Interactive Video Retrieval in the Age of Deep Learning – Detailed Evaluation of VBS 2019

Luca Rossetto, Ralph Gasser, Jakub Lokoč, Werner Bailer, Klaus Schoeffmann, Bernd Muenzer, Tomáš Souček, Phuong Anh Nguyen, Paolo Bolettieri, Andreas Leibetseder, Stefanos Vrochidis

***Abstract*—Despite the fact that automatic content analysis has made remarkable progress over the last decade – mainly due to significant advances in machine learning – interactive video retrieval is still a very challenging problem, with an increasing relevance in practical applications. The Video Browser Showdown (VBS) is an annual evaluation competition that pushes the limits of interactive video retrieval with state-of-the-art tools, tasks, data, and evaluation metrics. In this paper, we analyse the results and outcome of the 8th iteration of the VBS in detail. We first give an overview of the novel and considerably larger V3C1 dataset and the tasks that were performed during VBS 2019. We then go on to describe the search systems of the six international teams in terms of features and performance. And finally, we perform an in-depth analysis of the per-team success ratio and relate this to the search strategies that were applied, the most popular features, and problems that were experienced. A large part of this analysis was conducted based on logs that were collected during the competition itself. This analysis gives further insights into the typical search behavior and differences between expert and novice users. Our evaluation shows that textual search and content browsing are the most important aspects in terms of logged user interactions. Furthermore, we observe a trend towards deep learning based features, especially in the form of labels generated by artificial neural networks. But nevertheless, for some tasks, very specific content-based search features are still being used. We expect these findings to contribute to future improvements of interactive video search systems.**

***Index Terms*—Interactive Video Retrieval, Video Browsing, Video Content Analysis, Content-based Retrieval, Evaluations.**

I. Introduction

Video data has become ubiquitous in our daily lives as its usage ranges from traditional entertainment and broadcasting to camera systems used in manufacturing, medicine, smart cities, transportation, or even wearable devices. This creates the need for systems that enable efficient and effective storage, organization and management of video data, while at the same time allowing users to quickly satisfy a particular information need on top of large video collections. The different research aspects in video retrieval, ranging from indexing strategies to query formulation, and their importance for the field are outlined in several surveys [1], [2].

L. Rossetto is with University of Zurich, Switzerland
R. Gasser is with University of Basel, Switzerland
J. Lokoč and T. Souček are with Charles University, Prague, Czech Republic
W. Bailer is with JOANNEUM RESEARCH, DIGITAL, Graz, Austria
K. Schoeffmann, B. Muenzer and A. Leibetseder are with Alpen-Adria-Universität Klagenfurt, Austria
P. Nguyen is with City University of Hong Kong, Hong Kong, China
P. Bolettieri is with Italian National Research Council (CNR), Pisa, Italy
S. Vrochidis is with Information Technologies Institute/Centre for Research and Technology Hellas, Thessaloniki, Greece

Two particular challenges in video retrieval are the inherent lack of structure in video data and the richness of a video's content, especially if compared to still images. Most importantly, however, the sheer quantity and diversity of collected data keeps challenging video management and retrieval systems. To address these issues, automatic video analysis has become a major research field over the last decades, providing significant enhancements, and aiming at temporal segmentation [3], concept annotation [4], object and speech recognition [5], scene captioning and event detection [6]. All of which can be leveraged to enable efficient video retrieval.

However, despite the recent advances in machine learning, and especially deep learning, automatic content annotation can still not match the quality of manual annotation by a human expert. But even if perfect annotation was achieved, it fails to solve all potential problems end-users searching in a collection might encounter. There are several reasons for this: Firstly, the annotation process itself is context-specific as well as subjective and the outcome may differ between multiple users and search contexts (e.g., "*is it a medium sized or a large house?*"). Hence, users often fail to correctly infer automatically assigned labels for a given scene at query time. Secondly, user-generated queries are often incomplete, since humans tend to have difficulties expressing their information need or simply because their memory of a scene may lack the details required to reconstruct a sufficiently accurate query. Therefore, even though we may have tens of thousands of semantic concepts available, users often fail to select the 'correct one' to find the desired content when it counts. Finally, given a sufficiently general query, the number of matching scenes can potentially clutter the result set, making it difficult to separate noise from relevant items. All these aspects lead to a need for a human-computer cooperation, in which users interactively explore result sets, inspect items, and reformulate queries or try different search paths or mechanisms in a trial-and-error fashion.

The Video Browser Showdown (VBS) [7]–[9] – first held in 2012 – is an annual video search evaluation campaign that employs a competitive format, which allows participating teams to evaluate their state-of-the-art interactive video retrieval systems in direct comparison to one another. It provides a fair and live performance assessment of retrieval systems for the same search tasks, on the same dataset, in the same environment. The participants have to perform a large number of varying tasks over several hours, such as visual and textual known-item search (KIS) and textual Ad-hoc Video Search (AVS), which are either automatically evaluated by the VBS

competition server or manually assessed by live judges. After several sessions and about 8 hours of searching, the team with the highest number of total points is selected as the winner. In contrast to TRECVID, which aims at evaluation of automatic search performance with an inferred ranking-based measure and a pooled ground-truth, the VBS evaluates interactive search performance of expert and novice users with task-specific metrics and an exhaustive assessment of submissions. Consequently, the two evaluation campaigns can be seen as complementing each other. This is also why they collaborate and use the same dataset for AVS with partly overlapping tasks. A detailed overview and history of the VBS can be found in [10].

This paper summarizes the evaluation results of the Video Browser Showdown 2019, which is the 8th iteration of VBS and the first time it operates on the new V3C1 dataset [11] – a corpus of over 1000 hours of video data gathered from the web and designed to be representative of web video. The videos in the dataset span a wide range of content and visual styles as well as resolutions, codecs, and frame rates. The paper performs an in-depth analysis of task results and team submissions and gives insights into the most thorough and detailed interaction log analysis ever conducted during a VBS competition.

The remainder of the paper is structured as follows: Section II provides an overview of various deep-learning methods currently in use for video retrieval. Section III describes the tasks of the Video Browser Showdown and outlines the specific setting during the 2019 iteration. Section IV continues with a brief overview of the participating teams, their tools and strategies. The individual systems are described in more detail in the referenced publications. Section V then outlines on the competition results while Section VI goes into further detail by analyzing the action logs gathered from all participants. Finally, Section VII summarizes all presented results and draws conclusions.

II. Deep-learning Methods for Video Retrieval

Current video retrieval tools rely a lot on deep learning methods. Therefore, we provide a brief overview of recently proposed approaches that can be used in this domain. The descriptions of the participating video search tools presented in Section IV, and the referenced papers therein, provide further insights into how VBS 2019 participants make use of specific deep learning methods.

Deep-learning has influenced the area of video retrieval in various ways over the past few years. One area where it is readily applied is in the creation of new feature transformations, which augment or even replace the previously used, manually engineered features such as the well-known SIFT [12] feature descriptor. These feature learning approaches have led to several representations applicable to the visual domain in general [13], [14], as well as to more specific retrieval tasks, such as finding videos showing a specific person [15], for example. These methods produce a wide range of descriptor sizes, which has also led to the application of deep-learning methods to the problem of hashing [16] in order to construct compact representations of the video content. Many of these methods operate in the visual domain and are often trained on still images rather than videos. But an increasing number of approaches explicitly considers multiple modalities, such as visual, aural and textual information [17]–[19].

In order to improve the retrieval of videos based on visual sketches, several directions have been considered so far. While some methods focus on line sketches and perform transformations that capture both the semantics as well as the visual appearance of the sketch [20], others aim at spatial-semantic retrieval [21], [22] based on pixel-wise labelling using semantic segmentation of images [23], [24] and broader, colored sketches as an input. Other approaches, such as GauGAN [25], provide a glimpse at how generative models can be leveraged to generate realistic video scenes from user's sketches. However, generalizing this model to fit with a complex video dataset containing different types of people, objects, and activities is still an unsolved problem.

For the support of more traditional text-based queries, several deep-learning based approaches exist. The most straight forward is to automate the semantic annotation of content – which was often done manually before – by applying concept, object, and activity detectors to video segments or individual frames. For concept detection, a very common network architecture is ResNet [26] with a different number of layers (ResNet50, ResNet101, ResNet152), trained on one or several semantically labelled image datasets, such as ImageNet 1k [27], ImageNet Shuffle [13], TRECVID SIN Task [28], Research Collection [29], MS COCO [30], and MIT Places [31]. For object detection and localization, a currently popular architecture is FasterRCNN [32], commonly trained on MS COCO or the OpenImage [33] dataset. For activity detection, networks such as C3D [34] or P3D [35] can be used, which are commonly trained on different datasets including Sport1M [36], Kinetics [37], ActivityNet [38] or EventNet [39]. To generate richer textual descriptions rather than just a set of simple labels, scene captioning [40] methods are employed as well.

For text-based querying, embedding the query into the visual space for matching is also a promising approach. Consequently, the W2VV [41] and W2VV++ [42] models proved their potential in the TRECVID-AVS tasks in 2018 and 2019. There even exist extensions to language models such as BERT [43], which enable visual question answering, visual commonsense reasoning, referring expressions, and caption-based retrieval [44].

III. The Video Browser Showdown

The Video Browser Showdown [7]–[10] is an annual video retrieval competition – collocated with the International Conference on Multimedia Modelling – where researchers can evaluate the efficiency and effectiveness of their video retrieval approaches. The participating teams solve retrieval tasks as quickly as possible and submit their results to the VBS competition server[1], where they are scored based on correctness and time that has elapsed since the start of the task.

[1]source code available at https://github.com/klschoef/vbsserver/

Table I: Textual KIS Tasks used in VBS 2019. During the competition, the description is being displayed sentence by sentence, 0, 60 and 120 seconds into the task.

VBS ID	Query
Textual2019-10	A slow pan up from a canyon, static shots of a bridge and red rock mountain. A river is visible at the ground of the canyon. The bridge is a steel bridge, there is a road right to the mountain in the last shot.
Textual2019-11	A protest camp on a public square, with blue canvas cover, the middle shot shows the statue of a horseman from below. The last shot shows an improvised library and people reading. Behind the statue is a building with scaffolding.
Textual2019-12	Inside shot of a church, first moving towards a glass window, then turning left, showing a golden image of Mary and the Child. Wooden interior, there are flowers on the altar below the image of Mary. Handheld camera, a golden chandelier is hanging from the ceiling.
Textual2019-13	Interior of a radio studio, host in pink sweater on the left, talking. Guest in the right corner, in an intermediate shot two further guests are visible. Screens and microphones on the table, a dark green pin board in the background on the left.
Textual2019-14	Shot from a bike, first along a desert path, then on a street, overtaking riders on a tandem. On the desert road a plush flamingo sits on the front of the bike. The riders on the street wear yellow safety vests, first seen from behind, then from the side.
Textual2019-15	Close-up of a blond girl in front of a bookshelf, in between shots of her and elderly men walking. She wears a green vest, one of the old man wears dark and patterned clothes. In the final shot they walk off and she turns around.
Textual2019-18	A sequence of three starts with a paraglider, filmed from the view of the pilot. The paraglider is green/white with a blue stripe in between. In the last shot, the pilot's shadow is visible.
Textual2019-20	Close-up of hands playing the piano, then of hands using a tablet. The tablet shows musical score sheets. 'videoblocks' is superimposed over the shot with the tablet.

The tasks fall into one of three categories: *visual known-item search (visual KIS)*, for which participants have to find a unique 20 seconds video sequence within the collection based on a preview, *textual known-item search (textual KIS)*, for which participants have to find a unique, 20 seconds video sequence based on a textual description, and *ad-hoc video search (AVS)*, where participants are required to find as many video sequences as possible that satisfy a broader textual description. The latter task type is equivalent to the TRECVID Ad-hoc Video Search (AVS) task [45] and has a partial overlap in the queries that are being used.

All the aforementioned tasks have been created and can be classified along the taxonomy introduced in [8], which is based on the expected number of results and the prior knowledge users have about the scene, resulting in the table $\{target, class\} \times \{example, visual, textual, none\}$. While the assessment of submissions for tasks of the former two types require no human input, the submissions for AVS tasks are assessed manually on site by an independent jury. The scoring and the rational behind it are described in [8].

A. Competition structure in 2019

Like in previous years, VBS 2019 was split into a *private session*, with only the experts participating, and a *public session* held in front of an audience during the conference reception, with both *expert* and *novice* tasks. Experts are users from the participating teams (typically the developers themselves), while novices are users randomly selected from the audience, who are likely to have some prior knowledge of multimedia retrieval, but have no experience in using the particular tool other than a brief introduction they received prior to the session. During the public session, only visual KIS and AVS tasks were being solved, as they tend to be more entertaining for the audience, while in the private session all types of tasks were performed. Thus, AVS and visual KIS tasks were both part of the expert and the novice session. However, the AVS tasks were slightly adjusted for the novice session by making the descriptions more specific and therefore the number of potentially relevant scenes smaller. Therefore, the results for the AVS tasks with corresponding numbers in the two sessions cannot be compared directly.

Table II: AVS Tasks used in the VBS 2019 Expert (E) and Novice (N) sessions.

VBS ID		Find shots showing ...
AVS2019-10	E	a person jumping with a bike (not motorbike).
AVS2019-10N	N	a person jumping with a motorbike (not a bicycle).
AVS2019-11	E	bride and groom kissing.
AVS2019-11N	N	two people kissing who are not bride and groom.
AVS2019-12	E	a surfer standing on a surfboard.
AVS2019-12N	N	a surfer standing on a surfboard, not in the water.
AVS2019-13	E	people hiking.
AVS2019-13N	N	people walking in a gay pride parade.
AVS2019-14	E	inside a moving car.
AVS2019-14N	N	two people talking to each other inside a moving car.
AVS2019-16	E	two or more people talking to each other (outdoors).
AVS2019-17	E	people walking on a city square or street.
AVS2019-17N	N	people walking across (not down) a street in a city.
AVS2019-23	E	with snow or ice conditions (outdoors).
AVS2019-24	E	a single person playing a musical instrument.
AVS2019-25	E	electrical power lines.

Compared to VBS 2018 [8], there have been two small adjustments to the tasks: Firstly, the time to solve textual KIS tasks has been increased from 7 to 8 minutes to account for the difficulty of this task type. Secondly, the presentation of visual KIS tasks has been changed to make them more challenging. Visual KIS tasks try to mimic the case in which the searcher can recall a visual representation of the target scene from memory but does not have a sample at hand to compare against. To come closer to this scenario, the visual query clip is now displayed without distortion at first but then gradually blurred-out as time passes. Eventually, users can only determine the most salient properties of the scene and the main objects it depicts.

The 2019 iteration of the VBS competition also saw the introduction of a new dataset, the V3C1 [11] – the first shard of the Vimeo Creative Commons Collection (V3C) [46] – which consists of 1,000 hours of video content as it can be found in the wild. This new dataset replaces the previously used IACC.3, which is comprised of videos collected from the Internet Archive[2]. The IACC dataset [47] has been available for several years now and was shown to no longer represent the kind of video content commonly found on the Internet [48]. The V3C remedies this discrepancy in representativeness while also introducing videos with more modern content, codecs, resolutions and frame rates.

B. Tasks

The task selection process for VBS 2019 followed the same procedure as in earlier years [9]. Table I lists the textual KIS tasks used during the private session with the expert users. Each task description consists of three sentences, of which the first is displayed at the beginning and the other two are delayed and displayed 60 seconds and 120 seconds into the task. After 120 seconds, the complete description becomes visible.

[2]https://archive.org/

Table III: Overview of the different functionality implemented by the participating systems. ◯: functionality is implemented, ✓: functionality has been used during the competition. References to related work are included where applicable.

	vitrivr [49]	VIRET [50]	VIREO [51]	VISIONE [52]	ITEC [53]	VERGE [54]
Visual sketch	◯ [55]	✓ [56]	✓	✓	✓ [57]	
Motion sketch	◯ [55]					
Semantic sketch	✓	✓		✓		
Query by image	◯ [55]	✓ [58]	✓	✓	✓	✓ [59]
Concept labelling	✓[3] [60]	✓ [58]	✓	✓	✓	✓ [61]
Free text to concept matching						✓ [62]
Action labelling	✓ [63], [64]					
ASR	✓[4]					
OCR	✓[3]					
Metadata	✓		✓		✓	✓ [65]
Multi-modal query	✓	✓	✓	✓		✓
Temporal query[5]		✓				
Video playback	✓		✓	✓	✓	
Video preview[6]		✓			✓	✓
Shot context[7]	✓	✓	✓	✓	✓	✓
Video summary[8]		✓	✓	✓	✓	
Grid sorting[9]		✓			✓	
Query history					◯	✓
Result history	✓					
Custom shots		✓ [66]			✓	
Fast submission	✓	✓	✓		✓	✓
Collaboration	✓				◯	
SQL-Database	✓					
MongoDB-Database						✓
Custom indexing	✓	✓	✓	✓	✓	✓
Overall score	91	86	53	44	43	40

Table II lists the AVS tasks used in the expert and novice sessions. Due to the new V3C1 dataset, these tasks have been created specifically for VBS, as no TRECVID AVS tasks were available for the new collection. In line with the former TRECVID AVS tasks, the new tasks were designed to be sufficiently general. However, during the private session with the experts users it turned out that some of the tasks resulted in an extremely high number of relevant results, thus for the novice session, the tasks were rephrased to be more restrictive.

IV. Video Search Tools at VBS 2019

The following section briefly introduces the six different systems that participated in VBS 2019 and outlines the strategies employed by the teams when using their respective tool. A summary of the functionality implemented by the various systems is presented in Table III and Figure 1 gives an overview of their user interfaces.

[3]https://cloud.google.com/vision/
[4]https://cloud.google.com/speech-to-text/
[5]users describe two (usually different) consecutive frames/shots
[6]display of a subset of the frames without playback of the actual video
[7]information about preceding and following shots
[8]representative frames used to preview a video and enable fast navigation
[9]results organized in a 2D grid such that similar shots are close

A. vitrivr

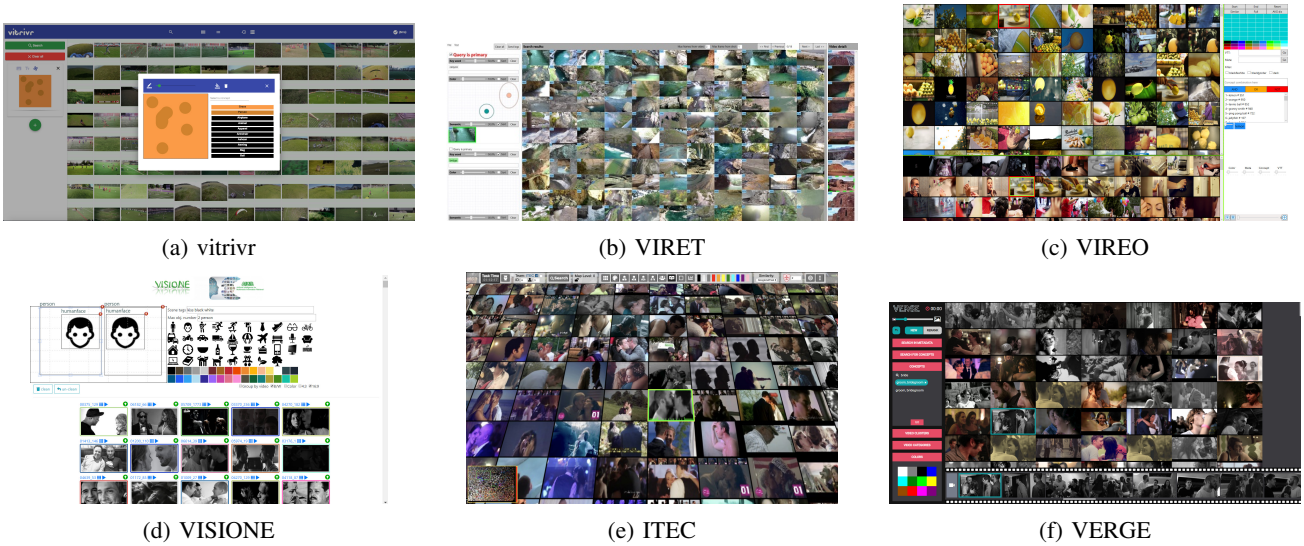
1) Overview: vitrivr is an open-source[10] multimedia retrieval stack [67] with support for content-based retrieval of several media types [68]–[70]. It supports a wide range of feature generation techniques, some of which are powered by various deep learning approaches. The vitrivr stack – together with its predecessor, the IMOTION system [71] – has been participating in VBS since 2015 [72]. For video, vitrivr generates features for (temporal) segments and their key-frames and it offers several query modes, such as query by sketch, query by example, as well as textual concept, caption, OCR and speech transcript search. In 2019, support for semantic sketch queries was added, in which semantic concepts are represented by a hand-drawn, colored free-form area. A detailed description of vitrivr's capabilities and their use during VBS 2019 can be found in [49] and [73].

2) Search strategy: The vitrivr team had a slightly different strategy for each of the three task types, all favoring textual over sketch-based query formulation. Since search for text-on-screen and speech transcripts are among the query modes vitrivr offers that provide the highest selectivity in a time-sensitive, competitive setting, the team always paid close attention to on-screen text, dialogue or lyrics in the visual KIS task target segment. In case no suitable dialog or text was available within the task, the strategy became the same as for textual KIS tasks, which predominantly relied on the selection of distinctive semantic concepts that could be recognized by one of the available detectors. To increase the selectivity, several concepts were used in combination where applicable. Both approaches still resulted in relatively large result sets in many cases and consequently, browsing was always an integral part of the search process. For AVS tasks, often only very few general concepts were used for the query in order to produce as many results as possible. From these results, the relevant ones were manually selected, paying attention to not select too many shots from the same video or shots that were temporally very close to one another. Redundant submissions were minimized by synchronizing a list of shots that had already been submitted across the two system instances. In a few cases, where the search results of a new query were not satisfying or the retrieval took longer than expected, the query history was used to browse through the previously retrieved results to increase the number of submissions. Since only two team members were allowed to operate a system instance each at any given time, the third team member could act as a coordinator, informing the searching members in case they were using the same query terms and suggesting alternatives.

B. VIRET

1) Overview: The VIRET tool [50], [66] is a frame-based, interactive video retrieval system relying on its own temporal segmentation using a TransNet Deep Convolutional Network architecture [74]. Query initialization approaches involve keyword search based on a set of 1243 supported labels (automatic annotation by retrained NasNet [58]) extended

[10]source code available at https://github.com/vitrivr



(a) vitrivr (b) VIRET (c) VIREO

(d) VISIONE (e) ITEC (f) VERGE

Figure 1: Screenshots of the participating tools

by additional hypernyms, query by simple color sketches (ALL/ANY color queries [56]), semantic sketches (regions with faces, considering NOT and ALL/ANY specification) and query by example images (from the result set or an external server). Each modality supports a second temporal query, taking into account the content of the nearby temporal context. The intermediate results of each query modality are filtered to the top-K results (filters are configurable) and the intersection of these intermediate results is returned as the overall result set, sorted by a selected modality. The top ranked frames are presented in a grid with easy temporal context inspection. By using the mouse wheel while hovering over a frame with the cursor, one can play a sequence of preceding/following frames and show representative context frames in a vertical stripe. In addition, the tool supports filtering of black & white frames and filtering by frame brightness, the number of displayed frames from a shot, and the number of displayed frames from a video.

2) Search strategy: The search strategy reflected the interface design focusing on multi-modal and temporal queries. Both users often relied on temporal keyword queries (e.g., "canyon" followed by "bridge"), complemented by color/face sketches and example images. One of the two tool instances also relied on an external search engine (e.g., Google Images) that was used to query for suitable example images using keyword search for subsequent use in an example-based query within the VIRET system. After the novice session, this feature was praised by the novice user controlling the tool. Over the course of the tasks, VIRET tool users inspected the temporal context of retrieved frames or opened their video summaries. For AVS tasks, the users collected promising candidates into a basket and submitted all the verified shots at once.

C. VIREO

1) Overview: VIREO is a concept-based interactive video search system that has been developed and has participated in VBS since 2017 [75]. The tool incorporates three retrieval modalities including query by color-sketch, query by metadata and query by concept [76]. The core is the query by concept modality with a concept bank supporting up to 15K concept labels from the categories human, animals, plants, objects, places and actions. To enable the fusion of results from different modalities, a simple weighting scheme is applied. In addition, a similarity search module is used to look for semantically similar shots from the video corpus. Furthermore, the tool supports filtering black-bordered and black & white shots to refine search results. The browsing interface of the tool presents the retrieval result in a zig-zag pattern where the more relevant candidate shots stay in the top rows and relevance decreases from left to right. Picking a candidate shot from this interface triggers its context view and video playback. In 2019, the VIREO team proposed a new browsing interface [77] to present the hierarchical representations of video shot clusters. In addition, they integrated a video-to-text module [78], which maps the textual query and video feature to the same latent space using a bi-directional GRU for video retrieval.

2) Search strategy: The two VIREO team members used the same tool with the same configuration in all VBS sessions, employing similar approaches for the different tasks. For visual KIS tasks, if the color distribution of the video frame was clearly discernible and exhibited saturated colors, the query by color-sketch modality was utilized. In other cases, they used the same approach as for textual KIS tasks, where queries with distinctive descriptions of the given topics were formulated. After that, manually browsing the results and picking the correct shot played a critical role. During the query formulation process, one team member focused on using the query by concept modality, whereas the other mainly used the video-to-text module. In AVS tasks, the advantages of the query by concept modality could be leveraged particularly well. The initial query mostly consisted of a single, distinctive concept. Once a relevant shot matching the query has been identified, the similarity search module could be employed to search for similar items for further investigation.

D. VISIONE

1) Overview: VISIONE is a content-based video retrieval system that participated to VBS for the very first time in 2019. It primarily leverages state-of-the-art artificial intelligence techniques to analyze the visual content and exploits highly efficient indexing techniques to ensure scalability. The system supports query by scene tag, query by object location, query by color sketch, and visual similarity search. For the scene tag search, it leverages the image tagging system proposed in [79], which is able to label images with about 15K concepts. VISIONE also uses YOLOv3 [80], YOLOv3-OpenImages [81], and YOLO9000 [82] object detectors, with about 9500 object tags. The R-MAC [83] descriptors are adopted as global image descriptors for the similarity search functionality. All descriptors (scene tags, dominant colors, object location, and visual descriptors) extracted from the video key-frames were encoded with a surrogate textual representation and efficient technologies for text retrieval were adopted for the indexing and searching phases [84]–[86]. The system's user interface provides a text box to specify the scene tags and a canvas for sketching objects and/or colors appearing in the target scene. 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 and/or colors. To speed-up drawing, the most common objects and the colors are grouped into a palette, from where they can be dragged & dropped onto the canvas. The user can modify the drawn bounding-boxes in order to refine the search. Moreover, so as to be more selective, the user can apply a range of filters, such as limiting the number of occurrences of specific objects in each of the results, or retrieving only black & white key-frames. Then, while browsing through the results, the user can leverage image similarity to refine the search or group the results by video to have a different view for inspection. Finally, to check if the selected key-frame matches the query, it is often helpful to display all the key-frames of a specific video (key-frame context) or to play the video starting from the selected candidate frame. A more detailed description of the VISIONE system can be found in [52].

2) Search strategy: The first step to initiate a search in VISIONE, for both KIS and AVS tasks, is to draw one or more bounding boxes of objects/colors, or to enter some scene tags. It is also possible to combine these two operations. We observed that during the competition, the most popular search strategy to address KIS tasks relied on query by object locations and scene tags. In contrast, for the AVS tasks, the image similarity search, query by scene tags, and query by object locations were mainly employed. Since image similarity search retrieves key-frames of similar visual content from different videos, that functionality turned out to be particularly useful during the AVS task. We also observed that in both the KIS and AVS tasks, the search by color sketch functionality was barely used since it seemed less stable and sometimes it negatively influenced the quality of the results. It is worth noting that at the time of the competition, the VISIONE system did not support some relevant functionality like any way of cooperation between the team members, simultaneous submissions, or query history. Thus, especially during the AVS tasks, the performance of the VISIONE system suffered from redundant submissions and the "slow" submission rate of the individual instances.

E. ITEC

1) Overview: The ITEC team employed their actively developed *diveXplore* system [87] (partly based on findings from [88]), a shot-based interactive video browser building on the concept of self-organizing feature maps, i.e., pre-calculated arrangements of video shots based on certain criteria such as deep features, colors, faces, text, etc. In addition, the tool offers several alternate shot retrieval possibilities: keyword-based search using metadata or tags extracted using a variety of convolutional neural networks, color-based search via scene sketching or HSV color filtering and shot similarity search based on deep features as well as *HistMap* [89] – a custom-built, region-based color descriptor. Participating since VBS 2017 [90], diveXplore has constantly been improved over the years and the version developed for VBS 2019, as detailed in [87], features an autopilot mode for browsing improvement and a more refined sketch search.

2) Search strategy: Similarly to other systems, applied search strategies varied from task to task. While textual KIS tasks were predominantly approached by textual deep feature or metadata search potentially combined with similarity search on retrieved results, visual KIS tasks first were assessed by their color composition. For color intensive or distinctive shots, sketch search or specific color feature maps were utilized. Otherwise, if said task appeared visually expressive, a concept-specific feature map served as a basis for narrowing the search space. If unsuccessful for a certain period of time, the same strategy as for textual KIS tasks was employed. AVS tasks were almost exclusively solved via keyword search with subsequent similarity search, where the team paid special attention to find shots that stem from different videos using diveXplore's group-by-video feature so as to to avoid being penalized by the scoring system. Generally, the two team members refrained from using the same keywords and tried to deploy different search strategies throughout the tasks. Result lists were carefully assessed before trying out different keywords or strategies. Finally, if the correct video had been found for any type of KIS task, the frame to submit was determined in a team effort with the occasional involvement of helpful bystanders.

F. VERGE

1) Overview: The VERGE tool is an interactive video search engine that has been participating to VBS since 2012. VERGE provides a frame-based representation of videos in a grid-like user interface for video retrieval. The following indexing and retrieval modules have been integrated into the VERGE system: concept-based retrieval, visual similarity search, automatic query formulation and expansion, clustering, text-based search and multi-modal fusion. On top of these modules, a re-ranking capability is provided, which allows the user to combine different modalities and consequently

Table IV: Overview of the scores for the individual task types per team (top-2 scores written in bold typeface)

Team	Visual KIS	Visual KIS Novice	Textual KIS	AVS	AVS Novice	Overall Score
vitrivr	**100**	**100**	**80**	**76**	**100**	**91**
VIRET	**99**	38	**100**	**100**	**91**	**86**
VIREO	69	20	41	45	88	53
VISIONE	52	45	21	67	35	44
ITEC	57	25	23	64	47	43
VERGE	50	**55**	31	40	22	40

functions as a multi-modal search module. To generate the visual description of the frames, a selection of visual concepts is detected for each of them, including the 1000 ImageNet concepts, 345 TRECVID SIN concepts, 500 event-related concepts, and 365 place-related concepts. The architectures used for these concepts are all based on deep convolutional neural networks. It should be noted that a mapping of free text to concepts is realized by using the pool of the concepts existing in the system. Moreover, each frame is globally represented by using the last pooling layer of a fine-tuned GoogleNet on 5055 concepts, thus allowing query by example visual search. Frames can also be globally described using the MPEG-7 color layout descriptor and by mapping them to an 8-color palette, color-based frame clustering is supported. As far as the video representation of the dataset is concerned, three modules are available. The first exploits the video text metadata, considers online databases (Wordnet, Babelnet) and replaces terms with their respective semantic concepts, in order to add versatility to the user query results. The video textual metadata is also used for topic modeling using Latent Dirichlet allocation. Finally, each video is represented by a vector incorporating the top-20 detected concepts and video similarity comparison is performed by computing the cosine distance between the videos.

2) Search strategy: The VERGE team members followed similar approaches in the VBS session, with some variations depending on the task. In all cases, the initial step was for one of the team members to search for a characteristic keyword between the visual concepts and for the other member to look at the video metadata. For AVS tasks, once results for the desired concept had been retrieved, the two members started submitting as many shots as possible, after some internal coordination to make sure that the two members focused on different videos. For textual KIS tasks, the results for the selected concepts were examined and the videos in the result set were investigated to find the one most similar to the query description. Finally, for visual KIS tasks, re-ranking by color and similarity by image have proven to be very useful.

V. Competition Results

This section presents the overall results of the competition as well as a discussion on the setting under which they were produced. Table IV lists an overview of the scores the

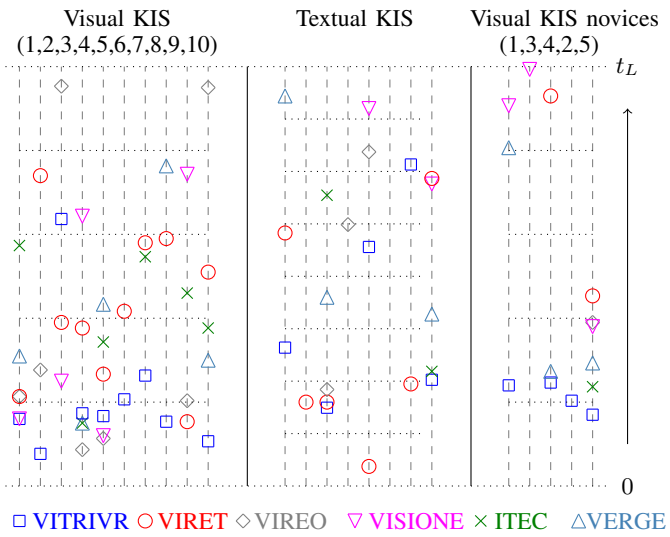


Figure 2: Time elapsed until the first correct submission per team for all scoring teams at VBS 2019. The tasks for novices were the same as for experts, except for their order. Time is shown on the vertical axis, increasing from bottom to top, the tasks are ordered horizontally from left to right and grouped by task type. Figure taken from [66].

teams achieved in the individual task types and highlights the two highest scores per task type. All task types are scored independently of one another. The function used to determine the score S for both visual and textual KIS tasks is given in Equation 1 where t denotes the time required for the correct submission and T is the total available time for the task. n_{ws} denotes the number of incorrect submissions made by a team prior to the correct one in the same task and p designates the constant penalty for an incorrect submission. The scores are normalized such that the best team receives 100 points per task type. The overall score is then determined by mean-averaging the scores achieved in the individual task types.

$$S = max\left(0, \left\lceil 50 + 50 \cdot \left(1 - \frac{t}{T}\right) - (p \cdot n_{ws}) \right\rceil\right) \quad (1)$$

All the scoring functions used are described in more detail in [8]. When looking at Table IV, one can see that with the exception of the novice visual KIS task, vitrivr and VIRET were the two highest scoring teams.

Figure 2 visualizes the time that had elapsed for each team before the first correct submission was registered. The total time for visual tasks was 5 minutes and for textual tasks 8 minutes. The large spread in submission times indicates, that it is still challenging for many teams to find the relevant video segment within the allotted time limit. We can also observe a high recall achieved by the VIRET team, which solved all expert visual KIS tasks and 75% of textual KIS tasks. This performance was achieved by frequent use of temporal and multi-modal queries targeting the scenes of interest. The query interface was also effectively used by the team of novice users, who were able to localize the target scenes on the

first page of the result set in four novice tasks (according to our log analysis). However, in two cases the correct frame was overlooked. We hypothesize that this was caused by the larger number of results displayed on a single page (88 or 140, depending on the tool instance).

The relatively low submission times of the vitrivr team were achieved by always trying a text-based query first, which has the lowest time requirement for query formulation and lookup of all the query modes supported by the system. In many cases, an early text query capturing a particularly salient aspect of the target scene was sufficient for the result set to contain the relevant item with a very low rank. This was especially true, when screen text or spoken text was involved. We also received the feedback from our novice user, that the user interface was very easy to use after just a brief introduction, which explains the high scores for the novice tasks.

The strategy employed by the VERGE team differs between the visual and the textual KIS tasks. Specifically, during the visual KIS task, the successful submissions usually occurred towards the middle or at the end of the given time limit since the VERGE team relied on image similarity. Thus, each relatively similar image lead to a slightly more similar one until the desired scene was finally found. In contrast, for the textual KIS task, the team leveraged the search for concepts and it become clear very quickly whether or not the required concepts are available or not. Therefore, the correct submissions either occurred in the first half of the given time frame or not at all.

VISIONE mainly suffered from a lack of a proper speech-to-text tools and also missed some of the image captioning tools used by other teams. The image analysis pipeline employed was often not powerful enough to solve some tasks, particularly in the textual KIS category. Furthermore, the user interface was too simplistic and lacked useful functionality such as a video preview, easy browsing of the results or a simple submission process that could cope with the volume of data produced during AVS tasks.

The VIREO team experienced three shortcomings in their system. Firstly, the system does not provide text-based retrieval on video speech, which was one of the keys to solve the visual KIS task. Secondly, solving the textual KIS task required tedious browsing activities. The reason is that the temporal information of the video was omitted, leading to ineffective retrieval. Thirdly, the user interface design only allows the user to submit shots within the same video while the scoring function for ad-hoc video search favors diversity rather than the number of submissions. Moreover, system errors occurred in the regular expert session and impaired the engagement of the one user who was the most experienced. Regardless of these shortcomings, however, the overall results demonstrate that this tool can compete with other tools.

Regarding results obtained by the ITEC team, it became apparent that the multitude of search strategies provided by their system overall negatively affected its performance. While for some tasks it was tempting to use strategies such as feature map browsing and sketch search, these approaches proved rather time-consuming either leading to large delays for completing tasks or to not solving them at all. Therefore, it proved to be more feasible to mainly conduct text-based retrieval in combination with similarity search. Additionally, the multitude of functionality provided with the ITEC system renders it less intuitive to use, which is reflected in its rather poor performance during the novice session.

VI. Logging and Analysis

During the competition, each team was required to log the actions performed by the systems and their users in order to gain additional insights into the search strategies that were employed by the different teams. This section presents the acquisition and analysis of these action logs.

A. VBS 2019 Log Format

As in 2018, all participants were required to send a log of the activities they performed (i.e., user interactions) during each task together with the results they retrieved to the VBS competition server, where that data would be stored for future analysis. Each of these submissions is enriched with basic information about the submitting team, team member (tool) and the time of the submission as perceived by the server.

While the log format in 2018 consisted of a simple sequence of letters, indicating the utilized functionality of a particular system, it was redesigned in 2019 so as to offer more flexibility and expressiveness. The new log format is JSON based and consists of a sequence of objects each describing an action. Each of these actions is described by five values: *timestamp*, *category*, *type*, *value* and *attributes*, with the latter two being optional. For the *category*, there are five possible values – namely, 'text', 'image', 'sketch', 'filter' and 'browsing' – which can be further specified by the *type* attribute. Each event belongs to exactly one category but can have multiple types. The options for the *type* attribute depend on the selection of *category*. For example, possible types for the 'sketch' category would be 'color', 'edge', 'motion' or 'semantic', depending on what the systems support and what the user chooses to employ in any particular moment. We tried to agree on a controlled vocabulary to use with this field prior to the VBS competition. The *value* field can be used to log the actual content of the query that was used in a particular instance, for example, the search string in a text query. The *attributes* can be used to encode additional information, such as the number of expected query results. However, both *attributes* and *value* are ignored in the following analysis.

B. Log Pre-processing

The log pre-processing was started with some basic data cleaning. We had to normalize the event timestamps, since different interpretations of the UNIX timestamp format seem to exist. Most importantly, however, there was a need to normalize the usage of the fields *category* and *type*. Even though we had agreed on a defined list of possible values for those fields, there still were some minor deviations from that standard. Most of these cases could be attributed to typing errors (e.g., 'bw' instead of 'b/w') whereas, in a few cases, teams came up with new terms that had not been included in

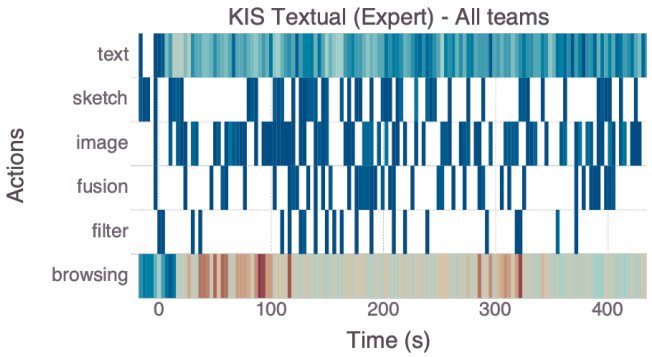


Figure 3: Heatmap of the different action categories over time for all textual known-item search tasks.

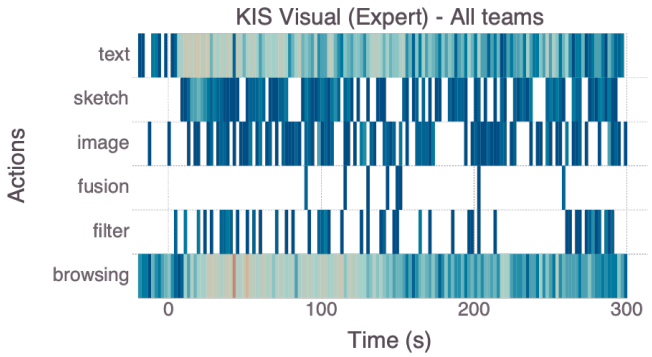


Figure 4: Heatmap of the different action categories over time for all expert visual known-item search tasks.

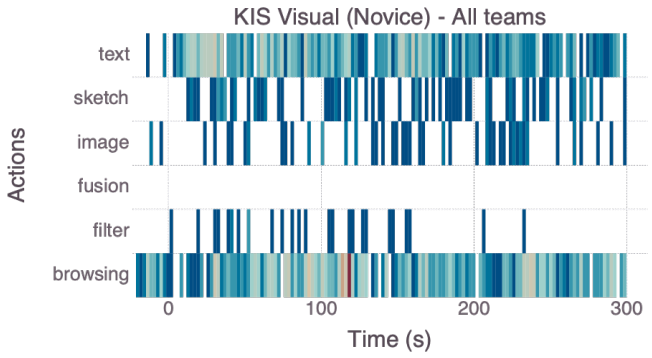


Figure 5: Heatmap of the different action categories over time for all novice visual known-item search tasks.

the vocabulary prior to the competition. Based on the feedback from the individual teams, we decided on a case-by-case basis, whether the new terms should be incorporated, changed or dropped.

In a second step, we corrected the timestamps for all the individual events that were logged. Since the logging of interactions takes place in the UI and therefore on local machines using different, local clocks, there was a minor yet notable shift between the timestamps submitted by the different team members. These shifts could sometimes make the difference between an event being counted towards a particular task or not. In order to correct for these shifts, we calculated the minimum difference between the submission time as reported by the server and the submission time logged by the respective tool and normalized each event timestamp by that difference. We did that for each combination of team and tool. The assumption here is, that there is a constant difference between the local clock of a particular machine and the server clock. The difference we can measure also includes other factors, such as delays caused by network transmission. This is why we chose the minimum value instead of the mean, so as to approximate this inherent difference as accurately as possible.

Last but not least, we decided to sample the submitted actions at a maximum rate of $2Hz$. That is, if multiple actions of the same category and type for the same team and tool were to be logged within a time window of $500ms$, only one of them would be considered. This reduces the weight of high frequency logging of certain types of UI interactions that are triggered by automatic events such as scrolling. From the data we could tell, that these types of events would otherwise cause a strong bias towards actions in the 'browsing' category.

C. Log Analysis

Based on the timestamps associated with each logged action, Figures 3, 4 and 5 present the temporal density of actions with respect to their category for the textual, visual expert and visual novice known-item search tasks respectively, showing lower densities in blue and higher densities in red. The timings for the AVS tasks could not be determined due to an overlooked implementation problem in the VBS server. These tasks are therefore not considered for this analysis.

There are some clear commonalities between the three figures, most notably the high density of browsing-related actions. Interestingly, also text actions were very frequent in comparison to the other modes of query formulation, irrespective of the type of task. Based on these action densities alone, it is however not possible to directly infer any differences in search strategies between the two user groups.

When comparing the action densities of novice users from Figure 5 to those of the expert users as shown in Figure 4, it can be seen that the peak of action density occurs a little later into the task. This is to be expected, since the novice users were, by design, not familiar with the systems and the details of the competition.

To illustrate the transitions between action categories, Figure 6 depicts bi-directional chord diagrams of sequential action pairs. The outer ring of such a diagram illustrates the fraction of this action type with respect to all actions, similar to a pie-chart. The inner ribbons indicate the amount of transitions between a given pair of actions. Since these are bi-directional diagrams, the color of the ribbons are not indicative of the source of a ribbon but rather used to make it easier to distinguish them. The arc-segment of each ribbon is relative to the outgoing fraction of an action category, unidirectional links therefore narrow to a point on one side of the ribbon,

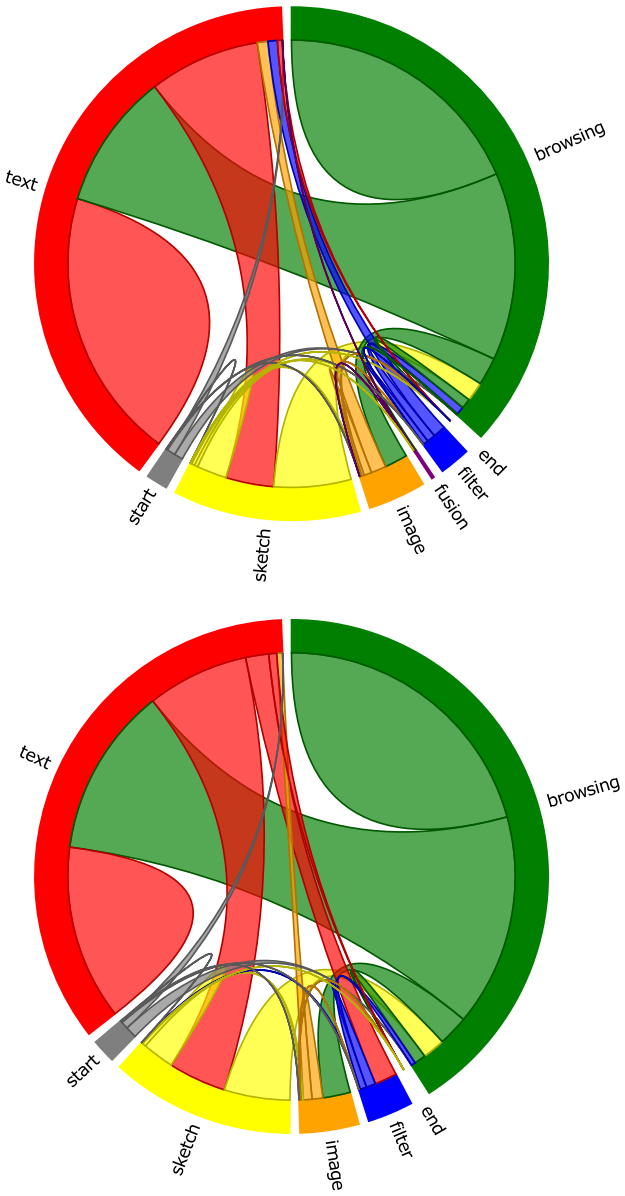


Figure 6: Bi-directional chord diagram of action bi-grams for expert (top) and novice (bottom) visual known-item search tasks.

since there is no link in the opposite direction.

For reasons outlined in Section VI-D, we do not present a similar analysis on a team and task-type level as such an analysis cannot be used to compare teams/tools using the current state of the logs.

When comparing the two diagrams in Figure 6, there is again little difference between the logged strategies employed by experts and novices. Users from both groups primarily alternate between textual queries and browsing, with more transitions towards browsing.

Figure 7 breaks the action categories down to their individual sub-types. On this level of detail, there is again little difference between the aggregated strategies of experts and novices. Both user groups spend the majority of their

Table V: Last non-browsing action before a successful submission per team. Please note that several non-browsing action log records are missing for VIREO and ITEC systems.

	vitrivr	VIRET	VIREO	VISIONE	ITEC	VERGE	Σ
fusion: weighting			1				1
image: globalfeatures		9	1	1	2	2	16
sketch: dominant colors		2		2			4
text: caption	6		1				7
text: concept	1	7	4	1	5	9	27
text: custom	7				1		8
text: localized object				6			6
text: metadata	1		1				2
text: ocr	3						3
Σ	18	18	8	10	8	11	

activities iterating on textual query representations, followed by the exploration of the retrieved results. These results are primarily browsed in their relevance-ordered representation, i.e., a ranked list. The second most common browsing activity is the exploration of the temporal context of a particular item, followed by the use of pre-computed video summaries or the videos themselves. The primary difference between the novice and the expert user group is that, similarly to what can be seen in Figures 5 and 4, the latter group makes more use of specific system functions, such as custom filters and result fusion instructions.

In summary, we can clearly see from the chord diagrams that users mainly switch between textual search and browsing, and typically also start with one of the search features – at least for KIS tasks.

The presented diagrams do however not directly show what action generated the results that, when browsed, led to a successful submission. This information is presented in Table V, which lists the last non-browsing actions in each task prior to a successful submission, aggregated by the type of action and team. The rows in this table refer to the same types of actions as are depicted in Figure 7. The entry for *text: custom* signifies the use of speech transcription for the vitrivr team whereas the same entry refers to a switch in concept maps for ITEC. This is due to a bug in the respective logging subsystems.

In Table V it can be seen that, aggregated over all teams, the action that led to the most successful submissions was the textual search for a particular semantic concept, followed by an image-based search using global features. For example, the VIRET system often used example images from the result set or external sources to search for the most similar representative frames (indicated by "image: globalfeatures"). That example image could potentially be a part of a more complex temporal and/or multi-modal query.

D. Discussion

After two years of interaction logging at VBS, we would like to take a look back and summarize what can and cannot be achieved with the current logging efforts. Whereas a first "naive" attempt based on a simple log format in 2018

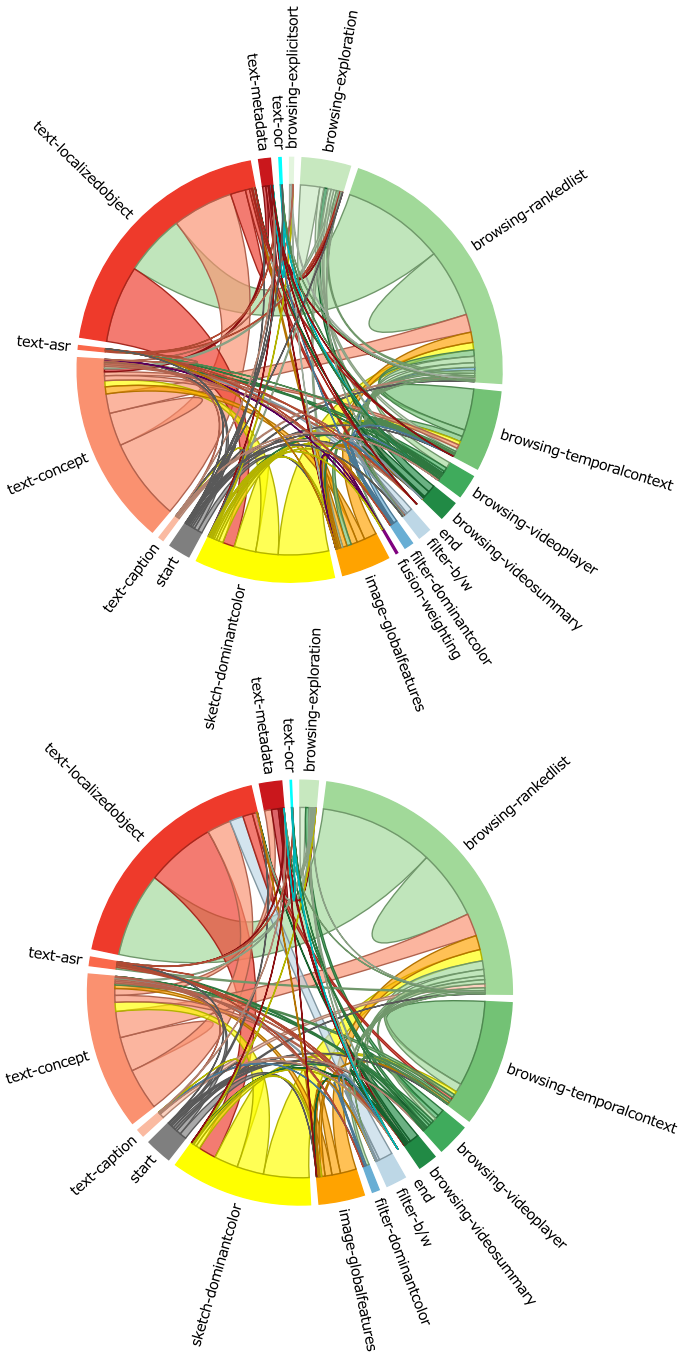


Figure 7: Bi-directional chord diagram of action category and type bi-grams for for expert (top) and novice (bottom) visual known-item search tasks.

demonstrated that simplified logging is a viable way to get some insight into the features that have been used during the competition, this second attempt has led to more thorough evidence of what actions were performed at a given time.

Provided with a simple, unified JSON based log format and a controlled vocabulary of action categories further divided into types, most of the VBS teams were able to integrate logging mechanisms into their tools to at least some extent. Hence, as the most remarkable achievement compared to the VBS era before interaction logging, the high-level statistics about search time and success rate (presented in Figure 2) measured on the server side, can be complemented with evidence of system components and models that have been used during the search. Hence, log analysis can finally reveal which models from the tool descriptions actually contribute to the final results in the competition. For example, it became very clear that speech transcription was used by the vitrivr team to successfully solve seven tasks, whereas the new query by semantic sketch functionality was rarely ever used at all. In addition, we can also estimate basic transitions between various modes of interaction during the competition. However, there are still several issues that prevent a more sophisticated analysis and comparative evaluation of the tools.

Firstly, despite the unified log format, there is still the issue of different logging implementations that log specific cases differently. In particular, the implementations can vary in how specific instances of implicit or transitive interactions are logged and what logging frequency is used. For example, a tool can skip the logging of some interaction type or a longer stream of consecutive actions of the same type, such as scroll actions. Another issue is with log records for a sequence of implicit query processing steps, e.g., search by text followed by an (implicit) filter to the top 10% of the dataset, where the tools can either log all steps or just the initial query formulation step. Some user interfaces also provide a more complex presentation logic where a single interaction on one panel may transitively trigger multiple changes, which again can be logged in different ways as with implicit actions. In addition, there is always the possibility that teams misunderstand (or incorrectly implement) the mapping of some type of interaction with regard to the list of suggested categories and types.

Secondly, the logs can be incomplete due to technical issues and missing log recovery protocols within the tools. For example, some logged records were lost due to network issues, while other tools that maintained logs just in main memory crashed during the competition.

Hence, using the current logging approach, it is not always possible to fully reconstruct the complete interaction history and to compare tools in terms of their level of interactivity and the variability of models that were employed. So far, the logs help to clarify how one particular task was solved by a specific team and which actions were used (or, more specifically, logged) during the competition. However, every attempt at using the current state of the logs to objectively and quantitatively assess and compare the efficiency and effectiveness of the tools would be futile.

Since newly participating tools and teams may also bring new approaches and may be revealing new issues, we currently admit that these logs should be thought of as informative in nature rather than being a means to obtain an exact measure of a tool's performance. Detailed comparative interaction studies require very comprehensive synchronization in the preparation of participating tools, which is hard to achieve for a larger event such as VBS. Regardless, in addition to interaction logging, we plan to introduce a new result set logging for VBS 2020, which captures the ranked list of results produced by each tool (i.e., store top k ranked frames and the employed

query settings). Such logs could be used to additionally compare the tools in terms of their ranking models as it would enable us to track the target frame of a task in the provided result set, e.g., as proposed in [66]. Such additional logs could provide valuable information on a tool's effectiveness from different perspective.

VII. Conclusion

In this paper, we presented a detailed analysis of the Video Browser Showdown 2019. The analysis based on the logged UI interactions per team as well as the overall scores of the competition show that there are still multiple approaches that can be used for efficient and effective, interactive video retrieval and that no dominating strategy has emerged. It can, however, also be observed that there is a clear increase in the use of retrieval approaches that are somehow based on deep learning methods, such as object recognition or scene captioning, and that textual input is the preferred method of query formulation. All participating teams relied, among other methods, on semantic annotations of various forms. Other methods based on deep learning approaches, such as the extraction of features with a high salience that can be represented as text, e.g., the transcript of a spoken dialog or text displayed in the video, have also been shown to be highly effective targets to base queries on. This effectiveness is probably a consequence of the intuitive and (near) loss-less query representation. Another promising approach for known-item search is temporal query formulation [66], for which users describe two consecutive shots and temporal closeness of results is taken into account.

Despite this trend, however, each team still spends most of the time during the competition on browsing the retrieved results and manually identifying and selecting the relevant item(s). This leads to the conclusion that it is still not feasible to efficiently formulate just a single query that is specific enough for the desired item to be ranked among the very top results in the list. This is especially true with the increased collection size of V3C1 and the inherently increased potential for false positives and noise. The need to go back and forth between querying and browsing also shows that interactive exploration and refinement are still a necessary part of the retrieval process. It is therefore not surprising, that UI/UX design considerations become important factors for the outcome of the competition. This can at least be interpreted into the results for the novice session, in which randomly selected users operated a tool they were not familiar with. For this session, the score discrepancies between the different teams tended to be larger than for the expert session tasks.

Finally, we can conclude that the further formalization and specification of the logging mechanism, as compared to the previous iteration of VBS [9] last year, enabled us to conduct a much more thorough and detailed analysis of user interactions. We have seen a substantial improvement of data quality and, hence, only little pre-processing was required. However, there are still some challenges such as the synchronization of time across participants as well as modelling the wide range of different modes of interaction into a common, controlled vocabulary. While the current logging mechanism already constitutes an improvement over the one used in 2018, it still requires further refinement in order to be able to produce detailed insights into the various search strategies employed by the teams as well as their relative effectiveness. Furthermore, it became apparent that complementing the interaction logs with logs of the results list might be highly beneficial for future analyses. Consequently, the experience encourages us to put even more thought and effort into further improving the logging system prior to next year's installment of VBS.

Regardless of the substantially larger V3C1 video collection comprising 1000 hours of video, as compared to the 600 hours for the previously used IACC.3, two systems still managed to solve 18 out of 23 evaluated known-item search tasks. Considering just visual expert known-item search tasks, the observed success rate was even higher for both systems, which relied mostly on query (re)formulation. This might also imply that the current setting used for visual known-item search, where users observe the "known" scene played in the loop on a data projector, is too easy to be considered a realistic case. Future instances of VBS might therefore reconsider the way in which these search tasks are presented.

Acknowledgments

Part of this work was supported by Czech Science Foundation (GAČR) project 19-22071Y, and by projects that have received funding from the European Unions Horizon 2020 research and innovation programme: V4Design under grant agreement No 779962, and MARCONI under grant agreement No 761802.

References

[1] K. Schoeffmann, F. Hopfgartner, O. Marques, L. Boeszoermenyi, and J. M. Jose, "Video browsing interfaces and applications: a review," *Spie Reviews*, vol. 1, no. 1, p. 018004, 2010.

[2] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, Nov 2011.

[3] S.-Y. Chien, Y.-W. Huang, B.-Y. Hsieh, S.-Y. Ma, and L.-G. Chen, "Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques," *IEEE Transactions on Multimedia*, vol. 6, no. 5, pp. 732–748, 2004.

[4] X. Ke, J. Zou, and Y. Niu, "End-to-end automatic image annotation based on deep cnn and multi-label data augmentation," *IEEE Transactions on Multimedia*, 2019.

[5] Y. Chen, C. Hao, A. X. Liu, and E. Wu, "Multi-level model for video object segmentation based on supervision optimization," *IEEE Transactions on Multimedia*, 2019.

[6] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, 2017.

[7] C. Cobârzan, K. Schoeffmann, W. Bailer, W. Hürst, A. Blazek, J. Lokoč, S. Vrochidis, K. U. Barthel, and L. Rossetto, "Interactive video search tools: a detailed analysis of the video browser showdown 2015," *Multimedia Tools Appl.*, vol. 76, no. 4, pp. 5539–5571, 2017.

[8] J. Lokoč, W. Bailer, K. Schoeffmann, B. Muenzer, and G. Awad, "On influential trends in interactive video retrieval: video browser showdown 2015–2017," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3361–3376, 2018.

[9] J. Lokoč, G. Kovalčík, B. Münzer, K. Schöffmann, W. Bailer, R. Gasser, S. Vrochidis, P. A. Nguyen, S. Rujikietgumjorn, and K. U. Barthel, "Interactive search or sequential browsing? a detailed analysis of the video browser showdown 2018," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 1, pp. 29:1–29:18, Feb. 2019.

[10] K. Schoeffmann, "Video browser showdown 2012-2019: A review," in *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, Sep. 2019, pp. 1–4.

[11] F. Berns, L. Rossetto, K. Schoeffmann, C. Beecks, and G. Awad, "V3c1 dataset: An evaluation of content characteristics," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. ACM, 2019, pp. 334–338.

[12] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *iccv*, vol. 99, no. 2, 1999, pp. 1150–1157.

[13] P. Mettes, D. C. Koelma, and C. G. Snoek, "The imagenet shuffle: Reorganized pre-training for video event detection," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ser. ICMR '16. New York, NY, USA: ACM, 2016, pp. 175–182.

[14] Y. Lou, Y. Bai, J. Lin, S. Wang, J. Chen, V. Chandrasekhar, L.-Y. Duan, T. Huang, A. C. Kot, and W. Gao, "Compact deep invariant descriptors for video retrieval," in *2017 Data Compression Conference (DCC)*. IEEE, 2017, pp. 420–429.

[15] Z. Dong, S. Jia, T. Wu, and M. Pei, "Face video retrieval via deep learning of binary hash representations," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[16] G. Wu, J. Han, Y. Guo, L. Liu, G. Ding, Q. Ni, and L. Shao, "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1993–2007, 2018.

[17] W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang, "Effective deep learning-based multi-modal retrieval," *The VLDB Journal – The International Journal on Very Large Data Bases*, vol. 25, no. 1, pp. 79–101, 2016.

[18] X. Xu, L. He, H. Lu, L. Gao, and Y. Ji, "Deep adversarial metric learning for cross-modal retrieval," *World Wide Web*, vol. 22, no. 2, pp. 657–672, 2019.

[19] X. He, Y. Peng, and L. Xie, "A new benchmark and approach for fine-grained cross-media retrieval," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1740–1748.

[20] O. Seddati, S. Dupont, and S. Mahmoudi, "Deepsketch 3," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22 333–22 359, 2017.

[21] R. Furuta, N. Inoue, and T. Yamasaki, "Efficient and interactive spatial-semantic image retrieval," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18 713–18 733, 2019.

[22] L. Rossetto, R. Gasser, and H. Schuldt, "Query by semantic sketch," *arXiv preprint arXiv:1909.12526*, 2019.

[23] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[24] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[25] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2014.

[28] W. Zhang, H. Zhang, T. Yao, Y. Lu, J. Chen, and C.-W. Ngo, "Vireo@trecvid 2014: instance search and semantic indexing," in *In NIST TRECVID Workshop*, 2014.

[29] S. Strassel, A. Morris, J. Fiscus, C. Caruso, H. Lee, P. Over, J. Fiumara, B. Shaw, B. Antonishek, and M. Michel, "Creating havic: Heterogeneous audio visual internet collection," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC12*, 2012.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[31] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[32] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[33] A. Kuznetsova, H. Rom, N. Alldrin, J. R. R. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, "The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale," *CoRR*, vol. abs/1811.00982, 2018.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[35] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," *CoRR*, vol. abs/1711.10305, 2017.

[36] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[37] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *CoRR*, vol. abs/1705.06950, 2017.

[38] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 961–970.

[39] G. Ye, Y. Li, H. Xu, D. Liu, and S.-F. Chang, "Eventnet: A large scale structured concept library for complex event detection in video," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 471–480.

[40] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[41] J. Dong, X. Li, and C. G. M. Snoek, "Predicting visual features from text for image and video caption retrieval," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3377–3388, Dec 2018.

[42] X. Li, C. Xu, G. Yang, Z. Chen, and J. Dong, "W2VV++: fully deep learning for ad-hoc video search," in *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, 2019, pp. 1786–1794.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[44] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *ArXiv*, vol. abs/1908.02265, 2019.

[45] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Qunot, J. Magalhaes, D. Semedo, and S. Blasi, "Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search," in *Proceedings of TRECVID 2018*. NIST, USA, 2018.

[46] L. Rossetto, H. Schuldt, G. Awad, and A. A. Butt, "V3C – A Research Video Collection," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 349–360.

[47] P. Over, G. Awad, A. F. Smeaton, C. Foley, and J. Lanagan, "Creating a web-scale video collection for research," in *Proceedings of the 1st workshop on Web-scale multimedia corpus*, 2009, pp. 25–32.

[48] L. Rossetto and H. Schuldt, "Web video in numbers-an analysis of web-video metadata," *arXiv preprint arXiv:1707.01340*, 2017.

[49] L. Rossetto, M. A. Parian, R. Gasser, I. Giangreco, S. Heller, and H. Schuldt, "Deep learning-based concept detection in vitrivr," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 616–621.

[50] J. Lokoč, G. Kovalčík, T. Souček, J. Moravec, and P. Čech, "VIRET: A video retrieval tool for interactive known-item search," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019, Ottawa, ON, Canada, June 10-13, 2019.*, 2019, pp. 177–181.

[51] P. A. Nguyen, C.-W. Ngo, D. Francis, and B. Huet, "Vireo@ video browser showdown 2019," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 609–615.

[52] G. Amato, P. Bolettieri, F. Carrara, F. Debole, F. Falchi, C. Gennaro, L. Vadicamo, and C. Vairo, "VISIONE at VBS2019," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 591–596.

[53] K. Schoeffmann, B. Münzer, A. Leibetseder, J. Primus, and S. Kletz, "Autopiloting feature maps: The deep interactive video exploration (diveXplore) system at vbs2019," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 585–590.

[54] S. Andreadis, A. Moumtzidou, D. Galanopoulos, F. Markatopoulou, K. Apostolidis, T. Mavropoulos, I. Gialampoukidis, S. Vrochidis,

V. Mezaris, I. Kompatsiaris *et al.*, "Verge in vbs 2019," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 602–608.

[55] L. Rossetto, I. Giangreco, and H. Schuldt, "Cineast: a multi-feature sketch-based video retrieval engine," in *Multimedia (ISM), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 18–23.

[56] J. Lokoč, T. Souček, and G. Kovalčík, "Using an interactive video retrieval tool for lifelog data," in *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*, 2018, pp. 15–19.

[57] A. Leibetseder, S. Kletz, and K. Schoeffmann, "Sketch-based similarity search for collaborative feature maps," in *International Conference on Multimedia Modeling*. Springer, 2018, pp. 425–430.

[58] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *CoRR*, vol. abs/1707.07012, 2017.

[59] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[60] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 652–663, 2017.

[61] F. Markatopoulou, V. Mezaris, and I. Patras, "Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[62] F. Markatopoulou, D. Galanopoulos, V. Mezaris, and I. Patras, "Query and keyframe representations for ad-hoc video search," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 2017, pp. 407–411.

[63] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 4724–4733.

[64] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[65] S. Albitar, S. Fournier, and B. Espinasse, "The impact of conceptualization on text classification," in *International Conference on Web Information Systems Engineering*. Springer, 2012, pp. 326–339.

[66] J. Lokoč, G. Kovalčík, T. Souček, J. Moravec, and P. Čech, "A framework for effective known-item search in video," in *In Proceedings of the 27th ACM International Conference on Multimedia (MM19), October 21-25, 2019, Nice, France*, 2019, pp. 1–9.

[67] L. Rossetto, I. Giangreco, C. Tanase, and H. Schuldt, "vitrivr: A flexible retrieval stack supporting multiple query modes for searching in multimedia collections," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1183–1186.

[68] L. Rossetto, I. Giangreco, R. Gasser, and H. Schuldt, "Open-source column: content-based multimedia retrieval using vitrivr," *ACM SIGMultimedia Records*, vol. 9, no. 3, p. 8, 2018.

[69] R. Gasser, L. Rossetto, and H. Schuldt, "Towards an all-purpose content-based multimedia information retrieval system," *arXiv preprint arXiv:1902.03878*, 2019.

[70] ——, "Multimodal multimedia retrieval with vitrivr," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. ACM, 2019, pp. 391–394.

[71] L. Rossetto, I. Giangreco, C. Tănase, H. Schuldt, S. Dupont, and O. Seddati, "Enhanced retrieval and browsing in the imotion system," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 469–474.

[72] L. Rossetto, I. Giangreco, H. Schuldt, S. Dupont, O. Seddati, M. Sezgin, and Y. Sahillioğlu, "Imotion - a content-based video retrieval engine," in *International Conference on Multimedia Modeling*. Springer, 2015, pp. 255–260.

[73] L. Rossetto, M. Amiri Parian, R. Gasser, I. Giangreco, S. Heller, and H. Schuldt, "Deep Learning-based Concept Detection in vitrivr at the Video Browser Showdown 2019 - Final Notes," *arXiv preprint arXiv:1902.10647*, 2019.

[74] T. Souček, J. Moravec, and J. Lokoč, "TransNet: A deep network for fast detection of common shot transitions," *CoRR*, vol. abs/1906.03363, 2019.

[75] Y.-J. Lu, P. A. Nguyen, H. Zhang, and C.-W. Ngo, "Concept-based interactive search system," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 463–468.

[76] P. A. Nguyen, Y.-J. Lu, H. Zhang, and C.-W. Ngo, "Enhanced vireo kis at vbs 2018," in *International Conference on Multimedia Modeling*. Springer, 2018, pp. 407–412.

[77] P. A. Nguyen, C.-W. Ngo, D. Francis, and B. Huet, "Vireo@ video browser showdown 2019," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 609–615.

[78] D. Francis, B. Huet, and B. Merialdo, "Gated recurrent capsules for visual word embeddings," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 278–290.

[79] G. Amato, F. Falchi, C. Gennaro, and F. Rabitti, "Searching and Annotating 100M Images with YFCC100M-HNfc6 and MI-File," in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, ser. CBMI '17. New York, NY, USA: ACM, 2017, pp. 26:1–26:4.

[80] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[81] ——, "YOLOv3 on the Open Images dataset," https://pjreddie.com/darknet/yolo/, 2018, [Online; accessed 28-February-2019].

[82] ——, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[83] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879*, 2015.

[84] G. Amato, P. Bolettieri, F. Falchi, C. Gennaro, and F. Rabitti, "Combining local and global visual feature similarity using a text search engine," in *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2011, pp. 49–54.

[85] G. Amato, F. Falchi, C. Gennaro, and L. Vadicamo, "Deep permutations: deep convolutional neural networks and permutation-based indexing," in *International Conference on Similarity Search and Applications*. Springer, 2016, pp. 93–106.

[86] G. Amato, F. Debole, F. Falchi, C. Gennaro, and F. Rabitti, "Large scale indexing and searching deep convolutional neural network features," in *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 2016, pp. 213–224.

[87] K. Schoeffmann, B. Münzer, A. Leibetseder, J. Primus, and S. Kletz, "Autopiloting feature maps: the deep interactive video exploration (diveXplore) system at vbs2019," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 585–590.

[88] K. Schoeffmann, D. Ahlström, and M. A. Hudelist, "3-d interfaces to improve the performance of visual known-item search," *IEEE Transactions on Multimedia*, vol. 16, no. 7, pp. 1942–1951, 2014.

[89] K. Schoeffmann, B. Münzer, M. J. Primus, S. Kletz, and A. Leibetseder, "How experts search different than novices–an evaluation of the diveXplore video retrieval system at video browser showdown 2018," in *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2018, pp. 1–6.

[90] M. J. Primus, B. Münzer, A. Leibetseder, and K. Schoeffmann, "The itec collaborative video search system at the video browser showdown 2018," in *International Conference on Multimedia Modeling*. Springer, 2018, pp. 438–443.