An Image Retrieval System for Video*

Paolo Bolettieri, Fabio Carrara, Franca Debole, Fabrizio Falchi, Claudio Gennaro, Lucia Vadicamo, and Claudio Vairo

Institute of Information Science and Technologies,
Italian National Research Council (CNR),
Via G. Moruzzi 1, Pisa, Italy
{firstname.surname}@isti.cnr.it

Abstract. Since the 1970's the Content-Based Image Indexing and Retrieval (CBIR) has been an active area. Nowadays, the rapid increase of video data has paved the way to the advancement of the technologies in many different communities for the creation of Content-Based Video Indexing and Retrieval (CBVIR). However, greater attention needs to be devoted to the development of effective tools for video search and browse. In this paper, we present Visione, a system for large-scale video retrieval. The system integrates several content-based analysis and retrieval modules, including a keywords search, a spatial object-based search, and a visual similarity search. From the tests carried out by users when they needed to find as many correct examples as possible, the similarity search proved to be the most promising option. Our implementation is based on state-of-the-art deep learning approaches for content analysis and leverages highly efficient indexing techniques to ensure scalability. Specifically, we encode all the visual and textual descriptors extracted from the videos into (surrogate) textual representations that are then efficiently indexed and searched using an off-the-shelf text search engine using similarity functions.

Keywords: content-based image indexing · neural networks · multimedia retrieval· similarity search · object detection

1 Introduction

Video data is the fastest growing data type on the Internet, and because of the proliferation of high-definition video cameras, the volume of video data is exploding. Visione [1] is a content-based video retrieval system that participated for the first time in 2019 to the Video Browser Showdown (VBS) [11], an international video search competition that evaluates the performance of interactive video retrievals systems. The VBS 2019 uses the V3C1 dataset that consists of 7,475 video files, amounting for 1000h of video content (1082659 predefined segments) [15] and encompasses three content search tasks: *visual Known-Item Search (visual KIS)*, *textual Known-Item Search (textual KIS)* and *ad-hoc Video*

* Work partially supported by the AI4EU project (EC-H2020 - Contract n. 825619)

Search (AVS). The visual KIS task models the situation in which someone wants to find a particular video clip that he has already seen, assuming that it is contained in a specific collection of data. In the textual KIS, the target video clip is no longer visually presented to the participants of the challenge but it is rather described in details by text. This task simulates situations in which a user wants to find a particular video clip, without having seen it before, but knowing the content of the video exactly. For the AVS task, instead, a textual description is provided (e.g. "A person playing guitar outdoors") and participants need to find as many correct examples as possible, i.e. video shots that fit the given description.

In this paper, we describe the current version of Visione, an image retrieval system used to search for videos, presented for the first time at VBS2019. After the first implementation of the system, as described in [1], we decide to focus our attention on the query phase, by improving the user interaction with the interface. And for that reason, we introduce a set of icons for the object location and, inspired by other system involved in VBS of the previous years (e.g. [10]), we integrate the query-by-color sketch. In the next sections, we describe the main components of the system and the techniques at the bottom of the system.

2 System Components

Visione is based on state-of-the-art deep learning approaches for the visual content analysis and exploits highly efficient indexing techniques to ensure scalability. In Visione, we use the keyframes made available by the VBS organizers (1 million segments and keyframes[1]), focusing our work on the extraction of relevant information on these keyframes and on the design of a clear and simple user interface.

In the following, we give a brief description of the main components of the system: the User Interface and the Search Engine (see Figure 1).

2.1 User Interface

The user interface, shown in the upper part of Figure 1, provides a text box to specify the keywords, and a canvas for sketching objects to be found in the target video. Inspired by one of the system on VBS2018, we integrate also the query-by-color sketches, realized with the same interface we used for the objects (canvas and bounding box). The canvas is split into a grid of 7×7 cells, where the user can draw simple bounding boxes to specify the location of the desired objects/colors. The user can move, enlarge or reduce the drawn bounding boxes for refining the search. In the current version of the system, we realize a simple drag & drop on the canvas using icons for the most common objects. Furthermore with the same mechanism we define a color palette available as icons, to facilitate the search by color: for each cell of the grid (7×7), we calculate the dominant colors using a K-NN approach, largely adopted in color based image segmentation [13].

[1] https://www-nlpir.nist.gov/projects/tv2019/data.html

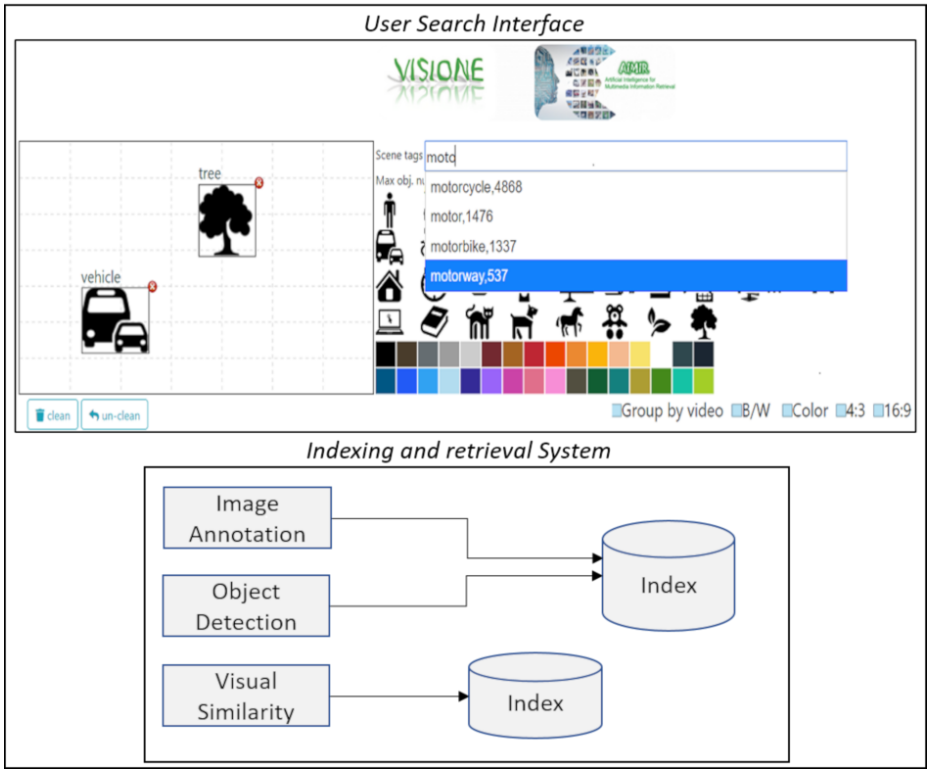

Fig. 1. The main components of Visione: the User Search Interface, and the Indexing and Retrieval System

Moreover, another new functionality added to the old system [1], is the possibility of using some filters, such as the number of occurrences of specific objects, and the type of keyframes to be retrieved (B/W or color, 4:3 or 16:9). At browsing time, the user browsing through the results can use the image similarity to refine the search, or group the keyframes (in the result set) that belong to the same video. Finally, the user interface offers the possibility to show for each keyframe of the result set, all the keyframes of the video of the selected keyframe, and play the video starting from the selected keyframe: this can help to check if the selected keyframe matches the query. A standard search in Visione, for all the tasks, could be done by drawing one or more bounding boxes of objects/colors, or by searching for some keywords, and often by combining them.

2.2 Search Engine

Retrieval and browsing require that the source material is first of all effectively indexed. In our case, we employ state-of-the-art deep learning approaches to extract both low-level and semantic visual features. We encode all the features

extracted from the keyframes (visual features, keywords, object locations, and metadata) into textual representations that are then indexed using inverted files. We use a text surrogate representation [6], which was specifically extended to support efficient spatial object queries on large scale data. In this way, it is possible to build queries by placing the objects to be found in the scene and efficiently search for matching images in an interactive way. This choice allows us to exploit efficient and scalable search technologies and platform used nowadays for text retrieval. In particular, Visione relies on the Apache Lucene[2].

In the next section, we describe in detail the techniques employed to obtain useful visual/semantic features.

3 Methodologies

Visione addresses the issues of CBVIR modeling the data using both the simple features (color, texture) and derived features (semantic features). Regarding the derived features, Visione relies heavily on deep learning techniques, trying to bridge the semantic gap between text and image using the following approaches:

- for **keywords search**: we exploit an image annotation system based on different Convolutional Neural Networks to extract scene attributes.
- for **object location search**: we exploit the efficiency of YOLO[3] as a real-time object detection system to retrieve the video shot containing the objects sketched by the user.
- for **visual similarity search**: we perform a similarity search by computing the similarity between the visual features represented using the R-MAC [17] visual descriptor.

Keywords. Convolutional Neural Networks, used to extract the deep features, are able to associate images with categories they are trained from, but quite often, these categories are insufficient to associate relevant keywords/tags to an image. For that reason, then, Visione exploits an automatic annotation system to annotate untagged images. This system, as described in [2], is based on YFCC100M-HNfc6, a set of deep features extracted from the YFCC100M dataset [16], created using the Caffe framework [8]. The image annotation system is based on an unsupervised approach to extract the implicitly existing knowledge in the huge collection of unstructured texts describing the images of YFCC100M dataset, allowing us to label the images without using a training model. The image annotation system also exploits the metadata of the images validated using WordNet [5].

Object Location. Following the idea that the human eye is able to identify objects in the image very quickly, we decide to take advantage of the new technologies available to search for object instances in order to retrieve the exacted video shot.

[2] https://lucene.apache.org/
[3] https://pjreddie.com/darknet/yolo/

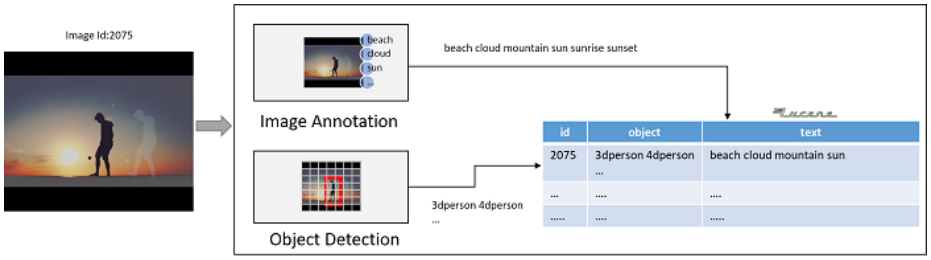

Fig. 2. Search Engine: the index for both object location and keywords.

For this purpose, we use YOLOv3 [14] as object detector, both because it is extremely fast and because of its accuracy. Our image query interface is subdivided into a 7×7 grid in the same way that YOLO segments images to detect objects. Each object detected in the single image I by YOLO is indexed using a specific encoding ENC conceived to put together the location and the class corresponding to the object ($cod_{loc}cod_{class}$). The idea of using YOLO to detect objects within video has already been exploited in VBS, e.g. by [18], but our approach is distinguished by being able to encode the class and the location of the objects in a single textual description of the image, allowing us to search using a standard text search engine. Basically for each image I entry on the index, we have a space-separated concatenation of ENCs, one for all the possible cells (cod_{loc}) in the grid that contains the object (cod_{class}) where:

- loc is the juxtaposition of row and col on the grid
- $class$ is the name of the object as classified by YOLO.

In practice, through the UI the users can draw the objects they are looking for by specifying the desired location for each of them (e.g., *tree* and *vehicle* in Figure 1). Meanwhile, for each object, the UI encodes appropriately the request to interrogate the index, marking all the cells in the grid that contain the object. For example, for the query in Figure 1, we will search for entries I on our index that contain the sequence $p_1 tree\ p_2 tree\ ...\ p_6 tree$, where p_i is the code of the i-th cell (with $1 \leq i \leq 6$ since the tree icon covers six cells). Note that, a cell of a sketch can contain multiple objects. As showed in Figure 2, for the image with id 2075, we extract both keywords (*beach, cloud, etcetera*), using the image annotation tool, and object location (*3dperson, etcetera*), exploiting the object detector, and later we index these two features in a single Lucene index.

Visual Similarity. Visione also supports visual content-based search functionalities, which allows users to retrieve scenes containing keyframes visually similar to a query image given by example. To start the search the user can select any keyframe of a video as query. In order to represent and compare the visual content of the images, we use the Regional Maximum Activations of Convolutions (R-MAC) [17]. This descriptor effectively aggregates several local convolutional features (extracted at multiple position and scales) into a dense and compact

global image representation. We use the ResNet-101 trained model provided by [7] as feature extractor since it achieved the best performance on standard benchmarks. To efficiently index the R-MAC descriptor, we transform the deep features into a textual encoding suitable for being indexed by a standard full-text search engine, such as Lucene: we first use the Deep Permutation technique [3] to encode the deep features into a permutation vector, which is then transformed into a Surrogate Text Representation (STR) as described in [6]. The advantage of using a textual encoding is that we can efficiently exploit off-the-shelf text search engines for performing image searches on large scale.

4 Results

For the evaluation of our system, we took advantage of the participation to the VBS competition, which was a great opportunity to test the system with both expert and novice users.[4] For each task, a team receives a score based on response time and on the number of correct and incorrect submissions.

KIS tasks. During the competition, the strategy used for solving both the KIS tasks was mainly based on the use of queries by object locations and keywords. Queries by color-sketch were used sparingly since they resulted to be less stable and sometimes degrades the quality of results obtained with the keywords/object search. As showed in Figure 3, for our system the textual-KIS task was the hardest, accordingly to the observation done by the organizers of the competition in [12], where they note that textual-KIS task is much harder to solve than visual tasks.

AVS tasks. In this tasks, keywords/object and the image similarity search functionalities were mainly used. In particular, the image similarity search resulted to be notably useful to retrieve keyframes of different videos with similar visual content.

We experienced how an image retrieval system could be useful for video search, for the (*Textual KIS*) the results were not particularly satisfying, but for the *AVS* task are very promising. A problem on the textual KIS is a too specific categorisation of the object which decreased the recall: sometimes users does not distinguish between car or trunk or vehicle and they may use one of them (as textual query) indistinctly. However, for the YOLO detector the difference is quite significant and this leads to low recall. Globally speaking, one of the main problem was due to a rather simple user interface. In fact, Visione was not supporting functionality like query history, multiple submissions at once, or any form of collaboration between the team members: this leads to redundant submissions and "slow" submission of multiple instances.

5 Conclusion

We described Visione, a system presented at the Video Browser Showdown 2019 challenge. The system supports three types of queries: query by keywords, query

[4] http://www.videobrowsershowdown.org/example-browsers/infos-and-results-2019/

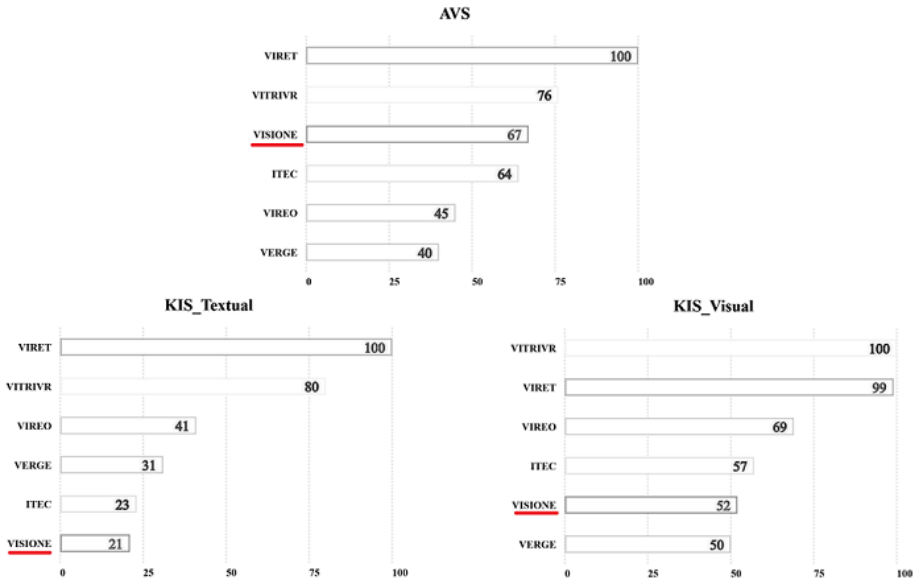

Fig. 3. The VBS2019 competition results for the three tasks AVS, KIS-textual and KIS-visual (score between 0 and 100). The bold line highlights the result of our system.

by object location, and query by visual similarity. Visione exploits state-of-the-art deep learning approaches and ad-hoc surrogate text encodings of the extracted features in order to use efficient technologies for text retrieval. From the experience at the competition, we ascertained a high efficiency regarding the indexing structure, made to support large scale multimedia access but a lack of effectiveness on keywords search. As a result of the system assessment made after the competition, we decide to invest a more effort on the keywords-based search, trying to ameliorate the image annotation part: we plan to integrate dataset of place, concept and categories ([19], [4], [9]), and automatic tools for scene understanding. Furthermore, we will improve the user interface to make it more usable and collaborative.

References

1. Amato, G., Bolettieri, P., Carrara, F., Debole, F., Falchi, F., Gennaro, C., Vadicamo, L., Vairo, C.: VISIONE at VBS2019. In: MultiMedia Modeling - 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8-11, 2019, Proceedings, Part II. pp. 591–596 (2019)
2. Amato, G., Falchi, F., Gennaro, C., Rabitti, F.: Searching and Annotating 100M Images with YFCC100M-HNfc6 and MI-File. In: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing. pp. 26:1–26:4. CBMI '17, ACM (2017)

3. Amato, G., Falchi, F., Gennaro, C., Vadicamo, L.: Deep permutations: Deep convolutional neural networks and permutation-based indexing. In: Similarity Search and Applications. pp. 93–106. Springer International Publishing, Cham (2016)
4. Awad, G., Snoek, C.G.M., Smeaton, A.F., Quénot, G.: Trecvid semantic indexing of video : A 6-year retrospective. ITE Transactions on Media Technology and Applications **4**(3), 187–208 (2016)
5. Fellbaum, C., Miller, G.: WordNet: an electronic lexical database. Language, speech, and communication, MIT Press (1998)
6. Gennaro, C., Amato, G., Bolettieri, P., Savino, P.: An approach to content-based image retrieval based on the lucene search engine library. In: Research and Advanced Technology for Digital Libraries. pp. 55–66. Springer Berlin Heidelberg (2010)
7. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. International Journal of Computer Vision **124**(2), 237–254 (2017)
8. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
9. Jiang, Y.G., Wu, Z., Wang, J., Xue, X., Chang, S.F.: Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(2), 352–364 (2018)
10. Lokoč, J., Kovalčík, G., Souček, T.: Revisiting siret video retrieval tool. In: MultiMedia Modeling. pp. 419–424. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-73600-6_44
11. Lokoč, J., Bailer, W., Schöffmann, K., Münzer, B., Awad, G.: On influential trends in interactive video retrieval: Video browser showdown 2015–2017. IEEE Transactions on Multimedia **20**(12), 3361–3376 (2018)
12. Lokoč, J., Kovalčík, G., Münzer, B., Schöffmann, K., Bailer, W., Gasser, R., Vrochidis, S., Nguyen, P.A., Rujikietgumjorn, S., Barthel, K.U.: Interactive search or sequential browsing? a detailed analysis of the video browser showdown 2018. ACM Trans. Multimedia Comput. Commun. Appl. **15**(1), 29:1–29:18 (2019)
13. Niraimathi, D.S.: Color based image segmentation using classification of k-nn with contour analysis method. International Research Journal of Engineering and Technology **3**, 1169–1177 (2016)
14. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
15. Rossetto, L., Schuldt, H., Awad, G., Butt, A.A.: V3c – a research video collection. In: MultiMedia Modeling. pp. 349–360. Springer International Publishing, Cham (2019)
16. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: YFCC100M: The new data in multimedia research. Communications of the ACM **59**(2), 64–73 (2016)
17. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879 (2015)
18. Truong, T.D., Nguyen, V.T., Tran, M.T., Trieu, T.V., Do, T., Ngo, T.D., Le, D.D.: Video search based on semantic extraction and locally regional object proposal. In: MultiMedia Modeling. pp. 451–456. Springer International Publishing (2018)
19. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)