A Cascaded Approach for Keyframes Extraction from Videos

Conference Paper *in* Communications in Computer and Information Science · November 2020

DOI: 10.1007/978-3-030-63426-1_8

A Cascaded Approach for Keyframes Extraction from Videos ⋆

Yunhua Pei[1], Zhiyi Huang[1], Wenjie Yu[1], Meili Wang[123], and Xuequan Lu[4]

[1] College of Information Engineering, Northwest AF University, China.
[2] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China.
[3] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling 712100, China.
wml@nwsuaf.edu.cn
https://cie.nwsuaf.edu.cn/szdw/fjs/2012110003/
[4] Deakin University, 221 Burwood Highway, Burwood, Victoria 3125, Australia
xuequan.lu@deakin.edu.au
http://www.xuequanlu.com/

Abstract. Keyframes extraction, a fundamental problem in video processing and analysis, has remained a challenge to date. In this paper, we introduce a novel method to effectively extract keyframes of a video. It consists of four steps. At first, we generate initial clips for the classified frames, based on consistent content within a clip. Using empirical evidence, we design an adaptive window length for the frame difference processing which outputs the initial keyframes then. We further remove the frames with meaningless information (e.g., black screen) in initial clips and initial keyframes. To achieve satisfactory keyframes, we finally map the current keyframes to the space of current clips and optimize the keyframes based on similarity. Extensive experiments show that our method outperforms to state-of-the-art keyframe extraction techniques with an average of 96.84% on precision and 81.55% on F_1.

Keywords: keyframe extraction · frame difference · image classification · video retrieval.

1 Introduction

Keyframes extraction, that is extracting keyframes from a video, is a fundamental problem in video processing and analysis. It has a lot of application fields like video coding, so it is important to design robust and effective keyframe extraction methods. Current methods are usually based on either pixel matrix or deep learning classification results [5, 6, 9, 10, 18, 19]. However, they still suffer from some limitations. More specifically, the keyframe extraction techniques based on pixel matrix are not capable of achieving decent accuracies, for example, when

⋆ This is a preprint

handling news videos[16]. Nevertheless, the involved keyframes extraction could take a considerable amount of time [15].

Motivated by the above issues, we propose a novel keyframe extraction approach in this paper. Given an input video, we first turn it into frames and perform classification with available deep learning networks. The classified frames are split into initial clips, each of which has consistent content. We then design an adaptive window length for frame difference processing which takes the computed initial clips as input and outputs. Also, we remove the frames with meaningless information for previous results, such as black screen. Eventually, to obtain desired keyframes, we map the current keyframes to the space of the current clips and optimize the keyframes based on similarity.

Our method is simple yet effective. It is elegantly built on top of deep learning classification and the frame difference processing. Experiments validate our approach and demonstrate that it outperforms or is comparable to state-of-the-art keyframe extraction techniques. The main contributions of this paper are:

- a novel robust keyframe extraction approach that fits various types of videos;
- the design of the adaptive window length and the removal of meaningless frames;
- a mapping scheme and an optimization method on determining keyframes.

Our source code will be released online.

2 Related Work

Keyframes extraction has been studied extensively. We only review researches mostly relevant to our work. Please refer to [2, 12] for a comprehensive review.

Some researchers introduced a keyframe extraction method for human motion capture data, through exploiting the sparseness and Riemannian manifold structure of human motion [17]. Guan et al. introduced two criteria, coverage and redundancy, based on keypoint matching, to solve the keyframe selection problem [3]. Kuanar et al. obtained keyframes with iterative edge pruning strategy using dynamic Delon Diagram for clustering [5]. Mehmood et al. used both viewer attention and aural attention to do the extraction [11].

Some researchers extract keyframes based on machine learning results. Yang et al. used an unsupervised clustering algorithm to first divide the frames, and then selected keyframes from the clustering candidates [18]. Yong et al. extracted keyframes by undergoing image segmentation, feature extraction and matching of image blocks, and the construction of a co-occurrence matrix of semantic labels [19]. Li et al. mapped the video data to a high-dimensional space and learnt a new representation which could reflect the representativeness of the frame [7].

Although Motion capture and machine learning are effective ways proved to extract high-quality keyframes through existing researches, there is still no one algorithm that can extract keyframes from these two perspectives at the same time and suit each kind of videos.

3 Method

3.1 Overview

Our keyframe extraction approach consists of four steps which are specifically:

1. Initial clips generation. We first obtain the classification results and split the classified frames into clips that respectively involve consistent content.
2. Adaptive Window Length for frame difference processing. We then design an adaptive window length for the frame difference method which takes the output of Step 1 as inputs and outputs the initial keyframes.
3. Meaningless frames removal. We refine the results of Step 1 and 2 by removing the frames with meaningless information (e.g., black screen).
4. Mapping and optimization. After removing meaningless frames, we finally map the current keyframes to the space of the current clips and perform keyframes optimization based on similarity, to achieve more representative keyframes.

3.2 Generating Initial Clips

ImageAI library [1] has four different deep learning networks (Resnet50, DenseNet-BC-121-32, Inception_v3, Squeezenet) separately trained on the imagenet-1000 dataset. The four networks in this paper are using default parameter settings, please reference [1] for more details. The classification $softmax\left(s_{i\,\max}\right)$ is defined as

$$softmax\left(s_i\right) = \frac{e^{s_i}}{\sum_{i=1}^{N} e^{s_i}} (i = 1, \ldots, N) \tag{1}$$

where s_i represents the score of the input x on the i_{th} category. After the calculation, we take the maximum recognition probability of an image as its label.

We turn the videos into consecutive frames which act as the input to the networks. Since one object in different scenes involve different meaning, it is necessary to treat them as independent scenes. Based on the image classification results, we simply split each video into a few clips, each of which continuously represent certain content, by simply checking if the labels of the current and next frames are the same or not.

3.3 Adaptive Window Length

The original frame difference method [13] simply sets the window length to 1 or a fixed value, which is prone to generate undesired keyframes. To solve this issue, introduce a formula to enable an adaptive length (L) calculation.

$$L = \frac{\sum (\mathit{frame})}{\mathit{Exp\_value}} \tag{2}$$

where $\sum(\mathit{frame})$ is the total number of frames in the video, and $\mathit{Exp\_value}$ denotes the expected number of keyframes.

We will describe how we achieve an adaptive threshold in experiments (Sec. 4.1), based on the videos.

3.4 Meaningless Frames Removal

Some consecutive frames may deliver little information, for example, black or white or other fuzzy colors representing non-recognizable items. As such, it is necessary to refine the results in Sec. 3.2 and Sec. 3.3. We judge whether a frame image has information or not by defining as follows, where mount is the total number of different colors in a frame. The *mount* threshold (i.e., T) is empirically set to 600 after testing on a hundred randomly selected 640x480 pixels pure color imagines from the Internet and experiment videos.

$$P = \begin{cases} 1, mount \geq T \\ 0, others \end{cases} \tag{3}$$

3.5 Mapping and Optimization

To obtain satisfactory results and reduce redundant results, we propose to map the results by Sec. 3.3 onto the space of the results after Sec. 3.2. After mapping, we perform the first optimization by computing the average similarity of each frame in a clip, and one with the highest average similarity is set as the keyframe of this clip. Finally, if two keyframes of two consecutive clips that are generated in Sec. 3.2 have a similarity above 50%, we conduct a second optimization by simply choosing the first keyframe and discarding the second keyframe. This is because sometimes actual vidoes have fast switch of camera shots which leads to repeated or similar scenes.

Similar to [4], the similarity is formulated as:

$$sim(G, S) = \frac{1}{M} \sum_{i=1}^{M} \left(1 - \frac{|g_i - s_i|}{Max\,(g_i - s_i)} \right) \tag{4}$$

where G and S are the values of the histogram after transforming the two images into regular images, respectively. M is the total number of samples in the color space. More information can be referred to [14]. The expected keyframe is the one with the greatest similarity, as follows $\arg\max_k \left(\frac{1}{|C|-1} \sum sim(G_k, S) \right)$, where (G_k, S) and $|C|$ are a pair of two frames (not identical) and the number of frames in the involved clip C, respectively. It is often not necessary to compare the similarity between frames of different clips due to the discontinuity and dissimilarity.

4 Experiments

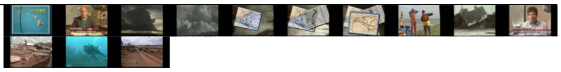
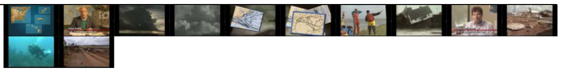
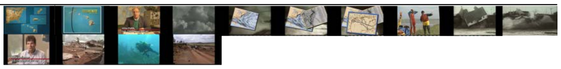
4.1 Experimental Setup

Data. A dataset of ten videos are used to validate the proposed method, which includes six types of video (i.e., advertisement, documentary and games).

Ground truth. Similar to previous research [7], three volunteers with multimedia expertise independently selected and merged the keyframes of each video.

If some images deliver the same information, they are merged by manually selecting the most representative one and discarding others. Table 1 illustrates an example. We perform this operation on all the videos and the results are shown as MK in Table 2.

Table 1: Ground truth keyframes generation example.

Volunteer A	
Volunteer B	
Volunteer C	
Merged	

We display the numbers of keyframes for initial keyframes and final keyframes based on the four classification networks in Table 2. It can be seen from Table 2 that for each video, the numbers of final keyframes decreased in general, which indicates that Sec. 3.4 and Sec. 3.5 refine the initial keyframes. Our method runtime is also shown in Tab 2 which sees $0.66 - -6.79$ times of the length of the videos.

Table 2: Results and runtime by using different networks (in seconds). MK: merged keyframes. IK: initial keyframes. FK: final keyframes. RT: runtime R: Resnet50, D: DenseNet-BC-121-32, I: Inception_v3 S: Squeezenet. AVG: average

Video	MK	IK$_R$	IK$_D$	IK$_I$	IK$_S$	FK$_R$	FK$_D$	FK$_I$	FK$_S$	RT$_R$	RT$_D$	RT$_I$	RT$_S$	RT$_{AVG}$
Ads_Audi(84s)	22	29	31	17	21	21	22	17	18	368.1	442.9	295.5	208.0	328.6
BBC...Camera(169s)	21	62	42	32	35	28	21	22	22	485.9	621.0	380.0	143.1	407.5
Highlight_soccer(101s)	30	31	41	28	28	25	33	23	23	598.3	685.5	533.2	398.2	553.8
lion vs zebra judo(18s)	3	64	55	64	100	3	2	2	2	88.7	104.2	81.4	37.9	78.0
MV_Gnstyle(252s)	41	36	46	30	30	31	40	26	26	426.0	518.8	350.0	166.7	365.4
Trailer...nonesub(161s)	46	37	23	30	26	18	15	18	16	284.5	315.8	225.5	115.3	235.3
Trailer...sub(147s)	45	47	63	54	63	31	38	34	40	606.9	652.3	490.7	392.9	535.7
Trailer_Pokemon(196s)	72	85	76	65	75	66	63	56	66	622.8	770.6	498.3	287.5	544.8
UGS10_003(77s)	14	33	27	34	35	12	11	11	12	366.9	426.0	293.2	166.3	313.1
Dragons Fight(87s)	10	27	25	27	27	6	7	9	8	270.1	316.7	234.8	128.1	237.4

Experimental setting. Our framework is implemented in a Lenovo Y7000 laptop with an Intel(R) Core(TM) i5-9300H64 2.3GHz CPU and a NVIDIA GeForce GTX 1050 Graphics card.

Window length. We take the separated clips of ResNet50 as an example, to remove the short clips within a certain thresholding number. We found that the remaining clips by setting this threshold to 10 can cover 90% of frames from the whole video.

4.2 Quantitative Results

As with previous works [7, 8, 11], the precision P, the recall R and the the average (F_1) of the P and R are employed as the evaluation metrics. They are computed as:

$$P = \frac{N_c}{N_c + N_f} \times 100\% \tag{5}$$

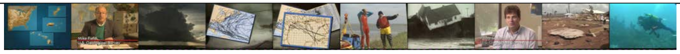
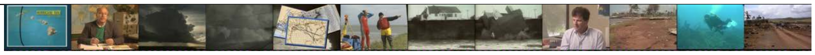
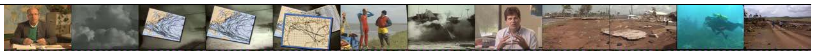
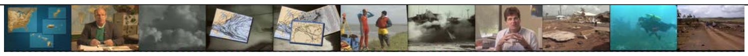
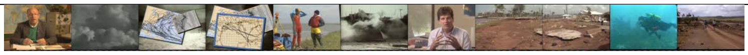
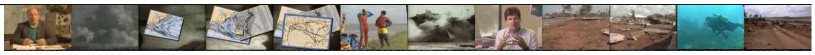
$$R = \frac{N_c}{N_c + N_m} \times 100\% \tag{6}$$

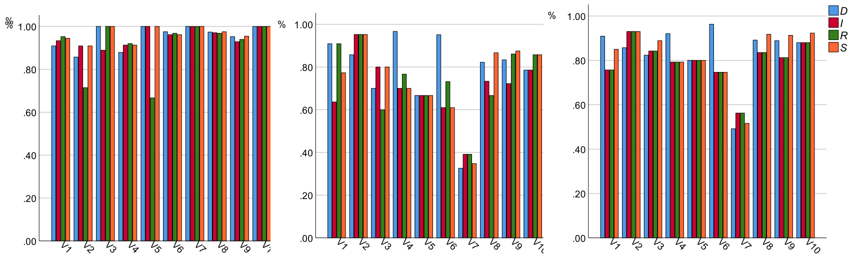
$$F_1 = \frac{2 * R * P}{R + P} \times 100\% \tag{7}$$

where N_c denotes the number of correctly extracted keyframes, and N_f refers to the number of incorrect keyframes. N_m is the number of missing keyframes. F_1 is a combined measure of R and P, and a higher value indicates both higher R and P.

Fig.1 shows the evaluation numbers for each video using four different networks, and we can observe that the accuracies of our method are high, except for only a few outliers (e.g., R network for V2 and V5). The average precision is 96.84%. Fig. 1(b) gives the recall numbers which are lower than precision numbers. We suspect that it has two reasons. The ground truth set of a video is simply removed high similiarity frames. Moreover, some blurry frames can result in misclassifications and further lower numbers of keyframes. As a result, $N_c + N_m$ becomes large and N_c becomes small, thus leading to relatively low recall numbers. Fig. 1(c) reflects that the overall performance is generally good, with an average of 81.55% for all videos and 84.69% for videos except the outlier video V7.

Table 3: Comparison with [7] and [8].

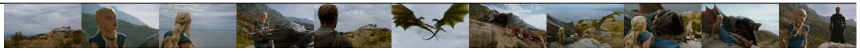
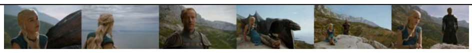
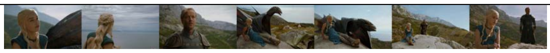
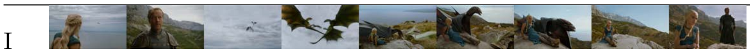
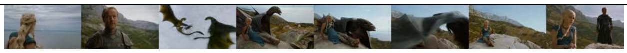
[7]	
[8]	
R	
D	
I	
S	



(a) Precision rate (b) Recall rate (c) F1 index

Fig. 1: Evaluation results. (a) Precision rate. (b) Recall rate. (c) F1 index. D:DenseNet-BC-121-32, I:Inception_v3, R:Resnet50, S:Squeezenet.

Table 4: Comparison with [11].

[11]	
R	
D	
I	
S	

Besides the above experiments that validate our approach, we also compare our method with state-of-the-art keframe extraction techniques [7,8,11]. Table 3 and 4 show some visual comparisons for our method and [7,8,11]. It can be seen from Table 3 that our extracted keyframes are very similar to current techniques [7,8]. Furthermore, our method can extract more representative keyframes, in terms of significant distinctions and front views. Table 4 shows that our method can extract fewer keyframes than [11] to describe the key content of the video. While the results by [11] seem a bit redundant, in terms of scene keyframes. Their scene keyframes occupied 36.4% while ours only took up 16.1%. This video is actually concentrated more on humans than pure scenes.

In addition to visual comparisons, we also conduct quantitative comparisons using the metrics mentioned above. The average P, R and F_1 numbers are listed in Tab. 5. Our average P numbers based on the four networks are the highest among all methods. Notice the numbers inside brackets are computed by excluding the outlier video V7.

Table 5: Metrics comparison by using different methods.

	P	R	F1
[7]	87.5%	84.0%	85.7%
[8]	92.0%	87.8%	89.9%
[11]	90.0%	80.0%	84.7%
R	92.8%(93.3%)	80.2%(79.3%)	86.0%(82.5%)
D	93.9%(94.3%)	80.2%(85.3%)	86.5%(89.2%)
I	93.5%(93.8%)	76.1%(74.3%)	83.9%(82.3%)
S	95.7%(95.7%)	83.2%(80.4%)	89.0%(87.0%)

5 Conclusion

We have proposed a novel framework for extracting keyframes from videos. Various experiments demonstrate that our approach is effective, and better or comparable to state-of-the-art methods.

One limitation is that it is challenging to classify burred images for existing deep learning networks, thus leading to undesired keyframes for videos with frequent blur. As the future work, we would like to investigate and solve this limitation, for example, by incorporating deblurring techniques into our framework.

Acknowledgement

This work was partially funded by Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China ($2018AIOT-09$). National Natural Science Foundation of China (61702433), Key Research and Development Program of Shaanxi Province ($2018NY-127$).

References

1. open source python library built to empower developers to build applications and systems with self-contained computer vision capabilities, https://github.com/OlafenwaMoses/frameAI
2. Asghar, M.N., Hussain, F., Manton, R.: Video indexing: a survey. International Journal of Computer and Information Technology **3**(01) (2014)
3. Guan, G., Wang, Z., Lu, S., Deng, J.D., Feng, D.D.: Keypoint-based keyframe selection. IEEE Transactions on Circuits and Systems for Video Technology **23**(4), 729–734 (April 2013). https://doi.org/10.1109/TCSVT.2012.2214871
4. Jiang, L., Shen, G., Zhang, G.: An image retrieval algorithm based on hsv color segment histograms. Mechanical & Electrical Engineering Magazine **26**(11), 54–57 (2009)
5. Kuanar, S.K., Panda, R., Chowdhury, A.S.: Video key frame extraction through dynamic delaunay clustering with a structural constraint. Journal of Visual Communication and Image Representation **24**(7), 1212–1227 (2013)
6. Kulhare, S., Sah, S., Pillai, S., Ptucha, R.: Key frame extraction for salient activity recognition. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 835–840. IEEE (2016)
7. Li, X., Zhao, B., Lu, X.: Key frame extraction in the summary space. IEEE transactions on cybernetics **48**(6), 1923–1934 (2017)
8. Liu, H., Li, T.: Key frame extraction based on improved frame blocks features and second extraction. In: 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). pp. 1950–1955. IEEE (2015)
9. Liu, H., Meng, W., Liu, Z.: Key frame extraction of online video based on optimized frame difference. In: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery. pp. 1238–1242. IEEE (2012)
10. Luo, Y., Zhou, H., Tan, Q., Chen, X., Yun, M.: Key frame extraction of surveillance video based on moving object detection and image similarity. Pattern Recognition and Image Analysis **28**(2), 225–231 (2018)
11. Mehmood, I., Sajjad, M., Rho, S., Baik, S.W.: Divide-and-conquer based summarization framework for extracting affective video content. Neurocomputing **174**, 393–403 (2016)
12. Milan Kumar Asha Paul, J.K., Rani, P.A.J.: Key-frame extraction techniques: A review. Recent Patents on Computer Science **11**(1), 3–16 (2018). https://doi.org/10.2174/2213275911666180719111118
13. Singla, N.: Motion detection based on frame difference method. International Journal of Information & Computation Technology **4**(15), 1559–1565 (2014)
14. Swain, M.J., Ballard, D.H.: Indexing via color histograms. In: Active perception and robot vision, pp. 261–273. Springer (1992)
15. Tang, H., Zhou, J.: Method for extracting the key frame of various types video based on machine learning. Industrial Control Computer **3**, 94–95 (2014)
16. Wang, S., Han, Y., Yadong, W.U., Zhang, S.: Video key frame extraction method based on image dominant color. Journal of Computer Applications **33**(9), 2631–2635 (2013)
17. Xia, G., Sun, H., Niu, X., Zhang, G., Feng, L.: Keyframe extraction for human motion capture data based on joint kernel sparse representation. IEEE Transactions on Industrial Electronics **64**(2), 1589–1599 (2016)
18. Yang, S., Lin, X.: Key frame extraction using unsupervised clustering based on a statistical model. Tsinghua Science & Technology **10**(2), 169–173 (2005)

19. Yong, S.P., Deng, J.D., Purvis, M.K.: Wildlife video key-frame extraction based on novelty detection in semantic context. Multimedia Tools and Applications **62**(2), 359–376 (2013)