Online Video Summarization with the Kohonen SOM in Real Time

Oleksii Gorokhovatskyi[0000-0003-3477-2132], Oleh Teslenko[0000-0003-3105-9323] ,Volodymyr Zatkhei[0000-0003-4426-7789]

Simon Kuznets Kharkiv National University of Economics, 9a Nauka Ave., Kharkiv 61166 Ukraine
oleksii.gorokhovatskyi@gmail.com

Abstract. In this paper, we propose an algorithm for the creation of an automatic video summary. It is based on the Kohonen's Self-Organizing Map as a method for training and clustering of frame features in online mode. The decision about whether the frame should be in summary depends on the stability of the last sequential clustering results. Three-way matching of images between automatic summary and corresponding user one is proposed and tested. Open Video and SumMe datasets were used for accuracy and performance comparison. It is shown, that the proposed approach can achieve real time summarization combining with its online properties without the requirement to see the whole video. The accuracy (measured by F1 scores) of the proposed approach can compete with batch processing methods. We also compared the performance to the state-of-the-art existing methods of online real time processing.

Keywords: video summarization, keyframe, self-organizing map, clustering, image features, matching, summary

1 Introduction and the related work

In recent years tremendous development of the information, computer and communication technologies made humans impossible to process all available and appearing data themselves. Especially, this is true for video content, e.g. every minute 300-500 hours of video (by different sources) are uploaded to YouTube, additionally, users watch over a billion hours every day.

Video summarization is a process of selecting some specific subset of keyframes (still images) or keyshots (small sequences of frames) from a video stream which preserves the main idea of video [1]. A summary should keep important frames of the initial video that creates the core summarization challenge – the same frames may be important and unimportant at the same time for different users, making in such a way the summary of video to be a quite subjective term.

A lot of researches [2–5] distinguish all summarization methods into two classes: unsupervised [6–8] and supervised [9–12]. A brief description of some known methods is below.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2608/paper80.pdf

Some researchers [13–15] used self-organizing maps or close approaches. Hierarchical Growing Cell Structures (GCS) methods are proposed in [13, 14] as an extension of the Kohonen's Self-Organizing Maps (SOM) that allows to build a flexible structure without knowing the number of classes a priori. A graphical user interface to investigate the construction of two-dimensional SOM is proposed in [15]. Unfortunately, these researches don't contain significant modeling results using at least a dataset of medium size.

One of the most popular summarization approach is VSUMM [7, 16]. The authors proposed the method based on the extraction of color features from video frames and unsupervised classification. Additionally, a new measure to compare automatic and user-defined summaries (Comparison of User Summaries, CUS) was presented.

The variety of other video summarizing methods, based on generative adversarial networks [17], long short-term memory networks [11, 12], attention-based [18, 19] and deep learning approaches [20], using of text annotations both with visual features [21], fuzzy-based incremental clustering [22] were proposed before.

Paper [8] describes the idea to generate video summaries in online mode immediately without seeing the entire video in quasi real time. This method includes the building of the dictionary via group sparse coding for some initial video frames, following by the reconstruction attempt of unseen frames. If reconstruction error is significant enough, a dictionary is updated and the current frame is added to summary.

Ideas proposed in [23] extend dictionary learning with the prediction of interestingness using global camera motion analysis and colorfulness. This approach seems not to be designed for online processing but focused on real time processing mainly.

The implementations of the abovementioned approaches [8] and [23] seem not to be available, as well as test videos for [8].

The contributions of the paper include:

- video summarization method based on the self-organizing maps, that can work in online mode (without seeing the whole video) in real time;
- new three-stage matching of two sets of frames, that includes both keypoint and raw image pixels comparison;
- selection of the keyframes based on Kohonen's SOM clustering stability;
- we performed the quality assessment of the proposed online summary generation method using the dataset, which was created by volunteers in online fashion also.

2 Problem statement

Formally the problem is to create the summary of the video as the set of still keyframes. We want to generate summary frames on the fly in online mode without seeing the whole video. The summarization method should be fast enough to fit real time processing requirement. We want to generate such a summary that is close to the corresponding one created by the human in the same online conditions.

3 Self-organizing maps

Kohonen's self-organizing map (SOM) [24] is one of the most popular neural network unsupervised clustering approaches. This type of network preserves the topology between input and output values and allows to map multidimensional input into low-dimensional (typically 1D or 2D) outputs.

The important property of the Kohonen network is that it is capable of online data processing when input samples come one by one followed by immediate clustering (classification) decision.

Training of Kohonen SOM network implies the update of all weights after the processing of each training sample one by one (online training) and may be done according to stages below. Let n be the quantity of outputs (known a priori) and we denote the quantity of features as m.

1. Initializing of weights for each neuron with the small random weights.

2. Selection for input features vector x.

3. Train the features vector while error for it is bigger than training epsilon (0.0001):

3.1 Find the closest (Best Matching Unit, BMU) ϑ neuron in terms of some distance, e.g. Euclidean.

3.2 Update weights w_{ij} for all n neurons according to (1):

$$w_{ij} = w_{ij} + \alpha(t)\theta(x_i, \vartheta, t)(x_i - w_{ij}) \tag{1}$$

where $\alpha(t) = 0.1e^{-0.001t}$ is the learning rate, t – training step, $\theta(x_i, \vartheta, t) = e^{-d^2/\sigma(t)}$ – the value of the neighborhood function between best matching unit vector ϑ and current neuron number i at the training step t, where $d = \left\| r_\vartheta - r_{x_i} \right\|$ is the distance between the coordinates of best matching unit r_ϑ and current one r_{x_i}, $\sigma(t) = n^{2-0.002t}$ is the radius of Gauss function, x – current input vector.

4. Repeat step 2 for all features vector increasing t each time.

One of the most important parameters to be set up for the usage of the SOM is the size of a one-dimensional or two-dimensional map. In this work, we used a one-dimensional map with 20 possible clusters to reach the required real time performance. Our experiments showed that the bigger quantity of clusters requires more training time, the lesser quantity doesn't catch the difference between frames well enough. The successful choice of this parameter allows to balance the quality and the performance of the method being proposed.

The training stage of the SOM continues until all clusters are trained and the quantity of already processed feature vectors is less than 1000.

4 Features and keyframes selection

4.1 Image features

The selection of features is an important step to represent the specific properties of each frame. The best choice from our point of view is such features, which can be calculated quickly and/or in parallel to preserve real-time processing.

We selected the common color features, obtained from floating non-intersection windowing of an image. Averaged R, G and B color components in range [0;1] are used as features from the single window.

We used square windows with size $w = 16$ or $w = 32$ in experiments and rescaled images accordingly to preserve the same length of features vector. The processing of building a feature vector for the entire image was performed in two parallel independent threads and merged at the end.

4.2 The selection of keyframes

We will define the keyframe as a frame that varies slightly during some quantity T of previously examined frames in a video stream. We define the quantitative measure of the variability in the two-step procedure.

The first step includes the clustering of a frame by SOM. So, we cluster the video stream frame by frame, counting the quantity of frames Q, belonging to the same cluster in a row. When this quantity becomes bigger than the predefined threshold T_0, we consider this part of a video stream as stable enough and select keyframe candidate from the middle with index $k^* = i - Q/2$, where i – is the number of the frame being processed.

At the next step we compare keyframe candidate k^* with previously added frame in order to avoid the addition of the very similar, belonging to the same cluster but having some frame with another cluster in between accidently. If frame candidate k^* and previously added frame k^*_{prev} are similar, we replace previous frame with a frame from an updated index: $k^* = (i - Q/2 - k^*_{prev})/2 + k^*_{prev}$. If they are not similar, candidate frame k^* is confirmed as keyframe.

Measuring the difference between frames k_1 and k_2 is based on the average difference between corresponding feature vectors f^1 and f^2 in LAB color space (c^1 and c^2):

$$d = \frac{1}{m}\sum_{i=1}^{m} \Delta E_{00}(c_i^1, c_i^2) \tag{2}$$

where $\Delta E_{00}(c_i^1, c_i^2)$ – is the CIEDE2000 difference between two colors [25]. We claim images to be similar if $d \le 1.5$, where 1.5 is the just noticeable difference (JND) for this metric according to [26].

So, we present the entire scheme of the suggested summarization method in Fig. 1. We denote as $T_0 = 42$ the quantity of frames, required to be classified as the same class in a row with SOM.

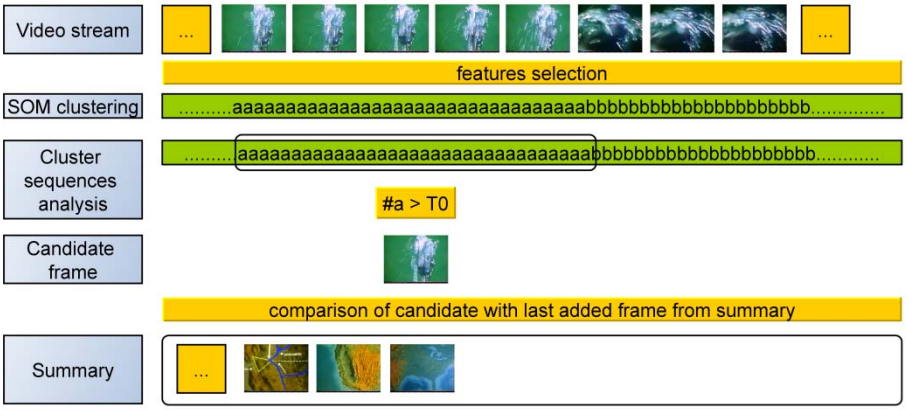

Fig. 1. The whole scheme of the proposed summarization method

5 Image matching

In order to check the quality of summarizing with the suggested approach we need to compare two sets of images, the first one contains images from our automatic summary, the second (ground truth or etalon) one contains images, proposed by humans.

We implemented the match between two sets in three consecutive stages.

The first stage includes the rough estimate of match candidate images with Fast Retina Keypoint (FREAK, [27]) descriptor. FREAK is built on FAST keypoint detector [28] and requires initial threshold t to be set up as a required difference between a central pixel and surrounding to identify central pixel as a corner. The bigger value t leads to the less quantity of keypoint being detected. We found $t = 20$ to be a good default value for experiments.

FREAK descriptors contain 512 bits, we compare them in cascades 128 bits each as proposed in [27]. A comparison of chains requires another threshold F_0, which allows some differences to appear, as the same bit arrays even for close images is the rare case. If corresponding bit chains in descriptors have more differences compared to the threshold, descriptors are considered being different and comparison stops. In our experiments we found threshold $F_0 = 32$ to be the good choice (25% bit values may differ between descriptors).

We suggest that two images probably match if the overall quantity of matched descriptors between them is greater than the quantity of non-matched.

When the list of matching candidates is ready, we compare each pair of candidates using comparison (2). We perform these comparisons on feature vectors, built on full size images with window size $w = 32$.

The impact of different F_0 values on the matching results is shown in Fig. 2. The first row contains some frames from the user summary. Three rows below contain some frames from an automatic summary and a corresponding F_0 value (24, 32 or 40). As one can see from Fig. 2, we have one successful match for $F_0 = 24$, three correct matches for $F_0 = 32$. In the last case when $F_0 = 40$ we have five matches, one of which is absolutely false (third frame) and one is quite subjective to make the decision about the match (last frame).

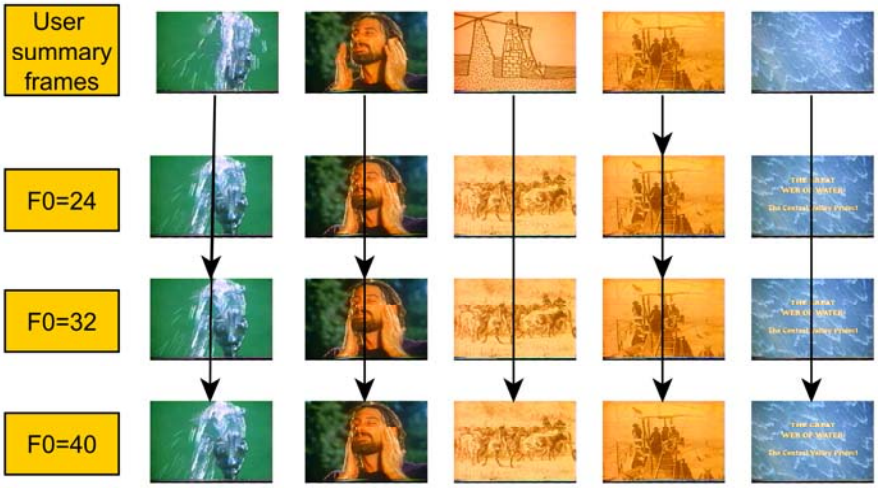

Fig. 2. Frames matching with different F_0 values

The last stage is independent of the previous ones and is used for the frames, for which FREAK descriptor is not effective enough. We process the closest (in the scope of Manhattan distance between frame numbers) frames from automatic and user summaries, which were not matched before. Some frames may be skipped earlier because they contain not enough FREAK descriptors. We analyze them with searching the difference between average R, G and B values of the entire full size image. Images are assumed to be similar if the average difference of all three R, G and B is less than 20.

6 Measuring quality of the automatic summary

We will estimate the quality of the suggested approach with measuring the CUS values like it was suggested in [7] and F1 scores, like it is described in detail in [18] and applied in [11, 12] and others.

Two summary quality CUS_A (accuracy) and CUS_E (error) metrics are proposed in [7]:

$$CUS_A = \frac{n_{mAS}}{n_{US}},\ CUS_E = \frac{\bar{n}_{mAS}}{n_{US}}, \tag{3}$$

where n_{mAS} is the quantity of matching keyframes from automatic summary, $\bar{n}_{mAS}$ is the quantity of non-matching keyframes from automatic summary, n_{US} is the total quantity of keyframes from user summary.

$F1$ score is calculated as $F1 = 2PR/(P+R)$, where P is the precision, F is the recall, calculated on the true positives (quantity of matched frames), false positives (quantity of frames which are present if automatic summary but absent in user one), false negatives (quantity of frames which are present in user summary but missing in automatic).

7 Experiments

SumMe [10] and TVSum [6] are the most popular datasets in video summarizing, but not suitable for us. They have importance assigned to each frame as a result of the user summary performed on the entire video. This is convenient to build etalon summaries from this user summaries just solving 0/1 knapsack problem for a fixed length of the summary, typically 15%. In our case we make the decision in online mode, so we don't know the final length naturally. Despite we could extend out an automatic summary to make the required length, criteria of frame selection in batch and online modes are very different.

So, we used dataset, gathered from Open Video dataset by [7, 16], that contains 50 color videos in MPEG-1 format (30 fps, 352x240 pixels) approximately 75 minutes in total.

User summaries were generated by 2 volunteers in online mode. We asked them to select important (from their point of view, no explanations required) frames while looking video-sequences without sound. The modification of previously selected summary frames was not allowed. Users made the decision about the importance of the frame without knowing the content of the entire video.

The specific of human behavior, called chronological bias, is described in [6]. The authors say that humans sometimes claim frames that appear earlier to be more important just because of chronologically, regardless of frame content. This effect leads to the selection of very close frames as important ones. To avoid the influence of chronological bias, we apply the elimination of duplicates in user summaries applying matching of frames, described above.

After we compared automatic summaries using the proposed approach with user summaries, created by volunteers in online mode and cleaned from duplicates, we got such average values: $CUS_A = 0{,}58$, $CUS_E = 0{,}38$, $F1 = 0{,}61$. Feature vector of length 210 was built on the size of images that two times less than original ones.

We have recalculated these scores for OV [16, 29], DT [30] and VSUMM [7] methods using our matching approach, results are presented in Table 1 and they are close to ones, shown in [7]. As one can see, our approach has much bigger CUS_E value compared to the same value for our user summaries, created in online mode ($CUS_E = 0{,}38$). That means, that the rules for creation of user summaries (online without seeing the entire video or the selection of keyframes after watching entire video) matters.

Table 1. CUS_A, CUS_E and $F1$ scores for the known approaches, suggested matching and known user summaries

Method	CUS_A	CUS_E	$F1$
OV	0,63	0,61	0,56
DT	0,44	0,37	0,48
VSUMM1	0,76	0,47	0,71
VSUMM2	0,62	0,35	0,65
Our (compared to existing summaries)	0,58	0,53	0,57
Our (compared to our user summaries, created in online fashion)	0,58	0,38	0,61

The total duration of the 50 videos [16] from Open Video dataset is about 75 minutes, we built the summary for all videos in 17 minutes. The average length of the summary is 0.3% of the entire video. One-dimensional map with 20 clusters was used.

We used SumMe [10] dataset to evaluate the processing speed. The duration of 25 videos from the SumMe dataset is approximately 70 minutes, time of processing, the quantity of keyframes in summary and length of the features vector are shown in Table 2.

First column contains the name of video and its duration in minutes and seconds. Values in the second column were calculated for the frames with size two times less, than original, in the third column – three times less. Values in the last column were calculated with specific downscaling factor for each video that limits the quantity of features (maximum allowed width of the frame is 200, height is 140). The empty value in the last column means that the result of processing corresponds to one of the values from previous columns.

Values in the last column in Table 2 show the ratio between the best processing time and the total time of a video. The processing of four videos (Air_Force_One, Eiffel Tower, Notre_Dame and Scuba) didn't satisfy real time requirement. The length of the summary decreases significantly with decreasing of feature vector length for one video (Bearpark_climbing).

Table 2. The performance of summarization for videos of SumMe dataset

Video (duration, m:s)	Time / Summary length / features vector length (0.5)	Time / Summary length / features vector length (0.33)	Time / Summary length / features vector length (200x140)	Ratio
Air_Force_One (3:00)	12:10 / 2 / 5841	7:47 / 2 / 2574	4:04 / 2 / 273	1.35
Base jumping (2:39)	2:38 / 19 / 1092	1:51 / 21 / 459	1:28 / 19 / 273	0.55
Bearpark_climbing (2:14)	4:11 / 8 / 2574	2:47 / 8 / 1092	1:46 / 3 / 273	0.79
Bike Polo (1:43)	3:58 / 11 / 2574	2:42 / 11 / 1092	1:42 / 11 / 273	0.99
Bus_in_Rock_Tunnel (2:51)	1:29 / 8 / 627	1:04 / 9 / 273	-	0.37
Car_over_camera (2:26)	1:46 / 4 / 798	1:08 / 4 / 251	1:08 / 4 / 351	0.46
Car_railcrossing (2:49)	6:06 / 8 / 2574	4:03 / 12 / 1092	2:37 / 13 / 273	0.93
Cockpit_Landing (5:02)	9:30 / 19 / 2574	6:52 / 16 / 1092	4:28 / 13 / 273	0.89
Cooking (1:27)	0:09 / 6 / 189	0:06 / 4 / 72	0:25 / 6 / 798	0.07
Eiffel Tower (3:20)	13:57 / 26 / 5841	9:35 / 25 / 2574	5:04 / 23 / 273	1.52
Excavators river crossing (6:29)	3:06 / 37 / 627	2:11 / 34 / 273	-	0.34
Fire Domino (0:55)	0:25 / 7 / 462	0:17 / 5 / 189	-	0.31
Jumps (0:39)	0:12 / 4 / 336	0:08 / 4 / 135	-	0.21
Kids_playing_in_leaves (1:46)	4:01 / 5 / 2574	2:42 / 5 / 1092	1:38 / 6 / 273	0.92
Notre_Dame (3:12)	13:09 / 18 / 5841	8:49 / 18 / 2574	4:40 / 15 / 273	1.46
Paintball (4:16)	7:13 / 5 / 2574	4:43 / 2 / 1092	2:48 / 5 / 273	0.66
Paluma_jump (1:26)	1:06 / 1 / 798	0:46 / 1 / 351	-	0.53
Playing_ball (1:44)	0:48 / 12 / 462	0:34 / 13 / 189	-	0.33

Playing_on_water_slide (1:42)	0:46 / 12 / 462	0:31 / 12 / 189	-	0.30
Saving dolphines (3:43)	6:06 / 2 / 1914	4:08 / 2 / 798	2:44 / 2 / 264	0.74
Scuba (1:14)	2:58 / 10/ 2574	1:59 / 8 / 1092	1:15 / 8 / 273	1.01
St Maarten Landing (1:10)	1:02 / 3/ 1092	0:42 / 2 / 459	0:34 / 3 / 273	0.49
Statue of Liberty (2:36)	1:36 / 7 / 798	1:04 / 7 / 351	-	0.41
Uncut_Evening_Flight (5:23)	10:32 / 50 / 2574	7:46 / 48 / 1092	5:04 / 47 / 273	0.94
Valparaiso_Downhill (2:53)	6:27 / 13 / 2574	4:10 / 13 / 1092	2: 28 / 12 / 273	0.86

We tested also the performance of the proposed online summarization method on the two long movies. First one contains 260222 frames and lasts 3 hrs. 00 min. 53 sec. (10853 seconds in total), the second one has 203882 frames and lasts 2 hrs. 21 min. and 43 sec. (8503 seconds in total).

The online summarization of the first movie took 3800 seconds for 336 features per frame, the length of the summary was 705 frames. Second video required 2209 seconds to build summary containing 585 frames using 252 features per frame.

8 Conclusion

We proposed an approach to generate the video summary in the form of keyframes, which are still images. It is based on the clustering of separate frames in online mode without the analysis of the whole video using Kohonen's self-organized maps. The frame from the video stream is selected as a summary frame if some quantity of frames $T_0 / 2$ after it and before were classified as the same cluster.

The proposed method was tested on Open Video dataset and the performance is tested on SumMe dataset. We showed that the quality is comparable to some batch video summarization methods, and the performance combined with the flexibility in the selection of the quantity of frame features allows achieving real time processing in most cases. The quality of the suggested method is better compared to the user summary created in online mode.

The investigation of the dependency of quality and performance on the size of SOM map may be the topic of future research.

References

1. Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J.: Rethinking the Evaluation of Video Summaries. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15-20 June 2019, pp. 7588-7596 (2019) doi: 10.1109/CVPR.2019.00778
2. Panda, R., Das, A., Wu, Z., Ernst, J., Roy-Chowdhury, A.K.: Weakly Supervised Summarization of Web Videos. In: 2017 IEEE International Conference on Computer Vision (ICCV), 22-29 October 2017, pp. 3677-3686 (2017)
3. Pan, G., Zheng, Y., Zhang, R., Han, D., Sun, D., Qu, X.: A bottom-up summarization algorithm for videos in the wild. EURASIP Journal on Advances in Signal Processing, vol. 2019, #15 (2019) doi: 10.1186/s13634-019-0611-y
4. Rochan, M., Wang, Y.: Video Summarization by Learning from Unpaired Data. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15-20 June 2019, pp. 7894-7903 (2019) doi: 10.1109/CVPR.2019.00809
5. Cai, S., Zuo, W., Davis, L.S., Zhang, L.: Weakly-supervised Video Summarization using Variational Encoder-Decoder and Web Prior. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol. 11218, pp. 193-210 (2018) doi: 10.1007/978-3-030-01264-9_12
6. Song, Y., Vallmitjana, J., Stent, A., Jaimes A.: TVSum: Summarizing Web Videos Using Titles. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7-12 June 2015, pp. 5179-5187 (2015) doi: 10.1109/CVPR.2015.7299154
7. De Avila, S. E. F., Lopes, A. P. B., da Luz, A. Jr., Araujo, A.D.L.: Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters, vol. 32, no. 1, pp. 56-68 (2011)
8. Zhao, B., Xing E.P.: Quasi Real-Time Summarization for Consumer Videos. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 23-28 June 2014, pp. 2513-2520 (2014) doi: 10.1109/CVPR.2014.322
9. Gygli, M., Grabner, H., Van Gool L.: Video summarization by learning submodular mixtures of objectives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7-12 June 2015, pp. 3090–3098 (2015) doi: 10.1109/CVPR.2015.7298928
10. Gygli, M., Grabner, H., Riemenschneider, H., Van Gool L.: Creating summaries from user videos. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8695, pp. 505–520 (2014) doi: 10.1007/978-3-319-10584-0_33
11. Zhang, K., Chao, WL., Sha, F., Grauman, K.: Video Summarization with Long Short-Term Memory. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol. 9911, pp. 766–782 (2016) doi: 10.1007/978-3-319-46478-7_47
12. Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarization with adversarial LSTM networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July 2017, pp. 1–10 (2017) doi: 10.1109/CVPR.2017.318
13. Koprinska, I., Clark, J.: Video Summarization and Browsing Using Growing Cell Structures. In: International Joint Conference on Neural Networks (IJCNN), 25-29 July 2004, pp. 2601-2606 (2004) doi: 10.1109/IJCNN.2004.1381056
14. Koprinska, I., Clark, J., Carrato, S.: VideoGCS - A Clustering-Based System for Video Summarization and Browsing. In: The Proceedings of the 6th COST 276 Workshop, Thessaloniki, Greece, May 2004, pp. 34-40 (2014)

15. Baerecke, T., Kijak, E., Nuernberger, A., Detyniecki, M.: Summarizing Video Information Using Self-Organizing Maps . In: 2006 IEEE International Conference on Fuzzy Systems, 16-21 July 2006 (2006) doi: 10.1109/FUZZY.2006.1681764
16. VSUMM (Video SUMMarization). URL: https://sites.google.com/site/vsummsite/home. Accessed: 01 March 2020.
17. Zhang, Y., Kampffmeyer, M., Liang, X., Zhang, D., Tan, M., Xing, E.: Dilated Temporal Relational Adversarial Network for Generic Video Summarization. Multimedia Tools and Applications , vol. 78, pp. 35237-35261 (2019) doi:10.1007/s11042-019-08175-y
18. Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D., Remagnino, P.: Summarizing Videos with Attention. In: Carneiro G., You S. (eds) Computer Vision – ACCV 2018 Workshops. ACCV 2018. Lecture Notes in Computer Science, vol 11367, pp. 39-54 (2019) doi: 10.1007/978-3-030-21074-8_4
19. Ji, Z., Xiong, K., Pang, Y., Li, X.: Video Summarization with Attention-Based Encoder-Decoder Networks. URL: https://arxiv.org/pdf/1708.09545.pdf. Accessed: 01 March 2020.
20. Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., Yokoya, N.: Video Summarization Using Deep Semantic Features. In: Lai SH., Lepetit V., Nishino K., Sato Y. (eds) Computer Vision – ACCV 2016. ACCV 2016. Lecture Notes in Computer Science, vol. 10115, pp. 361-377 (2017) doi: 10.1007/978-3-319-54193-8_23
21. Plummer, B.A., Brown, M., Lazebnik, S.: Enhancing Video Summarization via Vision-Language Embedding. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July 2017, pp. 1052-1060 (2017) doi: 10.1109/CVPR.2017.118
22. Pournazari, M., Mahmoudi, F., Moghadam, A. M. E.: Video Summarization Based on a Fuzzy Based Incremental Clustering. International Journal of Electrical and Computer Engineering (IJECE), vol. 4, No. 4, pp. 593-602 (2014)
23. Marvaniya, S., Damoder, M., Gopalakrishnan, V., Iyer, K. N., Soni, K.: Real-time video summarization on mobile. In: 2016 IEEE International Conference on Image Processing (ICIP), 25-28 September 2016, pp. 176-180 (2016) doi: 10.1109/ICIP.2016.7532342
24. Kohonen, T.: Self-Organizing Maps. Berlin: Springer-Verlag (1995)
25. Sharma, G., Wu, W., Dalal, E. N.: The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations. Color Research & Application, vol. 30(1), pp. 21-30 (2004) doi:10.1002/col.20070
26. Yang, Y., Ming, J., Yu, N.: Color Image Quality Assessment Based on CIEDE2000. Advances in Multimedia, vol. 2012 (ID of paper 273723) (2012) doi: 10.1155/2012/273723
27. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 16-21 June 2012, pp. 510-517 (2012) doi: 10.1109/CVPR.2012.6247715
28. Rosten, E., Drummond, T: Machine Learning for High-Speed Corner Detection. In: Leonardis A., Bischof H., Pinz A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol. 3951, pp. 430-443 (2006) doi: 10.1007/11744023_34
29. DeMenthon, D., Kobla, V., Doermann, D.: Video summarization by curve simplification. In: Proceedings of the sixth ACM international conference on Multimedia, September 1998, pp. 211-218 (1998) doi: 10.1145/290747.290773
30. Mundur, P., Rao, Y., Yesha, Y.: Keyframe-based video summarization using Delaunay clustering. International Journal on Digital Libraries, vol. 6, no. 1, pp. 219-232 (2006) doi: 10.1007/s00799-005-0129-9