VERGE in VBS 2021

Stelios Andreadis, Anastasia Moumtzidou, Konstantinos Gkountakos, Nick Pantelidis, Konstantinos Apostolidis, Damianos Galanopoulos, Ilias Gialampoukidis, Stefanos Vrochidis, Vasileios Mezaris, and Ioannis Kompatsiaris

Information Technologies Institute/Centre for Research & Technology Hellas, Thessaloniki, Greece
{andreadisst, moumtzid, gountakos, pantelidisnikos, kapost, dgalanop, heliasgj, stefanos, bmezaris, ikom}@iti.gr

Abstract. This paper presents VERGE, an interactive video search engine that supports efficient browsing and searching into a collection of images or videos. The framework involves a variety of retrieval approaches as well as reranking and fusion capabilities. A Web application enables users to create queries and view the results in a fast and friendly manner.

1 Introduction

VERGE is an interactive video search engine that integrates several retrieval modalities and provides users with a user interface (UI) for formulating different types of queries and visualising the most relevant shots and videos. After a multi-year participation in the Video Browser Showdown (VBS) competition [14], the engine has been adjusted so as to support the Ad-Hoc Video Search (AVS) and the Known Item Search Visual and Textual (KIS-V, KIS-T) tasks. This year two new search modalities are introduced, i.e. Face Detection (Section 2.4) and Activity Recognition (Section 2.6), while previously used methodologies are improved. In addition, the latest version of the VERGE UI (Section 3) is presented, where fewer search options enable the same assortment of retrieval modules, offering a more compact and friendly usage.

2 Video Retrieval Framework

The VERGE framework involves a multitude of search modalities, implemented as services, that can be used independently, fused or consecutively to rerank the top results. Through a UI the users are able to readily create queries and view the most relevant images or videos that match the criteria. A detailed description of the integrated retrieval methodologies follows in the next subsections, while the architecture of the framework is depicted in Fig. 1.

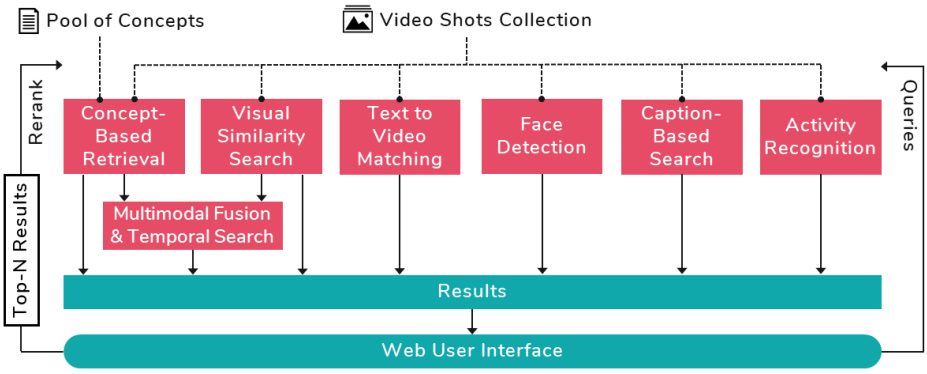

Fig. 1. The VERGE Framework

2.1 Visual Similarity Search

This module retrieves visually similar content using Deep Convolutional Neural Networks (DCNNs). These features are the output of the last pooling layer of the fine-tuned GoogleNet architecture presented in [13]. The dimension of the last pooling layer is 1024 and it is used as global image representation. Eventually, an IVFADC index database vector is created for fast binary indexing using these vectors and K-Nearest Neighbors are computed for the query image [7].

2.2 Concept-Based Retrieval

This module annotates each keyframe with a pool of concepts, which comprises 1000 ImageNet concepts, a selection of 300 concepts out of the 345 concepts of the TRECVID SIN task [12], 500 event-related concepts, 365 scene classification concepts, 580 object labels and 30 style-related concepts. To obtain the annotation scores for the 1000 ImageNet concepts, we used an ensemble method, averaging the concept scores from three pre-trained models that employ different DCNN architectures, namely the EfficientB3, EfficientB5 [15] and InceptionResNetV2. To obtain scores for the subset of 300 concepts from the TRECVID SIN task, we trained and employed two models based on the EfficientB1 and EfficientB3 architectures on the official SIN task dataset. For the event-related concepts we used the pre-trained model of EventNet [5]. Regarding the extraction of the scene-related concepts, we utilized the publicly available VGG16 model fine-tuned on the Places365 dataset [20]. Object detection scores were extracted using models pre-trained on the established MS COCO and Open Images V4 datasets, with 80 and 500 detectable objects, respectively. For the style-related concepts we employed the pre-trained models of [17]. Finally, to offer a cleaner representation of the concept-based annotations we employed various text similarity measures between all concepts' labels. After manual inspection of the text analysis results we formed groups of very similar concepts for which we create a common label and assign the max score of its members.

2.3 Text to Video Matching Module

The text to video matching module inputs a complex free-text query along with a set of video shots and returns a ranked list with the most relative video shots w.r.t. to the input textual query. For this, the method presented in [3] is utilized, in which a textual instance (e.g. a sentence) and a visual instance (i.e. a video shot) are represented into a new common feature space and therefore the direct comparison between free-text queries and video or image instances is feasible. The method utilizes an attention-based dual encoding neural network that uses two similar modules [1], each consisting of multi-level encoding for the video shot as well as for the natural language sentence, in parallel. For initial video shot representation, a pre-trained Resnet-152 model is used for every shot's keyframe whereas each word sentence is initially encoded as a bag-of-words vector. Then, both the sentence and the keyframe representations go through three different encoders (i.e. mean-pooling, attention-based [3], bi-GRU sequential model, and biGRU-CNN [9]). This multilevel encoding is used in order to project both text and video instances into a common feature space following the approach of [2]. When it comes to training data, two different datasets were combined, the TGIF [11] which contains approx. 100k short animated GIFs with one short description per each, and the MSR-VTT [19] consisting of 10k short video clips, each accompanied by 20 short descriptions.

2.4 Face Detection

This module is a specialization of object detection to human faces. For each shot, we extract the number of humans that appear. So, the user can easily distinguish the results of single-human or multi-human activities using as a searching parameter the number of persons. The selection of a face detector in contrast to a person detector is because in crowd-centred scenes the faces can be detected more efficiently as there are fewer occlusions. To address this, we select the implementation of BiFPN [16], a bidirectional feature pyramid network that allows easy and fast multi-scale feature fusion. The model is trained using Google Open Images [10] dataset, keeping only the defections of the "Human-face" class.

2.5 Video Captioning - Caption-Based Search

This module aims to generate for each shot a representative sentence/caption using words included in vocabulary, and thus the user can retrieve videos by simple text search. Video captioning approaches comprise two separate components: (i) a feature extractor that typically extracts the features of a video by sampling among the frames using a fixed number as a step, and (ii) an encoder-decoder that encodes the content and subsequently assigns it to words. To address this, an RNN-based neural network is used [4] based on [18] that takes into account the similarity of the words using semantic clusters. The model is pre-trained on MSR-VTT [19], a widely-known dataset in video captioning domain.

2.6 Activity Recognition

This module generates predictions of human-related activities for each shot, and thus, the user can filter the videos using activity-based keywords. A ranked list of 400 predefined human-related activities and the corresponding scores are generated using a deep learning-based approach. The model architecture is based on a 3D-Resnet Convolutional Neural Network (3D-CNN), similarly to [6] that encodes Spatio-Temporaly the input shots to human-related activities. In particular, the model architecture consists of a ResNet with 50 layers, the input video is fed in the form of: 112 [pixel] x 112 [pixel] x 3 [channel] x 16 [frame], and the model is pre-trained using Kinetics-400 dataset [8].

2.7 Multimodal Fusion and Temporal Search

This module fuses the results of two or more search modules, such as visual features (Section 2.1), visual concepts (Section 2.2) and color features, in a late fusion manner and retrieves similar shots with a two-step algorithm. The first step is the computation of a tensor L whose surfaces capture the similarity of the results between modality pairs, while the second involves the computation of the final ranked list. In the first step, the top-N results are retrieved for each modality, and thus, K lists are produced; one per modality. Each surface of the tensor is a 2D array with size $N \times N$ with values 0 or 1. In this sparse matrix, the value 1 is observed in the position (i, j) of the 2D surface if the i-th element of the first list coincides with the j-th element of the second list. The second step of the algorithm involves four stages to obtain the final result: (i) the extraction of a list from each 2D surface of L, (ii) the bi-modal ranking of the retrieved results, (iii) the merging of the rankings, and (iv) the duplicate removal for obtaining the final list. We further re-rank the top-N retrieved shots, by considering adjacent keyframes as temporally close, so as to perform temporal search.

3 VERGE User Interface and Interaction Modes

The VERGE UI integrates the above modalities in a friendly and efficient way. Aiming to be a compact and easy-to-use tool, this year the users are provided with fewer options in the UI, but without compromising the variety of alternative retrieval capabilities (e.g. one input field for all keyword-based searches).

As seen in Fig. 2, VERGE consists of three main components: (i) the dashboard menu on the left, (ii) the results panel that spans most of the screen, and (iii) the filmstrip on the bottom. The dashboard menu starts with a countdown timer that shows the remaining time for submission during VBS, a back button to restore previous results, and a switch button to select between obtaining new results and reranking. Then, it continues with four search options. The first option is a text input field where the user can type a free-text query and retrieve the most relevant video shots (Section 2.3). The second option offers a long list of concepts (Sections 2.2, 2.6), supporting autocomplete search and multiple selection. The next option can bring shots of a selected color; the user is able to pick

Fig. 2. The VERGE User Interface

a color from a pop-up palette either for the complete image or for certain parts of the image by coloring a 3x3 grid. The last option in the menu allows the user to retrieve shots where a specified number of people appear in them (Section 2.4). All the results are displayed in the central panel as single shots in a grid view or as groups of shots (videos). Hovering over a shot reveals two additional functionalities: getting visually similar images (Section 2.1) and submitting a shot to the contest. Clicking on it updates the filmstrip with the frames of the video it belongs to, while the button on the right can play the video.

To demonstrate the capabilities of VERGE in the VBS contest, we present some usage scenarios that tackle different types of queries. For a KIS-V query that shows a pink, cloudy sky, the user can select the concept "clouds" and rerank the results by color (Fig. 2). For a KIS-T query that reads "playing the drum in a subway station", the user can type the sentence in the free text search or alternatively combine the concepts "subway station/platform" and "drum". Lastly, the AVS query that asks for shots of a single kid smiling can be addressed with the concept "child", then a reranking by selecting one person to appear and, when a matching image appears, visual similarity can bring more relative results.

4 Future work

The usability and the effectiveness of the retrieval methodologies as well as the user interface will be evaluated during VBS 2021 and will identify the direction of future algorithms and implementations in VERGE.

Acknowledgements This work has been supported by the EU's Horizon 2020 research and innovation programme under grant agreements H2020-825079 MindSpaces, H2020-779962 V4Design, H2020-780656 ReTV, and H2020-832921 MIRROR.

References

1. Dong, J., Li, X., Xu, C., Ji, S., He, Y., et al.: Dual encoding for zero-example video retrieval. In: Proceedings of IEEE Conf. CVPR 2019. pp. 9346–9355 (2019)
2. Faghri, F., Fleet, D.J., et al.: VSE++: Improving visual-semantic embeddings with hard negatives. In: Proc. of the British Machine Vision Conference (BMVC) (2018)
3. Galanopoulos, D., Mezaris, V.: Attention mechanisms, signal encodings and fusion strategies for improved ad-hoc video search with dual encoding networks. In: Proc. of the ACM Int. Conf. on Multimedia Retrieval. (ICMR '20), ACM (2020)
4. Gkountakos, K., Dimou, A., Papadopoulos, G.T., Daras, P.: Incorporating textual similarity in video captioning schemes. In: 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC). pp. 1–6. IEEE (2019)
5. Guangnan, Y., L., Y., X., H., et al.: Eventnet: A large scale structured concept library for complex event detection in video. In: Proc. ACM MM (2015)
6. Hara, K., et al.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018)
7. Jegou, H., et al.: Product quantization for nearest neighbor search. IEEE transactions on pattern analysis and machine intelligence **33**(1), 117–128 (2010)
8. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
9. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
10. Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., et al.: Openimages: A public dataset for large-scale multi-label and multi-class image classification. https://storage.googleapis.com/openimages/web/index.html (2017)
11. Li, Y., Song, Y., Cao, L., Tetreault, J., et al.: TGIF: A new dataset and benchmark on animated gif description. In: Proceedings of IEEE CVPR 2016 (2016)
12. Markatopoulou, F., Moumtzidou, A., Galanopoulos, D., et al.: ITI-CERTH participation in TRECVID 2017. In: Proc. TRECVID 2017 Workshop. USA (2017)
13. Pittaras, N., Markatopoulou, F., Mezaris, V., Patras, I.: Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In: International Conference on Multimedia Modeling. pp. 102–114. Springer (2017)
14. Schoeffmann, K.: Video browser showdown 2012-2019: A review. In: 2019 Int. Conf. on Content-Based Multimedia Indexing (CBMI). pp. 1–4. IEEE (2019)
15. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
16. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (2020)
17. Tan, W.R., Chan, C.S., Aguirre, H.E., Tanaka, K.: Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. In: 2016 IEEE ICIP. pp. 3703–3707. IEEE (2016)
18. Venugopalan, S., Rohrbach, M., Donahue, J., et al.: Sequence to sequence-video to text. In: Proceedings of the IEEE ICCV. pp. 4534–4542 (2015)
19. Xu, J., Mei, T., Yao, T., Rui, Y.: MSR-VTT: A large video description dataset for bridging video and language. In: The IEEE Conf. on CVPR (June 2016)
20. Zhou, B., Lapedriza, A., et al.: Places: A 10 million image database for scene recognition. IEEE Trans. on PAMI **40**(6), 1452–1464 (2017)