REGULAR PAPER

Interactive video retrieval in the age of effective joint embedding deep models: lessons from the 11th VBS

Jakub Lokoč[1] · Stelios Andreadis[10] · Werner Bailer[4] · Aaron Duane[6] · Cathal Gurrin[5] · Zhixin Ma[11] ·
Nicola Messina[2] · Thao-Nhu Nguyen[5] · Ladislav Peška[1] · Luca Rossetto[8] · Loris Sauter[3] ·
Konstantin Schall[9] · Klaus Schoeffmann[7] · Omar Shahbaz Khan[6] · Florian Spiess[3] · Lucia Vadicamo[2] ·
Stefanos Vrochidis[10]

Abstract

This paper presents findings of the eleventh Video Browser Showdown competition, where sixteen teams competed in known-item and ad-hoc search tasks. Many of the teams utilized state-of-the-art video retrieval approaches that demonstrated high effectiveness in challenging search scenarios. In this paper, a broad survey of all utilized approaches is presented in connection with an analysis of the performance of participating teams. Specifically, both high-level performance indicators are presented with overall statistics as well as in-depth analysis of the performance of selected tools implementing result set logging. The analysis reveals evidence that the CLIP model represents a versatile tool for cross-modal video retrieval when combined with interactive search capabilities. Furthermore, the analysis investigates the effect of different users and text query properties on the performance in search tasks. Last but not least, lessons learned from search task preparation are presented, and a new direction for ad-hoc search based tasks at Video Browser Showdown is introduced.

Keywords Interactive video retrieval · Video browsing · Video content analysis · Content-based retrieval · Evaluations

1 Introduction

While video portals like YouTube can easily grow by hundreds of hours of video content per minute, there is still a shortage of effective video retrieval models providing access to contents of huge volumes. Whereas for domain-specific data, it is possible to collect large training datasets and try to train an effective model, for general domain datasets, it is hard to build a universal search approach. Nevertheless, there are attempts to train models with huge volumes of training pairs (item; text description) [1, 36, 45, 62]. Despite these efforts, it is worth noting that even with a perfect cross-modal text-video search model there are other limitations affecting search effectiveness. Three important factors are limitations of human memory (i.e., ability to remember all details), language skills and density of items in the dataset (i.e., ability to identify a correct item in a larger cluster).

The analysis of annual reports from TRECVID [57], Video Browser Showdown (VBS) [31], or Lifelog Search Challenge [25] shows that there still exist many practical situations where state-of-the-art retrieval approaches do not provide sufficient results even for far smaller datasets. Hence, it is essential to continue with the development of new video retrieval models/tools as well as with evaluation efforts providing important performance insights.

This paper presents a thorough analysis of the 11th instance of the Video Browser Showdown competition with participating tools combining interactive search interfaces and ranking approaches based on deep machine learning models. The competition setting consists of a known large video dataset (V3C [66]) and evaluation methodology allowing fair comparison of participating systems. During the competition, all the teams have access to a distributed evaluation server [64], where competition tasks are presented to participating teams at the same time and with the same time limit. In particular, three task categories are evaluated [52]:

– Visual known-item search (KIS-v), where teams observe a target video segment from the collection. No meta-data is provided, no cameras are allowed.
– Textual known-item search (KIS-t), where teams receive a text description of a target video segment. The text is gradually extended.
– Ad-hoc video search (AVS) tasks introduced with a short text description, where teams have to submit as many correct shots (matching the description) as possible.

A known-item search task is considered as solved by a team once the team submits a correct frame/shotID from the target segment. Incorrect submissions result in a penalty deducted from the score. The evaluation server knows the target segment and thus evaluates submissions in known-item search tasks automatically. In the ad-hoc search category, teams submit all items where team members think the item is correct. Since the ground truth is unknown for the whole collection, live judges are necessary to assess the submissions. Teams receive points for each correct submission (merging temporally close submissions into ranges counted only once), independent of whether other teams also found the same segment. The scoring is described in more detail in [31].

With these settings, the Video Browser Showdown hosted sixteen teams during the 2022 International Conference on Multimedia Modeling in Vietnam, where several teams (or members) participated remotely. All teams introduced a unique video search tool and some teams also implemented logging mechanisms. Hence, we collected a non-trivial amount of data from the competition which allows us to present the following key contributions, each in a separate section:

– A broad survey of multimedia search models and approaches participating in the 11th VBS.
– Overall summary of the competition results, showing success rate, submission times, and numbers of submissions.
– Thorough result log analysis of selected teams, revealing performance insights as well as query statistics.
– Analysis of ad-hoc search category, showing timeline statistics and also a new revision of the task category.
– Query specification methodology in connection with a qualitative study.

The last section concludes the paper and envisions future settings of the Video Browser Showdown.

2 Related work used by participating systems

VBS 2022 hosted many participating systems, each implementing different ranking models and search methods. A general overview of the systems and the approaches they employed is presented in Table 1. This section further summarizes important or unique methods used by each participant. For additional detail about any system, please see the corresponding publication referenced beside the system name in the overview table.

2.1 Concept search

This section summarizes the concept-based search approaches utilised by participating systems, including concepts detected for the whole image as well as localized information obtained from object detectors or semantic segmentation.

VISIONE [4], as in previous years [2, 3, 5], supports queries by object location appearing in a target scene. That is done by drawing simple diagrams on a canvas to specify objects (including their spatial locations). The object detection technique of *VISIONE* is based on three pre-trained DCNN models (i.e., VfNet [80], Mask R-CNN [27], Faster R-CNN [22]) for a total of 1460 object classes. Similarly, *VERGE* employs three different DCNN models (i.e., EfficientNet-B3, EfficientNet-B5 [74] for label-based search of video shots and InceptionResNetV2 [73]) for keyframes' enrichment with concept [56] and object [79] annotations.

IVIST [42], *AVSeeker* [41], *diveXplore* [43] use MS-COCO [46]. In particular, *IVIST* adopts an HTC [14] object detection model which is pre-trained on MS-COCO and supports an object query function to filter the frames which do not contain the query object categories. In contrast, *AVSeeker* [41] indexes object concepts on all keyframes into an Elasticsearch node, using categories from MS-COCO (detected by YOLOv4). This allows users to search for concepts using advanced query formulations such as customized AND/OR operators, fuzzy matching, negation, etc. These are provided by the "Query String Query" of Elasticsearch. Similarly, *diveXplore* [43] provides object search for object categories from MS-COCO (detected by YOLOv5).

VERGE [6] involves spatio-temporal human activity recognition using a 3D-CNN architecture. This construction relies on a three-step pipeline [24]: object detector, object tracker, and activity recognizer to identify human-related activities effectively.

As before, *VISIONE* [4] supports queries by color location [2, 3, 5]. A user can draw simple diagrams on a canvas

Table 1 List of participating teams and video search approaches

	Country	Overall score	Solved KIS tasks	Members	Concept search	Embedding	Temporal query	Relevance Feedback	Query by example	Other	Browsing			
											Ranked list	Video summary	Video player/preview	Temporal context
Vibro [33]	DE	300	30	2		✓	✓	✓	✓		✓	✓	✓	✓
CVHunter [49]	CZ	278	30	2		✓	✓	✓	✓		✓	✓	✓	✓
VISIONE [4]	IT	260	29	2	✓	✓	✓		✓		✓	✓	✓	✓
IVIST [42]	KR	249	25	2	✓	✓				✓	✓		✓	✓
AVSEEKER [41]	IE	207	25	2	✓	✓		✓	✓		✓		✓	✓
V-FIRST [75]	VN	200	26	2		✓	✓		✓		✓	✓		
VideoFall [60]	IE	197	25	2	✓	✓	✓		✓	✓	✓	✓	✓	
VERGE [6]	GR	176	24	2	✓	✓	✓		✓	✓	✓	✓	✓	
vitrivr [28]	CH	175	21	2	✓	✓	✓		✓	✓	✓	✓	✓	✓
VNUHCM [54]	VN	161	22	2		✓	✓		✓	✓	✓		✓	✓
VIREO [55]	SG	158	16	2	✓	✓				✓	✓			
AIClub@UIT [34]	VN	146	22	2		✓	✓		✓	✓	✓	✓		
Vitrivr-VR [71]	CH	137	20	2				✓	✓	✓		✓	✓	✓
DiveXplore [43]	AT	75	14	2	✓		✓				✓	✓	✓	
Exquisitor [40]	DK	72	14	1	✓			✓		✓		✓		
ViRMA [19]	DK	21	8	1	✓									✓

corresponding to colors appearing in a target scene. For the color annotation, two chip-based color naming techniques [11, 76] are employed. *vitrivr* maintains similar color-based sketches, as in previous iterations of the system [29, 30, 65, 67].

VIREO [55] supports color sketch queries using a grid of 48 cells (6x8), which users can individually fill with colors corresponding to their search target. Similarly, *VERGE* [6] maintains color-based queries using a grid of 9 cells (3x3). Video clips matching the colors in these positions are then assigned to a higher rank.

IVIST [42] enables a color query function to find frames depending on whether their top 3 dominant colors are included in the query colors or not. *VideoFall* [60] also affords users with color search based on specific dominant colors. The annotation for dominant colors in *VideoFall*, specifically 12 basic colors, is captured by integrating a *k*-means clustering algorithm to the set of frame-pixel values.

vitrivr [28] supports semantic-based sketches, as in previous iterations of the system. Semantic sketches are based on a DeepLab segmentation model [15] utilized as described in [65].

diveXplore [43] provides concept-based search for concepts in ImageNet-1000 [18] and Places365 (detected by EfficientNet-B2 [74]). Also, *Exquisitor* and *ViRMA* both support concept search for 12,988 ImageNet concepts, which were extracted for each keyframe using a pre-trained DCNN ResNet model [59]. To support its browsing and data model in VR, *ViRMA* further organises these concepts into a hierarchical structure using semantic relationships derived from WordNet [19]. Finally, in addition to the ImageNet concepts, *Exquisitor* also maintains search for activity concepts of Kinetics-700, extracted from the video shots using a pre-trained 3D-ResNet model [26].

VIREO [55] allows users to perform search using a bank of 16,263 concepts. These are extracted by the concept decoder of the dual task model [78].

VideoFall [60] affords users with textual search and the visible textual information in the frames is extracted using Google Vision API.[1]

2.2 Embedding

In this section, we discuss joint embedding approaches, which combine text and image/video processing architectures with the objective of mapping the same semantic information to similar vectors.

The top three scoring systems, *vibro*, *CVHunter* and *VISIONE*, including a large number of other systems (*AVSeeker*, *V-FIRST*, *VNUHCM* and *AIClub@UIT*) all use networks derived from CLIP [62]. The text query phrases are transformed into a joint text-image vector space with cosine similarity. Specifically, *VISIONE* uses TERN [58] for text-image retrieval.

In addition to this, *VISIONE* uses CLIP2video [20] for text-video retrieval. Similarly, *IVIST* and *VideoFall* use networks based on CLIP too [62], where input text queries are matched with videos in a joint text-video vector space.

VERGE's [6] text-video matching module translates a complex textual query and the videos into a joint latent space for direct comparison. Next, it utilizes the attention-based dual encoding network [21]. In contrast, *VIREO* [55] uses the dual-task model [78] for the same text-video retrieval task.

vitrivr and *vitrivr-VR* both rely on a custom visual-text co-embedding model [72] inspired by approaches like W2VV++ [44]. In comparison to CLIP-based approaches, the embedding models are much simpler, resulting in lower hardware requirements.

2.3 Temporal querying

Since VBS tasks can comprise longer target video sequences (up to 20 s), some systems can address multiple items in the target sequence at once using a temporal query.

vibro [33] employs a two-tab system in order to enable temporal queries. Each tab can formulate queries of the supported modalities and produces an individual ranked order of keyframes. If both tabs contain a query and a result list, consecutive sequences of keyframes from a single video are ranked according to the probability that the sequence contains content from the first tab's result list followed by content from the second tab in an adjustable time range.

CVHunter [49] supports two options to address a sequence of video segments: a context-aware ranker that supports unordered specification of target segments, and its special case, temporal query [50], where query parts are ordered in the same way as the searched sequence of segments. Both approaches require distances from all query parts to all selected frames. However, it is worth noting that based on the VBS log analysis, the context-aware ranker was rarely used.

VISIONE [4] uses a time quantization approach to support temporal queries, where each video is divided into intervals of 7 s. Given two queries, the temporal search is performed in two steps. First, the two queries are processed independently, and for each query, just the result with the maximum score is kept for each time interval. Second, the results of the two queries that are temporally close are then combined into pairs, and just a sample of distinctive pairs is kept in the final result list.

V-FIRST [75] simply allows the user to input two separate queries, then uses a weighted sum of the two queries to

---

generate ordered pairs of images in a video and return them for the user to browse.

VERGE [6] limits temporal queries to concepts; namely, the user is able to query for two concepts that should appear in subsequent shots of the same video. For each concept, a separate list of shot probabilities is created, then the intersection of concepts per video is computed, and finally, shots are re-ranked through an objective function.

vitrivr's [28] temporal queries are formulated using two or more blocks and upon presentation of the results, users have to switch to a dedicated temporal query result view [30]. In contrast, *vitrivr-VR*'s [71] temporal queries are formulated by grabbing and ordering small representations of query terms in virtual space. Temporal scoring is performed as described in [28]. The results are then presented to the user as stacks of temporally aligned segments that are relevant to the query.

VIREO [55] first measures the cosine similarity for the two successive queries independently. The two distributions of keyframes will be aggregated using a sliding window to produce the final probability.

Exquisitor's [40] temporal queries are defined by the user training two relevance feedback models, focusing on different aspects of the desired shot. Once the models are defined, the results of each model will highlight the shots which come from the same video. In addition, temporal constraints can be better utilized to specify the desired target shot [37].

2.4 Relevance feedback

Once results are displayed in a video retrieval system, relevance feedback tools enable users to provide feedback in the form of positive or negative examples. Compared to kNN-based browsing, this feedback updates the model or the current score rather than issuing a new independent query.

vibro [33] only uses relevance feedback for AVS tasks, where all presented results have to be marked as positive or negative by the user after an initial query, and all positive keyframes are used to produce a consecutive result list.

CVHunter [49] implements a Bayesian-like approach [16] to accumulate relevance scores for each representative frame in the dataset. A temporal variant [53] of this relevance feedback approach was successfully tested in the system as well.

V-FIRST [75] has an optional pseudo-relevance feedback feature, where it assumes the top-k (with $k = 10$) initial results are relevant and reformulates the query by taking their centroid. This can be useful to cluster a small set of correct answers to the top ranks.

Exquisitor [40] uses relevance feedback as its primary interactive approach for search, where it trains a linear SVM model to construct a hyperplane to retrieve the most relevant items [38]. With multiple modalities involved, an SVM model for each modality is used to get candidates, which are then fused using rank aggregation. For VBS 2022, *Exquisitor* uses two visual modalities, semantic concepts from ImageNet and actions from Kinetics-700.

2.5 Query by example

Many VBS systems allow query reformulation, where users select an example item from the currently displayed candidate set. The essential part of this method is a similarity model assigning a similarity score for two items.

vibro [33] uses a Swin [48] architecture that has been fine-tuned for content-based image retrieval for visual similarity search. The final embedding was binarized and concluded to 1024 bits for each vector.

For visual similarity of two items, *CVHunter*, *AVSeeker*, *V-FIRST*, *VideoFall* and *AIClub@UIT* all use the same CLIP feature vectors [62] as was used for text search.

VISIONE [4] supports both visual similarity search, where the user can use an image as a query to search for video keyframes visually similar to it, and a semantic similarity search, where an image can be used to retrieve video keyframes or video clips that are semantically similar to it. The visual similarity search is based on comparing GEM [63] features. For the semantic similarity CLIP2Video [20] and TERN [58] are used for searching video clips and video keyframes, respectively.

In *VERGE*'s [6] system, the visual similarity search module enables the retrieval of visually similar content starting from a query image and considers feature vectors produced from a fine-tuned GoogleNet architecture [61] and an effective IVFADC indexing structure [35].

vitrivr [28] provides two modalities for query-by-example. One enables users to simply upload a sample image to find visually similar items, and the other operates via a "more-like-this" button positioned next to results.

vitrivr-VR [71] allows querying by frames of already retrieved videos through a similarity search. The feature used for this more-like-this search can be configured and was set to simple color and edge features for VBS 2022.

VIREO [55] calculates the cosine similarity of the dual-task model's [78] embedding feature and indexes the KNNs for visual similarity search.

diveXplore [43] provides content similarity search with GoogleNet neural codes from ImageNet-1000, using the Manhattan distance to the selected example image.

2.6 Other

This section describes features and approaches which do not fall directly under any previous categories.

CVHunter [49] allows a fast inspection of top-k items from each video in the result set by pressing a number key (defining the k) on a numeric keyboard. For AVS tasks, the

tool supports fast selection of all visible items and selection of a database subset distinct from another team member.

IVIST [42] exploits a scene-text searching function to search frames that contain the query text in the corresponding scene by adopting PixelLink [17] and ASTER [70], so that users can try to find frames where specific scene texts exist. *VNUHCM* and *AIClub@UIT* also follow the approach of using textual information in video frames for retrieval.

VideoFall [60] introduces a method of submitting results in its video retrieval system, which involves two distinct interfaces prior to submission. The first interface is designed for users to input queries and explore the keyframe collection like normal, whereas the second interface is designed for users to verify the data received from the first interface and subsequently submit the final frame to the competition server.

VERGE [6] utilises a human and face detection module that aims to count the number of individuals in each frame by identifying their silhouettes and heads using the CrowdHuman dataset [69] and the YOLOv4 [12] deep neural network.

Both *vitrivr* and *vitrivr-VR* employ a novel query-by-pose approach [28, 71], allowing the specification of poses seen in the target clip. This pose-based query mode uses key-points extracted from segment keyframes using OpenPose [13]. In addition, vitrivr allows users to query by pose by dragging the key-points on a 2D canvas, while vitrivr-VR allows the posing of the key-points in 3D space, which are then projected with perspective on a camera-like canvas.

diveXplore [43] supports search for texts in OCR results detected with CRAFT [7, 8].

The *ViRMA* [19] prototype system employs a novel VR interaction approach by utilising the M^3 data model [23], which takes the media objects from the VBS dataset and maps them into a multidimensional media space based on their metadata. Users can then visualise the video data by filtering and dynamically projecting the multidimensional media space to the more familiar 3D space and then can explore this visualisation using virtual reality [19]. This type of 3D visualisation is effective at browsing and summarising a collection, but is less effective at search, which is likely why the ViRMA system did not perform well in VBS 2022.

2.7 Browsing

Table 1 contains four popular browsing approaches applied at VBS. The Ranked List simply refers to any visualization of the ranked result set. Video Summary refers to displaying a list of selected frames from a video. Video Player/Preview also refers to displaying frames, but at a higher frame rate (not just representative frames). Finally, the Temporal Context refers to the visualizations of resulting frames with the temporal neighborhood. Systems that have notable

variations on these four browsing approaches are discussed in this section.

vibro [33] allows browsing of result lists by displaying the 4000 most relevant keyframes to the current query in a list or on a 2D sorted map. Additionally, the entire keyframe collection can be explored with the help of a hierarchical graph [32]. A single click on each of the presented keyframes opens the corresponding video in the video section of the UI, where all keyframes are listed in chronological order, and the video can be viewed with a video player. Double clicks on keyframes will create a new result list and jump to the keyframes location in the exploration graph section.

VISIONE [4] groups the results by video so that one row (containing up to 20 frames) for each video is displayed in the browsing interface; the rows of videos and the frames in it are sorted according to the score given by the retrieval model. There is a menu on each frame that allows the user to do similarity searches, see the entire video starting with the selected frame, or see a preview of the video in a neighborhood of the selected frame.

IVIST [42] displays the top-100 lists of keyframes at once, organised into pages. Short video clips ($< 5s$), including each keyframe, are displayed as a GIF to provide temporal context. A keyframe can then be selected to display a video player function.

AVSeeker [41] generates a ranked-list of the top 2,048 keyframes that best match the query and groups them by video. The videos are then ranked by the average score of their top 3 best-matching keyframes to generate the final ranked list. These highest-scoring keyframes are also used as the preview of their corresponding video in the final result. Once a preview is clicked, a menu will pop up, which allows the user to see all matched keyframes, all keyframes of the video, and the video itself.

To expedite the process of elimination, *V-FIRST* [75] groups results by video up to a specific number of frames per video. Frames with high similarity are also removed to increase the variety of results.

VNUHCM [54] allows users to control the number of frames that are displayed. For each frame that is selected, a small video player of the corresponding timestamp is shown for users to interact with.

To allow fast and visually aided browsing within videos, *vitrivr-VR* provides a multimedia drawer view [71]. This video segment view, which resembles a VR drawer containing the most representative frames of the segments of a video, allows users to browse the segments of a video in 3D space simply by moving their hand through the drawer.

Exquisitor [40] displays the top 42 keyframes from its ranked list to the user. The user can either interact with the displayed keyframes to update the relevance feedback model, which will produce a new ranked list to get items from, or they can continue going through the current ranked

list one keyframe at a time or get the next 42 keyframes. For this VBS, *Exquisitor* does not provide a video player for the shots. Instead, it displays a shot with 1 to 5 frames depending on its length. In addition to this the next 2 shots' frames are also displayed below the selected shot [40]. Aside from the shot summary, a timeline browser for the video is available either as a vertical grid, or horizontal slider [39].

3 Overall results

In this section, we will discuss the final results of the VBS 2022 competition in detail. For this purpose, we analyze all three task types separately: KIS-v, KIS-t, and AVS (see Sect. 1).

We start with the overall scores, which are shown in Fig. 1 for all teams. We can identify three major groups of teams. The first group consists of the four top teams, who achieved more than 230 points. Among them is the *vibro* team, who was able to collect the maximum score in all three sessions: 100pts in KIS-v, KIS-t, and AVS, respectively. *vibro* is closely followed by *CVHunter* and *VISIONE*, who also reached a similar score for KIS-t (100pts and 90pts) and KIS-v (96pts and 100pts), but got fewer points for AVS (81pts and 74pts). The *IVIST* team, as the last one in this group, was also able to get the maximum score for AVS but achieved substantially fewer points in KIS-t (59pts), while scoring well in KIS-v (90pts).

In the second group, there are the teams that scored 210-137 points (*AVSeeker*, *V-FIRST*, *VideoFall*, *VERGE*, *vitrivr*, *VNUHCM*, *VIREO*, *AIClub*, and *vitrivr-VR*). For these teams, we can see a much lower and linearly decreasing score, with different difficulties per team. For example, the *vitrivr*, *VIREO*, and *AIClub* teams were challenged by
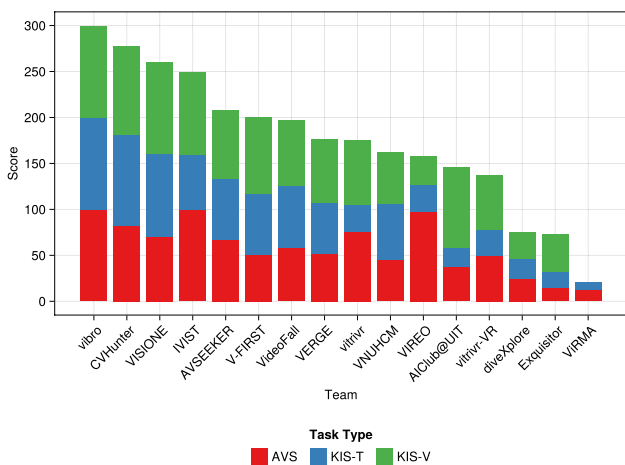
KIS-t, where they achieved only 30pts, 30pts, and 21pts, respectively. *VIREO* also had difficulties with KIS-v, where they achieved only 31pts.

Finally, in the third group, there are teams that were only able to collect up to 77 points: *diveXplore*, *Exquisitor*, and *ViRMA*. While *Exquisitor* was still okay in KIS-v (40pts), *ViRMA* could only score in KIS-t (9pts) and AVS (13pts). *diveXplore* scored in all three sessions but only with a low number of points (28pts, 23pts, and 24pts for KIS-v, KIS-t, and AVS).

From the number of submissions for KIS (Fig. 2), we can see that the teams in the first group were able to correctly solve 13 tasks in KIS-v, with only one wrong submission from *CVHunter*. The situation was different for KIS-t, where *vibro* submitted six wrong results, *CVHunter* one, *VISIONE* three, and *IVIST* two.

The teams in the second group were also very successful with KIS-v tasks but had substantially more wrong submissions (e.g., *V-FIRST* solved 12 KIS-v tasks correctly, but also had six wrong submissions). The *AIClub* team is an exception for KIS-v in this group: they solved 12 tasks correctly, with only two wrong submissions. For KIS-t, the situation was much worse than in the first group though: for many teams, the number of wrong submissions is higher than the number of correct ones (except *VERGE*, who could solve six KIS-t tasks correctly, with only one wrong submission).

In the last group of teams, the number of correct KIS submissions is generally low, except for *Exquisitor*, who could correctly solve six KIS-v tasks. It seems that *ViRMA* had serious difficulties with KIS-t, for which they submitted seven wrong submissions, while only one task could be solved correctly.

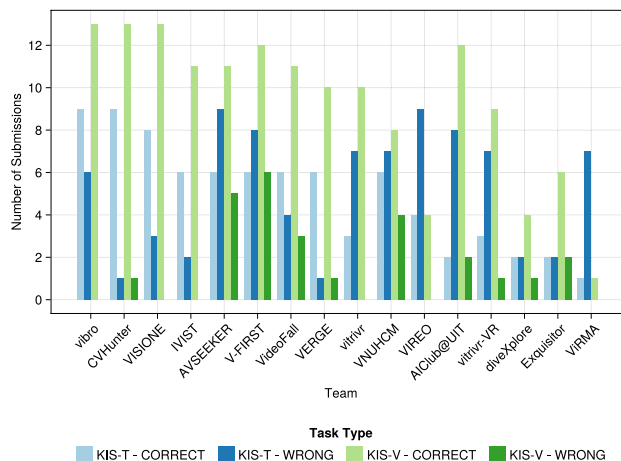When looking at the AVS tasks (Fig. 3), it is obvious that the *IVIST* team submitted most correct results (1851),

Fig. 1 Overall scores per team and task type

Fig. 2 Distribution of correct and incorrect submissions for known-item search tasks per team
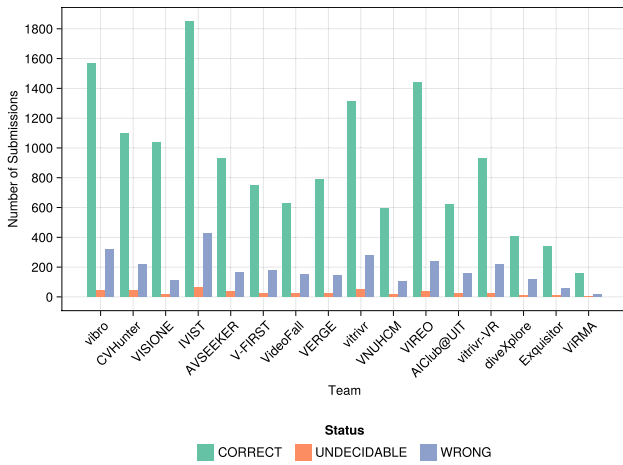
Fig. 3 Distribution of correct, incorrect, and undecidable submissions for Ad-hoc video search tasks per team

closely followed by *vibro* with 1568 submissions, while *CVHunter* and *VISIONE* submitted only 1095 and 1038 correct results, respectively. However, it is interesting that also most teams in the second group found many correct items for AVS: most notably *vitrivr* and *VIREO*, who submitted 1310 and 1437 correct AVS items, respectively. Most wrong submissions were made by the top-scorer in this session (*IVIST* with 425 wrong submissions). The number of undecidable submissions was generally low (at most 62), which is not only evidence of great team performance in general, but also proof of confidence of the AVS judging team.

It is worth noting that the submission time distribution provides deeper insights into how proficient each team's system performed during the real-time competition. The faster the system locates the target, the more efficient it is. As can be seen from Fig. 4, most teams had the shortest time to
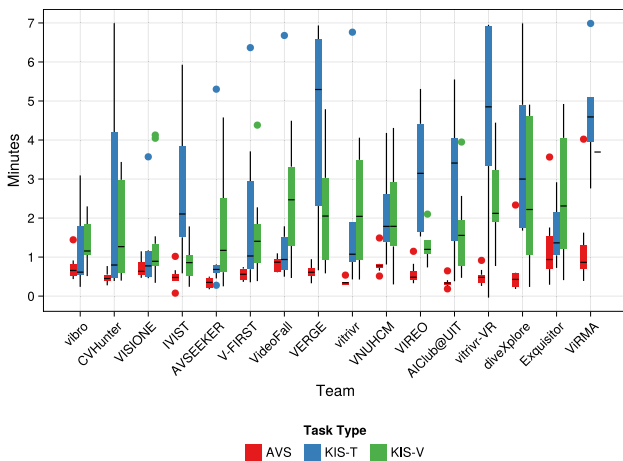
search for the AVS tasks regardless of the correctness, followed by the KIS-t and KIS-v tasks, respectively.

Figure 5 illustrates the distribution of time until the first correct submission across all teams and task type which excludes unsolved attempts. For the AVS tasks, the time is almost identical to the time to first submission meaning that many early submissions are correct. In contrast, the amount of time to find the correct answer for the other 2 tasks varies.

4 Analysis of KIS logs

As in previous years, during the competition, each team was asked to log the user interactions and the result sets of their queries for each task. Each team was given the choice of logging this data locally or sending it directly to the DRES competition server using a specific log format. In this section, we present the analysis of these logs to better understand the ranking performance of each system during KIS tasks.

The logs are in JSON format and each comprises the team identifier (in some cases, also the user identifier), timestamp, query description, and the list of top-ranked items that were retrieved for the query at hand. We report the analysis of result and query logs only for a subset of six teams (namely vibro, CVHunter, VISIONE, VERGE, vitrivr, vitrivr-VR). Unfortunately, the other teams did not log results in the common format or had incomplete/missing logs.

4.1 Log pe-processing

Of the six teams with logs, three saved the logs locally (vibro, CVHunter, VISIONE), while the others (VERGE, vitrivr, vitrivr-VR) sent the logs to DRES. We normalized the event timestamps of locally saved logs to the UNIX timestamp format used in DRES. Concerning possible
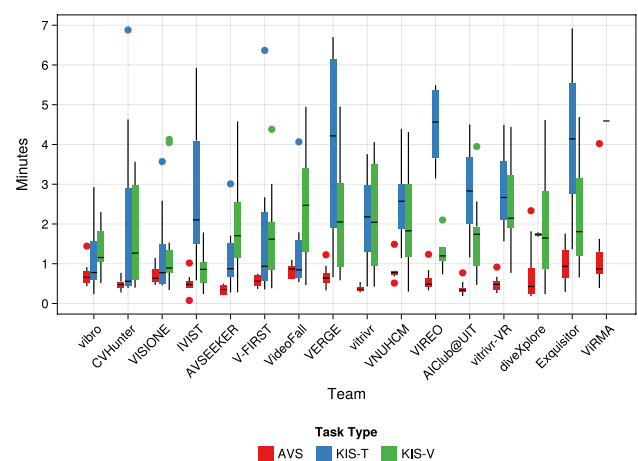

Fig. 4 Distribution of time until the first submission per team and task type

Fig. 5 Distribution of time until the (first) correct submission per team and task type

clock shifts between DRES and local clocks, we mostly rely on the synchronization performed by the OS. However, we carefully checked shifts in the submission timestamps and, according to our analysis, there might be only small differences (about 1 s). Therefore, we conclude that the presented times are consistent, and the slight shift does not affect the following analysis. We filtered data to only contain log entries that fall into the task duration interval and also removed all logs that comes after the correct submissions of respective teams.

Note that the set of logs collected may be incomplete (due to external circumstances or team log choices) and thus our analysis represents an approximation of all interactions and results of the tools. For example, VERGE experienced network problems during the competition and some of its logs were sent, but not received by the DRES server as evidenced by the fact that there are no logs of this team for an entire task (T6). Moreover, different teams used different logging parameters and units of retrieval. vibro logged only the top-1000 results for each query, while other teams logged the top-10,000 results. In our analysis we considered only the top-1000 results to keep the logging scale the same for all teams. Concerning the logging unit, vibro, CVHunter, and VISIONE logged frames; VERGE and vitrivr logged segments (predefined shots and custom shots with time intervals, respectively); vitrivr-VR logged both frame and predefined segments. In order to standardize the units of retrieval, we transform each of them into a temporal form. Specifically, if a frame is given, we convert the frame number into the corresponding physical time, using the frame rate metadata associated to each video, and we check if it ends inside the ground-truth interval. If a shot id has been logged, we convert the shot id to the corresponding temporal endpoints inside the video, using the provided shots metadata, and we check if the middle time of the submitted interval is inside the ground-truth interval. We note that during the competition a live judge was allowed to manually accept submissions from the same shot just outside (less than 3 s) the KIS ground truth segment boundary. However, these cases are rare and for the analysis of result log item correctness, the original official ground truth was utilized.

In the following analysis, it is also important to capture submissions not only at the level of the whole *team* but first and foremost at the level of the specific *user* who used the tool. This is important, as it may happen that if we collapse all the statistics of the whole team—two users as they were a single one—some inconsistencies may arise.

Throughout the following analysis, for the systems that logged the user ID (vibro, CVHunter, VISIONE, vitrivr-VR) and for each task, we labeled the user as *best* and *other*, where we define the best user as the one among the two that, for that particular task, obtained—ordered by decreasing importance—(i) the best shot rank, (ii) the best video rank,

(iii) the shortest time when the best shot was retrieved, (iv) the shortest time when the best video was retrieved. Each metric serves to perform a tie-break in case all the previous ones are equal among the two users. Using this formulation, the shot rank has primary importance. In fact, if the shot rank differs, we have that the best user is immediately the one having the lowest shot rank. In case any of these metrics are missing for that specific user and task, we set them to their maximum values (10^3 in case of ranks and a time longer than the task duration in case of times).

4.2 Comparison of retrieval models

One area of interest in comparing the system retrieval models is analyzing whether a correct item (frame or shot) of the searched video segment appeared in the top positions of the retrieved results. In this respect, Fig. 6 shows, for each system, the best-achieved rank of a correct item (frame or shot) before submission across all 23 KIS tasks (ten of which are textual KIS, reported in Table 2). The distribution of the minimum achieved rank by the *best* users reflects the overall teams' scores. For example, the best rank is below 100 in about 87% of cases for vibro and CVHunter, 78% for VISIONE, 68% for VERGE, 65% for vitrivr, and 30% vitrivr-VR. However, the minimum achieved rank of the *other* user is below 100 in about 52% of cases for vibro, 35% for CVHunter, 48% for VISIONE, and 9% vitrivr-VR. The considerably worse performance of the *other* users for some tasks may be caused by two main reasons: (i) we are not considering the logs after a team's correct submissions; (ii) a particular user may formulate better *initial* queries for some tasks than the other user.

In Fig. 7 we report, for each tool and each KIS task,

– The best achieved rank of a correct item (frame or video shot of the target video segment) in the top logged results of the *best* user in the considered task (before the team correct submission, if any);
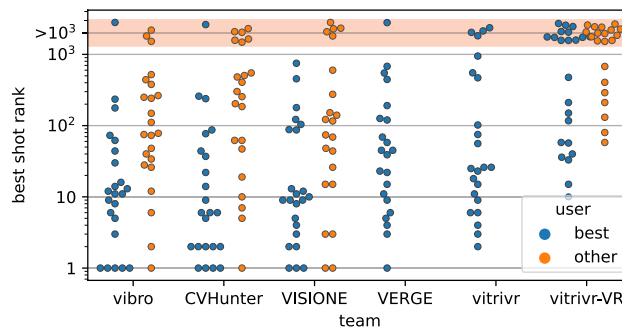

Fig. 6 Best rank of correct items appearing in result logs

Table 2 Textual KIS tasks

Name	Hints
T 1	Close-up of motorbike exhaust pipes being cleaned with a wet sponge.
	Two chromed pipes are visible, open on the left.
	The forearm of a man wearing black T-Shirt is visible a few times.
T 2	Two shots: first of a gorge with rocks hanging over a wooden walkway, second a wooden bridge seen from a creek.
	In the first shot, the creek is not visible, just the rocks on the right and trees and gorge on the left.
	Shot in autumn, with still some green trees but colored leafs on the ground, and boulders on both sides of the creek (in 2nd shot).
T 3	Almost static shot of a brown-white caravan and a horse on a meadow.
	The caravan is in the center, the horse in the back to its right, and there is a large tree on the right.
	The camera is slightly shaky, and there is a forested hill in the background.
T 6	Slow pan over a table with a glass, vase, leather cases and a wooden frame slate, then over a board with a timetable.
	The scene is poorly lit, and the text on the slate reads "Welcome to our Story", followed by a date.
	The timetable is titled "OUR WEDDING".
T 7	A girl riding a red bicycle, followed by a close-up shot of a termite trail on a tree root.
	The girl's head is not visible, she wears a blue shirt and short red pants, and has a bag and a tripod in a basket on the bicycle's handlebar.
	The focus in the shot of the termite trail gradually changes from back to front.
T 8	Shot of an opened magazine, showing a drawing of a bearded man on the right side, then a shot of a person standing in a street and holding different pages of an open magazine in front of the camera.
	The person in the street wears a blue T-shirt and light grey jacket, and is wearing a mask and sunglasses. There are white frames with black text messages flashing up inbetween.
	The drawing in the first shot is on black background, the man has a white beard, the title of the left page is "vote for Pedro".
T 9	Close-up shot of a yellow slug (naked snail) eating a green leaf with a tiny green branch.
	The leaf is in the lower center of the image, the slug curved in the right half.
	The slug and the leaf are on a bed of needles and small branches.
T 10	A shot of a man in a water slide, followed by two shots of two men trying to light a fire on a beach.
	The man slides down head first, and wears black bathing trunks.
	There is a circle of stones around the fire, and we do not see the heads of the two men.
T 11	View from an upper deck of a ship down to a lower deck and water, slowly changing the view to the front of the ship, where a man with a camera walks into view.
	The lower deck is on the left, with green floor and two red/orange chairs, and water is on the right.
	The man wears black trousers and a grey jacket.
T 12	A split screen shot of a building with a green facade with many different plants, static view on the left, detail view moving down on the right.
	The walls on the ground floor are concrete walls, partly covered with woodwork.
	The shadow of another building is moving down, until most of the building is in sunlight.

– The time t_f when the best rank of a correct item – as described above – was obtained;

– The best achieved rank of any item (frame or shot) of the correct video by the *best* user in the considered task (before the team correct submission, if any). Note that, in this case, the item may not overlap the target video segment even if it belongs to the correct video;

– The time t_v when the best rank of any item of the correct video was obtained;

– The time t_{cs} of the correct submission.

We can observe that the ranks and the overall competition scores of the top three teams are somewhat matched. In fact, vibro, CVHunter, and VISIONE are able to find the correct

video in the first 10 results more consistently. In general, browsing failures (red and orange cells) are most evident in the case of textual KIS or, as easily guessed, when the best rank of a corrected item is high. However, it is also interesting to note that for some tasks and tools, a correct video was in the top 10 results, but it was not correctly identified and submitted (e.g., vibro in task T2, VISIONE in task T11, VERGE in task V10, vitrivr in tasks T1 and T8, vitrivir-VR in task V2).

We also emphasize that the best rank and correct submission times can be from different team members. Hence, it might happen that the time between the occurrence of the best item and submission is unrealistically low. For example, vitrivr-VR had the best video rank 97 in time 104 s and

Tool			Task	T 1	T 2	T 3	T 6	T 7	T 8	T 9	T 10	T 11	T 12	V 1	V 2	V 3	V 4	V 5	V 6	V 7	V 8	V 9	V 10	V 11	V 12	V 13
vibro	correct frame/shot		rank	3	1	1	16	14	1	5	13	8	1	62	73	11	11	44	235	177	6	1	30	9	12	-
			t_f	59s	402s	9s	85s	6s	40s	17s	50s	95s	27s	31s	46s	52s	53s	47s	33s	30s	64s	31s	56s	64s	26s	-
	correct video		rank	1	1	1	16	14	1	5	13	8	1	62	49	8	11	37	235	7	6	1	30	9	12	1
			t_v	59s	402s	9s	85s	6s	40s	17s	50s	95s	27s	31s	59s	52s	53s	47s	33s	30s	64s	31s	56s	64s	26s	18s
	correct submission		t_{cs}	176s	-	36s	94s	14s	47s	35s	76s	123s	38s	63s	116s	110s	64s	119s	138s	78s	94s	59s	66s	69s	31s	63s
CVHunter	correct frame/shot		rank	2	14	1	259	1	2	2	1	87	6	37	77	6	22	44	6	2	9	5	1	-	2	239
			t_f	21s	44s	15s	27s	17s	268s	22s	20s	392s	40s	159s	95s	35s	74s	193s	111s	70s	191s	45s	23s	-	19s	16s
	correct video		rank	2	14	1	259	1	2	2	1	87	6	37	77	6	22	1	1	1	9	5	1	27	2	2
			t_v	21s	44s	15s	27s	17s	268s	22s	20s	392s	40s	159s	95s	35s	74s	193s	111s	32s	191s	45s	23s	22s	19s	16s
	correct submission		t_{cs}	34s	173s	33s	-	23s	278s	27s	25s	413s	62s	179s	120s	39s	85s	214s	194s	76s	198s	55s	29s	34s	24s	36s
VISIONE	correct frame/shot		rank	87	1	3	454	1	1	2	9	2	11	5	104	88	9	122	12	751	13	10	9	8	4	179
			t_f	9s	12s	145s	211s	15s	18s	13s	14s	339s	12s	67s	46s	29s	33s	88s	36s	27s	36s	40s	54s	16s	235s	16s
	correct video		rank	6	1	3	22	1	1	2	9	1	11	5	1	27	5	19	11	7	13	10	9	8	4	1
			t_v	9s	12s	145s	302s	15s	18s	13s	14s	339s	12s	67s	46s	91s	33s	88s	36s	27s	36s	40s	54s	16s	235s	16s
	correct submission		t_{cs}	39s	214s	155s	-	30s	54s	29s	28s	-	68s	73s	80s	92s	54s	248s	48s	31s	47s	52s	56s	20s	243s	29s
VERGE	correct frame/shot		rank	45	23	445	-	11	126	3	1	120	22	680	4	69	45	58	191	9	6	5	15	548	39	-
			t_f	279s	273s	333s	-	88s	106s	21s	29s	294s	384s	104s	195s	33s	115s	250s	144s	32s	64s	41s	20s	105s	23s	-
	correct video		rank	13	8	445	-	11	72	3	1	120	22	680	4	7	15	2	29	9	1	1	4	548	39	30
			t_v	192s	93s	333s	-	88s	106s	21s	29s	294s	384s	104s	195s	33s	21s	270s	164s	32s	64s	29s	185s	105s	23s	23s
	correct submission		t_{cs}	318s	386s	-	-	89s	-	40s	188s	-	402s	-	212s	150s	170s	297s	186s	45s	96s	44s	-	-	35s	87s
vitrivr	correct frame/shot		rank	-	9	947	75	2	6	15	25	18	469	551	-	4	6	-	102	23	56	26	11	-	3	26
			t_f	-	196s	37s	45s	22s	362s	100s	144s	139s	375s	161s	-	114s	41s	-	261s	30s	44s	154s	33s	-	239s	20s
	correct video		rank	9	9	947	75	2	6	15	25	18	469	551	244	4	6	46	102	23	56	26	11	1	3	26
			t_v	55s	196s	37s	45s	22s	362s	100s	144s	139s	375s	161s	30s	114s	41s	25s	261s	30s	44s	154s	33s	8s	239s	20s
	correct submission		t_{cs}	-	225s	-	-	26s	131s	-	-	-	143s	57s	103s	-	218s	227s	184s		57s	25s	244s	42s		
vitrivr-VR	correct frame/shot		rank	-	-	-	-	57	476	33	-	40	-	150	-	117	-	211	-	58	36	15	-	-	10	-
			t_f	-	-	-	-	71s	99s	86s	-	22s	-	231s	-	211s	-	31s	-	233s	30s	195s	-	-	89s	-
	correct video		rank	57	64	17	227	57	38	24	275	40	265	150	8	10	2	187	51	58	36	15	97	1	10	106
			t_v	213s	156s	191s	201s	71s	344s	86s	251s	22s	286s	231s	201s	79s	187s	31s	46s	233s	30s	195s	104s	22s	89s	73s
	correct submission		t_{cs}	-	-	-	-	160s	-	94s	-	269s	-	-	-	267s	194s	129s	-	-	129s	223s	106s	46s	114s	127s

Fig. 7 The table reports for each tool with logs (i) the best achieved rank of a *correct item* (frame or video shot); (ii) the time t_f in seconds when the best ranked correct item was retrieved; (iii) the best ranking of any frame/shot of the *correct video* (but not necessarily the correct video segment); (iv) the time t_v in seconds when the best ranked video frame/shot was retrieved; (v) the time t_{cs} of the tool's correct submission. Red values are for the best detected ranks of searched video frames/shots if the target video segment was not present in the logged result for a task. Green cells show the best achieved correct item with a rank less than 100. Yellow cells show the best achieved video item with rank less than or equal to 10. Red cells indicates browsing failures, when a correct item was in the first 1000 results but was not submitted. Orange cells are other browsing failures, when the correct video was present—but no correct frame or shot was present—and no correct submission was made

submission at time 106 s in task V10. On the other hand, long submission delays for both users just confirm issues with browsing.

Regarding vitrivr-VR, please also note that it uses an asynchronous workflow and allows users to browse very easily within multiple result sets at the same time, as well as within entire videos from different queries. The current logging format does not always allow the path from query to submitted result to be determined uniquely, and this explains why correct results submitted by vitrivr-VR do not always appear in the top-ranked logged query results or appear at a very high rank.

Some teams (e.g. VISIONE, vitrivr, and vitrivr-VR) display the search results by grouping together those from the same video and showing a limited number of items for each video in the browsing interface. Therefore, many correct submissions may have been generated from a video level-hit and not by scanning the top results from the highest to the lowest score (i.e., in the order in which they were logged and used in the analysis reported in this section). See for

example, VISIONE in task V7: even if the best correct item rank is pretty high (751), the correct video was displayed in the first page of the results (7-th row in the browsing interface).

4.3 Browsing efficiency

Figure 8 shows the relation between the rank of the first appearance of a correct item (frame or shot) in the logged result set and the elapsed time in second between this first appearance and the correct submission, if any, both for visual KIS (left-hand graph) and textual KIS (right-hand graph) tasks. Note that these graphs give an approximation of the real browsing time because (i) it is possible that a correct submission was made though inspecting the video and not the top-ranked frames/shots, (ii) the team user who first retrieved a correct item may not be the same who submitted the final correct answer (information on which team member made the correct submission is not available for all teams). Nevertheless, these graphs give some insight
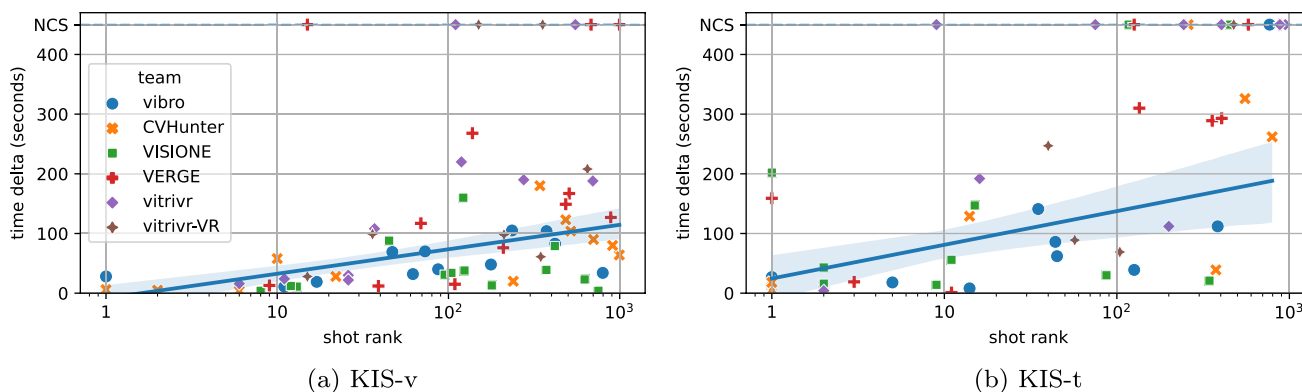
(a) KIS-v

(b) KIS-t

Fig. 8 Relation between the rank of the first occurrence of a shot in the result logs and time delta to correct submission, for both visual **a** and textual **b** KIS tasks. *NCS* stands for *Non-Correct Submissions* and corresponds to all the correct frames found in the result logs that

were not correctly submitted (either because of running out of time or incorrect submissions). The blue line is found through linear regression, and it is accompanied by the 95% confidence intervals

into how long it took users to find a correct item once it was present in the result set. This time clearly depends also on the specific system browsing capabilities and the user behavior (e.g., some user may prefer to check just a limited number of results and eventually reformulate the query instead of exhaustively inspecting the results set). Overall, we observed that—as expected—the time between the first appearance of a correct item and the submission tends to increase with the rank of the item. However, in textual KIS tasks, it happens more often that the rank of the first appearance is low (even 1), but the operator takes a long time before submitting a correct result (which we recall he/she has never seen before, but knows only a textual description). For example, in the graph in Fig. 8b, we can see five cases where the rank was less than 25, but the operator took more than two minutes to make a correct submission, and in one case, no correct result was submitted at all. These outliers are less frequent in the case of visual tasks—only one team in a single task, VERGE in V10, had a correct item in 15th position (obtained after only 20 s from the start of the task) but did not submit any correct results at the end. More generally, it is interesting to note that the variance of the time delta increases with the rank, as different strategies may be used by the team members., e.g. exhaustive inspection of a result set, query reformulation, or video-level browsing, just to guess a few.

4.4 Querying modalities

In this section, we aim to provide a more in-depth look on what kind of query modalities the individual teams actually used during the competition. In order to do so, we divided the query logs into six categories by summarizing the underlying analysis methods of the different teams. The outcome

was: *Text*, *Image*, *ODLS*, *OCR*, *ASR* and *Color*. *Text* includes joint-embedding queries for most of the logging teams and VERGE's concept search. *Image* groups methods such as query-by-example through visual similarity search and relevance feedback with global image embeddings. *ODLS* stands for object-detection-localisation and segmentation and includes those kind of queries that specified a number of objects or objects and their positions in an image. Since ODLS queries of the VISIONE system were often used in combination with the text modality, we keep also multi-modal combinations in the ODLS category. Examples are VISIONE's concept search and VERGE's number-of-object filter. Although *OCR* (optical character recognition) and *ASR* (automatic speech recognition) searches were formulated with text, the underlying analysis methods are fundamentally different compared to the other methods, which is why we assigned two additional categories. The last query type *Color* groups methods, where color was solely used for a search. Please note that >1000 used in tables/graphs may mean also that the searched target was filtered out.

Table 3 depicts per-team relative usage of individual query categories. Throughout all teams, pure text queries are by far the most used variant, while image queries are the second most used. Obviously, these results are affected by what modalities are actually implemented by individual teams, but nevertheless, text queries dominate considerably.

Additionally, Fig. 9 compares query modalities usage across the two task types KIS-t and KIS-v. Even though text is still the most frequently used category for both task types, query modalities such as image and ASR gain popularity in the KIS-v tasks, where audio-visual information is presented.

Nonetheless, a question may arise whether text queries are also effective apart from being popular. To clarify this,

Table 3 Percentages of query type use across all KIS tasks for each individual team

Team	Text	Image	ODLS	OCR	ASR	Color
vibro	69.9	27.8	0.0	0.0	0.0	2.3
CVHunter	82.2	17.8	0.0	0.0	0.0	0.0
VISIONE	78.5	0.0	21.1	0.0	0.0	0.3
VERGE	88.6	8.5	1.4	0.0	0.0	1.4
vitrivr	82.1	0.0	1.0	8.3	6.0	2.7
vitrivr-VR	96.9	0.0	0.0	3.1	0.0	0.0

Table 4 depicts the ranking of correct shots/frames per team and query type. In most cases, the performance of text queries is similar to or better than the performance of other query modalities. One notable exception is CVHunter, where Image queries for instance achieved 43% results within top 100, while only 19% of text queries were within the top-100. In this case, however, CVHunter often utilized relevance feedback queries, which incrementally refines the previous text search results, so the performance of Image queries is in a sense pre-conditioned by the performance of text queries.

4.5 Querying density

In this section, we focused on how individual teams divide their time between querying and other activity (e.g., browsing). First of all, we focused on whether the querying intensity changes in the course of the task duration. We divided each task into 1-minute intervals and counted the volume of per-team queries from this interval.[2]

The count of per-team queries decreases from approx. 3.6 in the first minute to approx. 2.1 in the last minute. This may indicate that in later stages, teams focus more on browsing, while earlier they try to re-formulate their search more. Even though, the differences are not so substantial, so we can also focus on per-team querying density in general. Table 4 contains the mean volume of queries per team per minute of their active participation.[3] Here, two main outliers are VISIONE, who made on average 5.4 queries per minute and vitrivr-VR, who only logged 1.9 queries per minute. In case of vitrivr-VR, the main cause is the tool design itself, which is much more focused on browsing than its competitors. In the case of VISIONE, the cause is that queries are evaluated "on-the-fly" at any user interaction with the search interface (even just moving or resizing an object in the canvas),

Fig. 9 Comparison of query type use across all teams for KIS-v and KIS-t tasks. Results are grouped by query type and task type

without the necessity of explicitly clicking on the "search" button.

Finally, we also measured users' reaction times, i.e., how fast did they construct the first query. Figure 10 depicts this statistic per team and task type. Notably, vitrivr-VR experienced significantly higher times for their initial queries than the rest of the teams. This is not unexpected, particularly considering the rise of text-based, cross-modal retrieval, since text entry in VR is still much slower than using conventional keyboards.

We also assumed that textual description is faster to process and therefore initial query times would be significantly smaller for textual KIS tasks. While this was true for some teams (vibro, VISIONE), the results were not conclusive in general. In fact, it is important to note that certain teams including VISIONE and vibro, occasionally did not manually type the textual query during textual KIS. Instead, they copied and pasted the query from the DRES visualization interface used to view the tasks. Unfortunately, this copy and paste action was allowed but not logged, so it is impossible to determine which teams relied on this method and how frequently they did so.

4.6 Analysis of textual queries

In this subsection, we focused on the properties of the textual queries. Specifically, we evaluated the length of textual

---

[2] We only kept those teams that did not yet solve the task, i.e., the timestamp of their correct submission was higher than the upper bound of respective interval (or they did not solve the task at all).

[3] Counting from the task start time to task end time or correct submission time, whichever comes first.

Table 4 KIS tasks query statistics per team and query type

Team	Query type	Query per minute	Words per Query	query length	Top-10	Top-20	Top-50	Top-100	Top-200	>1000
Vibro	Text	4.96	8.09	40.86	7.3	12.2	21.1	29.3	37.4	43.1
	IMAGE	–	–		10.2	14.3	20.4	30.6	32.7	51.0
CVHunter	Text	3.40	9.14	49.06	10.8	13.8	15.4	19.2	20.0	48.5
	IMAGE	–	–		10.7	14.3	39.3	42.9	57.1	32.1
VISIONE	Text	5.41	21.2	103.89	9.7	16.3	19.4	26.0	35.7	48.0
	ODLS	–	–		13.1	19.7	19.7	23.0	26.2	54.1
VERGE	Text	2.58	4.49	25.20	13.4	14.4	23.0	32.1	45.5	38.0
	IMAGE	–	–		11.1	11.1	11.1	27.8	38.9	55.6
vitrivr	Text	3.28	N/A	N/A	3.2	4.8	8.9	13.3	18.5	67.3
	OCR	–	–		0.0	0.0	0.0	4.0	12.0	88.0
vitrivr-VR	Text	1.91	5.70	28.37	0.0	1.6	3.8	7.0	10.3	80.0

Only the query types with 10+ per-team occurrences are depicted. For each team, the mean volume of queries, mean number of words and mean string length of textual queries are depicted. Then, for all pairs of a team and a query type, top-*K* denote percentage of queries, for which the target shot was within the first *K* results and > 1000 denote percentage of queries, where correct shot was not present in top-1000 results
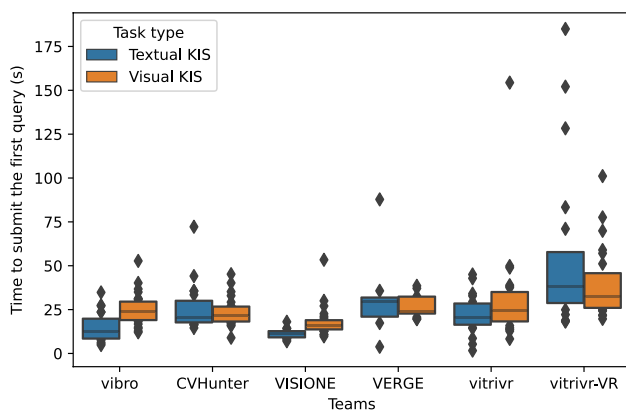
Fig. 10 Time to initiate the search with the first query. Results are grouped by team and task type

queries w.r.t. the number of words and number of characters, for which the mean values are depicted in Table 4. Note that vitrivr logs do not contain the text of the query, therefore we exclude them here. We observed some notable differences in textual querying strategies of individual teams: both VERGE and vitrivr-VR used on average shorter queries (w.r.t. both metrics), while VISIONE usually constructed much more complex queries - in average twice as large as its next competitor.

There are two main reasons behind this observation. First, VISIONE provided more extended and detailed textual descriptions of the searched scene compared to other teams. Second, a significant portion (about 83%) of VISIONE's queries were temporal textual queries (descriptions of two different scenes of the same video clip), which were

concatenated as a single textual query in this analysis. We also note that the vibro team did not log temporal queries as temporal fusion was an on the fly computed interface option for two independent queries.

We were also interested, in whether the additional effort coming with the construction of larger queries pays off, i.e., whether better ranking of correct items/videos are achieved. Nonetheless, as teams use different ranking models and per-team querying strategies also differ substantially, we have to resort to per team comparisons. Figure 11 depicts per team enhanced boxplots of queries, where the correct shot was *within*, or *outside* of *top-1000* results. In general, the differences were smaller than we expected, but even though, for vibro, CVHunter and VERGE teams, *within top-1000* queries were significantly larger (w.r.t. both number of words and number of characters; in case of *t*-test, *p*-values <= 0.04) than *outside top-1000* queries. Especially for vibro, the pattern was quite notable. However, just producing larger (more descriptive) queries might not lead to better results. This is illustrated by VISIONE, whose queries were largest in general, but lengths of *within top-1000* and *outside top-1000* queries were without statistically significant difference.

We also observed how much the initial results could be improved via subsequent text query reformulations. For this, we grouped all queries collected for each user and task and ordered them from first to last. We denote this as query sequences and grouped queries w.r.t. their position within the sequence. Figure 12 depicts enhanced boxplots for the ranks of correct shots. It can be seen that textual reformulation may lead to some notable improvements. On the other hand, results also reveal numerous browsing errors, where correct shots were within top 10 or top 100, but queries
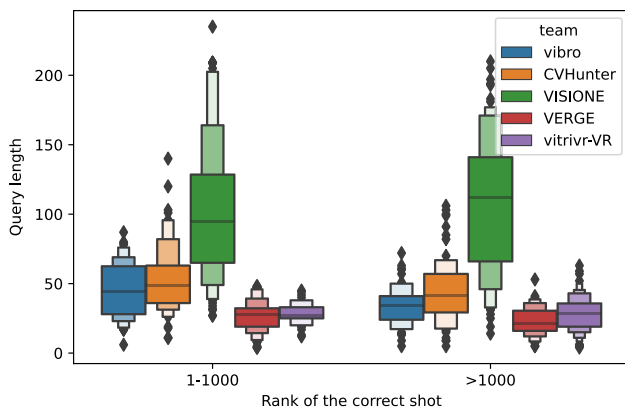
Fig. 11 Distribution of the number of words and string lengths for textual queries. Grouped by individual teams and ranks of the correct results
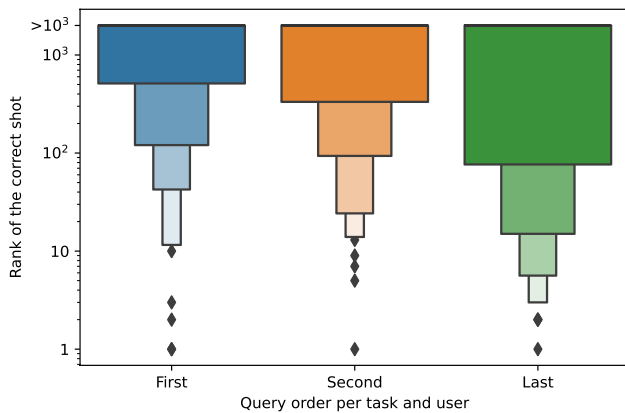

Fig. 12 Ranking distribution of correct shots w.r.t. query sequence position. "First" and "Second" denote the first text query per task and the next text query (i.e., first reformulation/extension). "Last" denotes the very last text query logged for a particular user and task. Note that only query sequences with length >= 3 are depicted

were reformulated anyway. This may indicate the necessity to focus more on the browsing capabilities of individual tools to prevent such oversights.

5 Analysis of AVS tasks

As in previous years, VBS organized another session focusing on ad-hoc search tasks. Specifically, 8 AVS tasks (see Table 5) were performed where teams are required to submit as many correct shots as possible. Figure 13 shows shares of correct submissions of all teams in all tasks. It is apparent that there is not one dominant team for all tasks. For example, looking at the top two AVS systems, vibro was way more effective than IVIST in the task a01, while in the task a10 the situation was reversed. Nevertheless, all top-performing AVS systems (vibro, IVIST, VIREO, and CVHunter) show an ability to solve a non-trivial share of the multi-set of correct submissions.

Figure 14 shows how the overall number of received correct submissions grows over time of each AVS task, while Fig. 15 shows the number of submissions in specific time slots.

In both graphs, the same submissions from *n* different teams count as 1. Hence, the graphs show the progress in the detection of new and unique correct scenes. The trend is similar for all tasks. After a first slow period (about 40 s, except task a07), there starts to be a continuous growth with occasional peaks.

Figures 16 and 17 present the number of submissions and the first *k* submissions by *n* teams. It is apparent that the overall number of submissions is always higher than the overall number of correct submissions (only unique submissions are counted). Regarding times to first submissions, there is not a clear difference between the time until the first submission and the first correct submission. Similarly, except for task a09, the times to the first submissions by 50% of teams are quite similar to the corresponding times in the correct submission graph. However, the times to the tenth submissions by 50% of teams are becoming lower than the times for correct tenth submissions by 50% of teams in tasks a02, a05, and a06. To sum up the analysis, there are differences in the complexity of AVS tasks. Some tasks are easier to solve for many teams, while others are way more

Table 5 AVS tasks

Name	Hint
a01	Find shots showing one person playing a guitar (other people but no other musicians may be visible)
a02	Find shots of one or more persons balancing on a bar, railing, rope or slackline, without any device under their feet
a04	Find shots of someone riding a horse or sitting on a horse (living animal)
a05	Find shots taken from any vehicle driving inside a tunnel, requiring part of the vehicle being visible
a06	Find outdoor shots showing a teddy bear (toy)
a07	Find shots of a waterfall, without people
a09	Find shots of one or more decorated trees (not just branches) that are not lit (inside or outside)
a10	Find shots of someone with their hands on a camera (not e.g. a phone-like device), filming or taking/preparing to take a picture
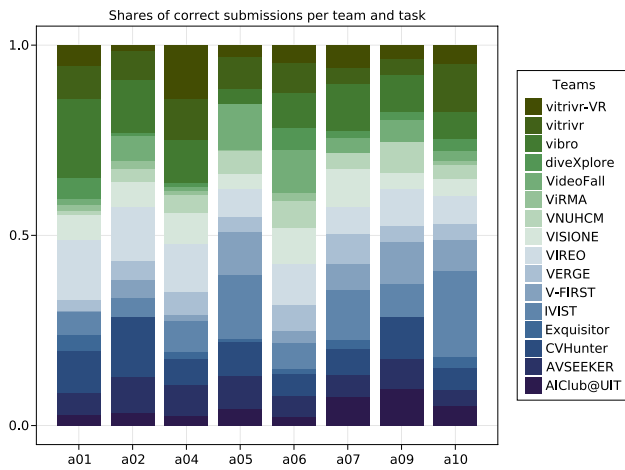
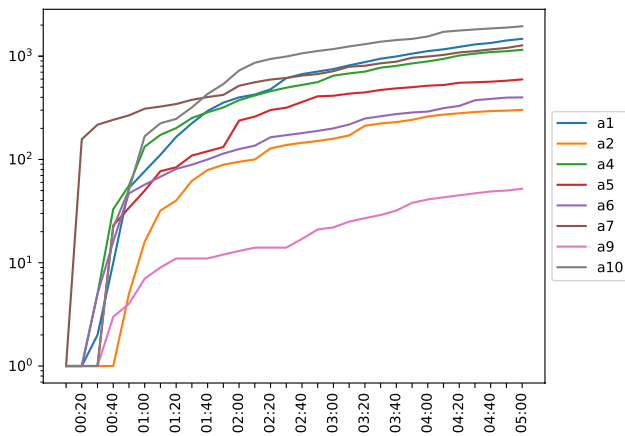**Fig. 13** Share of AVS submissions judged as correct per team and task

Fig. 14 Cumulative correct video submissions over time during an AVS task. Log scale on y axis
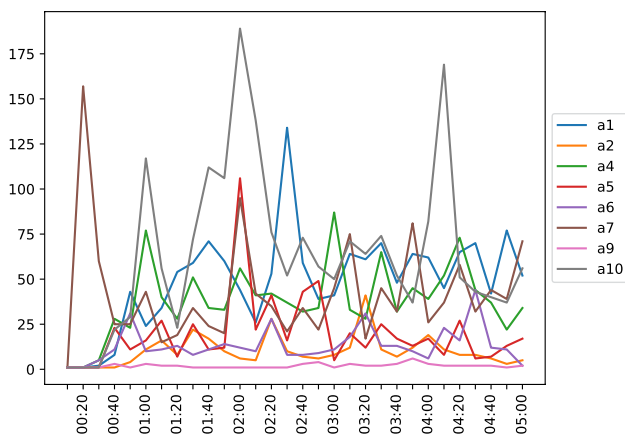

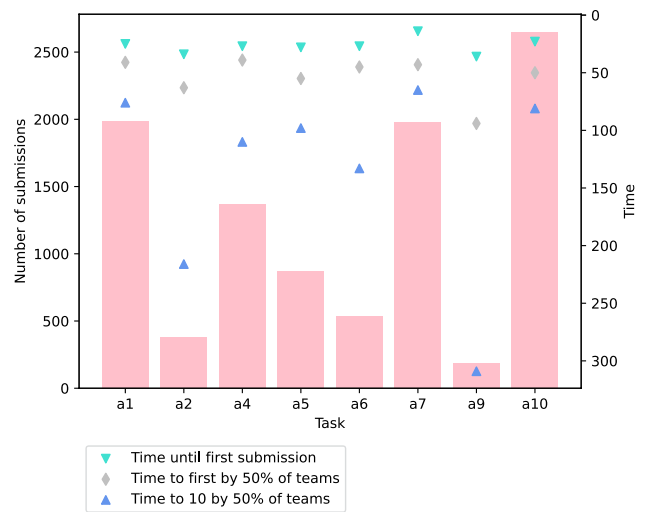**Fig. 15** Correct video submissions over time during an AVS task

Fig. 16 Selected AVS metrics per task. Higher y-axis values indicate that teams found it easier to find results to submit for a task
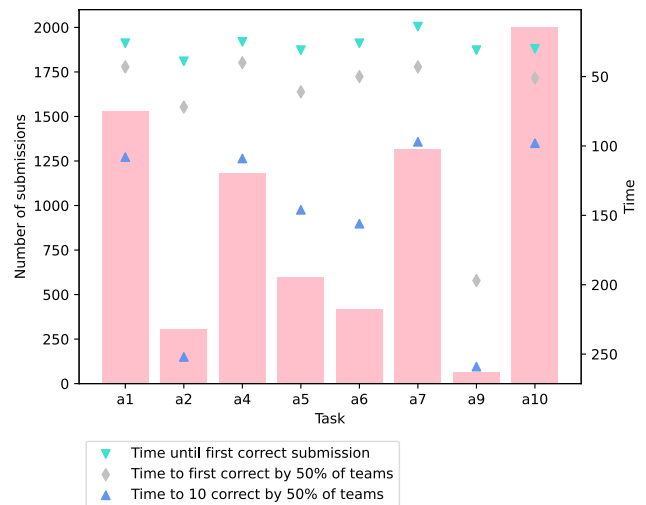

Fig. 17 Selected AVS metrics per task, looking at correct submissions. Higher y-axis values indicate that for a given task, it is easier to find results that judges deem correct

challenging and also interesting for VBS-like interactive search evaluations.

The VBS 2021 report [31] presented an observation that in several AVS tasks, there were many teams in disagreement with one judge. Since the text query preparation for VBS 2022 was more thorough (see Sect. 6), we also analyzed the agreement/disagreement stats in Table 6. Compared to the previous year, the data do not reveal a significant level of disagreement across seven tasks. Indeed, except a few exceptions (e.g., eleven teams against one judge in task a02), the teams mostly agreed with the

Table 6 Number of teams in agreement/disagreement with judges

Task	Number of Teams										
	1	2	3	4	5	6	7	8	9	10	11
a-1	1317/369	188/30	36/5	3/0	–	–	–	–	–	–	–
a-2	140/47	59/6	33/0	30/0	23/0	3/2	9/0	4/0	2/0	–	2/1
a-4	605/172	266/9	165/2	93/0	61/0	35/0	8/0	5/0	3/0	–	1/0
a-5	345/246	156/23	100/6	45/0	9/0	5/0	7/0	–	1/0	–	–
a-6	134/82	71/10	52/6	46/1	40/0	34/0	25/1	15/0	4/0	5/0	2/0
a-7	778/542	291/93	159/32	91/12	66/5	28/1	17/2	3/0	1/0	–	–
a-9	**44/74**	**9/19**	**3/5**	**2/3**	**1/5**	2/1	–	1/1	–	1/0	2/0
a-10	1419/512	371/38	157/6	85/1	44/0	17/0	7/1	3/0	1/0	–	–

Bold font highlights cases where the fraction is lower or equal to one (i.e., $\frac{\#agreement}{\#disagreement} \leq 1$)

judge's decision. Only in one task a09, there are cases where the teams disagreed with judges more often. However, this might also be caused by the task's difficulty and attempts to send at least something (the overall numbers are low).

5.1 New direction for Ad-hoc search at VBS

For many years, AVS tasks were evaluated at VBS in a similar fashion as at TRECVID. Teams were supposed to submit as many correct shots as possible, often overloading judges with thousands of submissions. The scoring function was designed to provide a high score for precision and (pooled) recall. However, there were also opinions questioning the current way AVS tasks are evaluated.

Although 100% recall with high precision of found shots is an important goal in various domains (security, endoscopy), for VBS, it is also highly important to first localize videos containing a correct shot. In other words, ad-hoc search can be divided into two task categories—localization of correct videos and effective search (e.g., advanced browsing) of the videos. From the organization's perspective, finding just one piece of evidence of video correctness (i.e., only the first correct shot) decreases the workload for judges and also simplifies discussions about a fair scoring function. Therefore, we have decided to focus on the video localization part of AVS tasks at the next VBS events.

A possible new scoring formula to compute AVS score per team can follow the objective of finding as many videos as possible, where the team must submit one correct shot from each video (i.e., evidence for each video). We note that the formula should integrate a penalty mechanism preventing floods of unverified submissions. In addition, a maximum limit of submissions per team could also be introduced as well (not considered currently). An example approach to defining the score f_t of a team t in an AVS task could be as follows:

$$f_t = 1000 \cdot \max \left(\frac{1}{|C|} \sum_{v}^{\mathcal{V}_t} (c_v - i_v \cdot p), 0 \right), \text{ where}$$

$i_v :=$ number of incorrect submissions before the first correct submission for video v, number of submissions in v else

$c_v :=$ 1 if correct submission for v, 0 else

$\mathcal{V}_t :=$ set of videos with a submission for team t

$p :=$ submission penalty constant (e.g. 0.2)

$C :=$ distinct correct videos across all teams

Although there are data to estimate the penalty from the VBS 2022 competition, we plan to carefully set this penalty based on more experiments. The reason is that the available data might be biased with respect to the AVS scoring formula used at VBS 2022.

While possibly suffering from the bias of teams optimizing for the scoring function used at the VBS 2022, an early analysis of this data shows an average of 1.39 incorrect submissions before the first correct submission per task, team, and video. The maximum number of incorrect submissions before the first correct submission for the corresponding video was 19. With the new scoring function in place, it will be interesting to see if more careful inspection of submissions can be encouraged.

6 Lessons from text query definition

In the purely physical VBS editions up to 2020, the judges for AVS tasks were seated next to each other in the room, and any questions concerning the ambiguity of handling queries were done informally among them. As reported in [31], a briefing with the judges was conducted for the first virtual VBS in order to discuss and refine the AVS queries, but was found insufficient to ensure consistency of judgements. Thus, the briefing of the judges for VBS 2022 was extended to consist of (i) a session discussing and refining the AVS

query texts like in 2021, (ii) a similar session for refining the KIS-t queries and (iii) a dry-run session in which the judges tried to solve the AVS tasks themselves and provided feedback about the query texts. Both the discussion and dry-run sessions were held as web conferences. Six judges participated in the sessions. More details and the evolution of this process can be found in Bailer et al. [9].

The creation procedure for creating the textual queries was unchanged from previous editions of VBS (see [51] for details). The queries were provided in a shared document that was made available to the judges in advance of the sessions. However, the ground truth was not included in the document.

Discussion session. In the discussion session, both AVS and KIS-t queries were covered. For AVS queries, the query text was read together with the judges and they were asked whether they could imagine scenes covered by the query, and request clarifications on possible interpretations of the queries that came to their mind. A reformulation that found consensus in the group was chosen for proposed changes. Where necessary, additional notes were recorded for later reference by the judges.

For KIS-t queries, the query was read together with the judges and the target clip was shown. As it was unclear which order would be better, reading the query first and watching the clip were tried, but first, reading the query seemed preferable. Then required changes and clarifications of the queries were discussed, watching the clip again if needed.

Dry-run. In order to perform the dry-run, SOMHunter V2 [77] was used as a browsing tool. The existing Docker deployment of the tool[4] was modified to run a set of independent instances (one per judge) on the same machine. A startup script took care of modifying configurations so that the Docker containers required by each instance would use a dedicated set of ports. The containers were hosted on an Amazon Web Services EC2 machine with 64GB RAM.

After a very brief introduction to the tool, the judges were given up to 10 min to explore and discuss one query. Searching was stopped once a larger number of results was collected. The results were analysed in order to understand what type of content could be found for the query, and which ambiguities and border cases may exist. Similar to the discussion session, consensus on reformulation of queries was found and additional notes were recorded where necessary. All but one of the AVS tasks have been solved by the judges in the dry-run session, which already provided a good indication that the tasks would be solvable in the competition.

Query improvements. Both sessions resulted in a number of changes to the originally proposed queries. As shown in

Fig. 18, the mean lengths of the queries increased after each of the sessions as details and clarifications were added. Figure 19 provides details about these changes on a word level, expressed as the number of changes per query. Most changes concerned nouns: more than 1.6 noun additions/changes per query were made for AVS queries, and almost 2 additions/changes for KIS-t queries. The numbers are slightly lower for adjectives and prepositions, but the pattern is similar to nouns. It is worth noting that while the number of changes is similar for both types, more additions were made for KIS-t queries, which indicates the higher need to adding details. Also, changes of words (typically finding a more precise term of easier to understand synonym) were only done in the discussion session, while after the dry-run only words were added. Examples were added for some AVS queries, but this occurs in less than 1 in 5 cases.
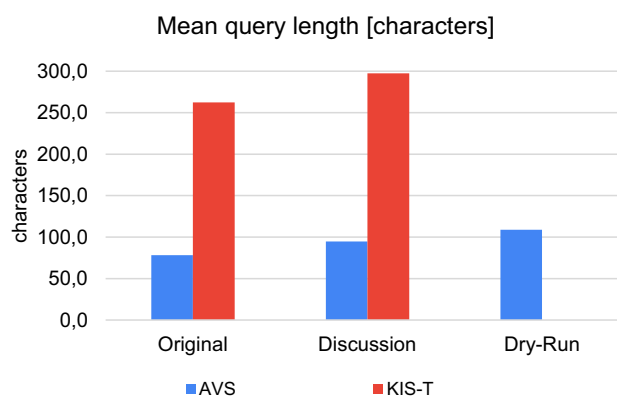

Fig. 18 Mean lengths of text queries: originally proposed queries, after discussion of the texts and after the dry-run (applies only to AVS)
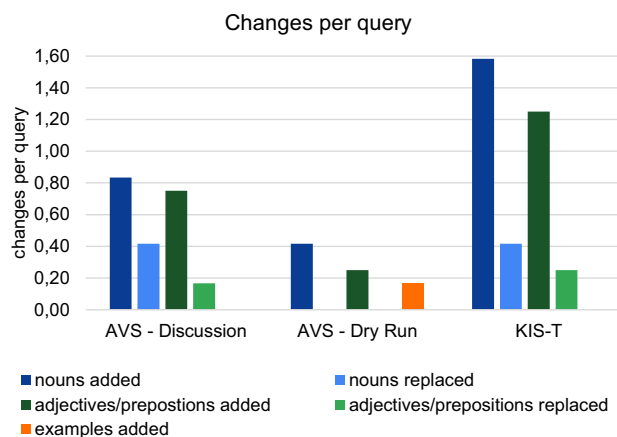

Fig. 19 Word-level changes to AVS an KIS-t queries made as a result of the discussion and of the dry-run. Adding examples only applies to AVS queries
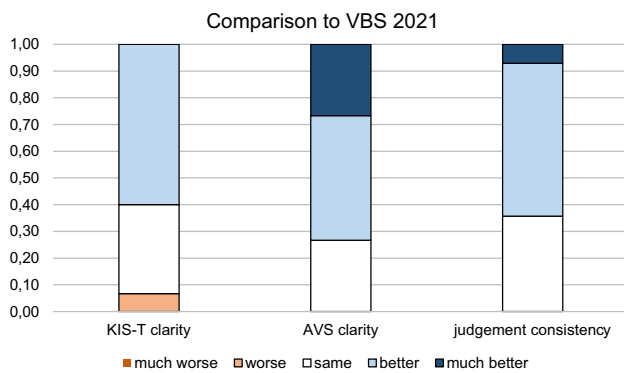
---

4 https://github.com/siret-junior/somhunter.

Fig. 20 Comparison of query clarity and judgement consistency between VBS 2021 and 2022

Fig. 21 Zipf's law in a video collection

Assessment. In order to assess the effectiveness of the modified judges briefing, we performed an online survey among all participating team members in the week after VBS 2022. We received 20 responses, of which 17 respondents stated they had participated in 2021. The repeated participants were asked to compare the clarity of the KIS-t and AVS queries as well as their perception of the consistency of the judgements of their AVS submissions on a 5-point scale ranging from much worse to much better. The responses to these questions are shown in Fig. 20. Roughly one-third of the respondents did not observe any changes, and one-eighth found the KIS-t queries less clear than in the previous year. But the majority of the respondents found the clarity of the descriptions as well as the judgement consistency better or much better. It is worth noting that for AVS clarity and judgement consistency, none of the respondents reported a decrease in the quality and also *much better* was chosen in some cases (which was not chosen for KIS-t). We believe that this is a consequence of performing the dry-run, which helped both improve AVS queries and ensure later judging quality.

7 Discussion and future challenges

A VBS-like evaluation is a unique large-scale experiment organized once per year. The observed results of the experiment provide a unique insight into the expected interactive search performance in KIS and AVS challenges with current state-of-the-art models. Based on the observed results, we would like to provide a summary of findings with a discussion on future directions.

– We start with the most resonating message (and not only within the VBS community [47]) that the CLIP model and its variants represent a game-changer in cross-modal search. The approach and its near-future potential (using
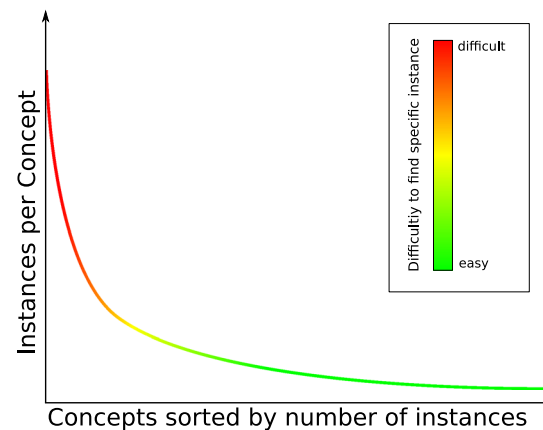
larger training datasets [68]) may break some assumptions that were made for challenges like VBS. This can be shown with Fig. 21 illustrating Zipf's law for the commonly observed distribution of concepts in image datasets [81]. For known-item search tasks, there were two challenges – concepts with a high number of occurrences often required non-trivial additional interaction, while rare concepts used to be hard to find with models trained with standard general-purpose training datasets. With CLIP based models and huge training datasets, rare concepts are now more likely to be known by content-based ranking approaches and thus often become findable with a free-form text query. Hence, one of the key remaining challenges for the future seems to be effective refinement and browsing in large clusters with many similar instances. We add that large clusters can also emerge if users do not actively remember all the details of the searched scene.

– Based on our survey on textual task quality, the concepts users imagine are sometimes culturally dependent. These issues indeed remain a challenge even with potentially much better ranking-models and represent an interesting task for future evaluations. We note that, so far, VBS has mostly data of users from Western cultures using CLIP, so the alignment of concepts between user and model may be high.

– Human interaction was still important even at this iteration of VBS, where top performing teams already used CLIP. Even despite the overall good ranking performance of the joint embedding models, one third to one half of issued text queries (for systems employing the CLIP model) ended up with a search scene outside of considered ranked results (i.e., rank > 1000, see Table 4). On average, users are able to improve their textual queries over time (Fig. 12) and also refine/modify the queries with different modalities (Fig. 9).

– Another future challenge are evaluations of the impact of users. We can observe that today's vision-language models are sensitive to differences in wording, so their use in video retrieval systems harms the retrieval consistency [10], i.e., the desirable property that a system returns consistent results for similar but differently phrased information needs. We are planning future evaluations with more users per team and more controlled settings (predefined start query) to further analyse the impact of users vs. tools.

– Another interesting challenge for future evaluations is to focus on different types of visual data. Interactive search in various visualizations of other types of data (e.g., images from human motion in RGB color space) might test generalization of presented approaches and systems for different domains, especially in situations where a good initial text query is not available.

8 Conclusions

The paper presents findings from the eleventh iteration of the Video Browser Showdown, where 16 teams participated with their interactive video search systems. The wide panorama of video analysis and retrieval approaches was used, as described in the related work section. The results confirm the effectiveness and reign of joint-embedding approaches, where CLIP-based models demonstrate impressive performance. The top three systems vibro, CVHunter, and VISIONE (according to the VBS ranking), were able to solve all visual known-item search tasks as well as almost all textual known item search tasks. Considering the size of the video collection, this is a great achievement compared to the previous several years. The result logs of six teams revealed that with multiple attempts to formulate a (mostly text) query, the teams were able to find known-items at good ranks. However, there emerged also several browsing/visualization issues where the teams overlooked a correct item with a good rank. The analysis of AVS tasks did not reveal a clear winner, although the average performance of vibro, IVIST, and VIREO teams were impressive. The analysis of agreement/disagreement with judges revealed a positive effect of the new query preparation process, which was supported by an online survey as well. For future VBS evaluations, we plan to make visual known-item search tasks harder (e.g., shorter targets segments or domain-specific collections) and reconsider AVS tasks (search for videos with a correct item).

Author Contributions Conceptualization: JL, LV, FS, WB, LS; Methodology: FS, LS, WB, JL; Data Curation: LV, NM, FS, LR, ZM, LS; Formal analysis and investigation: LV, NM, FS, LP, WB, KS, AD, LS; Writing - original draft preparation: JL, LV, NM, FS, LR, LP, WB, KS, OK, AD, LS; Software: LV, NM, FS, LR, ZM, AD, LS; Supervision: JL, LV, FS, LR.

Data and Code availability The data and code to reproduce graphs and tables of Sect. 4 are available at https://github.com/mesnico/VBS22-KIS-Analysis and 5 are available at https://github.com/sauterl/VBS22-AVS-Analysis.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. arXiv preprint arXiv:2204.14198 (2022)

2. Amato, G., Bolettieri, P., Carrara, F., Debole, F., Falchi, F., Gennaro, C., Vadicamo, L., Vairo, C.: VISIONE at VBS2019. In: International Conference on Multimedia Modeling, pp. 591–596. Springer (2019). https://doi.org/10.1007/978-3-030-05716-9_51

3. Amato, G., Bolettieri, P., Carrara, F., Debole, F., Falchi, F., Gennaro, C., Vadicamo, L., Vairo, C.: The visione video search system: Exploiting off-the-shelf text search engines for large-scale video retrieval. Journal of Imaging **7**(5) (2021). https://doi.org/10.3390/jimaging7050076

4. Amato, G., Bolettieri, P., Carrara, F., Falchi, F., Gennaro, C., Messina, N., Vadicamo, L., Vairo, C.: (2022) Visione at video browser showdown,: In: Huet, B., Gurrin, C., Tran, M.T., Dang-Nguyen, D.T., Hu, A.M.C., Huynh Thi Thanh, B., Huet, B. (eds.) Multi Media Modeling, pp. 543–548. Springer International Publishing, Cham (2022)

5. Amato, G., Bolettieri, P., Falchi, F., Gennaro, C., Messina, N., Vadicamo, L., Vairo, C.: VISIONE at video browser showdown 2021. In: International Conference on Multimedia Modeling, pp. 473–478. Springer (2021). Doi: https://doi.org/10.1007/978-3-030-67835-7_47

6. Andreadis, S., Moumtzidou, A., Galanopoulos, D., Pantelidis, N., Apostolidis, K., Touska, D., Gkountakos, K., Pegia, M., Gialampoukidis, I., Vrochidis, S., Mezaris, V., Kompatsiaris, I.: VERGE in vbs 2022. In: International Conference on Multimedia Modeling. Springer (2022)

7. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 4715–4723 (2019)

8. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9365–9374 (2019)

9. Bailer, W., Arnold, R., Benz, V., Coccomini, D., Gkagkas, A., Guðmundsson, G.T., Heller, S., Jónsson, B.T., Lokoč, J., Messina, N., Pantelidis, N., Wu, J.: Improving Query and Assessment Quality in Text-Based Interactive Video Retrieval Evaluation. In: Proceedings

of the 2023 ACM International Conference on Multimedia Retrieval. New York: Association for Computing Machinery, pp. 597–601 (2023). https://doi.org/10.1145/3591106.3592281

10. Bailey, P., Moffat, A., Scholer, F., Thomas, P.: Retrieval consistency in the presence of query variations. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 395–404 (2017)

11. Benavente, R., Vanrell, M., Baldrich, R.: Parametric fuzzy sets for automatic color naming. JOSA A **25**(10), 2582–2593 (2008)

12. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. CoRR arXiv:2004.10934 (2020)

13. Cao, Z., Hidalgo, G., Simon, T., Wei, S., Sheikh, Y.: Openpose: Real-time multi-person 2d pose estimation using part affinity fields. CoRR **abs/1812.08008** (2018)

14. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: Hybrid task cascade for instance segmentation. In: Conference on Computer Vision and Pattern Recognition pp. 4969–4978 (2019). https://doi.org/10.1109/CVPR.2019.00511

15. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)

16. Cox, I., Miller, M., Omohundro, S., Yianilos, P.: Pichunter: Bayesian relevance feedback for image retrieval. In: International Conference on Pattern Recognition, vol. 3, pp. 361–369. IEEE (1996). https://doi.org/10.1109/ICPR.1996.546971

17. Deng, D., Liu, H., Li, X., Cai, D.: Pixellink: Detecting scene text via instance segmentation. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), pp. 6773–6780. AAAI (2018)

18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009). https://doi.org/10.1109/CVPR.2009.5206848

19. Duane, A., Jónsson, B.T.: Virma: (2022) Virtual reality multimedia analytics at video browser showdown,: In: Huet, B.T., Gurrin, C., Tran, M.T., Dang-Nguyen, D.T., Hu, A.M.C., Huynh Thi Thanh, B., Huet, B. (eds.) MultiMedia Modeling, pp. 580–585. Springer International Publishing, Cham (2022)

20. Fang, H., Xiong, P., Xu, L., Chen, Y.: Clip2video: Mastering video-text retrieval via image clip. arXiv preprint arXiv:2106.11097 (2021)

21. Galanopoulos, D., Mezaris, V.: Attention mechanisms, signal encodings and fusion strategies for improved ad-hoc video search with dual encoding networks. In: International Conference on Multimedia Retrieval, pp. 336–340. ACM (2020). https://doi.org/10.1145/3372278.3390737

22. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448 (2015)

23. Gíslason, S., Jónsson, B., Amsaleg, L.: Integration of exploration and search: A case study of the m3 model. In: Proceedings of the International Conference on MultiMedia Modeling (MMM), Lecture Notes in Computer Science, pp. 156–168. Springer, Germany (2019). https://doi.org/10.1007/978-3-030-05710-7_13

24. Gkountakos, K., Touska, D., Ioannidis, K., Tsikrika, T., Vrochidis, S., Kompatsiaris, I.: Spatio-temporal activity detection and recognition in untrimmed surveillance videos. In: Proceedings of the 2021 International Conference on Multimedia Retrieval, pp. 451–455 (2021)

25. Gurrin, C., Zhou, L., Healy, G., Jónsson, B.Þ., Dang-Nguyen, D., Lokoc, J., Tran, M., Hürst, W., Rossetto, L., Schöffmann, K.: Introduction to the fifth annual lifelog search challenge, lsc'22. In: V. Oria, M.L. Sapino, S. Satoh, B. Kerhervé, W. Cheng, I. Ide, V.K. Singh (eds.) ICMR '22: International Conference on Multimedia Retrieval,

26. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6546–6555 (2018)

27. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)

28. Heller, S., Arnold, R., Gasser, R., Gsteiger, V., Parian-Scherb, M., Rossetto, L., Sauter, L., Spiess, F., Schuldt, H.: Multi-modal interactive video retrieval with temporal queries. In: International Conference on Multimedia Modeling, Lecture Notes in Computer Science. Springer (2022)

29. Heller, S., Arnold, R., Gasser, R., Gsteiger, V., Parian-Scherb, M., Rossetto, L., Sauter, L., Spiess, F., Schuldt, H.: Multi-modal Interactive Video Retrieval with Temporal Queries. In: MultiMedia Modeling, pp. 493–498. Springer International Publishing, Cham (2022)

30. Heller, S., Gasser, R., Illi, C., Pasquinelli, M., Sauter, L., Spiess, F., Schuldt, H.: Towards explainable interactive multi-modal video retrieval with vitrivr. In: Int. Conf. Multimed. Model., pp. 435–440. Springer, UK (2021)

31. Heller, S., Gsteiger, V., Bailer, W., Gurrin, C., Jónsson, B.Þ, Lokoc, J., Leibetseder, A., Mejzlík, F., Peska, L., Rossetto, L., Schall, K., Schoeffmann, K., Schuldt, H., Spiess, F., Tran, L., Vadicamo, L., Veselý, P., Vrochidis, S., Wu, J.: Interactive video retrieval evaluation at a distance: comparing sixteen interactive video search systems in a remote setting at the 10th video browser showdown. Int. J. Multim. Inf. Retr. **11**(1), 1–18 (2022). https://doi.org/10.1007/s13735-021-00225-2

32. Hezel, N., Barthel, K.U.: Dynamic construction and manipulation of hierarchical quartic image graphs. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, ICMR '18, p. 513-516. Association for Computing Machinery, New York, NY, USA (2018)

33. Hezel, N., Schall, K., Jung, K., Barthel, K.U.: Efficient search and browsing of large-scale video collections with vibro. In: B. Þór Jónsson, C. Gurrin, M.T. Tran, D.T. Dang-Nguyen, A.M.C. Hu, B. Huynh Thi Thanh, B. Huet (eds.) MultiMedia Modeling, pp. 487–492. Springer International Publishing, Cham (2022)

34. Ho, K., Dinh, V.X., Nguyen, H.Q., Le, K., Tran, K.D., Do, T., Mai, T.D., Ngo, T.D., Le, D.D.: Uit at vbs 2022: An unified and interactive video retrieval system with temporal search. In: MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6-10, 2022, Proceedings, Part II, p. 556-561. Springer (2022)

35. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transact. Patt. Anal. Mach. Intell. **33**(1), 117–128 (2010). https://doi.org/10.1109/TPAMI.2010.57

36. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International Conference on Machine Learning, p. 4904–4916. PMLR (2021)

37. Khan, O.S., Jónsson, B.T., Larsen, M., Poulsen, L., Koelma, D.C., Rudinac, S., Worring, M., Zahálka, J.: Exquisitor at the video browser showdown 2021: Relationships between semantic classifiers. In: MultiMedia Modeling: 27th International Conference, MMM 2021, Prague, Czech Republic, June 22-24, 2021, Proceedings, Part II, p. 410-416. Springer-Verlag (2021)

38. Khan, O.S., Jónsson, B.T., Rudinac, S., Zahálka, J., Ragnarsdóttir, H., Þorleiksdóttir, T., Guðmundsson, G.T., Amsaleg, L., Worring, M.: Interactive learning for multimedia at large. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I, p. 495-510. Springer-Verlag (2020)

39. Khan, O.S., Larsen, M.D., Poulsen, L.A.S., Jónsson, B.T., Zahálka, J., Rudinac, S., Koelma, D., Worring, M.: Exquisitor at the lifelog

Newark, NJ, USA, June 27 - 30, 2022, pp. 685–687. ACM (2022). https://doi.org/10.1145/3512527.3531439

search challenge 2020. In: Proceedings of the Third Annual Workshop on Lifelog Search Challenge, LSC '20, p. 19-22. Association for Computing Machinery (2020)

40. Khan, O.S., Sharma, U., Jónsson, B.T., Koelma, D.C., Rudinac, S., Worring, M., Zahálka, J.: Exquisitor at the video browser showdown 2022. In: MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6-10, 2022, Proceedings, Part II, p. 511-517. Springer-Verlag (2022)

41. Le, T.K., Ninh, V.T., Tran, M.K., Healy, G., Gurrin, C., Tran, M.T.: Avseeker: An active video retrieval engine at vbs2022. In: MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6-10, 2022, Proceedings, Part II, p. 537-542. Springer (2022)

42. Lee, S., Park, S., Ro, Y.M.: Ivist: Interactive video search tool in vbs 2022. In: MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6-10, 2022, Proceedings, Part II, p. 524-529. Springer (2022)

43. Leibetseder, A., Schoeffmann, K.: divexplore 6.0: Itec's interactive video exploration system at vbs 2022. In: International Conference on Multimedia Modeling, pp. 569–574. Springer (2022)

44. Li, X., Xu, C., Yang, G., Chen, Z., Dong, J.: W2VV++: Fully Deep Learning for Ad-hoc Video Search. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 1786–1794. ACM, Nice France (2019). https://doi.org/10.1145/3343031.3350906

45. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: European Conference on Computer Vision, pp. 121–137. Springer (2020)

46. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Computer Vision – ECCV, pp. 740–755. Springer (2014). https://doi.org/10.1007/978-3-319-10602-1_48

47. Lin, Z., Geng, S., Zhang, R., Gao, P., de Melo, G., Wang, X., Dai, J., Qiao, Y., Li, H.: Frozen clip models are efficient video learners. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer vision - ECCV 2022, pp. 388–404. Springer Nature Switzerland, Cham (2022)

48. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)

49. Lokoč, J., Mejzlík, F., Souček, T., Dokoupil, P., Peška, L.: Video search with context-aware ranker and relevance feedback. In: B. Þór Jónsson, C. Gurrin, M.T. Tran, D.T. Dang-Nguyen, A.M.C. Hu, B. Huynh Thi Thanh, B. Huet (eds.) MultiMedia Modeling, pp. 505–510. Springer International Publishing, Cham (2022)

50. Lokoč, J., Souček, T., Veselý, P., Mejzlík, F., Ji, J., Xu, C., Li, X.: A W2VV++ case study with automated and interactive text-to-video retrieval. In: International Conference on Multimedia. ACM (2020). https://doi.org/10.1145/3394171.3414002

51. Lokoč, J., Veselý, P., Mejzlík, F., Kovalčík, G., Souček, T., Rossetto, L., Schoeffmann, K., Bailer, W., Gurrin, C., Sauter, L., Song, J., Vrochidis, S., Wu, J., Jónsson, B.Þ.: Is the reign of interactive search eternal? findings from the video browser showdown 2020. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 17(3) (2021). https://doi.org/10.1145/3445031

52. Lokoč, J., Bailer, W., Schoeffmann, K., Muenzer, B., Awad, G.: On influential trends in interactive video retrieval: Video browser showdown 2015–2017. IEEE Transact. Multimed. 20(12), 3361–3376 (2018). https://doi.org/10.1109/TMM.2018.2830110

53. Lokoč, J., Peška, L.: A study of a cross-modal interactive search tool using clip and temporal fusion. In: Priya, D. (ed.) MultiMedia modeling - 29th international conference, mmm 2023, bergen, norway, January 9–12, 2023. Lecture Notes in Computer Science. Springer, UK (2023)

54. Luu, D.T., Quan, K.A.C., Nguyen, T.Q., Hua, V.S., Nguyen, M.C., Tran, M.T., Nguyen, V.T.: Cdc: Color-based diffusion model with caption embedding in vbs 2022. p. 575-579. Springer (2022)

55. Ma, Z., Wu, J., Hou, Z., Ngo, C.W.: Reinforcement learning-based interactive video search. In: B. Þór Jónsson, C. Gurrin, M.T. Tran, D.T. Dang-Nguyen, A.M.C. Hu, B. Huynh Thi Thanh, B. Huet (eds.) MultiMedia Modeling, pp. 549–555. Springer International Publishing, Cham (2022)

56. Markatopoulou, F., Mezaris, V., Patras, I.: Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation. IEEE Transact. Circuits Syst. Video Tech 29(6), 1631–1644 (2018)

57. Markatopoulou, F., Moumtzidou, A., Galanopoulos, D., Avgerinakis, K., Andreadis, S., Gialampoukidis, I., Tachos, S., Vrochidis, S., Mezaris, V., Kompatsiaris, I., Patras, I.: ITI-CERTH participation in TRECVID 2017. In: TREC Video Retrieval Evaluation. NIST (2017). https://doi.org/10.5281/zenodo.1183440

58. Messina, N., Falchi, F., Esuli, A., Amato, G.: Transformer reasoning network for image-text matching and retrieval. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 5222–5229. IEEE (2021)

59. Mettes, P., Koelma, D.C., Snoek, C.G.: The imagenet shuffle: Reorganized pre-training for video event detection. In: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ICMR '16, p. 175-182. Association for Computing Machinery (2016)

60. Nguyen, T.N., Puangthamawathanakun, B., Healy, G., Nguyen, B.T., Gurrin, C., Caputo, A.: Videofall - A Hierarchical Search Engine for VBS2022. In: MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6-10, 2022, Proceedings, Part II, p. 518-523. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-030-98355-0_48

61. 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). https://doi.org/10.1007/978-3-319-51811-4_9

62. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. CoRR (2021) arXiv:2103.00020

63. Revaud, J., Almazan, J., Rezende, R., de Souza, C.: Learning with average precision: Training image retrieval with a listwise loss. In: International Conference on Computer Vision, pp. 5106–5115. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00521

64. Rossetto, L., Gasser, R., Sauter, L., Bernstein, A., Schuldt, H.: A system for interactive multimedia retrieval evaluations. In: International Conference on Multimedia Modeling. Springer (2021). https://doi.org/10.1007/978-3-030-67835-7_33

65. Rossetto, L., Parian, M.A., Gasser, R., Giangreco, I., Heller, S., Schuldt, H.: Deep learning-based concept detection in vitrivr. In: International Conference on Multimedia Modeling, pp. 616–621. Springer (2019). https://doi.org/10.1007/978-3-030-05716-9_55

66. Rossetto, L., Schuldt, H., Awad, G., Butt, A.A.: V3C - A research video collection. In: International Conference on Multimedia Modeling, pp. 349–360. Springer (2019). https://doi.org/10.1007/978-3-030-05710-7_29

67. Sauter, L., Amiri Parian, M., Gasser, R., Heller, S., Rossetto, L., Schuldt, H.: Combining boolean and multimedia retrieval in vitrivr for large-scale video search. In: International Conference on Multimedia Modeling, pp. 760–765. Springer (2020). https://doi.org/10.1007/978-3-030-37734-2_66

68. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C.W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S.R., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5b: An open large-scale dataset for training next generation image-text models. In: Thirty-sixth

Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2022). https://openreview.net/forum?id=M3Y74vmsMcY

69. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018)

70. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER: An attentional scene text recognizer with flexible rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence 41(9), 2035–2048 (2019). https://doi.org/10.1109/TPAMI.2018.2848939

71. Spiess, F., Gasser, R., Heller, S., Parian-Scherb, M., Rossetto, L., Sauter, L., Schuldt, H.: Multi-modal video retrieval in virtual reality with vitrivr-vr. In: International Conference on Multimedia Modeling, Lecture Notes in Computer Science. Springer (2022)

72. Spiess, F., Gasser, R., Heller, S., Rossetto, L., Sauter, L., Schuldt, H.: Competitive interactive video retrieval in virtual reality with vitrivr-vr. In: International Conference on Multimedia Modeling, pp. 441–447. Springer (2021). https://doi.org/10.1007/978-3-030-67835-7_42

73. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence (2017)

74. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR (2019)

75. Tran, M.T., Hoang-Xuan, N., Trang-Trung, H.P., Le, T.C., Tran, M.K., Le, M.Q., Le, T.K., Ninh, V.T., Gurrin, C.: V-first: A flexible interactive retrieval system for video at vbs 2022. In: MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6-10, 2022, Proceedings, Part II, p. 562-568. Springer (2022)

76. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Transact. Image Process. 18(7), 1512–1523 (2009)

77. Veselý, P., Mejzlík, F., Lokoč, J.: Somhunter V2 at video browser showdown 2021. In: International Conference on Multimedia Modeling, pp. 461–466. Springer (2021). https://doi.org/10.1007/978-3-030-67835-7_45

78. Wu, J., Ngo, C.W.: Interpretable embedding for ad-hoc video search. In: Proceedings of the 28th ACM International Conference on Multimedia, MM '20, p. 3357-3366. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394171.3413916

79. Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B.: A survey of modern deep learning based object detection models. Digital Signal Processing p. 103514 (2022)

80. Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N.: VarifocalNet: An IoU-aware dense object detector. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2021)

81. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ADE20K dataset. Int. J. Comput. Vis. 127(3), 302–321 (2019). https://doi.org/10.1007/s11263-018-1140-0

Authors and Affiliations

Jakub Lokoč[1] · **Stelios Andreadis[10]** · **Werner Bailer[4]** · **Aaron Duane[6]** · **Cathal Gurrin[5]** · **Zhixin Ma[11]** · **Nicola Messina[2]** · **Thao-Nhu Nguyen[5]** · **Ladislav Peška[1]** · **Luca Rossetto[8]** · **Loris Sauter[3]** · **Konstantin Schall[9]** · **Klaus Schoeffmann[7]** · **Omar Shahbaz Khan[6]** · **Florian Spiess[3]** · **Lucia Vadicamo[2]** · **Stefanos Vrochidis[10]**

✉ Jakub Lokoč
jakub.lokoc@matfyz.cuni.cz

Stelios Andreadis
andreadisst@iti.gr

Werner Bailer
werner.bailer@joanneum.at

Aaron Duane
aadu@itu.dk

Cathal Gurrin
cathal.gurrin@dcu.ie

Zhixin Ma
zxma.2020@phdcs.smu.edu.sg

Nicola Messina
nicola.messina@isti.cnr.it

Thao-Nhu Nguyen
thaonhu.nguyen24@mail.dcu.ie

Ladislav Peška
ladislav.peska@matfyz.cuni.cz

Luca Rossetto
rossetto@ifi.uzh.ch

Loris Sauter
loris.sauter@unibas.ch

Konstantin Schall
konstantin.schall@htw-berlin.de

Klaus Schoeffmann
ks@itec.aau.at

Omar Shahbaz Khan
osk@oskhan.com

Florian Spiess
florian.spiess@unibas.ch

Lucia Vadicamo
lucia.vadicamo@isti.cnr.it

Stefanos Vrochidis
stefanos@iti.gr

[1] Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

[2] ISTI-CNR, Pisa, Italy

[3] Department of Mathematics and Computer Science, University of Basel, Basel, Switzerland

[4] Joanneum Research, Graz, Austria

[5] Dublin City University, Dublin, Ireland

[6] IT University of Copenhagen, Copenhagen, Denmark

[7] Klagenfurt University, Klagenfurt, Austria

[8] Departement of Informatics, University of Zurich, Zurich, Switzerland

[9] Visual Computing Group, HTW Berlin, Berlin, Germany

[10] Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), Thermi-Thessaloniki, Greece

[11] School of Computing and Information Systems, Singapore Management University, Singapore, Singapore