Interactive Multimodal Video Search: An Extended Post-Evaluation for the VBS 2022 Competition

Konstantin Schall[1*], Werner Bailer[2*], Kai-Uwe Barthel[1*], Fabio Carrara[3*], Jakub Lokoč[4*], Ladislav Peška[4*], Klaus Schoeffmann[5*], Lucia Vadicamo[3*] and Claudio Vairo[3*]

[1*]Visual Computing Group, HTW Berlin, Berlin, Germany.
[2*]JOANNEUM RESEARCH, Graz, Austria.
[3*]Institute of Information Science and Technologies (ISTI), CNR, Pisa, Italy.
[5*]Klagenfurt University, Klagenfurt, Austria.
[4*]Department of Software Engineering, Charles University, Prague, Czech Republic.

*Corresponding author(s). E-mail(s): konstantin.schall@htw-berlin.de;

Abstract

CLIP-based text-to-image retrieval has proven to be very effective at the interactive video retrieval competition *Video Browser Showdown* 2022, where all three top-scoring teams had implemented a variant of a CLIP model in their system. Since the performance of these three systems was quite close, this post-evaluation was designed to get better insights on the differences of the systems and compare the CLIP-based text-query retrieval engines by introducing slight modifications to the original competition settings. An extended analysis of the overall results and the retrieval performance of all systems' functionalities shows that a strong text retrieval model certainly helps, but has to be coupled with extensive browsing capabilities and other query-modalities to consistently solve known-item-search tasks in a large scale video database.

Keywords: Interactive video retrieval, video browsing, video content analysis, content-based retrieval, evaluations

Statements and Declarations

This work was partially funded by the European Commission under the projects "AI4Media - A European Excellence Centre for Media, Society and Democracy" (H2020 n^o 951911), "XRECO - XR mEdia eCOsystem" (HEU n^o 101070250), and Czech Science Foundation (GAČR) project 22-21696S. The authors have no competing interests to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article. The raw log data analysed during the current study is available in this repository: https://github.com/Visual-Computing/VBS22-Post-Evaluation. The repository also includes scripts to generate the presented results, tables and figures. We would also like to thank all colleagues and student that took the time to participate in this evaluation.

1 Introduction

Multimodal AI models, which learn relationships between natural language and images, have significantly improved automatic image content understanding and visual information retrieval in the last few years. One popular example of such a model is CLIP [1] (Contrastive Language-Image Pre-training), which has not only successfully demonstrated its great performance in matching text with images, but also it has been successfully used for image and video retrieval competitions. For instance, the top performing teams at the Video Browser Showdown (VBS) [2–4] competition, as well as at the Lifelog Search Challenge (LSC) [5], were all relying on CLIP models when building their retrieval engines. Interestingly, however, their performance in solving specific queries in these competitions was quite varying and the same task was not always solved by all the teams.

In this paper, we evaluate the CLIP-based retrieval performance of the top three systems that participated in the Video Browser Showdown (VBS) 2022 competition [2]. We set up a dedicated VBS-like competition with about 60 KIS (known-item search) tasks that needed to be solved by the teams. In order to level out user-based performance impact and test the three retrieval systems rather than individual users, each system is tested with four independent users. Additionally, in the first 45 seconds the teams were not allowed to change the text of the KIS query, and have to use the same text that is provided by the competition moderators. This specific setting allowed both, to measure system-level performance and find differences in the CLIP-based approaches.

We evaluate the three systems with several performance metrics (correct item rank, achieved VBS score, query frequency and mAP) and discuss the differences of the employed retrieval systems, which all operate on the same data to solve the same tasks.

Our results show that even though VISIONE has the best performing text-to-image search engine, VIBRO, the winning system of the Video Browser Showdown 2022 was able to secure the first place in this extended evaluation due to three main factors: one superuser with an outstanding performance, the support of extended browsing capabilities, and the use of a general-purpose nearest neighbor search model for image-based similarity queries. However, on a team-wide level, the performance differences between VIBRO and CVHUNTER (second place) are not statistically significant and a much larger amount of KIS tasks would have been needed to distinguish these two teams. This results indicate that even though a strong text-query method is capable of solving a large number of tasks, other features like image-based searches and visual browsing are very important to achieve a consistency in solving video-based known-item-search tasks.

2 Related Work

2.1 Interactive Retrieval Benchmarks

During the last decades, several highly recognized competitions emerged that provide benchmark datasets and unified evaluation procedures such that the participating approaches can be compared and ranked. For example, NIST organizes a respected TRECVID benchmark [6] focusing on different types of tasks like Ad-hoc search, Video to Text or Deep Video Understanding. The MediaEval benchmark [7] is another example of activities towards multimedia task description and standardization of evaluation methodology. Other competitions focus primarily on task categories, where not only ranking models but also good user interfaces are necessary for better performance. Out of many possible task categories [8], known-item search tasks became well established at the Video Browser Showdown [3, 9] and Lifelog Search [10, 11] challenges. Both competitions define known-item search tasks over a large dataset and organize annual meetings at the International Conference on Multimedia Modeling (MMM) and the ACM International Conference on Multimedia Retrieval (ICMR) respectively. The VBS challenge is the most related evaluation competition to this paper as the same dataset, task category, similar setting and evaluation procedures were used for the presented study. Furthermore, it was based on the results of VBS 2022 [2] that the top three systems were identified. The authors of the systems agreed to participate in a more comprehensive evaluation to reveal more insights to the performance of the systems and analyze the effect of different users.

One comparison of top-performing VBS teams was conducted previously [12], where SOMHunter and vitrivr, the two best-performing tools of VBS 2020 competition, were evaluated. In that study, SOMHunter significantly outperformed vitrivr, mainly due to the better text-to-image ranking model in combination with the used search strategy. Also, the authors conducted a bootstrap analysis to estimate the size of the study that would be necessary to reliably distinguish the best and the second-best team. In particular, to achieve a 95% confidence interval, approx. 20-25 tasks solved by 4-6 participants, or approx. 40 tasks solved by 2 participants were suggested. The dynamic nature of the field is shown by the fact that none of the tools mentioned is among the top 3 tools in the VBS 2022 competition. The currently evaluated tools have evolved in terms of query modalities, underlying retrieval models, as well as visualization options. Compared to the previous study, we altered the task settings and performed more in-depth analysis of user behavior, including usage statistics for various query paradigms. Finally, all three tools evaluated in this paper are much more similar in terms of the text-to-image retrieval model, which resulted in smaller performance differences.

2.2 Description of the Systems

Even though the performance of the video search systems vibro [13], CVHunter [14] and Visione [15] was quite similar in the VBS 2022, the video browsing tools have significant differences regarding their supported query modalities, underlying ranking models, presentation of retrieval results and browsing capabilities. However, the general approach of splitting up videos into segments (shots) and defining a representative frame (image) for each segment is used by all three systems with small differences in this procedure.

Considering all search related features of the three systems, the query types can be grouped as *Text*, *Image*, *Temporal*, *Multimodal* and *Other*. Starting with *Text*, all systems support rich text inputs by leveraging text-to-image models like CLIP [1]. vibro uses OpenAI's ResNet50x16 [16] CLIP-trained model and reduces the dimensionality of the 768-dimensional embeddings to 512 via PCA-whitening [17]. Additionally, the output is further quantized to byte-scale (INT8). While these steps might harm the text-to-image retrieval results, the memory footprint is greatly reduced. CVHunter also uses a CLIP-based model, the ViT-L/14 [18] variant that performed well in many benchmarks in the original paper. Visione is using a combination of two mutimodal joint embedding models: TERN [19] (for text-to-image retrieval) and CLIP2video [20] (for text-to-video retrieval).

Image-queries play an important role in vibro, since any image presented on the UI can be double-clicked to perform a new image-based search. A Swin-L@384 [21] model, pre-trained with ImageNet21k [22] (classification) was fine-tuned for content-based image retrieval with the ProxyAnchor loss function [23] and a combination of publicly available datasets with a total of over 100k classes. Furthermore, a simple binarization with threshold=0 per dimension was used to obtain memory efficient image embeddings. CVHunter uses the image embeddings from their CLIP model for image-as-example queries and implemented a Bayesian relevance feedback approach introduced in PicHunter [24]. A temporal variant of the model was supported as well [25]. Visione supports both visual and semantic similarity queries. The GEM [26] features are used to support visual similarity search. The features extracted using CLIP2video [20] are used to retrieve video clips that are semantically similar to a query video segment, while TERN [19] are used for searching video keyframes that are semantically similar to a query image.

Temporal-queries can be formulated for two consecutive shots with vibro, where each shot can be described by text or an image. CVHunter supports description of two temporally close video segments, where the relevance score of the first segment is combined with the relevance score of the best following segment within a search window. This aggregation can be further updated with temporal relevance feedback [25]. Visione uses a temporal quantization approach for combining two different queries and select results temporally close each other. Specifically, videos are divided into intervals of $T = 21$ seconds, and the best results for each query in each interval are retained. Only result pairs from the same video and with a temporal distance smaller than 12 seconds are then displayed in the UI.

The *Other* query modality category groups less commonly used features of the three systems. For VIBRO this includes color-based searches, i.e. a user can do multi-colored drawings on any selected image to modify the color layout of the image. CVHUNTER supports only text and image (kNN or relevance feedback) search. VISIONE also supports object and color-based queries. In the UI there is a canvas where the user can place objects and colors appearing in a target scene. To support this kind of query, three pre-trained *object detectors* (VfNet [27], Mask R-CNN [28], Faster R-CNN [29]) and two *chip-based color naming* [30, 31] are indexed.

On top of that, the VIBRO and VISIONE systems support merging of the previously described modalities (*Multimodal*-queries). However, for the case of VIBRO this was not used during this evaluation. VISIONE enables users to perform multimodal searches by combining textual queries and object/color-based queries. For instance, a user can specify objects in an image (e.g., a person and a dog) while also providing a textual description (e.g., "a man and a dog running in a park"). Moreover, users can issue two multimodal queries together to perform a temporal search, where the first query describes what happened before the second query.

VIBRO has two ways to display results of the current query. The first one is a simple list, arranged in scan-line order, sorted by the relevance to the query of each displayed item. The second one is the same result-list arranged on a 2D-grid with a SOM-like [32] algorithm, FLAS [33], using a combination of the image embeddings and a low-level descriptor to include color information in this sorting. The most relevant item will always be in the center. All items represent keyframes of all videos and none of the above display methods aggregate those keyframes into videos, leading to up to 1.7 million ranked items but only the most relevant 10,000 keyframes are displayed. In addition, VIBRO supports exploration of the entire keyframe collection by using of an exploration graph [13]. CVHUNTER allows to show top ranked selected frames or top ranked frames accompanied with their video context. For each displayed frame, it is possible to use playback of sampled video frames or show the whole video summary. Users can press a number on numeric keyboard to limit the number of displayed result set frames from each video. In the browsing interface of VISIONE, the search results are organized by videos, presenting one row per video containing up to 20 frames. The order of these video rows and the frames within them is determined by the retrieval model's scores. Each frame in the row has a menu that offers various options to the user. These options include conducting similarity searches, viewing the entire video starting from the selected frame, or getting a preview of the video around the chosen frame.

2.3 CLIP-Based Video Retrieval

The effectiveness of CLIP-based video retrieval is a well studied phenomenon and many different works use CLIP to produce video-level descriptors [20, 34–36]. The common idea of this field of research is to extract embeddings with CLIP-trained visual encoders from sub-sampled frames of each video (e.g. one frame per second) and then aggregate those frame-level embeddings to a single, video-level embedding. Those descriptors therefore allow more complex action-based textual queries. A simple aggregation method would be mean-pooling, but can be improved as seen in [20, 34, 35]. CLIP2Video [20] proposes to use a trainable transformer network [37] to achieve video-level features and XCLIP [34] presents a multi-grained contrastive learning module, to enhance the importance of frames that have a high affinity to some single words of the query sentence. Both methods start with CLIP pre-trained visual and textual encoders but fine-tune those networks in combination with the training of the weights of the newly introduced modules. Bain et. al. show that a parameter free, query-specific pooling approach can achieve very good results and outperforms CLIP2Video´s transformer-based aggregation, which used 19 million parameters. However, the downside of this approach is that all frame-level visual embeddings have to be stored to compute relevance scores for each textual query. This scores are then used to create a weighted-average pooling to form a query-specific video-level descriptor. Due to the nature of the V3C dataset and the VBS tasks, where short sequences have to be found in rather long videos, initial tests of the CVHunter and vibro teams showed that video retrieval models where only beneficial in specific tasks where actions where required to be described. However, in most task scenarios, it is more important to query particular easily distinguishable objects. This can better be achieved with the standard image-text CLIP models. To keep the memory footprint low, both teams have therefore decided to omit video level embeddings in their respective systems.

3 Extended Evaluation

3.1 Differences to the VBS Competition

The three introduced systems achieved very similar results at the main VBS competition. In order to get more detailed insights on their differences, we decided to introduce some changes in this extended evaluation. For reference, the typical VBS competition settings are described in [9].

Usually, a participating system is represented by up to two individual users at the main VBS event. If one user solves the current task, the team gets assigned a score and the second user does not longer have to solve the task. Those two users are often highly experienced in solving video retrieval tasks with their respective systems and usually compliment each other. Since this team-wide aggregation of performance makes it difficult to analyze user specific behaviour and performance, we omitted this default aggregation in this evaluation. Each team was asked to assign four users, the information about the users' experience can be found in Table 1.

Even though three types of tasks have to be solved at the main VBS event, the next change was to solely focus on the visual known-item-search (v-KIS) task category. This allowed us to perform a much higher volume of tasks, 57 compared to 10 at the VBS22 event for this particular category. The main purpose of the higher task volume was to obtain a much larger sample size and thus be able to draw more reliable conclusions about the performance of the respective systems.

The last change was the introduction of a predefined textual query and the restriction not to change this initial text for the first 45 second of each task. Since all three systems used different CLIP-based text-to-image retrieval models, we wanted to reduce the variance introduced by user-formulated queries and focus on a fair comparison of the systems' text-to-image retrieval performance. Furthermore, we hoped to gain insights on the browsing capabilities of the systems and the performance of retrieval models from other modalities such as image-as-example queries. Restricting the reformulation of text forced users to use other features of their video retrieval system, resulting in a more comprehensive evaluation process.

Table 1: Information on the 12 participants. Active VBS indicates participation as one of the two competition users. Passive VBS participation stands for working on the system, when it took part in a VBS competition.

User	Active VBS	Passive VBS	Experience with KIS tasks	Overall Score
vibro1	-	✓	✓	69.1
vibro2	✓	✓	✓	93.5
vibro3	✓	✓	✓	83.9
vibro4	-	-	-	86.0
cvhunter1	-	-	-	75.0
cvhunter2	✓	-	✓	84.0
cvhunter3	✓	✓	✓	85.8
cvhunter4	-	-	✓	85.6
visione1	✓	✓	✓	84.2
visione2	-	✓	✓	78.0
visione3	✓	-	✓	86.5
visione4	✓	✓	✓	85.4

3.2 Setup and Execution

The entire extended evaluation was conducted in a fully remote setting with DRES [38], a system for interactive multimedia retrieval evaluations. Since DRES has also been used at the Video Browser Showdown since 2020, the API communication has already been implemented for all of the three evaluated systems. The modified v-KIS tasks were displayed in the web-browser interface of DRES. Each task consists of one short segment of a single video from the V3C1 or V3C2 data sets [39] and a textual description of this clip. The users had a maximum of 300 seconds per task and each task is rated with a scoring function that assigns 0 to 100 points if a correct submission appears within the task time limit. The score consists of 50 points for solving a task, $(300 - t)/6$ points based on elapsed

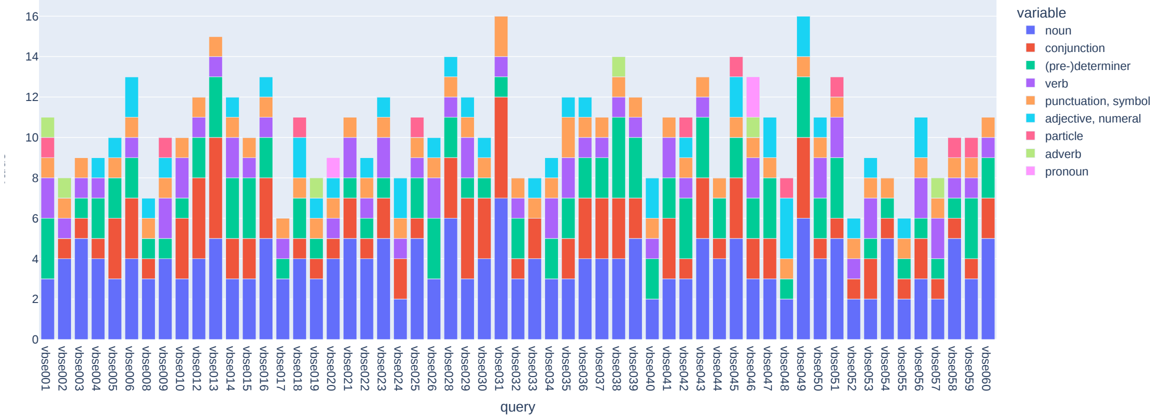

Fig. 1: Part of speech tagging analysis of the predefined queries.

submission time t, and a penalty for wrong submissions $x \cdot |WS|$, with $x = 10$ (one tenth of the maximum number of points).

3.3 Task Formulation

The target segments for the visual KIS tasks were selected following the established procedure described in [40]. While traditionally, visual KIS queries are typically 15-20s long, we wanted to have more short queries in this experiment. Thus, the selected queries have a mean duration of 8.2s (standard deviation 4.2s) and range from 2.6s to 21s.

For defining the predefined text queries to be initially used by the participants, an attempt was made to form a sentence with subject, predicate and object, and to add adjectives, quantifiers, etc. when necessary for a good description. The aim was to give a factual description of the main contents of the scene without being too specific, i.e., not as detailed as would be required for a textual KIS query. This should produce a result set after the initial query that is still large enough to use the browsing capabilities of the systems. Two VBS experts created the queries, each starting with queries for half of the tasks and the other reviewing and refining them. If necessary, details of the queries were discussed and jointly reformulated.

We performed an analysis of the predefined text queries using part-of-speech (POS) tagging from NLTK [41], using a coarser grouping (10 types) of the POS tags. The queries range in length from 3 to 16 words, and the typical query contains 3 nouns and 1-2 verbs. A plot of the POS tags is shown in Figure 1.

4 Analysis

During the system evaluation, each team maintained a record of user queries and the corresponding results for each task. In this section, we present a comprehensive analysis of these logs to gain a more in-depth understanding of system performance and user-interaction during the KIS tasks.

The logs are structured in JSON format, and each log contains details such as the team user identifier, timestamp, query description, and a list of ranked items retrieved by the systems for each specific query. To ensure data accuracy, we verified the consistency and synchronization of timestamps with the DRES local time and we filtered out records not related to active tasks. However, it's essential to acknowledge that circumstances beyond our control may have led to incomplete logs. For instance, VISIONE encountered issues recording logs of a single user in two tasks where the user did not submit any results. Furthermore, teams logged retrieved results up to a maximum rank of 10,000, but in certain cases, especially when using filters, the maximum rank may be less than 10,000 in the log files. As a result, the analysis using these logs should be considered an estimation of the system performance.

4.1 Overall results

Let us start with a simple binary metric, namely whether the user was able to solve a given task within the time limit. Of the 228 user-task pairs in total, VIBRO, CVHUNTER and VISIONE users managed to solve 199, 198, and 198 tasks, respectively. We can therefore conclude that there were no significant differences w.r.t. binary solved tasks metric and focus on the capability of individual tools to provide correct answers quickly and reliably.

For this, we used the same metric as in the VBS competition, denoted as *VBS score*. First, we focused on results, if all users solved the tasks independently. The mean per-user VBS scores were 73.02, 72.58, and 73.38 for VIBRO, CVHUNTER and VISIONE users (no statistically significant differences were found). Finally, we focused on the same scenario as in VBS competitions, i.e., all users of a single tool play as a team, and the score of the fastest team member (who found the correct solution for the task) is considered as the team score. With these settings, the mean per-team VBS scores were 87.85, 85.77, and 78.84 for VIBRO, CVHUNTER and VISIONE. The differences were statistically significant between VIBRO and VISIONE (p-value: 0.006 w.r.t. one-sided paired t-test), and between CVHUNTER and VISIONE (p-value: 0.049).

We also conducted a bootstrap analysis to verify the significance of the results and to estimate the necessary study sizes to reliably distinguish the performance of individual approaches. In particular, we draw k tasks, $1 \leq k \leq 200$ with repetition and calculated total per-team VBS scores for these tasks. Then, we evaluated whether each team was better than the other two. For each k, the task selection was repeated 500 times, and we report the percentage of cases, where one team was better than the other. Results of the bootstrap analysis confirmed the t-test values when we sampled the same volume of tasks as in the actual volume of evaluated tasks (i.e., $k = 57$). For these settings, VIBRO was better than CVHUNTER in 82% of cases and better than VISIONE in 100% of cases, while CVHUNTER was better than VISIONE in 95% of cases. The minimal necessary size of the study to reliably distinguish between VIBRO and VISIONE (w.r.t. 95% confidence) was $\sim$ 20 tasks. In order to estimate the necessary size of the study, where we can reliably distinguish VIBRO and CVHUNTER, one would need to extend far beyond the size of the conducted study. In particular, the bootstrap analysis suggests that the required study size would be $\sim$ 160 tasks.

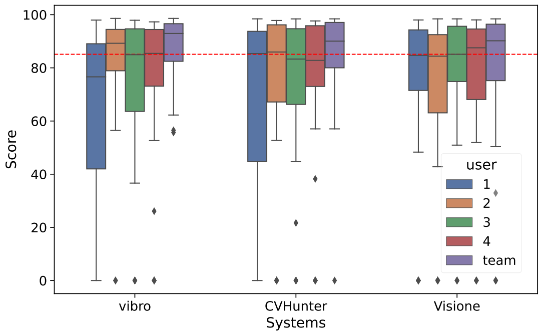

Fig. 2: Distribution of VBS scores for each system user over all the tasks, the results for "user"="team" are VBS scores when users of the same system are treated as a unique team. The red line is the median score over all the 12 users.

4.2 Individual Users vs System as a Team

Figure 2 displays the distribution of *VBS scores* for each user within each system, as well as the *team* scores computed based on the collective performance of users within the same system, acting as a unified team. The calculation of the team score takes into account the time of the first correct score submission of a team member, while at the same time imposes a penalty for all incorrect submissions of a team member before the first correct submission. A noticeable observation is that both VIBRO and CVHUNTER systems have a user (user 1 in both cases) who achieved significantly lower scores compared to other team members. In the case of CVHUNTER, this discrepancy can be attributed to the fact that cvhunter1 was a novice user. As for vibro1, it appears that this user encountered difficulties in resolving the queries. For VISIONE, the distribution of scores among users is more evenly distributed, although visione2 fell slightly behind compared to its team members. This could be due to the fact that visione2 had no prior competition experience, despite having contributed to the system development. Furthermore, it is worth noting that VIBRO secured the

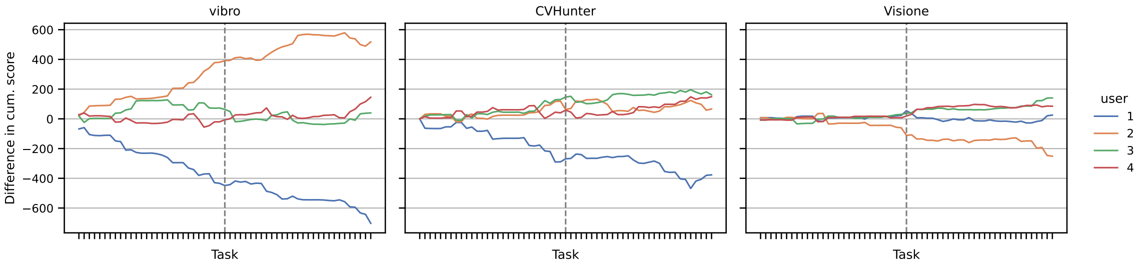

Fig. 3: Difference of the cumulative VBS score w.r.t. the user average cumulative score.

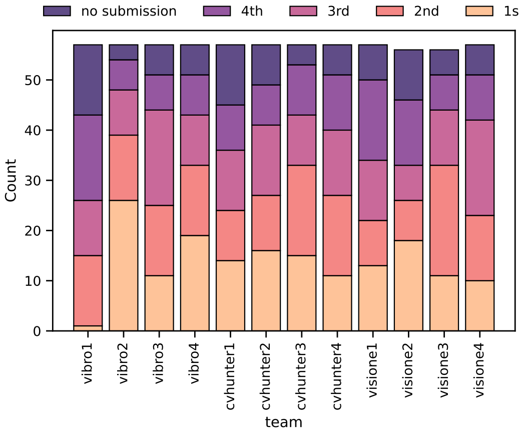

Fig. 4: Ranks of users related to their teams for each task.

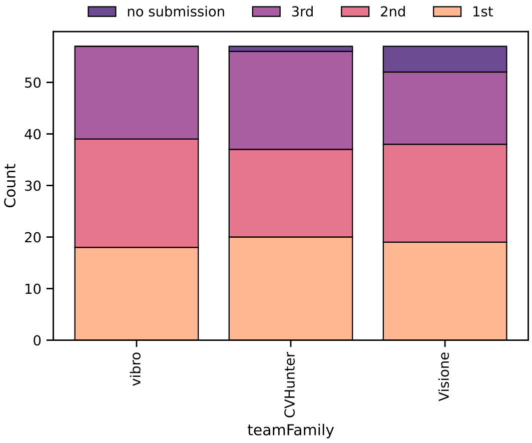

Fig. 5: Ranks of teams for each task.

first position both as a single user (vibro2) and as a team. On the other hand, VISIONE achieved the second position as a single user (visione3), but ranked third as a team, behind CVHUNTER.

Figure 3 presents the difference between the cumulative *VBS score* of each user and the average cumulative score for each system in the competition. We can observe that the VIBRO system has two users whose performance closely approaches the average score, an outstanding "superuser" (vibro2) who significantly outperforms the average, and another user (vibro1) who performs significantly worse than the rest of the team. Similarly, the CVHUNTER system exhibits a user (cvhunter1 who was a novice user) who achieved a significantly lower score compared to the others, while the overall performance of the remaining users is relatively consistent. In contrast, the VISIONE system demonstrates a more stable performance across all its users, with only a slight divergence observed in the final queries. In particular, this deviation was most evident with visione2, the only team member with no prior competition experience. The dotted vertical line represents the lunch break, and it is worth noting also that the queries in the morning and afternoon sessions were selected by different individuals. Interestingly, in the afternoon session, the difference between the cumulative scores of users and the average score tends to increase. This could be attributed to the selection of more challenging queries during the session, as well as potential fatigue experienced by the users.

See Figure 4 for an analysis of the ranks achieved by each user in the queries relevant to their respective teams. This includes the number of times they ranked first within their team and the occurrences of no submissions. A notable observation from the figure is that vibro2 consistently ranked first in his team and had the fewest instances of no submissions. Furthermore, it is evident that each system has one user with

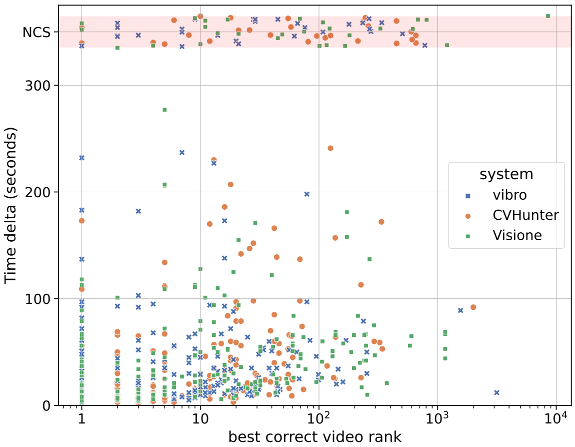

Fig. 6: Best video rank vs time delta between correct submission time and the time of the best video rank

a higher number of no submissions compared to their teammates. These users are vibro1, visione2, and cvhunter1.

It is worth noting that although visione2 had the highest number of no submissions within their team, he also ranked first most frequently. This indicates that despite his lack of experience in using the system, when he formulated the correct queries, he was the fastest among their team members in finding the correct results. Moving on to Figure 5, it presents the same plot but considers the users as a team, with the first user to find the correct answer being considered for each task. We can observe that the VIBRO team had zero instances of no submissions, indicating that at least one member of the team consistently found the correct result. The CVHunter team had a relatively low number of no submissions, while the Visione team experienced a higher number of no submissions. These findings align with the overall competition results, where the VIBRO team secured the first position, followed by CVHunter in second place, and Visione in third place.

We also investigate the correlation between the best video rank and the corresponding submission time for each task. We present the results in Figure 6, where the x-axis represents the best rank of the searched video, while the y-axis displays the time in seconds from the beginning of the task until the correct submission occurred. An important aspect to note in this plot is the presence of outliers. Under normal circumstances, when a video is ranked among the top positions (around 10), the submission time should be relatively low (below 100 seconds). However, it is evident that there are several instances across all three systems where the rank was below 10, but the submission time is unexpectedly high or even absent (indicating no submission). This discrepancy could be attributed to various factors, such as the frame displayed in the interface not being representative of the searched video or the user not identifying it promptly. Furthermore, it is interesting to observe that there are cases where the video was ranked very low (beyond the 1,000th position), but the submission time remains relatively low (below 100 seconds) in a few instances for Visione, a couple of instances for VIBRO, and once for CVHunter. In these cases, the browsing ability of the users proved to be beneficial in quickly finding the correct video despite its lower (initial) ranking. Overall, this figure highlights the variability in submission time and rank, indicating the influence of factors such as video representation, user perception, and browsing capabilities in the competition results.

Based on the analysis, we can draw several conclusions regarding the performance of each system and its respective teams in the competition. Vibro is probably the most effective system. The outstanding performance of the "superuser" vibro2 played a significant role in securing the team's first position. However, even as a novice user, vibro4 achieved the 3rd highest user score in the competition. In contrast, Visione demonstrated more consistent results among its users, which translated into a more balanced performance as a team (as observed in Figure 2). If a user struggled to find a specific video, it was likely that other team members faced similar difficulties. Consequently, the performance of Visione as a team is closely aligned with the collective performance of its members. CVHunter, on the other hand, exhibited a different dynamic. While the individual users' results were not particularly impressive (each user had a noticeable number of no submissions, comparable to Visione team members as reported in Figure 4), the team as a whole managed to compensate for these individual errors. This is evident from the relatively low number of no submissions achieved by the

CVHunter team (see Figure 5), ultimately securing their second-place position. This implies that the CVHunter system possesses sufficient flexibility to yield diverse results from different users utilizing the system.

4.3 User-Specific Interaction with Retrieval Models

To gain insights on the user-specific interactions with their systems, we first analyzed the individual queries that were formulated by each user in order to solve the tasks and divided all queries into three time ranges. The individual results are depicted in Figure 7 and show big differences in usage-patterns between and in-between the teams. The first time-frame is the first 45 seconds of each task, since this was the range where the predefined text was not allowed to be altered by the users and had to be used as the first query for each task. The number of completed tasks was 57, therefore the amount of text-queries that have been formulated in this time-range is close to this number. Discrepancies occur, since some users had problems with their systems during a few of the tasks or the system encountered a problem with the logging mechanism. Most users spent the first 45 seconds inspecting the initial queries results. This is especially true for Visione. For all four users, only a small amount of queries from other modalities than text are used during this time. For the other two systems, image-queries where used quite often, especially by the best performing user, vibro2. The second temporal category includes queries between 45 and 90 seconds into each task. This was the time that allowed users to rephrase the initial text description and therefore text was the most popular query modality here across all users. Again, the only exception is vibro2. However, both cvhunter2 and cvhunter3 also had a large proportion of image-queries during this time range. The last time-range includes the remaining time of the tasks, 90-300 seconds. It can be observed that a significant number of users shift towards query-modalities that were less frequently employed in the earlier time ranges. For example, temporal and multimodal queries gain popularity and a lot of users from CVHunter and vibro rely on images as the most dominant query-type. Outliers are the two users with the least experience (Table 1) vibro4 and cvhunter1.

Next, we analyzed the performance of the systems underlying retrieval models and used the mean average precision (mAP) metric as a performance measure. Since there is only one relevant video for each task, the average precision can simply be calculated as the reciprocal value of the rank of the first item from the current tasks video for each query. Given that only 10,000 items from the result lists were logged by the systems, the mAP is a robust metric for outliers or items that not had been logged. All but the initial text queries are additionally affected by the users query formulation abilities. Experience with the system might be such a factor. Therefore, we first compare the initial text-queries performance and the results of Table 2 show that Visiones text-query retrieval model yields significantly higher average precision scores across all of those queries. CVhunters superior performance over vibro can be explained by the use of a better model (ViT-L vs ResNet50x16) and vibro's compression and quantization of the embeddings.

Next, we investigate the performance of the retrieval systems in hard tasks, which are defined per system individually, specifically as tasks, where no user from the system could solve the task solely with the initial query. We can observe that the initial queries obtain far worse mAPs in this scenario and extensive browsing would be needed to find the relevant video. Once the 45 seconds have passed and the users are allowed to reformulate the given text, the user-formulated text queries provide considerably better, but still not sufficient results. This indicates that text-to-image models like CLIP fail to match images with text in this hard task scenarios. An example for such a task is "Flashing shots of a man on a bed and in front of a wall". Neither of the three system could solve the task with this given query. However, when looking at the performance of the second most popular query type, image, we can see that significantly superior mAPs could be achieved. Especially vibro's image retrieval engine performs very well during the hard tasks. Possible explanations are that due to vibro browsing capabilities it is easier to find fitting queries and the use of a retrieval specific image model that was designed to work on visual, rather than semantic aspects of the images.

Figure 8 shows the progression of each users mAP over time, where the mean is calculated

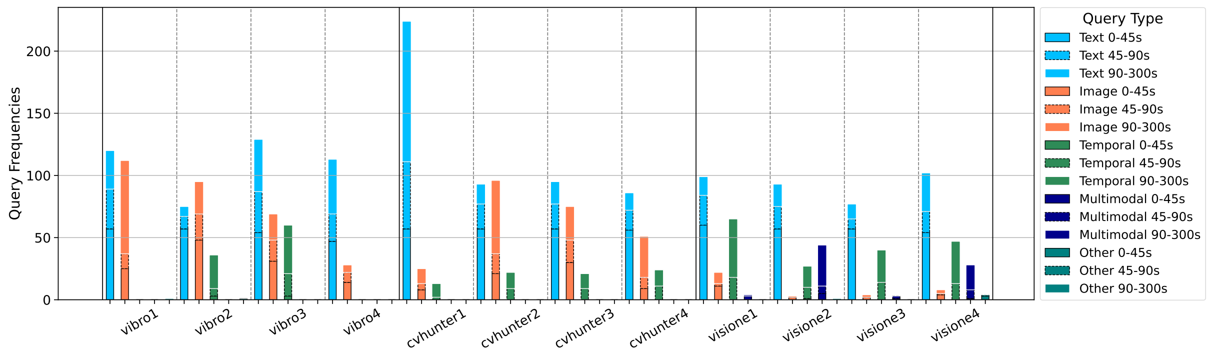

Fig. 7: Query type frequencies for each individual user

Table 2: Video mAP for pre-defined, initial text queries from all (A) and hard tasks (H). Hard tasks where defined per system and include tasks, where no user of the respective system could find the video with solely the initial query. The number after the slash indicates the amount of unique queries in the respective category.

System	Initial Text (A)	Initial Text (H)	User Text (H)	Image (H)
vibro	0.108/57	0.011/33	0.029/215	0.129/199
cvhunter	0.138/57	0.006/34	0.039/237	0.053/182
visione	0.183/57	0.004/40	0.024/128	0.056/28

across all tasks. We can observe that VISIONE users get a head-start for the aforementioned reasons at the beginning of the tasks but struggle to find queries that would significantly boost the rank of the relevant video afterwards. On the other hand, even though VIBRO and CVHUNTER users begin the tasks with lower mAP values, their systems are able to improve the relevant video rank through user formulated queries more often. Even though text-queries where not allowed up until the 45 seconds mark, vibro2 was able to achieve the best mAP at this point and on average, more than doubled this metric compared to the starting point of his initial query. This diagram also shows clear differences in the interaction between users and their systems. For example, given that three CVHUNTER users (2, 3 & 4) achieved very similar VBS scores at the end of the competition, cvhunter3 was consistently able to find queries that scored better mAPs compared to the other CVHUNTER users.

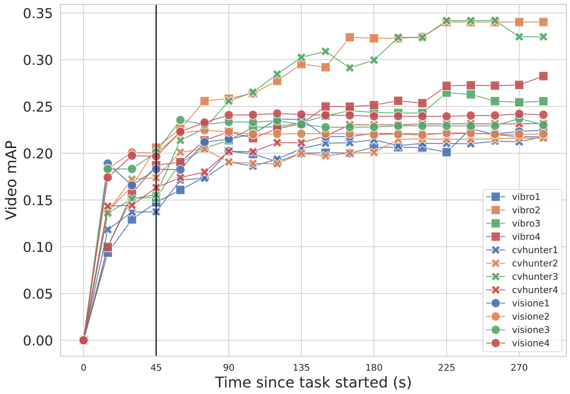

Fig. 8: Development of the video mAP over time for each user. Queries from all modalities where used and the mean was calculated across all tasks. Browsing actions like scrolling and switching views are not included in this Figure and the mAPs are solely computed from the logged result lists of each query.

4.4 Reformulation of the Pre-defined Text Queries

We analyse how the participants made use of the predefined text query, and the changes made to narrow down the content-set. We provide a visualization of the times of query changes and submissions per user and task in Figure 9. First, we see that some reformulations have taken place within the first 45s (where the predefined query should stay unchanged), which is mostly due to copy/paste errors and their correction. Most of these can be considered negligible, however, we observe that for example including a full stop or not may impact the result list created with CLIP.

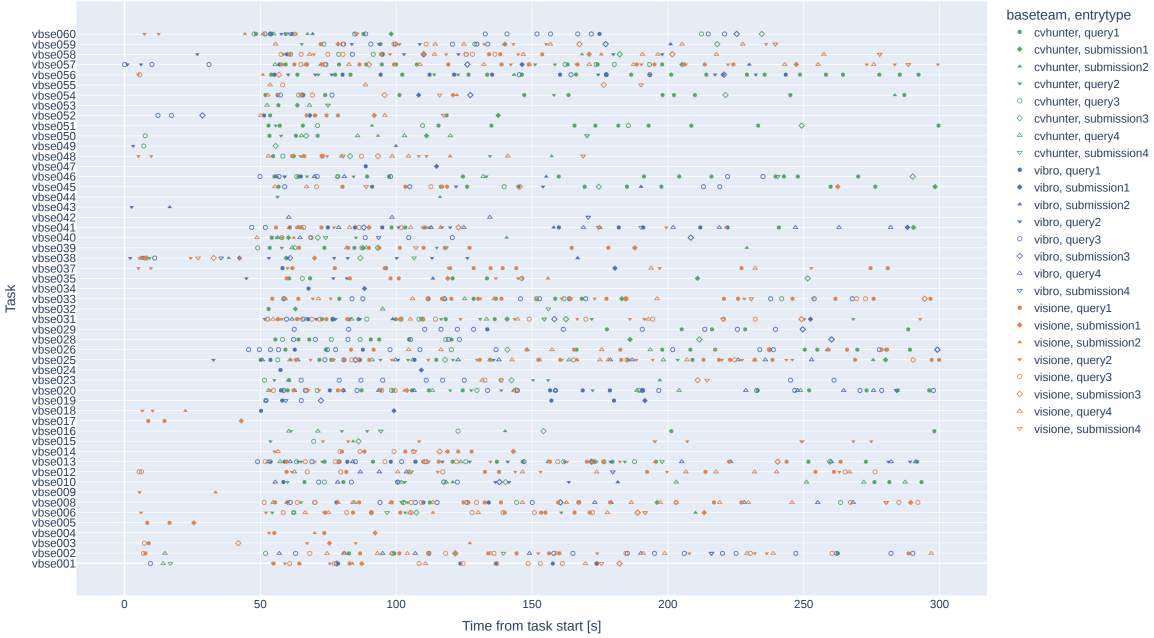

Fig. 9: Times of text query changes and submissions per query and user: colors denote the different teams, symbols the different users (filled: text query change, outline: submission).

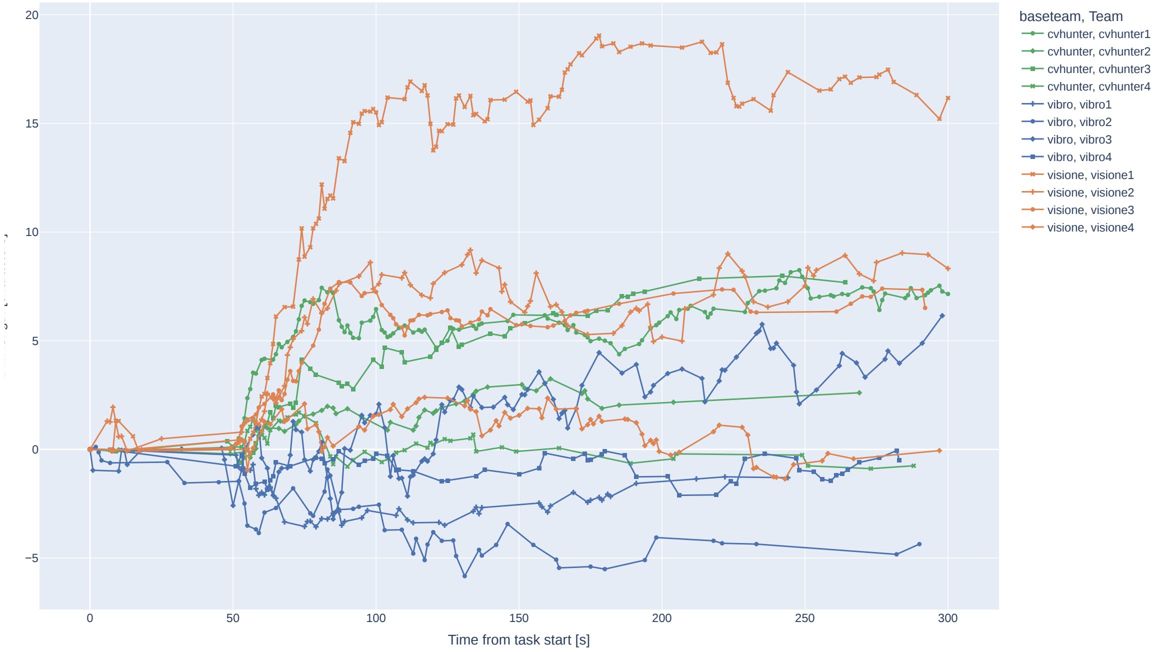

Fig. 10: Relative lengths of the text queries compared to the predefined query. Each line represents the mean of length differences at the specified time into the task working time across all tasks. For tasks that have already been solved at a particular time, the length is the last query is used in the mean calculation (in order to keep the number of queries considered constant).

The visualisation gives a good indication of the difficulty of tasks, and the amount of text query changes done by the different teams.

We also looked into how the text queries changed. The most common changes of queries involve adding adjectives or numerals (on average 0.5 per task, and quite consistent for all teams and over working time), as well as adding conjunctions and nouns. Here it is interesting that for vibro and CVhunter on average 0.5 of these types of words are added to the first modified query, which increases to on average 1.0 to the final query before submission. For Visione the average number is 0 over the working time, but with quite high variability in terms of added/removed words between team members and tasks.

In order to understand the trends in query reformulation applied by different users (or influenced by the tool) we analyse the lengths of the queries over the working time. Figure 10 shows the mean length differences (over all tasks) of queries per user over the working time, i.e. the length is expressed as the difference to the length of the predefined query. Each point in the plot means that the query changed for at least one task at that working time into the task. It becomes apparent from the figure, that for Visione and CVhunter the query lengths tend to increase for 3 out of 4 users, and stay similar for one user. In contrast, the query lengths rather tend to decrease for 3 out of 4 vibro users, and slightly increases for the other one. These observations seem consistent with those from other data, showing that Visione results hinge more on text search, while vibro users' success is often due to browsing capabilities.

5 Conclusion

This post-evaluation aimed to gain insights on performance differences between the three top-scoring teams at the interactive video retrieval competition VBS22. Even though the amount of KIS tasks was largely increased in this post-evaluation, the systems ranked in the same order, i.e. vibro first, CVhunter second and Visione third when aggregating the performance on a system level. Comparing the individual users showed a slightly different picture with two vibro users in the top-3 (first and third) and one Visione user ranked second, followed by two CVhunter users on fourth and fifth place. Analyzing the user specific interactions with their respective systems showed that Visione mostly relies on text queries and achieved the best text-to-image retrieval performance across the three systems. On the other hand, vibro and CVhunter performed a significantly larger amount of image-to-image queries, which is especially true for the more experienced users of the two systems. Vibro's success at this post-evaluation can be explained with three factors. First, the user vibro2 showed an outstanding performance at solving know-item-search tasks and greatly influenced vibro's overall VBS score. Second, compared to the other two systems, vibro offers more advanced browsing capabilities, which especially helped during the 45 seconds of each task, where the initial query-text was not allowed to be modified. Third, since vibro browsing mostly relies on visual embeddings of video keyframes, the use of a model optimized for general-purpose nearest neighbor search to extract those embeddings has proven to be especially beneficial in hard tasks, i.e. tasks where the CLIP-based text-queries failed to achieve good results. Additionally, the introduction of a pre-defined initial text query helped to compare the CLIP-based retrieval engines, and allowed to analyze the reformulation of this text. Even though reformulation was moderate, we observed that Visione formulated longer queries compared to the two other teams.

References

[1] Radford, A. *et al.* Meila, M. & Zhang, T. (eds) *Learning transferable visual models from natural language supervision.* (eds Meila, M. & Zhang, T.) *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139 of *Proceedings of Machine Learning Research*, 8748–8763 (PMLR, 2021). URL https://proceedings.mlr.press/v139/radford21a.html.

[2] Lokoč, J. *et al.* Interactive video retrieval in the age of effective joint embedding deep models: lessons from the 11th vbs. *Multimedia Systems* (2023). URL https://doi.org/10.1007/s00530-023-01143-5.

[3] Heller, S. *et al.* Interactive video retrieval evaluation at a distance: comparing sixteen

interactive video search systems in a remote setting at the 10th video browser showdown. *International Journal of Multimedia Information Retrieval* **11**, 1–18 (2022). URL https://doi.org/10.1007/s13735-021-00225-2.

[4] Lokoč, J. *et al.* Is the reign of interactive search eternal? findings from the video browser showdown 2020. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **17** (2021). URL https://doi.org/10.1145/3445031.

[5] Gurrin, C. *et al.* Kompatsiaris, I. Y. *et al.* (eds) *Introduction to the sixth annual lifelog search challenge, lsc'23.* (eds Kompatsiaris, I. Y. *et al.*) *Proc. International Conference on Multimedia Retrieval (ICMR'23)* (ACM, Thessaloniki, Greece, 2023).

[6] Awad, G. *et al.* Awad, G. (ed.) *An overview on the evaluated video retrieval tasks at trecvid 2022.* (ed.Awad, G.) *Proceedings of TRECVID 2022* (NIST, USA, 2022).

[7] Constantin, M. G., Hicks, S., Larson, M. & Nguyen, N.-T. MediaEval multimedia evaluation benchmark: Tenth anniversary and counting. *ACM SIGMM Records* **12** (2020).

[8] Lokoč, J. *et al.* Þór Jónsson, B. *et al.* (eds) *A task category space for user-centric comparative multimedia search evaluations.* (eds Þór Jónsson, B. *et al.*) *International Conference on Multimedia Modeling* (2022).

[9] Lokoč, J., Bailer, W., Schoeffmann, K., Münzer, B. & Awad, G. On influential trends in interactive video retrieval: video browser showdown 2015–2017. *IEEE Transactions on Multimedia* **20**, 3361–3376 (2018).

[10] Gurrin, C. *et al.* Oria, V. *et al.* (eds) *Introduction to the fifth annual lifelog search challenge, lsc'22.* (eds Oria, V. *et al.*) *ICMR '22: International Conference on Multimedia Retrieval, Newark, NJ, USA, June 27 - 30, 2022*, 685–687 (ACM, 2022). URL https://doi.org/10.1145/3512527.3531439.

[11] Tran, L. *et al.* Comparing interactive retrieval approaches at the lifelog search challenge 2021. *IEEE Access* **11**, 30982–30995 (2023). URL https://doi.org/10.1109/ACCESS.2023.3248284.

[12] Rossetto, L. *et al.* On the user-centric comparative remote evaluation of interactive video search systems. *IEEE MultiMedia* (2021). URL https://doi.org/10.1109/MMUL.2021.3066779.

[13] Hezel, N., Schall, K., Jung, K. & Barthel, K. U. Þór Jónsson, B. *et al.* (eds) *Efficient search and browsing of large-scale video collections with vibro.* (eds Þór Jónsson, B. *et al.*) *MultiMedia Modeling*, 487–492 (Springer International Publishing, Cham, 2022).

[14] Lokoč, J., Mejzlík, F., Souček, T., Dokoupil, P. & Peška, L. Þór Jónsson, B. *et al.* (eds) *Video search with context-aware ranker and relevance feedback.* (eds Þór Jónsson, B. *et al.*) *MultiMedia Modeling*, 505–510 (Springer International Publishing, Cham, 2022).

[15] Amato, G. *et al.* Þór Jónsson, B. *et al.* (eds) *Visione at video browser showdown 2022.* (eds Þór Jónsson, B. *et al.*) *MultiMedia Modeling*, 543–548 (Springer International Publishing, Cham, 2022).

[16] He, K., Zhang, X., Ren, S. & Sun, J. ... (ed.) *Deep residual learning for image recognition.* (ed....) *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[17] Philbin, J., Chum, O., Isard, M., Sivic, J. & Zisserman, A. ... (ed.) *Object retrieval with large vocabularies and fast spatial matching.* (ed....) *2007 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, 2007).

[18] Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR* (2020).

[19] Messina, N., Falchi, F., Esuli, A. & Amato, G. ... (ed.) *Transformer reasoning network for image-text matching and retrieval.* (ed....)

2020 25th International Conference on Pattern Recognition (ICPR), 5222–5229 (IEEE, 2021).

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

[21] Liu, Z. *et al.* Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030* (2021).

[22] Russakovsky, O. *et al.* Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)* (2015).

[23] Kim, S., Kim, D., Cho, M. & Kwak, S. ... (ed.) *Proxy anchor loss for deep metric learning.* (ed....) *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[24] Cox, I., Miller, M., Omohundro, S. & Yianilos, P. ... (ed.) *Pichunter: Bayesian relevance feedback for image retrieval.* (ed....) *International Conference on Pattern Recognition*, Vol. 3, 361–369 (IEEE, 1996). URL https://doi.org/10.1109/ICPR.1996.546971.

[25] Lokoc, J. & Peska, L. Dang-Nguyen, D. *et al.* (eds) *A study of a cross-modal interactive search tool using CLIP and temporal fusion.* (eds Dang-Nguyen, D. *et al.*) *MultiMedia Modeling - 29th International Conference, MMM 2023, Bergen, Norway, January 9-12, 2023, Proceedings, Part I*, Vol. 13833 of *Lecture Notes in Computer Science*, 397–408 (Springer, 2023). URL https://doi.org/10.1007/978-3-031-27077-2_31.

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

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

[28] He, K., Gkioxari, G., Dollár, P. & Girshick, R. ... (ed.) *Mask r-cnn.* (ed....) *Proceedings of the IEEE international conference on computer vision*, 2961–2969 (2017).

[29] Girshick, R. ... (ed.) *Fast r-cnn.* (ed....) *Proceedings of the IEEE international conference on computer vision*, 1440–1448 (2015).

[30] Van De Weijer, J., Schmid, C., Verbeek, J. & Larlus, D. Learning color names for real-world applications. *IEEE Transactions on Image Processing* **18**, 1512–1523 (2009). URL https://doi.org/10.1109/TIP.2009.2019809.

[31] Benavente, R., Vanrell, M. & Baldrich, R. Parametric fuzzy sets for automatic color naming. *JOSA A* **25**, 2582–2593 (2008). URL https://doi.org/10.1364/JOSAA.25.002582.

[32] Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics* **43**, 59–69 (1982).

[33] Barthel, K. U., Hezel, N., Jung, K. & Schall, K. Improved evaluation and generation of grid layouts using distance preservation quality and linear assignment sorting. *Computer Graphics Forum* **n/a** (2023).

[34] Ma, Y. *et al.* X-clip: End-to-end multi-grained contrastive learning for video-text retrieval 638–647 (2022). URL https://doi.org/10.1145/3503161.3547910.

[35] Bain, M., Nagrani, A., Varol, G. & Zisserman, A. A clip-hitchhiker's guide to long video retrieval (2022). 2205.08508.

[36] Ali, A., Schwartz, I., Hazan, T. & Wolf, L. Video and text matching with conditioned embeddings 1565–1574 (2022).

[37] Vaswani, A. *et al.* Guyon, I. *et al.* (eds) *Attention is all you need.* (eds Guyon, I. *et al.*) *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, Inc., 2017). URL https://proceedings.

neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[38] Rossetto, L., Gasser, R., Sauter, L., Bernstein, A. & Schuldt, H. Lokoc, J. *et al.* (eds) *A system for interactive multimedia retrieval evaluations.* (eds Lokoc, J. *et al.*) *International Conference on Multimedia Modeling* (Springer, 2021). URL https://doi.org/10.1007/978-3-030-67835-7_33.

[39] Rossetto, L., Schuldt, H., Awad, G. & Butt, A. A. Kompatsiaris, I. *et al.* (eds) *V3C - A research video collection.* (eds Kompatsiaris, I. *et al.*) *International Conference on Multimedia Modeling*, 349–360 (Springer, 2019). URL https://doi.org/10.1007/978-3-030-05710-7_29.

[40] Lokoč, J. *et al.* Interactive search or sequential browsing? a detailed analysis of the video browser showdown 2018. *ACM Trans. Multimedia Comput. Commun. Appl.* **15** (2019). URL https://doi.org/10.1145/3295663.

[41] Bird, S., Klein, E. & Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit* (" O'Reilly Media, Inc.", 2009).