al. [20] designed a hierarchical weighted fusion deep CNN framework that assigns a score to each frame by detecting whether the video frame contains a specific object or not.

By analyzing the above two-stage approach, inspired by the recommendation algorithm, we design a novel keyframe extraction paradigm. First, video features are artificially constructed from multi-dimensional video frames such as color, texture and shape. The purpose of reducing four-dimensional video data to serialized two-dimensional features is achieved. Second, the importance of each frame is obtained using a recommendation model (a deep learning model, part of the structure is commonly used in the field of recommendation algorithms to deal with serialized multidimensional features). Finally, keyframes are obtained based on the ordering of importance. In summary, the contributions of this paper are as follows.

- We propose a paradigm called keyframe recommendation based on feature intersection and fusion to solve the keyframe extraction problem. This paradigm introduces the approach of recommendation algorithms for processing structured textual data to visual data.
- We propose an algorithm for jointly extracting intra-frame image features based on multiple image descriptors and an algorithm for jointly extracting inter-frame distance features based on multiple distance calculation methods. The algorithm comprehensively describes the frame by constructing rich image features for each frame using multiple descriptors and describes the video sequence by calculating inter-frame distance features.
- A deep learning model for inputting multi-dimensional long sequence data and outputting importance scores is proposed, which adds dimensional bit flags and multi-head attention mechanisms to the skeleton of the feature fusion structure commonly used in recommendation models.
- KFRFIF was compared with other methods. The results of the comparison experiments on HMDB-51 [32], UCF-101 [33] and SumMe [34], TVSum [35] datasets show that the proposed KFRFIF outperforms the state-of-the-art methods in terms of the average F scores, which are 67.9%, 72.0%, 53.6%, 61.4%, respectively.

The rest of this paper is organized as follows. The related work is reviewed in the next section. The subsequent section describes the process by which the recommended method introduces the key extraction problem followed by which the procedure of the proposed approach is detailed. Experimental results are presented in the penultimate section. The last section summarizes and discusses the work.

Related work

We review the state-of-the-art keyframe extraction methods in terms of two stages of keyframe extraction technique development.

The first stage of artificially constructing video features has the following four main sub-themes.

(a) Shot-based measure
Similarity features are frequent in shot-based methods, and the keyframe extraction in the following three studies relies on the computation of frame similarity. In the first step, Priya et al. [21] clustered the frames sequentially into shots using feature extraction, continuous value construction step of shot-boundary detection process, and shot-frame clustering technique. The second step selects clusters with large dispersion rates for inter cluster similarity analysis (ICSA) and extracts keyframes based on subshots using ICSA. The method finally obtains a good F-score for shot-boundary detection. In the first step, Mounika Bommisetty et al. [9] estimated the feature similarity between gradient sizes of consecutive frames to determine the boundaries of lens transitions. The second step selects the frame with the largest mean and standard deviation as the keyframe for that shot. The method evaluates excellent results in terms of figure of merit, detection percentage, accuracy, and missing coefficient. Thakre et al. [10] noticed that for some video clips, static thresholding may lead to keyframes with added error information, so the authors' team proposed an adaptive thresholding keyframe-selection algorithm to segment the clips efficiently and obtain representative frames. The method has been extensively evaluated on more than 200 video clips, and the results are accurate and satisfactory.
As research progresses, more features for extracting effects and conditions for filtering keyframes are introduced into the shot-based approach. Bommisetty et al. [11] fully referred to the linear-transformation invariance characteristics of the PCC and the scale and rotation invariance of the CMs. The first step detects the boundaries of the shots using a combined feature set (PCC and CM). The second step selects the frame with the highest mean and standard deviation from each shot as the keyframe. The main advantage of the method is its ability to detect sudden and gradual lens transitions. Omidyeganeh et al. [22] addressed a new approach to the keyframe-extraction problem using the generalized Gaussian density (GGD) parameters of wavelet transform subbands along with Kullback–Leibler distance (KLD) measurement. The first step uses the KLD between the GGD feature vectors to select the lens and cluster boundaries. The second step is also based on the similarity features of frames to locate keyframes. The method accuracy is experimentally proven to be high.
The shortcoming of the shot-based keyframe-extraction method is that it cannot handle shots with high motion intensity. It is easy to select too many keyframes when dealing with shots having little motion variation, whereas for shots with more motion variation, the content cannot be adequately described with one or two keyframes.

(b) Clustering-based measure
To improve the video-frame clustering effect, affine-propagation clustering [12], density clustering [23], spectral clustering [13], hierarchical clustering [24], Markov chain based clustering and adjacent matrix based clustering [36] are combined with a certain feature of the frame to achieve keyframe extraction, respectively. Sun et al. [12] proposed a new method for keyframe extraction through the affine-propagation clustering algorithm. This clustering algorithm relies on the interframe similarity metrics based on human part features. The method adaptively finds the best keyframes of the video from the information distribution of the video itself and runs quickly. Finally, the evaluation of keyframe-based sequence reconstruction is well validated by the results. Xu et al. [23] assumed that the motion is fitted by curve slices. The segmentation points are judged to be in clustering based on the fact that some frames are selected as segmentation points several times, but other frames near these frames are selected as segmentation points less often. They applied the density clustering algorithm to select keyframes from the clusters, and the center of density is clearly the best candidate keyframe. The method achieves impressive results. Ioannidis et al. [13] argued that using a single image descriptor (color, texture, etc.) to extract keyframes is not always effective because no single descriptor surpasses the others in all video cases. The authors proposed a method for the weighted fusion of several descriptors, automatically estimating the weight of each descriptor. These weights reflect the relevance of each descriptor to a particular video shot. They are used to form a combined similarity matrix, which is then used as an input to a spectral clustering algorithm and, ultimately, to obtain keyframes. Numerical

experiments using various videos have shown that this method is effective in summarizing video shots regardless of the characteristics of the visual content of the video. Fei et al. [24] proposed a combination of sparse selection (SS) and mutual information-based clustering hierarchical [25] clustering (MIAHC) to generate valid keyframes. The SS algorithm is first applied to the original video sequence to obtain candidate keyframes, which are used as initial clusters. The improved MIAHC is then further processed to eliminate redundant images and generate the final keyframes. The proposed method overcomes the problems of information redundancy and computational complexity that plague SS methods while also reducing the computation time for clustering large videos.

The shortcoming of the clustering-based approach is that the number of clusters and cluster centers need to be set in advance before clustering. In the case of uncertain video content, setting the number and centers of clusters in advance is very difficult.

(c) Measurements based on linear features

Each frame can be represented as a linear combination of keyframes, whereby Mei et al. [14] improved the convex-relaxation-based sparse dictionary selection method. The method uses a sparse constrained L-0 criterion and the keyframes are directly selected as sparse dictionaries. The most significant advantage of this method is that all video frames can be reconstructed well. The method is also highly real time, so the authors further developed an online version. Li et al. [26] proposed a new shot-boundary detection algorithm that uses sparse coding to learn the dictionary from a given video and update the atoms in the dictionary. Specifically, the method follows the concept that different shots cannot be reconstructed with the learned dictionary, i.e., each shot in the video needs to learn the dictionary corresponding with it. After determining the shot boundaries, a representative keyframe is selected from each shot. This method is shown to be the powerful keyframe-extraction algorithm on VSUMM and YouTube datasets.

Ma et al. [27] noticed that the sparse subset selection-based algorithms proposed in the past do not consider local or global relationships between frames. Therefore, the authors' proposed a similarity-based block sparse subset-selection method. The method applies a specially designed transformation matrix on the kernel block sparse subset-selection model to characterize the global interframe relationships by similarity. The proposed method has the following advantages: the global relationships between all frames can be considered by the similarity between each frame and any other frame, and the local relationships of neighboring frames are further characterized by block sparse coding. The method is shown to be superior to other methods based on sparse subset selection.

(d) Motion-based measure

A class of keyframe-extraction algorithms have also been proposed by some scholars based on the properties of object motion features [28]. These algorithms learn the features of the original video motion by modeling, mapping, and intelligent algorithms and obtain keyframes from that feature.

To reduce the reconstruction error and control the optimum compression rate during keyframe extraction, Zhang et al. [29] proposed a keyframe-extraction method for human motion-capture data based on a multipopulational genetic algorithm. The fitness function is defined to satisfy the objectives of minimum reconstruction error and optimum compression rate, where multiple initial populations are subject to co-evolution. The multipopulational genetic algorithm considers global and local searches. Experimental results show that the algorithm can effectively extract keyframes from motion-capture data and satisfy the required reconstruction error. Xia et al. [15] proposed a new model, called the joint kernel sparse representation (SR), which can correctly model two important features of motion data, i.e., sparsity and Riemannian manifold structure. The unreasonable distribution and redundancy of extracted keyframes can be successfully solved by SR modeling. Liu et al. [16] constructed a pose saliency curve by learning the feature representation of human motion. They extracted the best keyframes from the initial keyframes (local maxima of the curve) according to the reconstruction-error optimization algorithm. Experiments demonstrate that this method can effectively extract keyframes with high visual perception quality and low reconstruction error, thereby better meeting the needs of real-time analysis and compression of motion-capture data. Drawing upon motion features, Clinton et al. [37] presented a pioneering framework designed to precisely extract the potential motion manifold from a complete sequence of keyframes, employing keyframe-based constraints. This framework encompasses a crucial stage dedicated to identifying the potential motion subspace, known as the keyframe coding stage. Their innovative method exhibits a remarkable visual resemblance to authentic ground motion.

For the motion-based keyframe-extraction method, the motion object state changes frequently. Due to the diversity of motion targets and similarity of movements, missing is easy keyframes if only motion features are considered, whereas the deviation of feature extraction may be large.

The extraction of video features in the second stage is left to the neural network.

Kiziltepe et al. [17] proposed a novel keyframe extraction method based on action templates by effectively combining CNN and RNN, which identifies the information regions of each frame and selects keyframes based on the similarity between these regions. Experiments demonstrate that the method can significantly improve the accuracy of downstream video classification tasks. Muhammad et al. [20] designed a hierarchical weighted fusion deep CNN framework and proposed a new keyframe extraction method accordingly. The method first extracts discriminative features from the deep CNN for shot segmentation. Then, memory features and entropy features of the image are predicted from the CNN model. The extracted features are efficiently computed by a hierarchical weighted fusion mechanism to generate an aggregation score. Finally, the aggregation score is used to compose an attention curve to locate salient keyframes. The effectiveness of the framework is finally validated experimentally. Kar et al. [19] successfully combined CNNS with the MIL framework to detect high scoring keyframes in videos based on RGB and optical streaming data using a two-stream network containing both spatial and temporal networks. The method learns how to pool these discriminative and informative frames while discarding most of the uninformative frames when performing a single temporal scan of the video, and the results of the detection are used for action recognition. Mahas Seni et al. [30] first applied Generative Adversarial Networks (GAN) for keyframe detection in video, which uses CNN to extract features of each frame and then a novel generative adversarial framework to detect keyframes. The framework consists of a summarizer and a discriminator, which is an autoencoder Long Short-Term Memory (LSTM) network designed to first select video frames and then decode the obtained summaries to reconstruct the input video. The discriminator is another LSTM designed to distinguish between the original video and the video reconstructed by the summarizers. The method is evaluated on four benchmark datasets and its performance is very competitive compared to fully supervised state-of-the-art methods.

Based on the overview of related work, we note that in the stage before deep learning was introduced, many image descriptors were used in the academic sessions to extract video features such as SIFT operators, PCC, CMs, color histograms and image entropy. At the same time, this stage requires distance calculation of the extracted features to distinguish the importance of each frame. There are several distance calculation methods that can be used to measure frame features, such as Euclidean distance and KLD; however, deep learning based keyframe extraction methods do not take full advantage of these frame features and frame spacing calculation methods. Based on the above analysis, we are inspired by recommendation algorithms and adopt this new perspective of the recommendation problem to analyze the keyframe extraction problem.

Association of keyframe extraction and recommendation

First, we need to clarify the meaning of keyframe extraction and recommendation. Indeed, the goal of keyframe extraction is to obtain from a video a collection of video frames that completely summarize the video content and have a small number of frames [38]. The meaning of recommendation is more complex and needs to be combined with specific scenarios to illustrate, for example, in shopping recommendation scenarios, it is necessary to filter out what users are more likely to need based on the attribute characteristics of the product (price, origin, category) combined with the user's purchase record and browsing record. In general, both keyframe extraction and recommendation hope to be able to filter out the part of the sample with more prominent importance in the overall sample, and this similarity of purpose is the basis for being able to use the framework of the recommendation method to solve the keyframe extraction problem.

Second, we need to map the keyframe extraction to the elements of recommendation.

- The object of the recommendation method is a collection of goods, which has features such as price, category, origin, etc. The object of keyframe extraction is the collection of videos, which has intra-frame image features and inter-frame distance features.
- The filtering basis of the recommendation method includes the purchase browsing records of the current user and the purchase browsing records of other users. Correspondingly, the screening basis of keyframe extraction is the importance annotation record of the current annotator for each frame of the whole video dataset, and at the same time, the importance judgement of other annotators can also be used as learning data.

Third, to achieve the purpose of applying the recommendation method to keyframe extraction, we need to be explicit about the specific steps of the recommendation method.

$$\begin{cases} X \xrightarrow{\text{feature engineer}} E(X) \\ H \xrightarrow{\text{collaborative filtering}} C(H) \\ F(C(H),\ E(X)) \xrightarrow{\text{mining}+\text{intercross}+\text{fusion}} R \end{cases} \tag{1}$$

Formula (1) represents the process of obtaining the recommendation result using the recommendation method. First,

the feature of recommendation object X is artificially constructed by feature engineering to obtain $E(X)$. Second, the user's historical data H is cleaned to obtain data $C(H)$ for collaborative filtering. Finally, the two kinds of data $E(X)$ and $C(H)$ are mined, crossed and fused by the recommendation model, and finally the recommendation result R is obtained.

Finally, we migrated the recommendation method to the field of keyframe extraction, and built a keyframe recommendation method, whose essence is to represent video frames by features, use the annotation of keyframes by multiple users, transform keyframe extraction into supervised learning tasks, and use deep learning model to output keyframes. Specially, the overall keyframe set K can be identified by performing the following steps.

- First, we get the intra-frame image features. Starting from the three dimensions of image features (color, texture and shape) [39], the color features of each frame of the video data set are calculated, including: color histogram, Gray-level Co-occurrence Matrix [40], SIFT key points [41]. Finally, based on the above three features, 192-dimension intra-frame image features are constructed.
- Then, the inter-frame distance feature reflects the distribution of the video data, which is very helpful in distinguishing the importance of frames. The 34-dimensional inter-frame distance feature is obtained by calculating three distances between the current frame and the previous frame, the current frame and the first frame, respectively. The three distances are divided into Euclidean distance, SIFT key point matching number response distance and color histogram difference distance, respectively.
- Eventually, we built a recommendation model. By mining, intersecting and fusing intra-frame image features and inter-frame distance features as well as multi-user annotation information, we obtain the high-level features of each frame and the importance of each frame. Finally, we get the set K of keyframes of the video dataset.

Proposed method

Architecture of the proposed KFRFIF

According to the above formulation of keyframe recommendation, this section designs an intra-frame image feature extraction algorithm, innovates an inter-frame distance feature extraction algorithm, and constructs a deep learning keyframe recommendation model based on feature cross and fusion mechanism. Accordingly, the keyframe recommendation method (KFRFIF) based on feature cross-fusion is proposed. Figure 1 shows the architecture of KFRFIF. Module A is used to obtain the in-frame image features f_i (see "Algorithm for extracting intra-frame image features based on the combination of multiple image descriptors" section). Module B is used to obtain the inter-frame distance feature f_d (see "Algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods" section). Module C is used to implement a feature intersection and fusion mechanism based on the features f_i and f_d to obtain the recommended values for the keyframe set K (see "Recommendation model based on feature intercross and fusion" section).

Algorithm for extracting intra-frame image features based on the combination of multiple image descriptors

Intra-frame image feature construction for keyframe recommendation algorithms is the process of mining and aggregating image features from video frames through feature engineering. The importance of this process for keyframe recommendation is mainly reflected in the following aspects.

- Keyframe recommendation requires processing serialized video data with a timeline. If high arithmetic cost is required to process the four-dimensional tensor directly using deep learning models, two-dimensional feature data can be obtained by extracting the image features of each frame through multiple image descriptors. This approach achieves dimensionality reduction of the data input to the model and greatly reduces the learning time.
- Artificially constructing theoretically solid and rich content features can avoid falling into the black box of deep-learning models [42], improve the interpretability of keyframe-recommendation algorithms, and be an effective complement for the self-learning features of neural networks.

Through the above analysis, to provide a comprehensive description of each frame from multiple dimensions such as color, texture and shape, we propose an algorithm for extracting intra-frame image features based on the combination of multiple image descriptors (IFCMI).

Firstly, IFCMI detects the SIFT key points of the video frame, and the key points are counted to form a distribution according to 40*30 partitions separately to obtain 64-dimensional shape features. Secondly, IFCMI counts the number of pixels of the video frame on 64 luminance partitions to form 64-dimensional luminance features. Finally, IFCMI counted the 4*4 size Gray Level Co-occurrence Matrix of the video frame in the four directions of 0°, 45°, 90°, and 135° and expanded it to form the 64-dimensional texture features. The key points found by SIFT are some very prominent points that do not change due to lighting, affine transformation and noise, such as the corner point, the edge

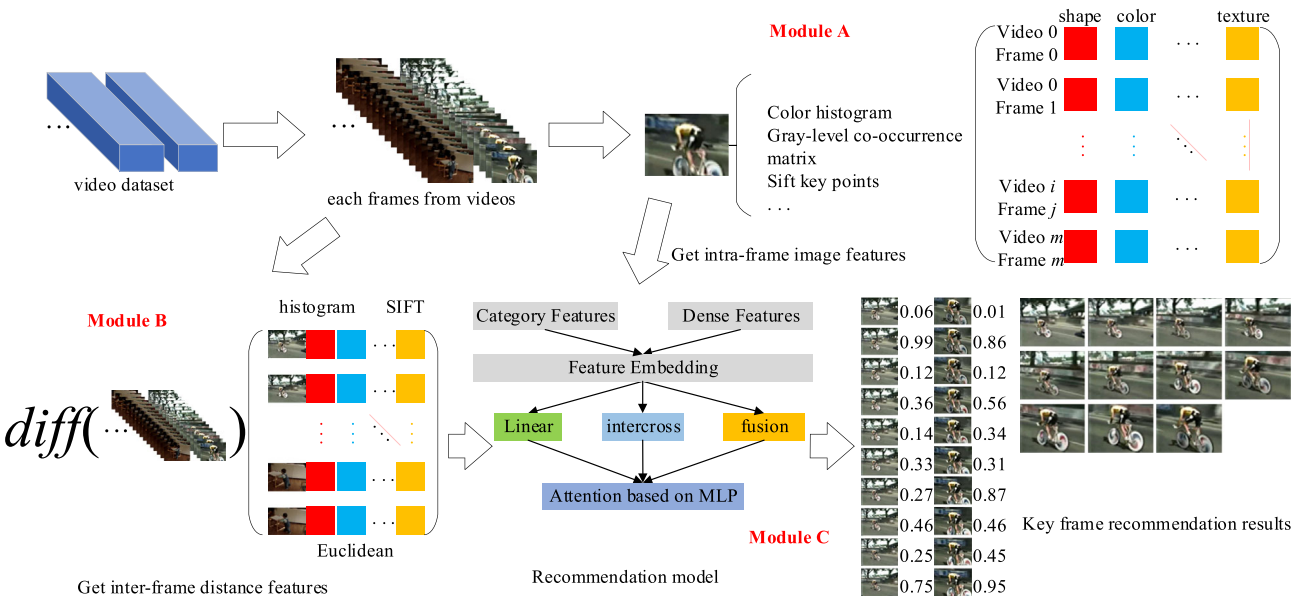

Fig. 1 Architecture of KFRFIF

point, the bright point in the dark area and the dark point in the bright area, etc., and their number reflects the criticality of the frame to a certain extent. Similarly, video frame brightness and texture are also important factors for filtering keyframes. The pseudocode of IFCMI is shown in Algorithm 1.

Algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods

Inter-frame distance is one of the main bases for determining the importance of a frame. Calculating the distance between the current frame and the first frame can measure the degree

Algorithm 1 Algorithm for extracting intra-frame image features based on the combination of multiple image descriptors

Input. Video v_i in the dataset D_v.
Output. Set of image features intra-frame F_i.
Step 1. Slice frame f_i, and size initialization (320,240,3).
Step 2. Construct the shape feature S_h of f_i:
Step 2.1. Calculate the coordinates (K_x,K_y) of all SIFT key points in frame f_i.
Step 2.2. Divide the frame f_i into 40*30 size slices and count the number of SIFT key points distributed in the 64 slices.
Step 2.3. Expand the SIFT key points distribution to 64 dimensions to form the intra-frame shape feature S_h.
Step 3. Construct the color feature C_h of f_i:
Step 3.1. The color histogram features C_h on the gray color space are calculated by Equation $C_h = \frac{1}{m \times n} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \delta(f_{ij})$, reflecting the statistical distribution of colors and the underlying hue, where f_{ij} is the pixel value of pixel point (i, j), $\delta(.)$ is the accumulation of the number of points of the same pixel value, and $m \times n$ is the size of the video frame f_i.
Step 3.2. The greyscale histogram is divided into 64 partitions according to luminance, the number of pixels in each partition is counted, and finally this luminance distribution is expanded to 64 dimensions to form the intra-frame color feature.
Step 4. Texture features T_m for the construction of f_i:
Step 4.1. The gray level co-occurrence matrix GLCM is calculated from Equation $P(i, j|d, \theta) = \{(x, y)|f(x, y) = i, f(x + dx, y + dy) = j\}$, where the formula $P(.)$ denotes the probability of another pixel with grayness j at a distance d and direction θ from a pixel with grayness i.
Step 4.2. Set the angle as 0°, 45°, 90°, and 135° respectively, set the pixel point with statistical distance as 1, and calculate to get four GLCMs of 4*4 size.
Step 4.3. Expanding each of the above four GLCMs ultimately results in 64-dimensional intra-frame texture features T_m.
Step 5. The shape feature S_h, color feature C_h and texture feature T_m are spliced together to form the intra-frame image features F_i.

of deviation of the current frame from the first frame. By analyzing this distance in clusters, those with the same degree of deviation are grouped in the same cluster, then it is clear to find out which clusters have a greater degree of deviation from the first frame. Since the first frame of a video is often not rich in information, the most informative category of frames is found. If one chooses to calculate the distance between the current frame and the previous frame can reflect the magnitude of change in video content. By comparing this distance with a threshold, frames above the threshold can be considered as having a strong change in the video content. The most important type of frames can also be found since the parts of the video that have changed intensely often contain key information. The extraction of inter-frame distance features is analyzed below in terms of mathematical expressions.

Inspired by the approach that the total set of video frames can be represented linearly by a subset of keyframes [43], the process of obtaining keyframes using the keyframe-extraction algorithm can be considered as the transformation of the video collection to its subset with less loss of video information.

The role of inter-frame distance features in the keyframe extraction problem is found by analyzing the mathematical expressions before and after the transformation of the video to a subset of keyframes. We first represent all the information of the video as a matrix B

$$\begin{cases} B = \{x_1, x_2, \ldots, x_N\} \in R^{d \times N} \\ x_i = \{ip_i, t_i\} \in r^{d \times 1} \\ d = 3 \times h \times w + 1 \end{cases}, \tag{2}$$

where each column vector x_i comprises image pixel information ip_i, and time information t_i represents a frame vector. The task of keyframe extraction is to find an optimum subset $\overline{B}$

$$\overline{B} = \{x_1, x_2, \ldots, x_n\} \in R^{d \times n}, \tag{3}$$

where the relationship between B and $\overline{B}$ is the relationship between the video and its keyframes. We perform an equivalent substitution for the column vector x_i of matrix B which means that all the information of the current frame is represented as the sum of all the information of a given frame and the gap between the two frames.

$$\begin{cases} x_i = x_j + dx_{i,j} \\ \Delta ip,\ \Delta t = dx_{i,j} \end{cases}. \tag{4}$$

We consider setting Δt in Eq. (4) as a constant value of 1, which means that the column vector x_i is obtained from the previous frame x_{i-1} computation, when the original video matrix B is replaced with matrix B_1

$$B_1 = \left\{x_1, x_1 + dx_{2,1}, \ldots, x_{N-1} + dx_{N,N-1}\right\}. \tag{5}$$

By the same token, we also consider x_j in Eq. (4) as a constant value x_1, which means that the column vector x_i is obtained from the first frame x_1 calculation when the original video matrix B is replaced with matrix B_2

$$B_2 = \left\{x_1, x_1 + dx_{2,1}, \ldots, x_N + dx_{N,1}\right\}. \tag{6}$$

Comparison of matrix B and B_1 in Eq. (5), B_2 in Eq. (6) reveals that after representing frame information in terms of inter-frame distances the overall information of the video can still be retained, while the information needed to filter the keyframes can be drastically reduced.

In summary, to distinguish keyframes from general frames as much as possible and to fully express the distribution of frames in the video, we have selected three different inter-frame distance calculation methods to achieve the extraction of inter-frame distance features and proposed an algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods (IFCMD).

First, IFCMD detects the SIFT key points of the video frames and matches the SIFT descriptor of the current frame with the previous frame and the first frame, respectively, and the number of SIFT key points matches obtained is used as the distance feature between the two frames. Second, IFCMD converts the video frames to grey-scale images and obtains the inter-frame distance feature by calculating the Euclidean distance between the 2D matrix of the current frame and the previous frame also the first frame, respectively. Finally, the video frames are divided into 80*60 size slices, and the grey scale histograms are counted separately on each slice. The difference of the grey-scale distribution between the current frame and each slice of the previous frame and the first frame, respectively, is taken as the slice distance gap, then the distance between two frames is characterized as a combination of 16-dimensional differences. The pseudo-code of IFCMD is shown in Algorithm 2.

Algorithm 2 Algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods

Input. Video v_i of the video dataset D_v
Output. Set of distance features inter-frame F_d
Step 1. Slice frame f_i, and size initialization (320,240,3).
Step 2. Calculate the Euclidean distance between two frames D_e.
Step 2.1. The current frame f_i, the previous frame f_{i-1} and the first frame f_1 are converted to a grey scale image which is a 2D matrix (320,240).
Step 2.2. According to Equation $D_e(X,Y) = \sqrt{\sum_{i=1}^{n\times m}(x_i - y_i)^2}$, where X, Y are the pixel matrices of the two frames, x_i, y_i are the values of each pixel point in matrix X,Y, and $m \times n$ is the size of the video frame f_i, the Euclidean distances E_d and E_o are calculated for f_i and f_{i-1}, f_i and f_{i-1}, respectively.
Step 3. Count the number of SIFT key points matches between two frames in response to the distance between the two frames D_s.
Step 3.1. Find the SIFT key points and descriptors for the current frame f_i, the previous frame f_{i-1} and the first frame f_1.
Step 3.2. The SIFT points with similar features between two frames are found and counted by k-nearest neighbor matching algorithm.
Step 3.3. The number of SIFT key points matching the current frame and the previous frame is noted as the distance between the two, as is the case for the first frame.
Step 4. Calculate the difference in the grey scale histogram of each slice between two frames, the combination of the differences constitutes the distance between the two frames D_c
Step 5. The three distances D_e, D_s and D_c are spliced into the inter-frame distance feature F_d.

In Step 3.2, the strategy for counting the number of matches for the SIFT descriptor is that when the distance between the optimal matches is less than 70% of the distance of the suboptimal matches, we can assume that the two SIFT key points of the optimal matches reach a match and can be counted.

In step 4, the partition of the grey scale histogram is set to 256, which means that the number of pixel points at each luminance is counted. Setting the partition size to 80*60 then the distance combination D_c has 16 components.

Recommendation model based on feature intercross and fusion

The intra-frame image feature acquisition algorithm and inter-frame distance feature acquisition algorithm for keyframe recommendation obtains a combination of features that includes both category features such as the number of SIFT key points as well as dense features such as Euclidean distances. To explore the implicit relationships between features and use these relationships to better achieve keyframe recommendation, we propose a keyframe recommendation model based on feature intersection and fusion mechanism, as shown in Fig. 2.

First, the model takes the structure of a general recommendation model as its skeleton (including the linear combination feature part, the low-order cross-feature part and the high-order feature extraction part). Second, the model incorporates structures that have been shown to work well in recent research results such as multi-head attention mechanisms, compressed interaction networks, etc. Then, we add learnable position markers in the embedding layer to locate the feature order, and improve the simple attention structure in the output layer to synthesize the outputs of different order features. Finally, the cross and fusion of features is successfully achieved.

(a) Feature input

The input types of the features are divided into two types: category and dense.

$$F_{\text{input}} = [s_1, s_2, \ldots, s_n, d_{n+1}, d_{n+2}, \ldots, d_m],$$

where each feature component denotes a feature field, for category features, s_i denotes a discrete form of feature, and for dense features, d_j denotes a scalar single value.

(b) Feature embedding

Different embedding methods are used for these two types of input feature types, in which the input features of dense type are first normalized and second both types are embedded, respectively.

$$\begin{cases} [e_1, e_2, \ldots, e_n] = V_i[s_1, s_2, \ldots, s_n] \\ [e_{n+1}, e_{n+2}, \ldots, e_m] = V_m N_m[d_{n+1}, d_{n+2}, \ldots, d_m] \end{cases}$$

$$[x_1, x_2, \ldots, x_m] = [e_1, e_2, \ldots, e_m] + [p_1, p_2, \ldots, p_m],$$

where V_i denotes the embedding vector that encodes the input features s_i in one-hot coding. For dense features first normalize N_m them second embed them with feature vector V_m. p_i is the positional flag of the features, which we use to indicate the relative order of the features.

(c) Feature linear

The first-order part of the features, i.e., the linear combination of features, is an important part of the model output layer. We use a linear neural network to extract the first-order features.

$$y_l = w_0 + \sum_{i=1}^{n} w_i x_i, \tag{7}$$

where w_i is the parameter to be learned, and x_i is the input feature. The linear regression model assumes that the features

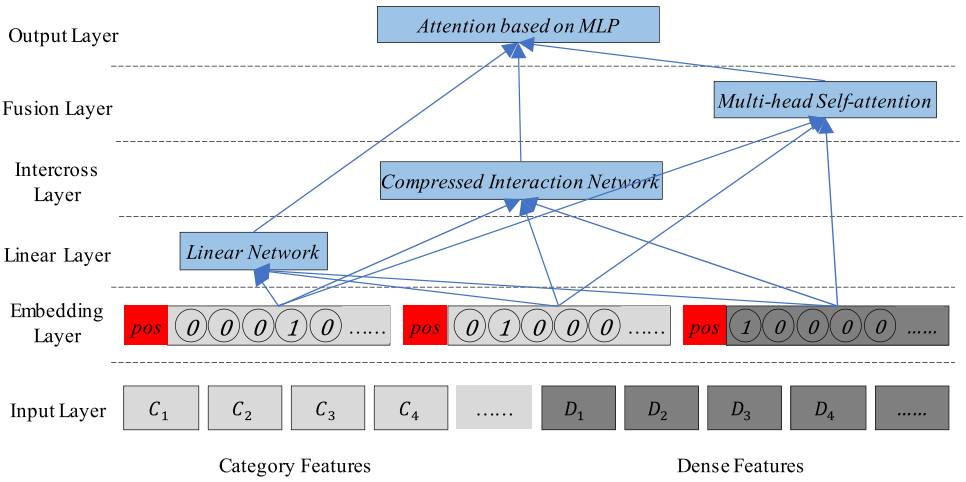

Fig. 2 Feature intercross and fusion based on Deep Structured Semantic Model

are independent of each other; however, there is an implied correlation between the features, so that higher order features can be formed by intersection between the features.

(d) Feature intercross

To model the dependencies between the features and characterize the second-order features of the input data, the factorization machine model is a usual choice.

$$y_c = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j>=i}^{n} w_{ij} x_i x_j,$$

where the factorization machine model [44, 45] constructs the second-order feature weights of features x_i and x_j by w_{ij}. The factorization machine has a simple structure, but the two features have different weights for the recommendation result. Directly calculating the inner product is not an excellent solution, and in contrast the multi-head attention mechanism is more effective for extracting low-order cross-features.

The process of obtaining second-order cross-features through the mechanism of multi-head attention can be expressed as follows.

$$\varphi^h(x_i, x_j) =< W_q^h x_i, W_k^h x_j >$$

$$\alpha_{i,j}^h = \frac{\exp(\varphi^h(x_i, x_j))}{\sum_{l=1}^{m} \exp(\varphi^h(x_i, x_l))}$$

$$\widetilde{x}_i^h = \sum_{l=1}^{m} \alpha_{i,l}^h W_v^h x_l,$$

$$\widetilde{x}_i = \widetilde{x}_i^1 + \widetilde{x}_i^2 + \ldots + \widetilde{x}_i^H$$

$$y_c^{(2)} = ReLU(\widetilde{x}_i + W_i x_i)$$

$$y_c = \left[y_c^{(2)}, y_c^{(3)} \right], \tag{8}$$

where $\varphi^h(e_i, e_j)$ is the attention function, and in this paper, we use the feature inner product to define the similarity between feature i and feature j. Since $\widetilde{e}_i^h$ is a combination of feature i and its associated features, it represents the learning of a new combined feature by self-attention. For the ith embedding, splice its output at H Attention head and then use the residual $e_i^{(2)}$ as its second-order cross output. Meanwhile, in this paper we choose the splicing of second-order and third-order cross-features as the final output.

(e) Feature fusion

Feature fusion modules often use deep neural networks (DNNs) to extract higher order features. However, the compressed interaction network (CIN) module in xDeepFM [46] is superior in terms of performance, so we briefly review the CIN modules used here.

$$Z_k = hadamard(x_0, x_{k-1}),\ x_0 \in R^{m \times D},\ x_{k-1} \in R^{H_{k-1} \times D}$$

$$x_k = W^k Z_k,\ Z_k \in R^{H_{k-1} \times m \times D},\ x_k \in R^{H_k \times D}$$

$$y_f = [x_4, x_5, \ldots x_k], \tag{9}$$

where each layer of CIN crosses the input feature x_0 with the output result x_{k-1} of the previous layer to obtain a $H_{k-1} \times m \times D$ dimensional vector, which is then compressed. The crossover results are linearly transformed using the parameter W^k to obtain a feature representation of H_k fields.

(f) Feature output

For aggregated first-order and multi-order features, we weight the features using a simple attention mechanism [47].

$$F_{\text{output}} = [y_l, y_c, y_f]$$

Table 1 Dataset overview

Dataset	Action category	Number of videos	Video length
SumMe [34]	–	25	1–6 min
TVSum [35]	–	50	2–10 min
HMDB-51 [32]	51	6766	1–0.5 min
UCF-101 [33]	101	13,320	1–3 min

$$y_{\text{interaction}} = \sum_{i=1}^{n}\sum_{j\geq i}^{n} w_{ij}x_0^{(i)}x_0^{(j)},\ x_0^{(i)} \in F_{\text{output}},$$

$$\begin{cases} a'_{ij} = ReLU(W_l w_{ij} x_0^{(i)} x_0^{(j)} + b_l) \\ a_{ij} = \frac{\exp(a'_{ij})}{\sum_{i=1}^{n}\sum_{j\geq i}^{n}\exp(a'_{ij})} \end{cases}$$

$$\widetilde{e}_i = \widetilde{e}_i^1 + \widetilde{e}_i^2 + \ldots + \widetilde{e}_i^H$$

$$y_{\text{attention}} = a_{ij}\, y_{\text{interaction}}$$

$$y_{\text{output}} = \text{Sigmoid}(y_{\text{attention}}), \tag{10}$$

where Eq. (10) first splices the output of Eqs. (7)–(9), second calculates the attention scores using MLP network and finally gets the recommendation results based on the attention parameters using Sigmoid function.

Experiments

Datasets

To fully evaluate the effectiveness of the proposed method, two video datasets commonly used for the keyframe extraction problem (SumMe dataset and TVSum dataset) and two larger action video datasets (HMDB-51 dataset and UCF-101 dataset) are selected to organize the experiments in this paper (Table 1).

TVSum is a dataset that validates video summarization techniques. The dataset consists of 50 videos of different genres (e.g., news, how-to, documentary, vlog, selfie) and 1000 crowdsourced annotations (20 per video) that rate the high importance of the shot, while SumMe consists of 25 videos with 15 human annotations per video. HMDB-51 and UCF-101 are commonly used datasets in the field of action recognition.

The videos in the HMDB-51 and UCF-101 datasets are generally shot from a fixed space, accompanied by an unhurried camera movement, with the aim of demonstrating the complete process of an action as much as possible. In this work, we choose the HMDB-51 and UCF-101 datasets to validate the ability of our method for capturing key actions in videos. Compared with the HMDB-51 dataset, the UCF-101 dataset has a highly similar organizational structure, which can be regarded as an extension of the former to verify the robustness of our method.

Table 2 Running environment

Operating system	Ubuntu 20.04
Hardware	GPU: Quadro GV100 and RTX 2060
Memory	32 GB
Software	Python3.8 Tensorflow-gpu2.10

Experimental settings

(a) Running environment: The hardware environment for conducting experiments in this paper is shown in Table 2

(b) Training set: To ensure the consistency of data distribution in the training and validation sets, the videos of each action category in the HMDB-51 and UCF-101 datasets are randomly divided according to the ratio of 8:2. The videos are then transformed into a collection of video frames by the frame-slicing algorithm to finally form the training and validation sets. Meanwhile, the videos in the TVSum and SumMe datasets are also partitioned into training and validation sets in the same ratio and subjected to frame slicing. It is worth noting that for the HMDB-51 and UCF-101 datasets there are enough positive samples so it is not necessary to enter all the negative samples into the training, so the number of positive and negative samples entered into the model is guaranteed to be 1:2 (see "Experimental results and analysis" section for details).

(c) Ground-truth construction: Unlike other research areas, the keyframe extraction task is somewhat subjective. The judgement of the importance of frames relies on subjective human evaluation, but the evaluation of the methodology has to exclude personal factors as much as possible, thus requiring large-scale data annotation. For the SumMe and TVSum datasets, the dataset builders have created human annotations for them, and our experiments will make use of these human annotations. Each experiment will calculate the performance of our method based on the annotations of different people separately, and then take the average performance of all annotators as the final result. For the HMDB-51 and UCF-101 datasets, we have added three new manual annotations for each of their videos.

(d) Matching rules: In this research, the strictest matching strategy is used, i.e., ground-truth construction evaluates the video frame as critical when and only when the recommendation algorithm evaluates it as critical.

Ground-truth construction evaluates it as noncritical when the recommendation algorithm evaluates it as noncritical. Only the above two cases are matched [48].

(e) Compression ratio and threshold: Since the final output of the model is an importance score for a frame between 0 and 1, whether each frame can be considered as a keyframe depends on two values, a threshold α and a compression ratio CR. In terms of the threshold α, a video frame is recognized as a keyframe if the importance score is greater than or equal to α. In this paper, α defaults to 0.5.

(f) Objective metrics: Because of the non-equilibrium nature of the positive and negative sample sizes for the keyframe screening task, we use only the $F1$ score to objectively assess the quality of the keyframe recommendations, for which we need to calculate the precision P and the recall R first.

$$P = \frac{Num_{mk}}{Num_{rk}},$$

where the overall keyframe-recommendation precision P is the ratio of the number of matches between the recommended keyframes and ground-truth construction (Num_{mk}) to the number of all recommended keyframes (Num_{rk}).

$$R = \frac{Num_{mk}}{Num_{gtk}},$$

where the recall rate R is the ratio of Num_{mk} to the number of keyframes in ground-truth construction (Num_{gtk}).

$$F = \frac{2 * P * R}{P + R},$$

where F score take into account the results of the model's accuracy and completeness calculations, favoring the indicator with the smaller value. In this paper, we calculate the average of all the labeler's F score and the best one among them separately.

Ablation study

To observe the performance of each part of the proposed keyframe recommendation method, we conducted experiments on the SumMe dataset for structural ablation of the model and for feature ablation of the two feature construction methods. Taking the best and average of 15 annotation results, the statistical results of F score are shown in Tables 3 and 4.

Table 3 shows that deleting one of the feature-intercross, feature-fusion, and feature-output based on the attention mechanism results in a lower objective evaluation criterion F score compared to the full model. This phenomenon validates the effectiveness of the model components. In contrast, feature-intercross, feature-fusion, and feature-output based on the attention mechanism abatement resulted in a decrease of 4.5, 5, 2.7 and 2.9, 5.2, 3.4 for the best F score and average F score, respectively. This phenomenon indicates that: (1) The ablation of the three structures resulted in a minimum decrease in model performance of 4.9% ((3) in the table) and a maximum decrease of 9.7% ((2) in the table), which can be regarded as a significant decrease, highlighting the importance of the model's individual structures. (2) Higher order features obtained from the feature fusion structure were more useful than the other two structures in recommending keyframes, showing that the higher order features are the most important keyframe recommendation factor. (3) The best F score has a greater decrease than the average F score indicating that each structure of the model improves the generalization ability and robustness of the model.

Table 4 verifies the validity of the six features in Sections "Algorithm for extracting intra-frame image features based on the combination of multiple image descriptors" and "Algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods" in the feature ablation experiments. The features that lead to more decrease in average F score include 1.4 (1), 3.2 (2), 2.5 (3), 8.8 (4), 7.5 (5), and 3.2 (6). The above six features belong to the color feature, texture feature, shape feature of the intra-frame image and the SIFT distance, histogram distance and Euclidean distance of the inter-frame distance, respectively, which illustrates that all the various features we use can have a significant impact on the acquisition of keyframes. Throughout the table, the distance features are generally more important than the image features, especially the SIFT distance feature and the Euclidean distance feature both of which resulted in the best F score and the average F score decreasing by 10.8%, 16.4% and 10.4%, 13.9%, respectively. This decrease indicates that the inter-frame distance plays a crucial role in the discrimination of keyframes.

Comparison test and analysis

Baseline

We compare the proposed method with several state-of-the-art methods, including vision-based methods and recommendation-based methods. Vision-based methods: a flexible summary detection network method [49], an attention-based encoding and decoding method [50], and a combined global and local attention method [51].

Table 3 Structure ablation study

Model part	*F* score	
	Best	Average
(1) Remove feature-intercross in Section "Recommendation model based on feature intercross and fusion" (d)	50.1	49.7
(2) Remove feature-fusion in Section "Recommendation model based on feature intercross and fusion" (e)	49.6	48.4
(3) Replacing feature-attention with feature-summation in Section "Recommendation model based on feature intercross and fusion" (f)	51.9	50.2
(4) Full model	54.6	53.6

Table 4 Feature ablation study

Removed features	*F* score	
	Best	Average
(1) Remove Shape feature in Section "Algorithm for extracting intra-frame image features based on the combination of multiple image descriptors"	52.3	52.2
(2) Remove Color feature in Section "Algorithm for extracting intra-frame image features based on the combination of multiple image descriptors"	51.2	50.4
(3) Remove Texture feature in Section "Algorithm for extracting intra-frame image features based on the combination of multiple image descriptors"	52.2	51.1
(4) Remove SIFT distance in Section "Algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods"	48.7	44.8
(5) Remove Euclidean distance in Section "Algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods"	48.9	46.1
(6) Remove Histogram distance in Section "Algorithm for extracting inter-frame distance features based on the combination of multiple distance calculation methods"	51.8	50.4
(7) No ablation	54.6	53.6

Recommendation-based methods: xDeepFM model [46], AutoInt model [52] are two deep learning recommendation model methods constructed based on compressed interaction network and based on multi-head attention mechanism, respectively.

Experimental results and analysis

First, we compare the proposed method with the above vision-based baseline algorithm on SumMe dataset, TVSum dataset. The statistics of the average *F*-scores after running our method 15 and 20 times based on all the labelled results, respectively, are shown in Table 5.

Table 5 shows that our proposed keyframe recommendation method based on feature crossover and fusion ranks second in terms of evaluation metrics on the datasets SumMe and TVSum, respectively. For the SumMe dataset we outperformed M-AVS's 44.4 and DSNet's 50.2 with a mean *F*-score of 53.6, and underperformed PGL-SUM's 55.6. For the TVSum dataset we outperformed M-AVS's 61.0 and PGL-SUM's 60.1 with a mean *F*-score of 61.4 and underperformed DSNet's 62.1. The performance evaluation of the two datasets shows that our method achieves a high level of performance compared to the state-of-the-art baseline methods. Although our method did not reach the optimal performance, the novel framework used in our study is still of great value

Table 6 shows how our recommendation model compares to the two state-of-the-art baseline models. For the SumMe dataset our model outperforms xDeepFM (best 50.1 and average 48.3) and AutoInt (best 52.5 and average 47.6) with a best of 54.6 and an average of 53.6. For the TVSum dataset our performance is also optimal with a best of 63.2 and an average of 61.4.

For the HMDB-51 dataset and the UCF-101 dataset, which are characterized by a much larger data size than the other two datasets, we therefore reorganized the training samples. 6766 videos are in HMDB-51, where the ratio of positive and negative sample frames is 1:8 and the number of positive samples reaches 11,862 frames. Since the number of positive samples of keyframes is sufficient, we dropped some negative samples to make the ratio of the two 1:2. 13,320 videos in UCF-101, where the ratio of positive to negative samples is 1:7, and the number of positive sample frames

Table 5 Comparison of our proposed keyframe recommendation method with several visual methods

Method	SumMe		TVSum	
	F-score		*F*-score	
	Best	Average	Best	Average
M-AVS	–	44.4	–	61.0
DSNet	–	50.2	–	**62.1**
PGL-SUM	–	**55.6**	–	61.0
KFRFIF (Ours)	54.6	53.6	63.2	61.4

Table 6 Comparison of our proposed keyframe recommendation method with several recommendation methods

Method	SumMe		TVSum		HMDB-51		UCF-101	
	F-score		*F*-score		*F*-score		*F*-score	
	Best	Average	Best	Average	Best	Average	Best	Average
xDeepFM	50.1	48.3	61.9	60.8	68.2	64.2	67.3	65.5
AutoInt	52.5	47.6	62.0	61.1	65.1	59.2	69.2	62.1
KFRFIF (Ours)	54.6	53.6	63.2	61.4	68.2	67.9	75.2	72.0

reaches 212,568 frames, we similarly reconstructed the samples for the input model with a ratio of 1:2 for the positive and negative samples.

For the HMDB-51 dataset our model outperforms xDeepFM (best 68.2 and average 64.2) and AutoInt (best 65.1 and average 59.2) with a best of 68.2 and an average of 67.9. For the UCF-101 dataset our performance is also optimal with a best of 75.2 and an average of 72.0. As the video size increases, our model performs better on the HMDB-51 dataset and even the UCF-101 dataset, and we believe the most likely reasons are as follows.

- Our model has a clear hierarchical structure that extracts first-order, low-order, and high-order features, and subtle differences between video frames in a video clip are extracted when the training samples are increased, which can easily be ignored by other models with insufficient feature extraction capabilities.
- Our model uses the attention mechanism to filter the features of each order, while labelling the order of the features with sequential bit flags, allowing the model to increase in depth and be able to deal with larger sized datasets.
- We use an optimization approach to deal with the imbalance between positive and negative samples of the data; the larger the amount of data, the more negative samples are reduced and the less the model is disturbed.

To demonstrate the features of our proposed method more intuitively, we show the results of five different keyframe detections on the HMDB-51 dataset, using our method and the ground truth construction model, as well as other recommended models, respectively. As shown in Fig. 3, the demonstrated videos are randomly selected. While our method may not select exactly the same frames as the ground truth build, the keyframes we detect are very visually similar to the ground truth build and seem reasonable. The video shows human hand-clapping movements, and omitting any of the keyframes would result in a degradation of the quality of the keyframe extraction. For example, the keyframes generated by the xDeepFM model lose the third hand-clapping action, and therefore lose the footage of key information. In addition, there are more redundant shots in the keyframes generated by the other 2 recommended models compared to our method. Meanwhile, the video pacing becomes smoother as our method advances the first hand-clapping node compared to the ground truth construction.

In conclusion, the comparable performance of experimental results reflects the advantages of our approach, which can be inferred as follows.

- Frames are represented by feature engineering, and the representation learned by the recommendation algorithm effectively reflects the representativeness of the frames.
- Our manually constructed features are content rich. For keyframe recommendation, capturing the video content comprehensively is crucial.
- Due to the imbalance of positive and negative samples in keyframes, expanding the number of videos will increase the number of positive samples so that the negative samples can be appropriately reduced. This processing method can improve the performance of keyframe recommendation.

Fig. 3 Comparison of keyframe detection results of our method with Ground Truth Construction and other recommended models on the HMDB-51 dataset

Conclusion and discussion

With the increase in mobile devices, people are becoming more accustomed to shooting videos or obtaining information from videos. This phenomenon is directly causing a dramatic increase in the number of videos. If one can improve keyframe-extraction methods, video data can be more easily managed, retrieved, and previewed. Thus, our proposed keyframe recommendation algorithm can: recommend possible keyframes from a large number of videos. For this purpose, this paper implements the application of a recommendation framework to the keyframe extraction problem, resulting in the KFRFIF method. In our approach, model input samples are obtained by mining image features of frames and inter-frame distance features. The importance score obtained by learning these two features through the recommendation model can effectively reflect the representativeness of the frames and fully describe the video content. We further create ground truth for each video to provide an objective evaluation of keyframe recommendation. The experimental results show that the proposed KFRFIF outperforms the comparative baseline in terms of keyframe extraction accuracy on the HMDB-51 and UCF-101 datasets, and achieves similar performance to the state-of-the-art methods on the SumMe and TVSum datasets. Inevitably, some interesting works need to be discussed.

- Recommendation algorithms depend heavily on the acquisition of features. The six features we use have a total of 234 dimensions, and it is worth future research to discern the validity of these dimensions.
- Experimentally, KFRFIF can effectively recommend keyframes in continuous action class videos. Combining KFRFIF with some action recognition tasks [53, 54] can eliminate the need for manual pre-slicing of video data, resulting in a more natural and coherent approach to these tasks. Some improvements to the constructed features may be needed for shot switching class videos to work better in the future.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under Grant Nos. 62373116 and 62163007, and the Guizhou Provincial Science and Technology Projects under grant QKHZC [2023]118, PTRC[2020]6007-2, and [2021]439.

Data availability The data that support the findings of this study are available on request from the corresponding author.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Wang W, Shen J, Li X, Porikli F (2015) Robust video object cosegmentation. IEEE Trans Image Process 24:3137–3148. https://doi.org/10.1109/TIP.2015.2438550
2. Venugopala PS, Nayak AA, Sarojadevi H, Chiplunkar NN (2015) Various challenges in video watermarking for android mobile devices. In: 2015 IEEE int. conf. inf. process. ICIP, pp 248–253
3. Lu X, Zheng X, Li X (2017) Latent semantic minimal hashing for image retrieval. IEEE Trans Image Process 26:355–368. https://doi.org/10.1109/TIP.2016.2627801
4. Castro A, Villagra VA, Garcia P, Rivera D, Toledo D (2021) An ontological-based model to data governance for big data. IEEE Access 9:109943–109959. https://doi.org/10.1109/ACCESS.2021.3101938
5. Jiang Y-G, Wang J, Wang Q, Liu W, Ngo C-W (2016) Hierarchical visualization of video search results for topic-based browsing. IEEE Trans Multimed 18:2161–2170. https://doi.org/10.1109/TMM.2016.2614233
6. Sidiropoulos P, Mezaris V, Kompatsiaris I (2014) Video tomographs and a base detector selection strategy for improving large-scale video concept detection. IEEE Trans Circuits Syst Video Technol 24:1251–1264. https://doi.org/10.1109/TCSVT.2014.2302554
7. Tu Z, Li H, Zhang D, Dauwels J, Li B, Yuan J (2019) Action-stage emphasized spatiotemporal VLAD for video action recognition. IEEE Trans Image Process 28:2799–2812. https://doi.org/10.1109/TIP.2018.2890749
8. Roselinkiruba R, Saranya Jothi C, Tamil Thendral M, Hemalatha R (2023) Secure video steganography using key frame and region selection technique. Int J Inf Technol 15:1299–1308. https://doi.org/10.1007/s41870-023-01180-3
9. Mounika Bommisetty R, Khare A, Siddiqui TJ, Palanisamy P (2021) Fusion of gradient and feature similarity for keyframe extraction. Multimed Tools Appl 80:15429–15467. https://doi.org/10.1007/s11042-020-10390-x
10. Thakre KS, Rajurkar AM, Manthalkar RR (2016) Video partitioning and secured keyframe extraction of MPEG video. In: 1ST Int. conf. inf. secur. priv., vol 78, pp 790–798. https://doi.org/10.1016/j.procs.2016.02.058
11. Bommisetty RM, Prakash O, Khare A (2020) Keyframe extraction using Pearson correlation coefficient and color moments. Multimed Syst 26:267–299. https://doi.org/10.1007/s00530-019-00642-8
12. Sun B, Kong D, Wang S, Li J (2018) Keyframe extraction for human motion capture data based on affinity propagation. In: 2018 IEEE 9th annu. inf. technol. electron. mob. commun. conf. IEMCON, pp 107–112
13. Ioannidis A, Chasanis V, Likas A (2016) Weighted multi-view keyframe extraction. Pattern Recognit Lett 72:52–61. https://doi.org/10.1016/j.patrec.2016.01.027
14. Mei S, Guan G, Wang Z, Wan S, He M, Feng DD (2015) Video summarization via minimum sparse reconstruction. Pattern Recognit 48:522–533. https://doi.org/10.1016/j.patcog.2014.08.002
15. Xia G, Sun H, Niu X, Zhang G, Feng L (2017) Keyframe extraction for human motion capture data based on joint kernel sparse representation. IEEE Trans Ind Electron 64:1589–1599. https://doi.org/10.1109/TIE.2016.2610946
16. Liu Y, Chen L, Lin Z (2022) Keyframe extraction for motion capture data via pose saliency and reconstruction error. Vis Comput 39(1):4943–4953. https://doi.org/10.1007/s00371-022-02639-3
17. Kiziltepe RS, Gan JQ, Escobar JJ (2021) A novel keyframe extraction method for video classification using deep neural networks. Neural Comput. Appl 35(34):24513–24524. https://doi.org/10.1007/s00521-021-06322-x
18. .Mahasseni B, Lam M, Todorovic S (2017) Unsupervised video summarization with adversarial lstm networks. In: Proc. IEEE conf. comput. vis. pattern recognit., pp 202–211
19. Kar A, Rai N, Sikka K, Sharma G (2017) Adascan: adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. In: Proc. IEEE conf. comput. vis. pattern recognit., pp 3376–3385
20. Muhammad K, Hussain T, Tanveer M, Sannino G, de Albuquerque VHC (2020) Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks. IEEE Internet Things J 7:4455–4463. https://doi.org/10.1109/JIOT.2019.2950469
21. Priya GGL, Domnic S (2014) Shot based keyframe extraction for ecological video indexing and retrieval. Ecol Inf 23:107–117. https://doi.org/10.1016/j.ecoinf.2013.09.003
22. Omidyeganeh M, Ghaemmaghami S, Shirmohammadi S (2011) Video keyframe analysis using a segment-based statistical metric in a visually sensitive parametric space. IEEE Trans Image Process 20:2730–2737. https://doi.org/10.1109/TIP.2011.2143421
23. Xu C, Yu W, Li Y, Lu X, Wang M, Yang X (2021) Key frame extraction for human motion capture data via multiple binomial fitting. Comput Animat Virtual Worlds. https://doi.org/10.1002/cav.1976
24. Fei M, Jiang W, Mao W, Song Z (2016) New fusional framework combining sparse selection and clustering for key frame extraction. IET Comput Vis 10:280–288. https://doi.org/10.1049/iet-cvi.2015.0237
25. Zhou Y, Zhang X, Ding F (2021) Hierarchical estimation approach for RBF-AR models with regression weights based on the increasing data length. IEEE Trans Circuits Syst II Expr Briefs 68:3597–3601. https://doi.org/10.1109/TCSII.2021.3076112
26. Li J, Yao T, Ling Q, Mei T (2017) Detecting shot boundary with sparse coding for video summarization. Neurocomputing 266:66–78. https://doi.org/10.1016/j.neucom.2017.04.065
27. Ma M, Mei S, Wan S, Wang Z, Feng DD, Bennamoun M (2021) Similarity based block sparse subset selection for video summarization. IEEE Trans Circuits Syst Video Technol 31:3967–3980. https://doi.org/10.1109/TCSVT.2020.3044600
28. Li Y, Yang G, Su Z, Li S, Wang Y (2023) Human activity recognition based on multienvironment sensor data. Inf Fusion 91:47–63. https://doi.org/10.1016/j.inffus.2022.10.015
29. Zhang Q, Zhang S, Zhou D (2014) Keyframe extraction from human motion capture data based on a multiple population genetic algorithm. Symmetry-Basel 6:926–937. https://doi.org/10.3390/sym6040926

30. Yan X, Gilani SZ, Qin H, Feng M, Zhang L, Mian A (2018) Deep keyframe detection in human action videos. https://doi.org/10.48550/arXiv.1804.10021
31. Banerjee A, Kumar E, Ravinder M (2024) Particle swarm optimized deep spatio-temporal features for efficient video retrieval. Int J Inf Technol. https://doi.org/10.1007/s41870-024-01733-0
32. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: a large video database for human motion recognition. In: 2011 international conference on computer vision, pp 2556–2563. https://doi.org/10.1109/ICCV.2011.6126543
33. Soomro K, Zamir AR, Shah M (2012) UCF101: a dataset of 101 human actions classes from videos in the wild. https://doi.org/10.48550/arXiv.1212.0402
34. Gygli M, Grabner H, Riemenschneider H, Van Gool L (2014) Creating summaries from user videos. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Comput. Vis.—ECCV 214. Springer International Publishing, Cham, pp 505–520
35. Song Y, Vallmitjana J, Stent A, Jaimes A (2015) TVSum: summarizing web videos using titles. In: 2015 IEEE conf. comput. vis. pattern recognit. CVPR, pp 5179–5187. https://doi.org/10.1109/CVPR.2015.7299154
36. Pandian AA, Maheswari S (2024) A keyframe selection for summarization of informative activities using clustering in surveillance videos. Multimed Tools Appl 83:7021–7034. https://doi.org/10.1007/s11042-023-15859-z
37. Mo CA, Hu K, Long C, Wang Z (2023) Continuous intermediate token learning with implicit motion manifold for keyframe based motion interpolation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. CVPR, pp 13894–13903. https://doi.org/10.48550/arXiv.2303.14926
38. Li X, Zhao B, Lu X (2018) Key frame extraction in the summary space. IEEE Trans Cybern 48:1923–1934. https://doi.org/10.1109/TCYB.2017.2718579
39. Kuncheva LI, Yousefi P, Almeida J (2017) Comparing keyframe summaries of egocentric videos: closest-to-centroid baseline. In: Proc. 2017 seventh int. conf. image process. theory tools appl. IPTA 2017
40. Kamal K, Qayyum R, Mathavan S, Zafar T (2017) Wood defects classification using laws texture energy measures and supervised learning approach. Adv Eng Inf 34:125–135. https://doi.org/10.1016/j.aei.2017.09.007
41. Hannane R, Elboushaki A, Afdel K (2016) Efficient video summarization based on motion SIFT-distribution histogram. In: 2016 13TH int. conf. comput. graph. imaging vis. CGIV., pp 312–317. https://doi.org/10.1109/CGiV.2016.67
42. Tafannum F, Shopnil MNS, Salsabil A, Ahmed N, Alam MGR, Reza MT (2021) Demystifying black-box learning models of rumor detection from social media posts. In: 2021 IEEE 12th annu. ubiquitous comput. electron. mob. commun. conf. UEMCON, pp 358–364. https://doi.org/10.1109/UEMCON53757.2021.9666567
43. Chen C, Li D, Yan J, Yang X (2022) Modeling dynamic user preference via dictionary learning for sequential recommendation. IEEE Trans Knowl Data Eng 34:5446–5458. https://doi.org/10.1109/TKDE.2021.3050407
44. Mao X, Mitra S, Swaminathan V (2017) Feature selection for FM-based context-aware recommendation systems. In: 2017 IEEE int. symp. multimed. ISM., pp 252–255. https://doi.org/10.1109/ISM.2017.42
45. Wen N, Zhang F (2020) Extended factorization machines for sequential recommendation. IEEE Access 8:41342–41350. https://doi.org/10.1109/ACCESS.2020.2977231
46. Lian J, Zhou X, Zhang F, Chen Z, Xie X, Sun G (2018) xDeepFM: combining explicit and implicit feature interactions for recommender systems. In: Proc. 24th ACM SIGKDD int. conf. knowl. discov. data min., pp 1754–1763. https://doi.org/10.1145/3219819.3220023
47. Wang Y, Yang G, Li S, Li Y, He L, Liu D (2023) Arrhythmia classification algorithm based on multi-head self-attention mechanism. Biomed Signal Process Control 79:104206. https://doi.org/10.1016/j.bspc.2022.104206
48. Nimmagadda P, Sudhakar K, Rajasekar P (2023) Perceptual video summarization using keyframes extraction technique. In: 2023 3rd international conference on innovative practices in technology and management (ICIPTM). IEEE, pp 1–4. https://doi.org/10.1109/ICIPTM57143.2023.10118236
49. Zhu W, Lu J, Li J, Zhou J (2021) DSNet: a flexible detect-to-summarize network for video summarization. IEEE Trans Image Process 30:948–962. https://doi.org/10.1109/TIP.2020.3039886
50. Ji Z, Xiong K, Pang Y, Li X (2018) Video summarization with attention-based encoder–decoder networks. http://arxiv.org/abs/1708.09545. Accessed December 18, 2023
51. Apostolidis E, Balaouras G, Mezaris V, Patras I (2021) Combining global and local attention with positional encoding for video summarization. In: 2021 IEEE int. symp. multimed. ISM, IEEE, Naple, Italy, pp 226–234. https://doi.org/10.1109/ISM52913.2021.00045
52. Song W, Shi C, Xiao Z, Duan Z, Xu Y, Zhang M, Tang J (2019) AutoInt: automatic feature interaction learning via self-attentive neural networks. In: Proc. 28th ACM int. conf. inf. knowl. manag., pp 1161–1170. https://doi.org/10.1145/3357384.3357925
53. Li L, Yang G, Li Y, Zhu D, He L (2023) Abnormal sitting posture recognition based on multi-scale spatiotemporal features of skeleton graph. Eng Appl Artif Intell 123:106374. https://doi.org/10.1016/j.engappai.2023.106374
54. Yang G, Yang S, Luo K, Lan S, He L, Li Y (2023) Detection of non-suicidal self-injury based on spatiotemporal features of indoor activities. IET Biom 12(2):91–101. https://doi.org/10.1049/bme2.1211012