A Video Library System Using Scene Detection and Automatic Tagging

Lorenzo Baraldi(✉), Costantino Grana, and Rita Cucchiara

Dipartimento di Ingegneria "Enzo Ferrari", Università degli Studi di Modena e Reggio Emilia, Via Vivarelli 10, 41125 Modena, MO, Italy
{lorenzo.baraldi,costantino.grana,rita.cucchiara}@unimore.it

Abstract. We present a novel video browsing and retrieval system for edited videos, based on scene detection and automatic tagging. In the proposed system, database videos are automatically decomposed into meaningful and storytelling parts (i.e. *scenes*) and tagged in an automatic way by leveraging their transcript. We rely on computer vision and machine learning techniques to learn the optimal scene boundaries: a Triplet Deep Neural Network is trained to distinguish video sequences belonging to the same scene and sequences from different scenes, by exploiting multimodal features from images, audio and captions. The system also features a user interface build as a set of extensions to the eXo Platform Enterprise Content Management System (ECMS) (https://www.exoplatform.com/). This set of extensions enable the interactive visualization of a video, its automatic and semi-automatic annotation, as well as a keyword-based search inside the video collection. The platform also allows a natural integration with third-party add-ons, so that automatic annotations can be exploited outside the proposed platform.

Keywords: Scene detection · Tagging · Video browsing · Interfaces

1 Introduction

Video is currently the largest source of internet traffic: after having been used mainly for fun and entertainment in the last decades, it is now employed into novel contexts, like social networks, online advertisement, and education.

Unfortunately, the vast majority of video content available on the web is not provided with annotations, and is therefore cumbersome to retrieve. Furthermore, video browsing platforms like Youtube, Facebook and Dailymotion treat the video as an indivisible entity, so that the user receives no help in finding the portion of the video that really interests him. The user must either watch the entire video or move from one portion of the video to another through seek operations. In the case of educational video clips, which are usually longer than the average user-generated video, this becomes even more evident, and fining a short segment on a specific topic of interest often becomes intractable.

In this paper, we propose a system which tries to address this limitation by exploiting computer vision and machine learning techniques. In particular,

C. Grana and L. Baraldi (Eds.): IRCDL 2017, CCIS 733, pp. 56–67, 2017.
DOI: 10.1007/978-3-319-68130-6_5

we rely on scene detection, a pattern recognition technique which enables the decomposition of a video into semantically coherent parts. In the case of scene detection, therefore, the objective is that of automatically segmenting an input video into meaningful and story-telling sequences, using perceptual and semantic features and exploiting editing rules or clustering algorithms. It is straightforward to see that scene detection can enhance video access and browsing, as it transforms a long video into a sequence of homogeneous parts. Also, it enables a fine-grained search inside the video itself, with which short sequences could be more easily retrieved with textual queries. Finally, it is worth to mention that sequences from the original video can also be exploited to create presentations or video lectures, thus enhancing the re-use of video collections.

In the case of broadcast edited videos, scenes represent one of the three levels at which the video can be decomposed. Edited videos are indeed made by sequences of shots, which in turn are frames taken by the same camera. Shots can then be grouped, according to their semantic meaning, into scenes. From this perspective, scene detection can be seen as the task of grouping temporally adjacent shots, with the objective of maximizing the semantic coherence of the resulting segments. The computer vision pipeline which will be described in the rest of this paper relies on this assumption, by creating an embedding space in which shots can be projected according to their relative semantic similarity, and exploiting a temporal clustering technique which is in charge of defining the final temporal segmentation of the video. We should also notice that using shots instead of the single frames as the basic unit of computation enables a reduction in computational complexity, and at the same time assures a quasi-optimal decomposition of the video, since shots usually have a uniform semantic content. This granularity level is also beneficial for visualization, since representative keyframes can be extracted from shots.

Using a scene detection algorithm that we have recently proposed in literature [6], and thanks to the application of Speech-to-Text techniques, it has been possible to automatically annotate a set of 500 educational broadcast videos taken from the large Rai Scuola archive[1]. Also, we developed a browsing and retrieval interface on top of a commercial ECMS, namely eXo Platform, from which the results of the automatic annotation can be browsed and manually refined. The interface has been developed as part of the Città Educante project, cofunded by the Italian Ministry of Education, with the aim of providing new technologies for education.

The rest of this paper is organized as follows: in Sect. 2 we give an overview of relevant works related to the topic of this paper; Sect. 3 describes the main algorithmic components of the system, by giving details on the scene detection approach and on the retrieval strategy we employ, while showing several examples and screenshots from the actual user interface. Finally, Sect. 4 draws the conclusion of the work.

[1] www.raiscuola.rai.it.

2 Related Work

In this section, we give a brief overview of the works related with the two main features of our system, namely video decomposition and video retrieval.

Video Decomposition. Since more than 15 years, automatic video decomposition has been categorized into three categories [9]: *rule-based methods*, that consider the way a video is structured in professional movie production, *graph-based methods*, where shots are arranged in a graph representation, and *clustering-based methods*.

Rule-based approaches consider the way a scene is structured in professional movie production. Of course, the drawback of this kind of methods is that they tend to fail in videos where film-editing rules are not followed, or when two adjacent scenes are similar and follow the same rules. Liu *et al.* [13], for example, propose a visual based probabilistic framework that imitates the authoring process. In [8], shots are represented by means of key-frames, clustered using spectral clustering and low level color features, and then labeled according to the clusters they belong to. Scene boundaries are then detected from the alignment score of the symbolic sequences, using the Needleman-Wunsch algorithm.

In graph-based methods, instead, shots are arranged in a graph representation and then clustered by partitioning the graph. The Shot Transition Graph (STG) [22] is one of the most used models in this category: here each node represents a shot and the edges between the shots are weighted by shot similarity. In [16], color and motion features are used to represent shot similarity, and the STG is then split into subgraphs by applying the normalized cuts for graph partitioning. Sidiropoulos *et al.* [18] introduced a STG approximation that exploits features from the visual and the auditory channel.

Clustering-based solutions assume that similarity of shots can be used to group them into meaningful clusters, thus directly providing the final story boundaries. With this approach, a deep learning based strategy has recently been proposed [4]. In this model, a Siamese Network is used together with features extracted from a Convolutional Neural Network and time features to learn distances between shots. Spectral clustering is then applied to detect coherent sequences.

Video Retrieval. Lot of work has also been proposed for video retrieval: with the explosive growth of online videos, this has become a hot topic in computer vision. In their seminal work, Sivic *et al.* proposed Video Google [20], a system that retrieves videos from a database via bag-of-words matching. Lew *et al.* [11] reviewed earlier efforts in video retrieval, which mostly relied on feature-based relevance feedback or similar methods.

More recently, concept-based methods have emerged as a popular approach to video retrieval. Snoek *et al.* [21] proposed a method based on a set of concept detectors, with the aim to bridge the semantic gap between visual features and high level concepts. In [3], authors proposed a video retrieval approach based

on tag propagation: given an input video with user-defined tags, Flickr, Google Images and Bing are mined to collect images with similar tags: these are used to label each temporal segment of the video, so that the method increases the number of tags originally proposed by the users, and localizes them temporally. In [12] the problem of retrieving videos using complex natural language queries is tackled, by first parsing the sentential descriptions into a semantic graph, which is then matched to visual concepts using a generalized bipartite matching algorithm. This also allows to retrieve the relevant video segment given a text query. Our method, in contrast to [3], does not need any kind of initial manual annotation, and, thanks to the availability of the video structure, is able to return specific stories related to the user query. This provides the retrieved result with a context that allows to better understand the video content.

3 The System

The proposed system is comprises three main components: a scene detection algorithm, which is in charge of performing a temporal segmentation of the input video into coherent parts, an automatic tagging algorithm, and a retrieval module, with which users can search for scenes inside a video collection.

3.1 Scene Detection

The decomposition of a video into semantically coherent parts is an intrinsic multi-modal task, which cannot be solved by applying heuristic rules, or a-priori defined models due to the variety of boundaries which can be found in professionally edited video. The definition of a hand-crafted rules would indeed be very time consuming, and would probably lead to poor results in terms of localization accuracy. We therefore choose to rely on machine learning, and to build a deep learning architecture for temporal video segmentation which can learn the optimal way of segmenting the video by learning from examples annotated by different users. On a different note, to tackle the multi-modal nature of the problem, we employ a combination of multi-modal features which range from the frames and the audio track of the video, to the transcript of the speaker.

The video is firstly decomposed into a set of chunks taken by the same camera (i.e. shots), using an open source shot detector [1]. Given that the content of a shot is usually uniform from a semantic point of view, we can constrain scene boundaries to be a subset of shot boundaries, therefore reducing the problem of scene detection to that of clustering adjacent shots. This preliminary decomposition also reduces the computational efforts needed to process the entire video, given that few key-frames can be used as the representative of the whole shot. Similarly, features coming from other modalities can be encoded at the shot level by following the same homogeneity assumption. For each shot of the video, we extract different features, in order to take into account all the modalities present in the video.

Visual Appearance Features. We encode the visual appearance of a shot, and information about the timestamp and the length of a given shot. Visual appearance is extracted with a pre-trained Convolutional Neural Network which is shortened by cutting out the last fully connected layers. This extracts high level features from the input image, which can be a rich source of information to identify changes in visual content between one portion of the video and another. Given that a single key-frame might be too poor to describe a shot, we uniformly sample three key-frame from the input shot, and take the pixelwise maximum of the network responses.

Visual Concept Features. Using a part-of-speech tagger, we parse the transcript obtained with standard speech-to-text techniques, and retain unigrams which are annotated as *noun*, *proper noun* and *foreign word*. Those are then mapped to the Imagenet corpus by means of a skip-gram model [15] trained on the dump of the Italian Wikipedia. By means of this mapping we build a classifier to detect the presence of a visual concept in a shot. Images from the external corpus are represented using feature activations from pre-trained deep convolutional neural networks (CNN), which can extract rich semantic information from an input image [10]. In our case, we employ the VGG-16 model [19], which is well known for providing state-of-the-art results on image classifications, and for its good generalization properties [17]. Then, a linear probabilistic SVM is trained for each concept, using randomly sampled negative training data; the probability output of each classifier is then used as an indicator of the presence of a concept in a shot.

Keeping a shot-based representation, we build a feature vector which encodes the influence of each concept group on the considered shot. Formally, the visual concept feature of shot s, $\mathbf{v}(s)$, is a K-dimensional vector, defined as

$$\mathbf{v}(s) = \left[\sum_{t \in M(\mathcal{T})} \delta_{t,i} \cdot f_{M(t)}(s) e^{-\frac{(u_t - u_s)^2}{2\sigma^2}} \right]_{i=1,\dots,K} \tag{1}$$

where $\mathcal{T}$ is the multiset of all terms inside a video, $\delta_{t,i}$ indicates whether term t belongs to the i-th concept group, u_t and u_s are the timestamps of term t and shot s. M is the mapping function to the external corpus, and $f_{M(t)}(s)$ is the probability given by the SVM classifier trained on concept $M(t)$ and tested on shot s.

Textual Concept Features. Textual concepts are as important as visual concepts to detect story changes, and detected concept groups provide an ideal mean to describe topic changes in text. Therefore, a textual concept feature vector, $\mathbf{t}(s)$, is built as the textual counterpart of $\mathbf{v}(s)$

$$\mathbf{t}(s) = \left[\sum_{t \in \mathcal{T}} \delta_{t,i} \cdot e^{-\frac{(u_t - u_s)^2}{2\sigma^2}} \right]_{i=1,\dots,K} \tag{2}$$

We thus get a representation of how much each concept group is present in a shot and in its neighborhood.

Audio Features. Audio is another meaningful cue for detecting scene boundaries, since audio effects and soundtracks are often used to underline the development of a scene or a change in content. We extract MFCCs descriptors [14] over a 10 ms window. The MFCC descriptors are aggregated by Fisher vectors using a Gaussian Mixture Model with 256 components.

Multi-modal Fusion. The overall feature vector for a shot is the concatenation of all the previously defined features. A Triplet Deep Network is then trained on ground-truth decompositions by minimizing a contrastive loss function: at each training iteration, the network processes a triplet of examples, namely an anchor example, a positive and a negative example. The anchor example is randomly selected from the available shots in the database, the positive one is constrained to be part of the same scene of the anchor, and the negative sample is selected from a different scene. Each of the sample is embedded by the same function into a common, multimodal, embedding space. The contrastive loss function then forces the distance between the anchor and the positive shot to be smaller than the distance between the anchor and the negative. This, during training, promotes the creation of an embedding function suitable for the task.

At test time, the network has learned to distinguish similar and dissimilar shots, and can be therefore employed to perform scene detection. In particular, our clustering algorithm relies on the minimization of variances inside each scene. For further details, the reader is encouraged to read the paper in which the technique was proposed [6].

3.2 User Interface

While the temporal segmentation step is carried out off-line, and its results are saved into a database for browsing, we also build an appropriate user interface for visualization. In particular, we extend a popular Enterprise Content Management System (ECMS), namely eXo Platform, which is largely used to build enterprise intranets and dynamic portals. We exploit the extension capabilities of eXo Platform and develop an add-on which can visualize videos decomposed into scenes. Every time a video is uploaded on the platform, a remote web service is called to perform the automatic decomposition of the video into scene, and to extract key-frames for visualization. Each video can then be visualized in a time line fashion, were each is scene is presented by means of the key-frames it contains.

Figures 1 and 2 show two sample screenshot from the proposed interface. As it can be noticed, we built two different views, one showing the list of available videos, each presented with one of its keyframes, and one for the browsing of the video itself, in which the actual temporal segmentation is shown. It can also be noticed that the visualization of each scene is enriched by a set of tags. These

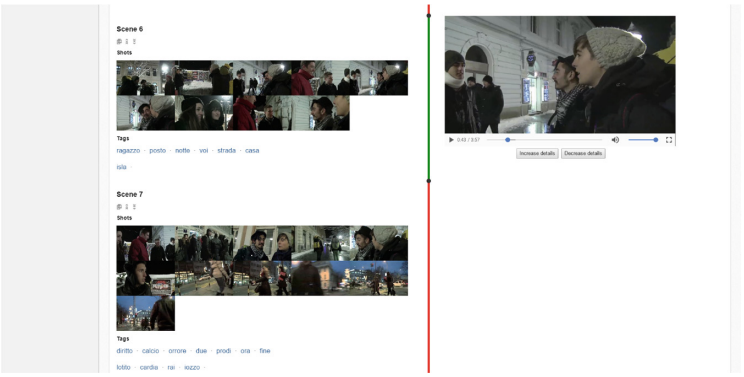

Fig. 1. Sample screenshot of the interactive visualization interface.

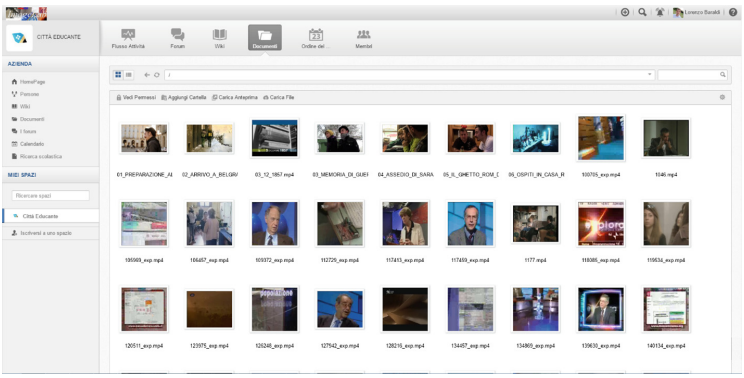

Fig. 2. Visualization of a collection of videos.

are obtained by parsing the transcript of the video, and extracting nouns and proper nouns with a NLP tagger trained for the Italian language.

Of course, even though generally precise, the automatic decomposition in scene might not always be correct, or appreciated by the final user, also considering the subjective nature of the task. Therefore, the output of the algorithm can not always be satisfactory for the user. For this reason, the interface allows the user to refine the automatic annotation, merging adjoining scenes together. Data collected from this manual annotation feature could be exploited in further works, both to extend the training set and to use it an a relevance feedback loop.

3.3 Retrieval

The ability to index parts of a video is an essential feature of a video browsing platform, as it enables a fine-grained search which is also important for video re-use. In developing this feature, we wanted video clips to be indexed at the

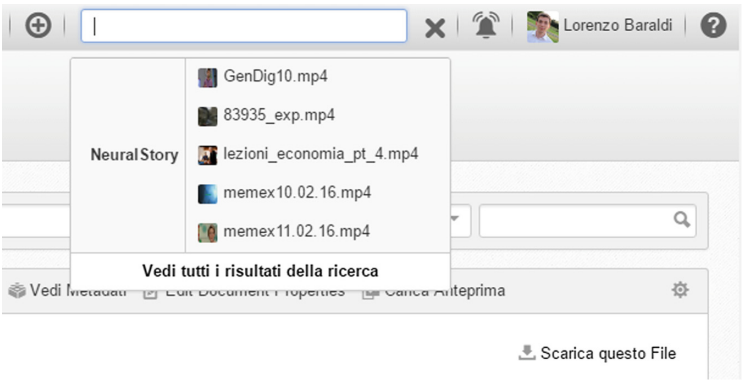

Fig. 3. eXo Platform search form, enhanced with results from the video collection.

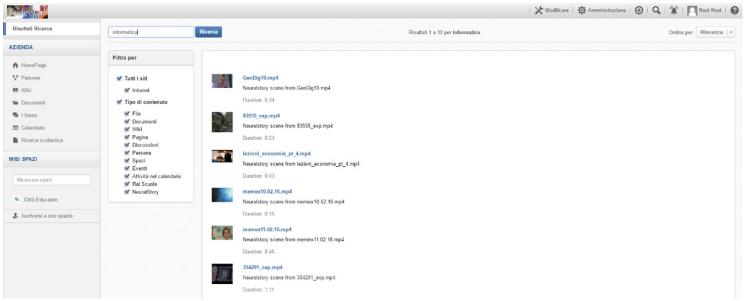

Fig. 4. Screenshot of the results page.

scene level, so that users can search inside video clips and not only among video clips. Secondly, we constrained the indexing system to be fully automatic, and therefore chose to rely on the transcription solely, rather than exploiting user-generated annotations. This allows our retrieval strategy to be enough precise, and content oriented, as most of the extracted keywords will focus on the topics addressed during the video, rather than on what is actually shown in the video.

From an implementation point of view, we extended the built-in search capabilities of eXo, by developing a component which can search inside the video collection, given a textual query. This is done by building a Search Connector component, which is called by eXo itself every time a user performs a textual search. The Connector, in turn, searches for the given query inside the video database, and matches the query with the available tags. Of course, more sophisticated techniques could be used, even though they are outside the scope of this paper.

For each retrieved scene, a thumbnail is also selected among the key-frames of the video by using a semantic and aesthetic criterion [5], so that the user can be confident about the result of his research by simply looking at the provided

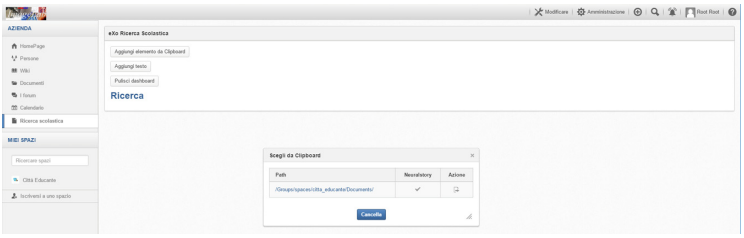

Fig. 5. Incorporation of a scene in a third-party extension (selection from the clipboard).

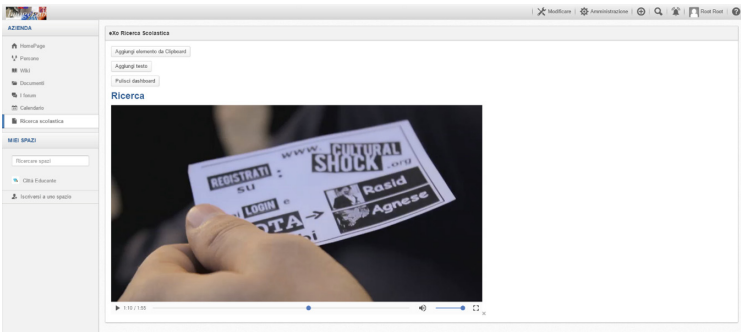

Fig. 6. Incorporation of a scene in a third-party extension.

thumbnails. Moreover, since results are presented in terms of scenes, the user is redirected directly to the video portion of interest for his query, without the need to search within the video of interest. Figures 3 and 4 show, respectively, the search interface and the results page.

3.4 Integration with Third-Party Add-Ons

One key feature of the system is the possibility to integrate our temporal segmentation, retrieval and visualization tools into third part components. This makes the developed system integrable with other software, which is a desirable property in most of the use cases, ranging from portals to project demonstrators.

Indeed, all the extensions which have been presented in the previous sections are designed to be naturally integrated with the underlying data structure of eXo, namely the Java JCR, and with third-party applications. In particular, full-length videos or portions of them can be retrieved from the database by any application inside the platform. Also, at an higher level, videos and scene managed by our application can be copied and pasted into the eXo Clipboard. This is a cross-application clipboard, which can be read an written across different applications and add-ons on the platform, and which provides an effective way to transfer content from one application to another.

Finally, for the purpose of demonstration, we developed a simple portlet which can showcase the benefit of using data provided by our algorithm outside the specific user interface we built for visualization. The demo portlet allows the insertion of video portions within a simple canvas, by exploiting the eXo Clipboard. A dialog shows the contents of the clipboard with the selected scenes by the user while navigating in the database: these can then be pasted on the canvas and displayed (Figs. 5 and 6).

Beside this simple example, the same extension capabilities showcased by our application can be used for integrating with other add-ons. In particular, this will also be beneficial in the context of the Città Educante project, in which the eXo Platform will be used as the enabling tool of the final demonstrator [2,7].

4 Conclusion

This paper has presented a video browsing and retrieval system for edited videos, which has been developed in the context of a national project, Città Educante. The main distinguishing feature of the system, with respect to other video browsing approaches, is that videos are automatically parsed and decomposed into meaningful segments (called scenes), by means of a novel scene detection algorithm which exploits state of the art multi-modal descriptors and machine learning techniques. In particular, it relies on a Triplet Deep Network which learns a multi-modal semantic embedding space in which shots from the input video can be projected, and on a temporal clustering algorithm which provides the final segmentation into scenes. The web-based interface enables the interactive visualization of a video, its automatic and semi-automatic annotation, as well as a keyword-based search inside a video collection. Finally, it is worth mentioning that using the proposed algorithm it has been possible to automatically annotate a set of 500 educational broadcast videos taken from the large Rai Scuola archive, which can be browsed and retrieved inside the internal portal of the Città Educante project. As a future work, it will also be possible to exploit the corrections and annotations provided by the users, as a source of additional training data, and to build a human-in-the-loop system which can possibly provide better temporal segmentation results.

References

1. Apostolidis, E., Mezaris, V.: Fast shot segmentation combining global and local visual descriptors. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6583–6587 (2014)
2. Balducci, F., Grana, C., Cucchiara, R.: Affective level design for a role-playing videogame evaluated by a brain-computer interface and machine learning methods. Vis. Comput. **33**(4), 413–427 (2017)
3. Ballan, L., Bertini, M., Serra, G., Del Bimbo, A.: A data-driven approach for tag refinement and localization in web videos. Comput. Vis. Image Underst. **140**, 58–67 (2015)

4. Baraldi, L., Grana, C., Cucchiara, R.: A deep siamese network for scene detection in broadcast videos. In: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference (MM 2015), pp. 1199–1202 (2015). http://doi.acm.org/10.1145/2733373.2806316
5. Baraldi, L., Grana, C., Cucchiara, R.: Scene-driven retrieval in edited videos using aesthetic and semantic deep features. In: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (ICMR 2016), pp. 23–29 (2016). http://doi.acm.org/10.1145/2911996.2912012
6. Baraldi, L., Grana, C., Cucchiara, R.: Recognizing and presenting the storytelling video structure with deep multimodal networks. IEEE Trans. Multimed. **19**(5), 955–968 (2017)
7. Bolelli, F., Borghi, G., Grana, C.: Historical handwritten text images word spotting through sliding window HOG features. In: 19th International Conference on Image Analysis and Processing (2017)
8. Chasanis, V.T., Likas, C., Galatsanos, N.P.: Scene detection in videos using shot clustering and sequence alignment. IEEE Trans. Multimed. **11**(1), 89–100 (2009)
9. Hanjalic, A., Lagendijk, R.L., Biemond, J.: Automated high-level movie segmentation for advanced video-retrieval systems. IEEE Trans. Circ. Syst. Video Technol. **9**(4), 580–588 (1999)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
11. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: state of the art and challenges. ACM Trans. Multimed. Comput. Commun. Appl. (TOMCCAP) **2**(1), 1–19 (2006)
12. Lin, D., Fidler, S., Kong, C., Urtasun, R.: Visual semantic search: retrieving videos via complex textual queries. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 2657–2664, June 2014
13. Liu, C., Wang, D., Zhu, J., Zhang, B.: Learning a contextual/multi-thread model for movie/TV scene segmentation. IEEE Trans. Multimed. **15**(4), 884–897 (2013)
14. Logan, B., et al.: Mel frequency cepstral coefficients for music modeling. In: ISMIR (2000)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
16. Rasheed, Z., Shah, M.: Detection and representation of scenes in videos. IEEE Trans. Multimed. **7**(6), 1097–1105 (2005)
17. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813 (2014)
18. Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., Trancoso, I.: Temporal video segmentation to scenes using high-level audiovisual features. IEEE Trans. Circ. Syst. Video Technol. **21**(8), 1163–1177 (2011)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint (2014). arXiv:1409.1556
20. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: IEEE International Conference on Computer Vision, pp. 1470–1477. IEEE (2003)

21. Snoek, C.G., Huurnink, B., Hollink, L., De Rijke, M., Schreiber, G., Worring, M.: Adding semantics to detectors for video retrieval. IEEE Trans. Multimed. **9**(5), 975–986 (2007)
22. Yeung, M.M., Yeo, B.L., Wolf, W.H., Liu, B.: Video browsing using clustering and scene transitions on compressed sequences. In: IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology, pp. 399–413 (1995)