2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Key frame extraction based on sparse coding with deep frame features

Yujie Li[1], Ning Liu[1], Depeng Chen[1], Shuxue Ding[1], and Benying Tan[1*]
[1] School of Artificial Intelligence, Guilin University of Electronic Technology, China
*Corresponding author, email:by-tan@guet.edu.cn

Abstract. Key frame extraction based on sparse coding can present the entire video with a small number of key frames, reducing the redundancy of the video. However, existing sparse coding-based methods use raw video frame features, which leads to high computational complexity and significant time consumption. In this paper, we propose a novel key frame extraction method based on sparse coding and deep frame features (KSC-DFF) to address these challenges. First, we obtain deep frame features using a deep neural network, which can reduce the dimensionality of the input video data and generate deep frame features such as the main object features of the frame. To automatically extract deep frame features, a YOLO-based deep neural network called YOLO-MLP was designed for video feature extraction. Then, we used sparse coding to extract key frames based on deep frame features, which can reduce information redundancy and computation time while maintaining high accuracy. Experimental results on SumMe demonstrate that the proposed KSC-DFF outperforms the existing methods with an increase of 49.4% and a time reduction of nearly 98% compared to the conventional sparse coding-based method SMRS.

Keywords: Key frame extraction, Sparse coding, Deep learning, Feature extraction, YOLO-MLP.

1 Introduction

Due to improved network designs and more storage capacity, video has been used increasingly in numerous applications in recent years, particularly in cell phones. Over 500 hours of video are posted on the Internet every minute [19]. The increasing demand for video material in the coming decades is predicted to sustain this significant expansion in the number of films on the Internet [7, 8, 29]. With the rapid development of video data in many fields, efficient key frame extraction is urgently required for video indexing, browsing, and retrieval [31, 30].

Inspired by sparse representation, sparse coding, which retains essential content from the original video while selecting only a limited number of frames, is a common technique used for key frame extraction. Specifically, sparse codingbased methods for key frame extraction excel in preserving crucial information while minimizing the number of selected frames [27, 12]. The pixel information of the original video

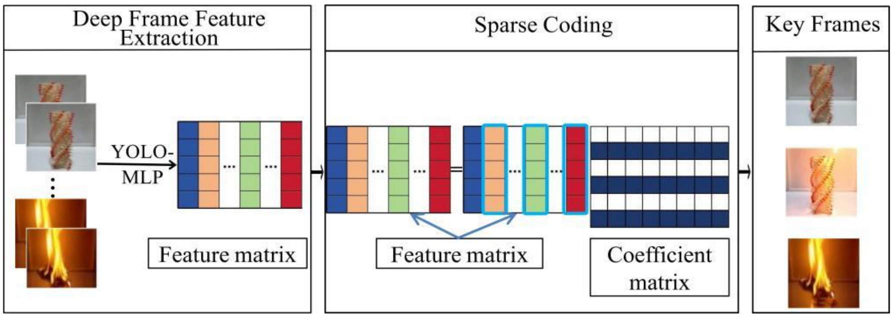

Fig. 1. Overview of the proposed KSC-DFF. Deep Frame Feature Extraction: YOLO-MLP converts the source video frame into a feature matrix;Sparse Coding: A Key frame index is identified by the non-zero rows of the sparse coefficient matrix; Key Frames: the key frames are obtained by the index of the key frame.

frames is used as input in existing sparse coding-based techniques, which have high computational complexity and time cost. Tan et al. used convolutional neural networks to compress video frames [22]. However, the compressed dimensions of the frame features were fixed at 1000. We plan to explore the effects when videos are compressed to various sizes.

In recent years, deep learning-based approaches have also been widely applied to key frame extraction, leveraging convolutional neural networks (CNNs)[17,26] to learn high-level semantic representations from raw video frames. These methods can capture complex spatio-temporal dependencies and have shown promising performance in identifying informative frames. However, such deep learning models typically require large-scale labeled datasets for training, which are expensive and time-consuming to annotate. Moreover, the performance of these models heavily relies on the diversity and quality of the training data, and they may not generalize well across different video domains without additional fine-tuning. These limitations hinder their practicality in scenarios where annotated data is scarce or computational resources are limited.

This paper proposes a novel framework called KSC-DFF (as shown in Figure 1). We use a sparse representation of the video content through sparse coding using these deep frame features. We aim to extract key frames that minimize redundancy and efficiently summarize video material using a sparse coding technique. We compared the integration of our proposed method with some current techniques based on the sparse representation algorithm SMRS [3]. KSC-DFF can first extract pertinent visual data from video frames. Furthermore, intricate and deep information can be collected to produce rich video content representations. Our method leverages these properties to enhance the effectiveness and efficiency of key frame extraction in video summarizing tasks by including them in sparse coding. The main contributions of this study are as follows:

1. We propose a novel key frame extraction method, KSC-DFF, which replaces the raw video frame with the deep frame feature obtained by YOLO-MLP. KSC-DFF can reduce the computation time while maintaining high accuracy compared with traditional sparse coding-based methods.

2. To produce a more thorough and rich representation of video features, we propose a complete deep frame feature extraction method (YOLO-MLP) with multi-scale fusion and deep feature extraction.

3. The proposed approach achieved competitive performance compared with the state-of-the-art key frame extraction methods on the SumMe dataset.

2 Related works

Conventional methods are either shotor segment-based, meaning that the input video is split into small shots or segments using change or segment detection algorithms before extracting key frames. The location of the shot switch is typically the main focus of shot-based key frame extraction techniques, which identify the frame at the switch point as the key frame. By comparing each frame in the shot to the reference frame, these algorithms first identify the reference frame and then identify and select the key frames [9, 21]. Significant visual or content elements and the locations of content or theme changes in a video are usually the focus of clip-based key frame extraction techniques. Key segments are chosen from the vicinity of each key frame to leverage dynamic patterns over short time intervals, thus enhancing discriminative power [6]. For example, uniform sampling (Uni.) often serves as a baseline for assessing key frame extraction techniques. Li et al. [11] proposed a method to identify and segment foreground items in a movie to extract key frames that are similar or of higher quality. A method for reference-based key frame extraction and clustering [16]. The interframe difference technique, which is frequently applied in conventional methods, depends on pixel-level fluctuations, which may not accurately represent the significance of the video material. It is also prone to noise and interference, which leads to less precise key frame extraction. Conventional techniques can also introduce noise and interference into a video, making the recovered key frames less steady and trustworthy.

Moreover, key frame extraction can be performed using deep neural networks [17, 26, 13]. To choose the key frames, Deep Semantic Feature Video Summarization (DFS) [17] uses the deep characteristics included in video frames. In contrast, deep semantic features from the Visual Geometry Group (VGG) are used in VGG-based video summarization [20] to extract key frames based on the extraction of SIFT features and optical flow [25, 4, 10]. An early study [25, 10] described an optical flow video and used the similarity between consecutive frames to detect local minimum changes. Subsequent studies have enhanced this pipeline by employing key frames detection for feature extraction [10]. Qu et al. used a knowledge graph to obtain key semantic information of video frames for extracting key semantic frames [18]. This method gathers key points to obtain key frames from a video, and it uses SIFT descriptors to extract local features. The disadvantage of these techniques is that if the

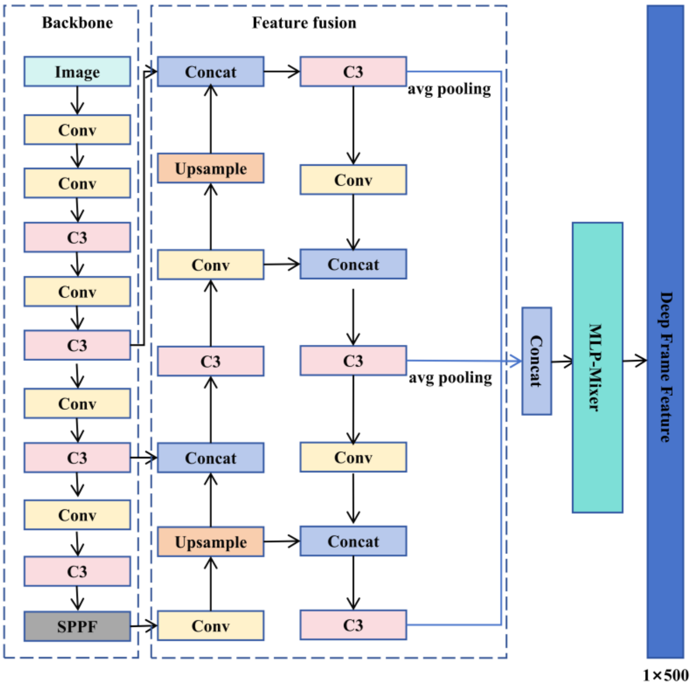

Fig. 2. Overview of the proposed YOLO-MLP. The final deep frame features are produced by feeding the extracted feature maps into the MLP-Mixer.

video contains the same content, they may extract key frames that are similar to one another. Ensuring proper key frames and adequate compression for video processing remains challenging.

Recently, numerous key frame extraction methods have been devised based on sparse coding [3, 32, 22, 28, 14, 12]. Among these, the SMRS algorithm [3] selects essential frames using a reconstruction problem formulation with paradigms as sparse constraints, demonstrating efficacy in video categorization and summarization. Other techniques, such as the determinantal metrics-based SC-det algorithm [22], first use VGG16 to extract features from video frames and then use a microscopic sparse constraint to extract video frames. In place of the conventional paradigm, a recent study presented DSSC-log [12], which employs a nonconvex group log regularizer and creates a workable decomposition technique to learn the structured sparse coefficient matrix. To enhance the sparse method, we employed deep frame features for sparse coding to achieve better key frame extraction performance.

3 Proposed Model

We extract deep frame features to obtain rich and deep video information and then use a sparse coding technique to choose key frames, drawing inspiration from [20, 22]. As shown in Figure 2, we initially extract deep frame features using YOLO-MLP. The original video frames are compressed using YOLO-MLP, and we evaluate this method on three sparse coding-based methods: SMRS [3], SC-det [22], and DSSC-log [12]. The average F-measures on the SumMe dataset are 0.242 (SMRS), 0.230 (SC-det), and 0.241 (DSSC-log), respectively. The combination of YOLO-MLP and SMRS achieves the best performance. Thus, SMRS is used as our sparse coding-based key frame extraction mechanism. A detailed description of the proposed KSC-DFF is as follows.

3.1 Deep Frame Feature Extraction

Instead of using raw pixel information from video frames, we decided to leverage picture features to reduce the computational complexity and increase the performance [1]. We use a trained YOLO-MLP network to compute the deep features of each frame. YOLO-MLP is an improved YOLOv5s network for better handling of the key frame extraction task. The network uses a multi-scale fusion technique to detect targets of various sizes and scales. Videos frequently contain scenes in which objects shift in size from small to large or from distant to near. YOLOv5s performs exceptionally well in identifying objects of various sizes, making it a better choice for the early phases of our feature extraction process. After one average pooling concat, we use the input component of the head in the trained YOLOv5s network [24], as shown in Figure 2.

A fully connected multilayer perceptron (MLP) structure is used as the fun- damental building block for visual tasks. We use MLP-Mixer [23] to integrate Concat features. MLP-Mixer, developed by Google Research, can effectively integrate multi-scale information in the input data by introducing the mechanisms of ”token mixing” and ”channel mixing.” This enhances the capacity of the model to adjust to various sizes and feature levels. We employ deep frame features rather than raw pixels as the input for sparse coding. The proposed YOLO- MLP can be extended to other tasks using its plug-and-play functionality.

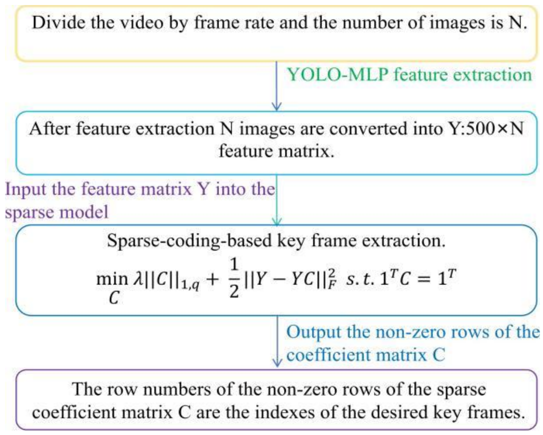

Fig. 3. Illustration of workflow for KSC-DFF.

3.2 Key Frame Extraction Using Sparse Coding

The previous sparse coding-based key frame extraction method considers video frames as data and creates a dictionary matrix of Y for each video frame. We create the following sparse coding cost function for key frame extraction [10], which employs the dictionary matrix Y as the deep frame features extracted by YOLO-MLP:

$$min\ \lambda\ ||C||_{1,q} + \frac{1}{2}\ ||Y_f - Y_f C||^2_f\ \ s.t.\ 1^T C = 1^T\ , \tag{1}$$

where **C** is the sparse representation matrix, $\|\cdot\|_2$ is the Frobenius norm, $\|\mathbf{C}\|_{1,q}$ is the sum of the ℓ_q norms of the rows of **C**, and λ is the trade-off between the sparse measure and approximation error. We set $\lambda = \frac{\lambda_0}{\alpha}$, where λ_0 is analytically computed from the data [10] and α is a hyper-parameter. The selected key frames are columns whose indices correspond to the non-zero elements of **C**. For details of equation (1), please refer to equation (20) in SMRS [10].

3.3 Key Frame Extraction Using Sparse Coding

The goal of YOLO-MLP is to vectorize each video frame and stitch them together to create a feature matrix Y of the video. Sparse coding is then used to extract key frames. Y is regarded as the input signal, and C is the coefficient matrix obtained by sparse coding. The key frame indexes we are searching for are the row numbers of non-zero rows in the coefficient matrix. To make the proposed algorithm more readable, we illustrate its workflow in Figure 3.

4 Experiments

We evaluated our proposed key frame extraction technique (KSC-DFF) on 25 SumMe videos [5]. In particular, we compared KSC-DFF with SMRS and other state-of-the-art algorithms. In our experiments, we ran the codes on a PC with a 3.2 GHz Inter(R) i7-8700 CPU and 16 GB of RAM on the Microsoft Windows 10 operating system.

4.1 Dataset

SumMe is a video summary dataset covering vacations, events, and campaigns. It consists of 25 videos ranging in length from 30 s to 6 min, each with at least 15 person annotations, for a total of 390 annotations.

4.2 Metrics

F-measure [15] is used as the first metric to evaluate the key frame extraction. The higher the F-measure, the better the result.
Summary length. We use summary length as the second metric. The shorter the summary length, the fewer key frames are selected and the better the algorithm performance. The summary length is defined as follows:

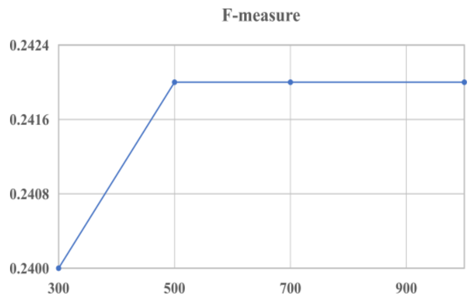

Fig. 4. The various output of YOLO-MLP vectors m representing the F-measure average overall video frames.

$$pInd = dedup(sInd)\ , \tag{2}$$

$$Summary\ length\ (S\text{ - }length) = \frac{len(pInd)}{N}\ , \tag{3}$$

where dedup(·) refers to a redundancy elimination algorithm based on vector distances, designed to remove highly similar samples. sInd denotes the non-zero row index chosen by the algorithm and pInd is the index after de-similarization. len(pInd) is the number of indices.

4.3 Implementation Details

YOLO-MLP is our improved YOLOv5s with MLP-mixer, and the detailed implementation is as follows: the input size of the image is (640 × 640) after the multi-scale fusion part to obtain the feature map (896 × 20 × 20). We convert the deep frame features of each frame into a 500-dimensional vector. Following feature extraction for all L frames, a feature matrix $Y \in R^{500 \times L}$ is constructed. Each column of this matrix comprises an 500-dimensional vector that represents the deep frame features. Notably, the optimal value for 500, determined by achieving the highest F-measure, as depicted in Figure 4, is selected.

The parameter α serves as a hyperparameter, playing a crucial role in balancing the tradeoff between the approximation error and the sparsity measure. During our experiment, α was chosen offline based on the performance evaluation of both summary length and F-measure. Figure 5 illustrates the variations in summary length and F-measure of the proposed KSC-DFF for different values of α. Through rigorous experimentation, we determined the optimal $\alpha = 10^x$ (where x = 4), which yielded superior performance in terms of both summary length and F-measure.

4.4 Performance on SumMe Dataset

The summary lengths of videos from the SumMe dataset are listed in Table 1. On average, our proposed algorithm demonstrated superior performance compared to SMRS. Specifically, the proposed method excels at extracting more condensed key frames, resulting in reduced summary lengths.

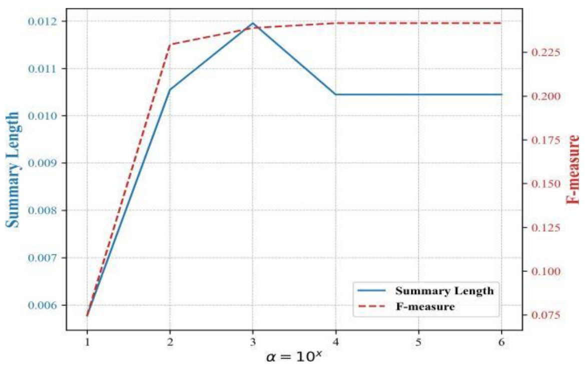

Fig. 5. F-measure variation of the proposed KSC-DFF approach for summary duration and all videos at various α values.

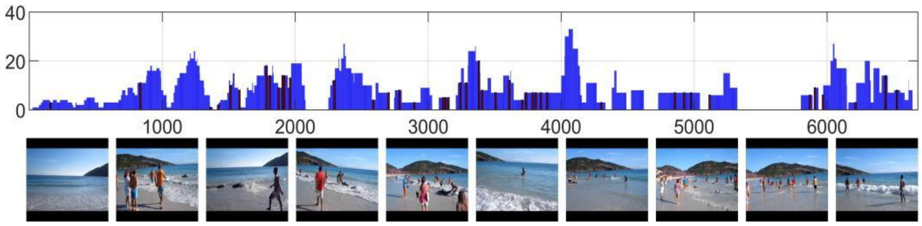

Fig. 6. Key frames of the "Saving dolphins" video generated by the proposed KSC- DFF.

Table 3 shows that the proposed KSC-DFF can achieve competitive results compared with some outstanding methods. Regarding F-measure, the proposed KSC-DFF is not a supervised method but still exhibits a good average F- measure, ranking second. The proposed algorithm can obtain a higher F-measure with a nearly 49.4% increase compared with SMRS on the SumMe dataset. Thus, the proposed KSC-DFF with deep frame features can obtain more accurate key frames with a high F-measure than other sparse coding-based methods, such as SMRS and SC-det. Although the results obtained by the proposed KSC-DFF are worse than DSSC-log in some videos, the running times of KSC-DFF are the lowest among the sparse coding methods as shown in Table 2. Note that, "-" in Table 3 means no results. The related videos, such as "Air Force One", are too long, and DSSC-log cannot obtain results because it is out of memory.

An analysis of the running time of the SumMe dataset is shown in Table 2. We compare the running times of using YOLO-MLP on three sparse coding-based methods: SMRS, SC-det, and DSSC-log. "+" indicates the related sparse coding-based method with YOLO-MLP added. Three example videos with increasing frames from the SumMe dataset are selected. As shown in Table 2, after applying YOLO-MLP, the running times are significantly reduced. The input for the SMRS method is the original image enlarged to one dimension. One of the videos named "Playing ball" is an example, which contains 6096

Table 1. Summary length for KSC-DFF and SMRS.

Video name	Frames	S-length(%)	
		SMRS	KSC-DFF
Base jumping	4729	0.59	0.68
Bearpark climbing	3341	1.98	0.93
Bike Polo	3064	1.54	1.34
Bus in Rock Tunnel	5131	0.55	0.62
Car railcrossing	5075	0.0002	0.73
Cooking	1286	2.20	2.64
Excavators river cross	9721	0.77	0.54
Fire Domino	1612	1.01	1.92
Jumps	950	2.00	2.53
Kids playing in leaves	3187	1.20	1.29
Paintball	6096	0.09	0.34
Paluma jump	2574	1.18	1.48
Playing ball	3120	1.15	1.19
Playing on water slide	3065	1.79	1.11
Saving dolphins	6683	1.16	0.81
Scuba	2221	14.82	1.22
St Maarten Landing	1751	1.31	0.86
Statue of Liberty	3863	1.12	0.70
Valparaiso Downhill	5178	1.64	1.18
Mean		1.90	**1.16**

frames. It first scales each image frame to (224×224) and then expands it to a (1×50176) vector. The input matrix $Y : (50176 \times 6096)$ is made up of 6096 video frames. In contrast, our proposed approach, KSC-DFF, compresses each video frame into a vector of (1×500), allowing us to work at a quicker pace with an input matrix of $Y : (500 \times 6096)$. Our approach preserves more information while reducing the input matrix. Moreover, the proposed KSC-DFF can achieve better results with lower computational time compared to SMRS. The running time for video "Jumps" by KSC-DFF is 27 seconds with a reduction of 98% compared to 1567 seconds by SMRS.

We use YOLO-MLP to compress the original video frames and evaluate the method on three sparse coding-based methods and the results are shown in Table 5. As shown in Table 5, following YOLO-MLP compression, SMRS+ achieves the highest F-measure among the three sparse coding algorithms.

To verify the performance of the proposed method, we pick out the "Save the Dolphins" (which contains 6683 frames) video from the SumMe dataset as the leading example. As shown in Figure 6, blue bars indicate the user's selection results and red bars show the key frames selected by our proposed method. The horizontal axis represents the video frame numbers, and the vertical axis represents the sum of the user scores. The key frames contain the sea, dolphins, and people. We selected more key frames between 1500 and 5000 frames, the frames in this interval are rapidly changing, this period is when the dolphin is being rescued and is the most important part of this video. At the bottom of Figure 6, we can see the whole process of the dolphin stranding to being found by humans to the successful rescue.

Table 2. Runtime on three sparse coding approaches before and after YOLO-MLP compression.

Video	Frames	Running time/s		
		SMRS/+	SC-det/+	DSSC-log/+
Jumps	950	**1567/27**	1648/52	149/58
Bike Polo	3064	**13877/1717**	18037/1617	2548/1017
Paintball	6096	**85687/5407**	54498/12001	12286/6212

Table 3. F-measure of various key frame extraction approaches, namely SC-det, SMRS, DSSC-log, Uni., VGG, Attn., Intr., and DFS.

Video name	F-measure								
	SC-det [22]	Uni. [17]	VGG[20]	Attn. [2]	Intr. [5]	DFS [17]	SMRS [22]	DSSC-log [12]	Ours
Air Force one	0.026	0.060	0.239	0.215	0.318	0.316	0.025	-	0.282
Base jumping	0.207	0.247	0.062	0.194	0.121	0.077	0.157	0.289	0.194
Bearpark climbing	0.210	0.225	0.134	0.227	0.118	0.178	0.234	0.289	0.311
Bike Polo	0.212	0.190	0.069	0.076	0.356	0.235	0.191	0.249	0.246
Bus in Rock Tunnel	0.204	0.114	0.120	0.112	0.135	0.151	0.198	0.255	0.224
Car over camera	0.352	0.245	0.048	0.201	0.372	0.132	0.087	0.240	0.292
Car railcrossing	0.175	0.185	0.139	0.064	0.362	0.328	0.179	-	0.179
Cockpit Landing	0.127	0.103	0.190	0.116	0.172	0.165	0.127	-	0.247
Cooking	0.200	0.076	0.285	0.118	0.321	0.329	0.148	0.281	0.261
Eiffel Tower	0.225	0.142	0.008	0.136	0.295	0.174	0.205	-	0.227
Excavators river cross	0.254	0.107	0.030	0.041	0.189	0.134	0.223	-	0.223
Fire Domino	0.205	0.103	0.124	0.252	0.130	0.022	0.102	0.288	0.313
Jumps	0.274	0.054	0.000	0.243	0.427	0.015	0.304	0.300	0.273
Kids playing in leaves	0.263	0.051	0.243	0.084	0.089	0.278	0.217	0.274	0.186
Notre Dame	0.167	0.156	0.136	0.138	0.235	0.093	0.193	-	0.198
Paintball	0.298	0.071	0.270	0.281	0.320	0.274	0.068	0.225	0.246
Paluma jump	0.089	0.058	0.056	0.028	0.181	0.428	0.093	0.263	0.225
Playing ball	0.237	0.123	0.127	0.140	0.174	0.194	0.200	0.277	0.220
Playing on water slide	0.155	0.075	0.092	0.124	0.200	0.183	0.163	0.252	0.174
Saving dolphins	0.066	0.146	0.103	0.154	0.145	0.121	0.060	0.281	0.190
Scuba	0.099	0.070	0.160	0.200	0.184	0.154	0.096	0.264	0.277
St Maarten Landing	0.434	0.152	0.153	0.419	0.313	0.015	0.245	0.279	0.369
Statue of Liberty	0.160	0.184	0.098	0.083	0.192	0.143	0.139	0.216	0.174
Uncut Evening Flight	0.159	0.074	0.168	0.299	0.271	0.168	0.186	-	0.238
Valparaiso Downhill	0.232	0.083	0.110	0.231	0.242	0.258	0.212	0.275	0.278
Mean	0.201	0.124	0.127	0.167	0.234	0.183	0.162	-	**0.242**

4.5 Ablation experiment

To comprehensively evaluate the contribution of each component in our proposed framework for key frame extraction, we conducted an ablation study using the F1-measure as the primary evaluation metric. We compared the baseline YOLO-based model, a standalone MLP-based approach, and our full YOLO + MLP integration. As shown in Table 4, the YOLO + MLP model consistently outperforms the other configurations across all test samples, achieving the highest average F1 score. This notable improvement highlights the complementary strengths of the spatial-temporal modeling capability of YOLO and the global semantic representation learned by the identification of semantically meaningful and temporally representative key frames. These results strongly validate the effectiveness of our joint architecture in enhancing the precision and robustness of key frame extraction.

Table 4. Ablation results of MLP-based, YOLO-based, and YOLO + MLP on F1-Measure for key frame extraction.

Video name	F-measure		
	MLP-based	YOLO-based	KSC-DFF
Air Force One	0.288	0.225	0.282
Base jumping	0.203	0.208	0.194
Bearpark climbing	0.237	0.209	0.311
Bike Polo	0.230	0.221	0.246
Bus in Rock Tunnel	0.247	0.187	0.224
Car railcrossing	0.156	0.158	0.179
Cockpit Landing	0.245	0.243	0.247
Cooking	0.292	0.233	0.261
Eiffel Tower	0.237	0.161	0.227
Excavators river crossing	0.243	0.225	0.223
Fire Domino	0.326	0.199	0.313
Jumps	0.347	0.146	0.273
Kids playing in leaves	0.136	0.267	0.186
Notre Dame	0.224	0.206	0.198
Paintball	0.239	0.298	0.246
Playing on water slide	0.180	0.188	0.174
Saving dolphins	0.184	0.257	0.190
Scuba	0.224	0.199	0.277
St Maarten Landing	0.314	0.243	0.369
Statue of Liberty	0.167	0.176	0.174
Uncut Evening Flight	0.210	0.178	0.237
Valparaiso Downhill	0.212	0.262	0.278
Car over camera	0.232	0.226	0.292
Paluma jump	0.146	0.206	0.225
Playing ball	0.191	0.212	0.220
Mean	0.228	0.213	**0.242**

5 Conclusion

In this paper, we proposed a novel key frame extraction approach, KSC-DFF, to consider deep frame features to achieve better results. First, we extract deep frame features with richer, deeper information, such as included object features. The deep frame features are obtained by YOLO-MLP networks, which employ MLP-Mixer to integrate the multi-scale information of the feature fusion networks. Then, we can effectively extract the key frames of the video by sparse coding. Key frames can be estimated automatically according to the nonzero rows in the learned sparse coefficient matrix.Experimental results show that our proposed KSC-DFF method performs well in extracting more accurate key frames from SumMe videos compared to most state-of-the-art techniques. Our approach not only accelerates sparse representation key frame extraction but also ensures high accuracy. Additionally, it is the fastest among sparse coding methods. However, our method is not universally suitable for all sparse coding techniques. In future work, we will explore different feature extraction methods to improve generalization.

Table 5. F-measure of three sparse coding methods after YOLO-MLP compression.

Video name	F-measure		
	SC-det+	DSSC-log+	SMRS+(Ours)
Air Force One	0.231	0.289	0.282
Base jumping	0.210	0.210	0.194
Bearpark climbing	0.239	0.296	0.311
Bike Polo	0.209	0.247	0.246
Bus in Rock Tunnel	0.196	0.227	0.224
Car railcrossing	0.179	0.174	0.179
Cockpit Landing	0.193	0.248	0.247
Cooking	0.203	0.237	0.261
Eiffel Tower	0.226	0.226	0.227
Excavators river crossing	0.216	0.218	0.223
Fire Domino	0.236	0.310	0.313
Jumps	0.274	0.273	0.273
Kids playing in leaves	0.189	0.170	0.186
Notre Dame	0.260	0.204	0.198
Paintball	0.260	0.280	0.246
Playing on water slide	0.262	0.177	0.174
Saving dolphins	0.197	0.193	0.190
Scuba	0.217	0.264	0.277
St Maarten Landing	0.336	0.345	0.369
Statue of Liberty	0.190	0.173	0.174
Uncut Evening Flight	0.257	0.239	0.237
Valparaiso Downhill	0.242	0.277	0.278
Car over camera	0.295	0.306	0.292
Paluma jump	0.209	0.225	0.225
Playing ball	0.230	0.222	0.220
Mean	0.230	0.241	**0.242**

References

1. Chaari, L., Batatia, H., Dobigeon, N., Tourneret, J.Y.: A hierarchical sparsity- smoothness bayesian model for l 0+ l1+ l2 regularization. In: 2014 IEEE Inter- national Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1901–1905. IEEE (2014)
2. Ejaz, N., Mehmood, I., Baik, S.W.: Efficient visual attention based framework for extracting key frames from videos. Signal Processing: Image Communication 28(1), 34–44 (2013)
3. Elhamifar, E., Sapiro, G., Vidal, R.: See all by looking at a few: Sparse modeling for finding representative objects. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 1600–1607. IEEE (2012)
4. Guan, G., Wang, Z., Lu, S., Da Deng, J., Feng, D.D.: Keypoint-based keyframe selection. IEEE Transactions on circuits and systems for video technology 23(4), 729–734 (2012)
5. Gygli, M., Grabner, H., Riemenschneider, H., Van Gool, L.: Creating summaries from user videos. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13. pp. 505–520.Springer (2014)
6. Hong, S., Ryu, J., Im, W., Yang, H.S.: D3: recognizing dynamic scenes with deep dual descriptor based on key frames and key segments. Neurocomputing 273, 611– 621 (2018)
7. Hussain, T., Muhammad, K., Ding, W., Lloret, J., Baik, S.W., De Albuquerque, V.H.C.: A comprehensive survey of multi-view video summarization. Pattern Recognition 109, 107567 (2021)
8. J, C.: Hours of video uploaded to youtube every minute 2007–2019. https://www.statista.com/statistics/259477/ (2020)
9. Jain, S., Gonzalez, J.E.: Fast semantic segmentation on video using block motion- based feature interpolation. In: Proceedings of the European Conference on Com- puter Vision (ECCV) Workshops. pp. 0–0 (2018)
10. Kulhare, S., Sah, S., Pillai, S., Ptucha, R.: Key frame extraction for salient ac- tivity recognition. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 835–840. IEEE (2016)
11. Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: 2011 International conference on computer vision. pp. 1995–2002. IEEE (2011)
12. Li, Z., Li, Y., Tan, B., Ding, S., Xie, S.: Structured sparse coding with the group log-regularizer for key frame extraction. IEEE/CAA Journal of Automatica Sinica 9(10), 1818–1830 (2022)
13. Lu, H., Jin, L., Luo, X., Liao, B., Guo, D., Xiao, L.: Rnn for solving perturbed time-varying underdetermined linear system with double bound limits on residual errors and state variables. IEEE Transactions on Industrial Informatics 15(11), 5931–5942 (2019)
14. Luo, X., Shang, M., Li, S.: Efficient extraction of non-negative latent factors from high-dimensional and sparse matrices in industrial applications. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). pp. 311–319. IEEE (2016)
15. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natu- ral images and its application to evaluating segmentation algorithms and measur- ing ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 416–423. IEEE (2001)

16. Nasreen, A., Shobha, G.: Reducing redundancy in videos using reference frame and clustering technique of key frame extraction. In: International Confer- ence on Circuits, Communication, Control and Computing. pp. 348–440 (2014). https://doi.org/10.1109/CIMCA.2014.7057840
17. Otani, M., Nakashima, Y., Rahtu, E., Heikkil¨a, J., Yokoya, N.: Video summarization using deep semantic features. In: Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13. pp. 361–377. Springer (2017)
18. Qu, Z., Zhang, L., Wang, X., Cao, B., Li, Y., Li, F.: Ksf-st: Video captioning based on key semantic frames extraction and spatio-temporal attention mechanism. In: 2020 International Wireless Communications and Mobile Computing (IWCMC). pp. 1388–1393 (2020). https://doi.org/10.1109/IWCMC48107.2020.9148294
19. Savran Kızıltepe, R., Gan, J.Q., Escobar, J.J.: A novel keyframe extraction method for video classification using deep neural networks. Neural Computing and Appli- cations 35(34), 24513–24524 (2023)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. Sujatha, C., Mudenagudi, U.: A study on keyframe extraction methods for video summary. In: 2011 International Conference on Computational Intelligence and Communication Networks. pp. 73–77. IEEE (2011)
22. Tan, B., Li, Y., Ding, S., Paik, I., Kanemura, A.: Dc programming for solving a sparse modeling problem of video key frame extraction. Digital Signal Processing 83, 214–222 (2018)
23. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al.: Mlp-mixer: An all- mlp architecture for vision. Advances in neural information processing systems 34, 24261–24272 (2021)
24. Ultralytics: YOLOv5: An open source, real-time object detection system. https://github.com/ultralytics/yolov5 (2020)
25. Wolf, W.: Key frame selection by motion analysis. In: 1996 IEEE international con- ference on acoustics, speech, and signal processing conference proceedings. vol. 2, pp. 1228–1231. IEEE (1996)
26. Wu, D., Luo, X., Shang, M., He, Y., Wang, G., Zhou, M.: A deep latent factor model for high-dimensional and sparse matrices in recommender systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems 51(7), 4285–4296 (2019)
27. Xia, G., Chen, B., Sun, H., Liu, Q.: Nonconvex low-rank kernel sparse subspace learning for keyframe extraction and motion segmentation. IEEE transactions on neural networks and learning systems 32(4), 1612–1626 (2020)
28. Xin, L., Yuan, Y., Zhou, M., Liu, Z., Shang, M.: Non-negative latent factor model based on β-divergence for recommender systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems 51(8), 4612–4623 (2019)
29. Zhang, Y., Zhang, S., Li, Y., Zhang, J., Cai, Z., Shui, L.: Surveillance video key frame extraction based on center offset. Comput Mater Contin 68(3), 4175–4190 (2021)
30. Zhong, R., Wang, R., Zou, Y., Hong, Z., Hu, M.: Graph attention networks adjusted bi-lstm for video summarization. IEEE Signal Processing Letters 28, 663– 667 (2021)
31. Zhu, W., Huang, Y., Xie, X., Liu, W., Deng, J., Zhang, D., Wang, Z., Liu, J.:Autoshot: A short video dataset and state-of-the-art shot boundary detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2237–2246 (2023)
32. Zou, D., Chen, X., Cao, G., Wang, X.: Unsupervised video matting via sparse and low-rank representation. IEEE transactions on pattern analysis and machine intelligence 42(6), 1501–1514 (2019)