Information Retrieval Journal (2023) 26:6
https://doi.org/10.1007/s10791-023-09425-2

Multimodal video retrieval with CLIP: a user study

Tayfun Alpay[1] · **Sven Magg**[1] · **Philipp Broze**[2] · **Daniel Speck**[1]

Received: 20 May 2023 / Accepted: 12 September 2023 / Published online: 29 September 2023

Abstract
Recent machine learning advances demonstrate the effectiveness of zero-shot models trained on large amounts of data collected from the internet. Among these, CLIP (Contrastive Language-Image Pre-training) has been introduced as a multimodal model with high accuracy on a number of different tasks and domains. However, the unconstrained nature of the model begs the question whether it can be deployed in open-domain real-word applications effectively in front of non-technical users. In this paper, we evaluate whether CLIP can be used for multimodal video retrieval in a real-world environment. For this purpose, we implemented IMPA, an efficient shot-based retrieval system powered by CLIP. We additionally implemented advanced query functionality in a unified graphical user interface to facilitate an intuitive and efficient usage of CLIP for video retrieval tasks. Finally, we empirically evaluated our retrieval system by performing a user study with video editing professionals and journalists working in the TV news media industry. After having the participants solve open-domain video retrieval tasks, we collected data via questionnaires, interviews, and UI interaction logs. Our evaluation focused on the perceived intuitiveness of retrieval using natural language, retrieval accuracy, and how users interacted with the system's UI. We found that our advanced features yield higher task accuracy, user ratings, and more efficient queries. Overall, our results show the importance of designing intuitive and efficient user interfaces to be able to deploy large models such as CLIP effectively in real-world scenarios.

Keywords Video retrieval · Self-supervised learning · CLIP

✉ Tayfun Alpay
tayfun.alpay@hitec-hamburg.de

Sven Magg
sven.magg@hitec-hamburg.de

Philipp Broze
p.broze@nachtblau.tv

Daniel Speck
daniel.speck@hitec-hamburg.de

[1] HITeC - Hamburger Informatik Technologie-Center e.V., University of Hamburg, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

[2] Nachtblau GmbH, Straßenbahnring 18, 20251 Hamburg, Germany

1 Introduction

Very recently, training large language models (LLMs) such as GPT-3 (Brown et al., 2020) with self-supervised learning (SSL) has led to versatile open-domain models such as e.g. InstructGPT (Ouyang et al., 2022) or ChatGPT (OpenAI, 2022), leading to a large range of machine-learning-assisted real-world applications. Despite these advances in natural language understanding, current search engines and retrieval systems largely remain keyword-based.

Video retrieval is a problem that can particularly benefit from language interpretation since bridging modalities can often lead to ambiguous interpretations. A large number of research has recently been focusing on improving vision-language models for zero-shot transfer. Despite these advances, current approaches for multimodal retrieval largely still rely on training how to combine different expert models. While these approaches have shown great success on benchmark datasets, they have severe issues with open-domain scalability and maintenance due to the need to train multiple large models independently.

Recently, CLIP has been introduced as a pre-trained model trained on large amounts of vision-language data by learning to associate images with text (Radford et al., 2021). In this paper, we explore how CLIP can be used for video retrieval to allow more natural user input and relevance feedback than with keyword-based search. We introduce IMPA (Intelligent Media Production Assistant), a novel video retrieval system that uses CLIP to interactively evaluate multimodal queries that can be used with any dataset. In addition, we conducted, to our knowledge, the first user study for video retrieval with CLIP in a real-world scenario while using TV production professionals as participants. We give them known-item search tasks, which, based on our expert interviews, represent a typical usage scenario during TV production, e.g. when retrieving stock footage from a video archive or searching for short scenes in long unedited raw footage.

Our main contributions are as follows:

1. We introduce IMPA, a scalable open-domain multimodal video retrieval system powered by CLIP, with a focus on a powerful but easy-to-learn UI and features (see Fig. 1).
2. We designed a user study with a usability test based on known-item search tasks.
3. We evaluated IMPA by extensively collecting data during the test (i.e., task submissions, interaction logs, questionnaires, interviews). Our evaluation focused on A/B testing our features, user perception, and overall interface usage and task accuracy.

2 Related work

2.1 Self-supervised learning

The main goal of self-supervised learning (SSL), and zero-shot transfer in particular, is to leverage large amounts of unlabeled data for the purpose of training general-purpose models that can be used in various downstream tasks such as e.g. visual question answering, image captioning, and image/video retrieval (Yu et al., 2020).

This is commonly achieved by defining pre-training tasks (e.g. masked language modeling and sentence-image alignment) to encourage the emergence of task-agnostic

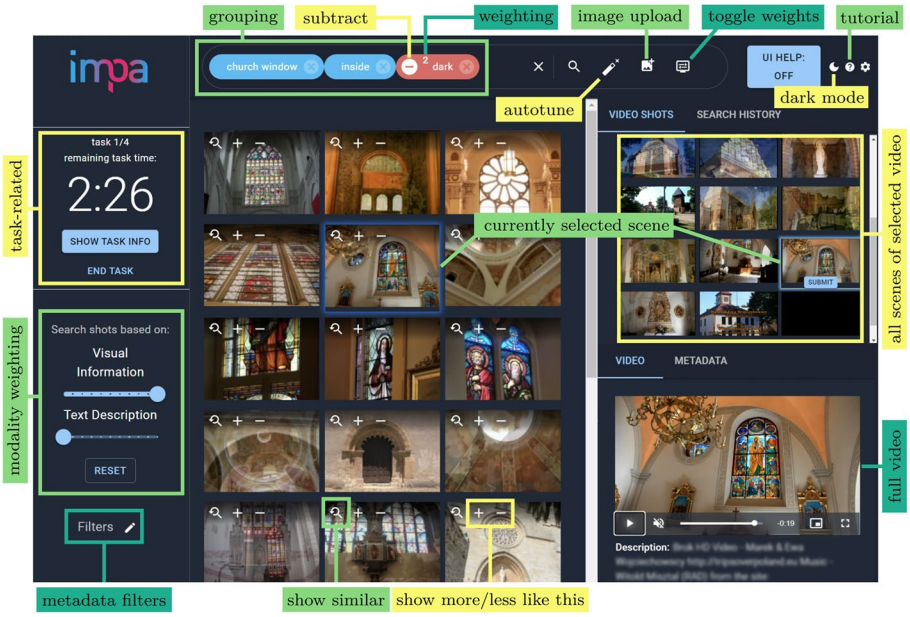

Fig. 1 The IMPA user interface (UI) showing search results (on the V3C1 dataset) for the example query $\mathbf{q} = +f(\text{“church window”}) + f(\text{“inside”}) - 2 \cdot f(\text{“dark”})$, i.e. non-dark video frames showing church windows from the inside. The UI is divided into four main components: the *Left Sidebar* (under the logo), the *Topbar* (the query interface to the right of the logo), the *Right Sidebar* with additional video context for the selected shot, and the *Gallery* which is used to display the retrieved results and offers a shot preview when hovering over each entry with the mouse. Note that the task-related information and “submit” button are only present for the user study.

representations. Recent advances in both vision and language models have given rise to a number of different ‘vision-and-language’ approaches for zero-shot transfer.

Most approaches use a single-stream model where language and vision representations are fed into the same Transformer layer (most notably, VisualBERT (Li et al., 2019a), VL-BERT (Su et al., 2019), Unicoder-vl (Li et al., 2020a), InterBERT (Lin et al., 2020)). Some approaches (ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019)) suggest a dual-stream architecture with separated modality encoding, while an analysis by Bugliarello et al. (2021) shows that neither approach comes out on top of the other.

Subsequent models explore variations on feature representations such as object labels (OSCAR (Li et al., 2020b)), pixel-level (Pixel-BERT (Huang et al., 2020)) and patch-level (VILT (Kim et al., 2021)) alignment, or by using scene graphs (ERNIE-ViL (Yu et al., 2021)).

CLIP (Radford et al., 2021) simplifies the pre-training process of previous SSL approaches significantly. The main idea is to pre-train a simple dual-stream model with a contrastive objective by learning to recognize which images belong to which captions (and which don’t). Since it has a joint embedding space, the model can not only be used to reason from images to text, but also from text to images. Trained on large amounts of data crawled from the internet, the model is (without fine-tuning) able to solve a large variety of downstream tasks such as OCR, image retrieval, action recognition, geo-localization, object classification, sentiment analysis, and more.

2.2 Zero-shot transfer for videos

Pre-training vision-language models for video-based downstream tasks resembles image-based approaches, except for their temporal processing of video frames. VideoBERT (Sun et al., 2019b) is similar to VisualBERT, in that it is a single-stream model that doesn't differentiate between its input's modality. Dual-stream models, on the other hand, use dedicated video encoders as backbones, e.g. ResNet 3D CNNs (HowTo100M Miech et al. (2019)), S3D with contrastive bidirectional Transformers (CBT (Sun et al., 2019a)), and Vision Transformers such as with ViViT (Arnab et al., 2021)), UniVL (Luo et al., 2020), or the Video Swin Transformer (Liu et al., 2022).

Besides integrating vision and language features, some approaches use additional features. Coming naturally for video data, the Video-Audio-Text Transformer (VATT (Akbari et al., 2021)) includes raw audio input. However, it is also possible to use more high-level features. As an example, Mithun et al. (2018) project audio and spatio-temporal motion features into a common "activity-text space". The collaborative experts model by Liu et al. (2019) and the Multimodal Transformer (MMT (Gabeur et al., 2020)) even use seven expert models for audio, actions, faces, OCR, scene, speech, and objects. However, transferable SSL approaches focusing on two modalities already require large amounts of data. Increasing the number of modalities further complicates the collection of large-scale data, and as a consequence, of training zero-shot models.

Therefore, some recent approaches only focus on few large-scale pretrained models. Luo et al. (2021) fine-tune CLIP, achieving state-of-the-art results on a number of video retrieval datasets. Portillo-Quintero et al. (2021) simplify the usage of CLIP for video retrieval even further by exploring both frame averaging and frame sampling as methods for temporal feature aggregation, finding rather small differences and an overall competitive performance to more complicated models.

2.3 Video retrieval systems

While machine-learning-assisted video retrieval systems can be fully automated without human intervention and feedback (Awad et al., 2020), these types of setups rarely work in open-domain systems, particularly in industrial applications, where final human supervision of the results is paramount. Consequently, video retrieval systems of the past decade have largely focused on improving interactive searches (Lokoč et al., 2021) by using intelligent systems as *assistants* and implementing *relevance feedback*. The idea behind relevance feedback is to allow users to tell the system whether the results of the initial query are relevant or not. Using this feedback, the system can iteratively improve the results by continually refining its hypotheses about the user intent. This can be particularly important in a multimodal setup to clear up misunderstandings and dissolve ambiguities.

Partly driven by recent SSL advances, visual-textual embedding spaces to match queries with frames have become increasingly popular in video retrieval systems (Awad et al., 2020; Rossetto et al., 2021). As an example, the W2VV++ model projects sentence representations into a video feature space for ad-hoc video search applications (Li et al., 2019b) and has been shown to improve existing systems such as SOMHunter (Veselỳ et al., 2021) with temporal and localized text queries.

Generally, multimodal systems have increasingly dominated the previous years' video retrieval challenges such as the Video Browser Showdown (VBS) and TRECVID.

Reflecting contemporary model capabilities, these systems do generally have separately trained models for different modalities (such as text, images, audio) and features (e.g. querying based on manual sketching, color-selection), giving the user some control in how these modalities are weighted to get a final result. For example, an analysis of the VBS 2019 has shown that VITRIVR (Heller et al., 2021) had a competitive advantage due to an integration of ASR and OCR systems, while sketch-based features, offered by systems such as VIRET (Lokoč et al., 2019), were rarely adopted by the users. The analysis also finds that a text search is often sufficient to solve known-item tasks, even though additional features and modalities can be important in rare situations. Most importantly, some user interface (UI) designs can have a very negative impact on novice users, leaving them confused. Designing an efficient and intuitive UI can be particularly challenging for novel features before any usability testing.

Different to the current state-of-the-art in video retrieval systems, we focus on using CLIP as a single multimodal model, implementing previously established core features but packaging the model in a simple interface. Concurrent to our work, other video retrieval systems powered by CLIP are being developed and under review for VBS 2023 (Dang-Nguyen et al., 2023). Based on currently available information,[1] our system most noticeably differs in its user interface design: we allow multimodal queries with vector algebra capabilities directly from a single text input field by using React Chip components. Other systems use multiple UI elements to accomplish this, not allowing the user to mix text with images in the search field.

3 The IMPA video retrieval system

3.1 Multimodal encoding

Our objective is to perform video retrieval on shot-level. Each shot is defined as a contiguous sequence of video frames, uninterrupted by a cut (video edit). Based on a database of videos, we perform boundary detection to segment the videos into shots. Each shot is represented by a single frame (as proposed by Portillo-Quintero et al. (2021)), specifically the center frame from each shot. While this effectively limits our system to non-temporal queries, we predict a low negative impact on the overall usability: using multiple frames has the side effect of overrepresenting longer shots with a potentially negligible impact on large video datasets (Karpathy et al., 2014). In addition, previous studies indicate that both novice and advanced users do not often use complex temporal expressions for known-item search (Rossetto et al., 2021).

We construct a database of visual embeddings $\mathbf{e}_i^{(\mathbf{V})}$ from shot frames v_i using CLIP's ViT-B/32 image encoder $f_V(v_i) = \mathbf{e}_i^{(\mathbf{V})}$ ($dim_v = 512$). Likewise, we use the Transformer text encoder $f_T(t_i) = \mathbf{e}_i^{(\mathbf{T})}$ to encode text t_i alongside each shot. Depending on the underlying data, this can e.g. be captions, transcripts, or video descriptions. We do not perform any fine-tuning with additional data as we are only interested in the usability and transfer of the originally published model. Based on this database of embeddings, we perform retrieval by ranking the embeddings based on closest distance to the query vector $\mathbf{q}$.

[1] https://videobrowsershowdown.org/teams/vbs-2023-systems/ (accessed 18 May 2023).

Furthermore, we exploit the fact that CLIP's text and vision encoder operate in the same representational space, allowing both image-to-text and text-to-image inference, by providing users with the ability to refine and extend previous queries by adding positive and negative examples from the search results.

3.2 User interface

The user interface is accessible as a web app in any modern web browser. The project uses a scaleable client–server architecture to separate features into two independent but interacting services, deployed in two separate containers for the frontend and backend, ensuring flexibility for managing resources and self-contained development updates. The frontend container serves the web app from an nginx server which receives all requests from the UI, thereby acting as a proxy server for the backend container and forwarding network requests. The backend container runs a python uWSGI application built on top of the Flask web framework, providing a REST API via a Web Server Gateway Interface. The user interface is built with JavaScript libraries, primarily React and Material UI (MUI) for interface design, and Redux for state management. For an increased usability, we implemented a tutorial and guided task mode for our usability test, including help text in interactive tooltips for user guidance, pop-ups, and system notifications.

The UI is divided into 4 main components with the following features (compare Fig. 1):

- Topbar: editable input "groups" supporting both text and image input (using Chip[2] components), arbitrary image URLs as input, image upload, manual and automatic weighting of input groups, dynamic query updating based on relevance feedback
- Gallery: video shot preview on mouse hover, features for relevance feedback (adding results as negative or positive examples to the previous query), result selection and navigation
- Right Sidebar: full video context of selected result, shot navigation within video, video player, metadata display, search history
- Left Sidebar: sliders for modality weighting, search filters based on metadata

The modality weighting sliders allow users to weight which data source shall have a higher impact on the results. While images represent video frames, the text modality can e.g. be used for transcript search, captions, or video descriptions. Since we use a separate index for each modality, querying on both modalities returns two separate lists of results (ranked by individual distances to the query $\mathbf{q}$). Consequently, we combine the distances $d_i^{\mathcal{V}}$ from the vision index with those from the text index $d_i^{\mathcal{T}}$, using a weighted average of user-defined weights $\alpha, \beta \in [0, 1]$:

$$d_i = \left(\frac{\alpha \cdot d_i^{\mathcal{V}} + \beta \cdot d_i^{\mathcal{T}}}{2} \right) \tag{1}$$

If a shot i is only part of *one* modality's results list $\mathcal{M} \in \{\mathcal{V}, \mathcal{T}\}$, it is approximated as the average distance of that modality's result list $\overline{d_i}^{\mathcal{M}}$. We perform this merging of modalities

[2] https://mui.com/material-ui/react-chip/ (accessed 18 May 2023).

in the state management of the user interface to avoid the need for additional backend requests.

Search filters can be set based on a list of known metadata attributes. Since metadata is stored per video, shot-level frame embeddings $\mathbf{e}_i^{(\mathbf{V})}$ returned from a query $\mathbf{q}$ have to be mapped to the corresponding metadata attributes. If there are no frame embeddings matching the currently selected filters, we iteratively increase the search space of the embeddings five times, while adjusting HNSW hyperparameters or until HTTP timeouts occur.

3.3 Multimodal queries and relevance feedback

3.3.1 Vector algebra

By normalizing the embedding vectors, queries can be arbitrarily combined with different modalities. As such, an embedding vector $\mathbf{e}_i$ can either result from using CLIP's image encoder $f_V(v_i) = \mathbf{e}_i^{(\mathbf{V})}$ on an image v_i or from using the text encoder $f_T(t_i) = \mathbf{e}_i^{(\mathbf{T})}$ on a text string t_i. Every query $\mathbf{q}$ is then constructed by aggregating up to $n \geq 1$ separate embedding vectors $\mathbf{e}_i$, which can be individually weighted by a scalar w_i and signed by $\sigma_i \in \{-1, 1\}$:

$$\mathbf{q} = \sum_i^n \sigma_i w_i \mathbf{e}_i \tag{2}$$

Consequently, we enable basic vector algebra in queries by allowing users to specify the sign σ_i of each embedding $\mathbf{e}_i$. The ability to add and subtract embeddings opens up possibilities for complex but precise queries. For example, the query $\mathbf{q} = +f(\text{“red car”}) - f(\text{“traffic”})$ evaluates into a single representation $\mathbf{q}$, after using the encoder $f(\cdot)$, to retrieve frames showing images of red cars **without** visible traffic. Most importantly, this introduces relevance feedback (see Fig. 2) which allows users to refine their queries by explaining which results are relevant and which are not.

It is important to note that we leave it open to the user how many embeddings they encode for their query. Not forcing all input into a single query vector enables the communication of crucial nuances and distinctions by *grouping* expressions. For an example, consider the semantic differences between the queries $\mathbf{q}_1 = f(\text{“white house”})$ and $\mathbf{q}_2 = f(\text{“white”}) + f(\text{“house”})$, where $\mathbf{q}_1$ is significantly more likely to retrieve images of the residency of the US president and $\mathbf{q}_2$ of houses with a white exterior. The ability to *weigh* each group allows additional nuance. While each group evaluates to a single embedding, it is visually represented by a single UI Chip.

While vector algebra offers a powerful tool for users with a mathematical background, the average non-technical user might be unfamiliar with it. Consequently, the concept of grouping and weighting expressions serves as an intuitive abstraction for manipulating embeddings, concealing the underlying complexity.

3.3.2 Heuristic fine-tuning of queries

One issue with allowing users to freely do vector algebra is that it can quickly lead to mathematically valid but completely unexpected results, especially when used for relevance feedback. As an example, consider the simple case in which a user executes the query $\mathbf{q}$, which gets evaluated to the embedding $\mathbf{e}_q$, leading to a list of resulting frame embeddings

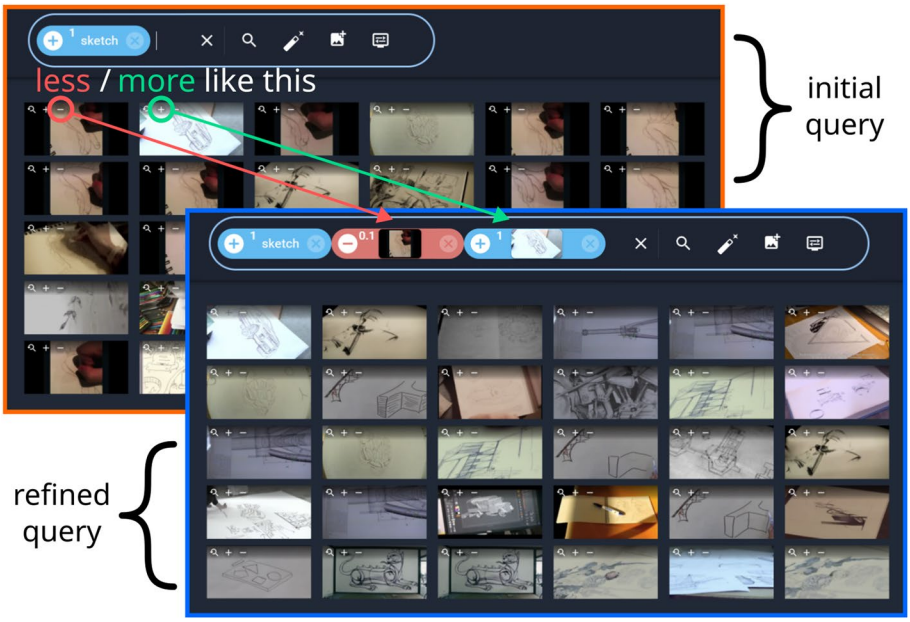

Fig. 2 Example for our implementation of relevance feedback in IMPA: The initial query for "sketches" (top) returns a variety of video frames (each representing a shot from the database). Clicking "-" on the first result removes this image's characteristics (i.e., a visible hand and a vertical aspect ratio), while clicking "+" on the second result adds more results showing sketches on white paper. Note how the refined query (bottom) is built by visually appending these two images to the input field, allowing text and image inputs side by side.

$\mathbf{e}_1^{(\mathbf{V})}, \mathbf{e}_2^{(\mathbf{V})}, \cdots$. Let's further assume that the user wants to make a minor adjustment to the retrieved results list by picking an image at the top of the list for subtraction to exclude a certain type of image from the results. Since the results are ranked by similarity to $\mathbf{e}_q$, results near the top have a very close distance to $\mathbf{e}_q$. With $\mathbf{e}_q \approx \mathbf{e}_i^{(\mathbf{V})}$ (and low enough i), the new query $\mathbf{q}'$ would, therefore, evaluate to $\mathbf{q}' = \mathbf{e}_q - \mathbf{e}_i^{(\mathbf{V})} \approx 0$. Consequently, the new results would effectively ignore $\mathbf{e}_q$ even though it is a positively weighted part of the current query. This query can be "fixed" by either including more positive examples or by re-weighting individual inputs to avoid query vectors close to 0.

To practically solve this issue for the most prevalent cases, we developed some general heuristics to automatically tune the weights inside user queries. The main reason for taking this approach is to keep IMPA, particularly its interface, accessible to non-technical users as much as possible.

In general, our heuristics for automatic fine-tuning of vector weights assume queries of the nature $\mathbf{q}' = \mathbf{q} + \sum_i^n \sigma_i w_i \mathbf{e}_i$ with $\mathbf{q} \approx \mathbf{e}_i$, $w_i = 1$, i.e. in practical terms, the resulting images of a query are used to construct a new query. Based on this, we count the number of positive vectors $\phi_+(\mathbf{q})$ and negative *image* vectors $\phi_-(\mathbf{q})$ (counting all *text* vectors as a single positive vector) and consider the following cases:

1. if $\phi_-(\mathbf{q}) \geq \phi_+(\mathbf{q})$ (i.e., $\sum_i \sigma_i \leq 0$):
   Set $w_i = \frac{\phi_+(\mathbf{q})-1}{\phi_-(\mathbf{q})}$ for each $\mathbf{e}_i$ with $\sigma_i = -1$,

2. if $\phi_+(\mathbf{q}) = 1, \phi\_(\mathbf{q}) = 1$:
   Set $w_i = c_1 := 0.1$ for each $\mathbf{e}_i$ with $\sigma_i = -1$,
3. if $\phi_+(\mathbf{q}) = 1, \phi\_(\mathbf{q}) > 1$:
   Give a hint that accuracy can be improved by providing at least $\phi\_(q) - 1$ positive examples,
4. if $\phi_+(\mathbf{q}) = 1, \phi\_(\mathbf{q}) > \frac{1-c_2}{c_1}$ with $c_2 := 0.8$:
   Give a hint that accuracy can be improved by providing at least $\frac{1-c_2}{c_1}$ examples,

where c_1 and c_2 are empirically determined constants that determine how much the new query can deviate from the original query when users provide too many negative examples. Note that the last two cases do not necessarily *always* lead to unexpected results. As a consequence, we simply advise the user to provide more (positive) examples as this is guaranteed to improve the result. While these heuristics are not exhaustive, we have found that the first heuristic in itself does already cover the most frequently occurring issues. Nevertheless, we make the autotuning of weights optional, allowing users to turn the feature off (see also Fig. 1).

3.4 Optimized indexing and retrieval

To be able to scale video retrieval performance with large amounts of data, we do not compare queries to every frame in the database. Instead, we perform approximate similarity search, balancing retrieval speed with recall. Building on the approach by clip-retrieval (Beaumont, 2022), we index all embedding vectors using FAISS (Johnson et al., 2019) with one index per modality. For the deployment of the usability test with the V3C1 dataset (see subsection 4.1)), we use Hierarchical Navigable Small Word (HNSW) graphs for indexing and performing approximate k-nearest neighbor similarity search. HNSW has been shown to offer high recall and efficient performance in highly clustered data (Malkov and Yashunin, 2020). We organize the quantized embeddings with 2^{14} clusters, retrieving 5000 results per user query of which the 30 highest ranked are displayed to the user on the first page. Additionally, we enable multimodal queries by mapping video and text embeddings to the same index. For efficient data retrieval during runtime, we use a compressed cache consisting of video metadata, scene boundaries, and embedding vectors with Apache Parquet, a columnar storage format for efficient data storage and retrieval (Vohra, 2016).

4 User study

To perform usability testing of IMPA within a real-world scenario, we designed a user study with TV production professionals. We tasked them with finding given video scenes in a large database within a limited amount of time. In doing so, we A/B tested the user interface functionality with a within-subjects design and evaluated how participants interact with the UI elements and rate the system's different aspects.

4.1 Dataset

We used the V3C1 dataset (Rossetto et al., 2019) to provide users with a database of searchable videos. The dataset is composed of around 1 Mio. shots from 7475 videos with

an average length of 8 min, totaling 1.3 TB in size and 1000 h of content. The dataset has been used successfully since 2019 in both the VBS (Rossetto et al., 2021) and the TRECVID challenges. We utilized the provided shot boundaries to feed the center frame of each shot to CLIP in order to generate visual embeddings to represent the scenes. To further improve the quality of the embedding database, we used OpenCV (Bradski, 2000) to detect and remove blurry frames and mostly monochrome images. This step is of particular importance as we have found that CLIP can sometimes favor single-colored images over more characteristic images (e.g. for the query "The Simpsons", ranking a fully yellow image higher than one with the characters).

The V3C1 metadata provides video descriptions which we used to generate text embeddings of each video. Together with the center frame embeddings, this served as the basis for multimodal query functionality. We used the remaining metadata (such as e.g. categorization tags, duration, aspect ratio, title) to display context information in the UI below the video player and to allow the users to filter videos based on these variables.

4.2 Participants

To test the overall system stability and to fine-tune the difficulty of the tasks, we have conducted a prestudy using 6 participants. After some necessary adjustments, the main study was done with the help of 17 participants of which 4 had technical difficulties or did not complete the experiment. The demographics of the remaining participants were ages 21–56 ($M = 36.30$, $SD = 11.35$), 61.5% male, 92% fluent in English, and 76% with a higher education background. The participants were chosen based on invitations from 7 different German TV production companies related to news. 76% comprised professional video editors, the remaining participants were a mix of journalists, software engineers, and video archivists. Asked to rate their professional skills on a scale from 1 to 5, our participants were highly familiar with video editing ($M = 4.54$, $SD = 0.78$), researching information/media using Google ($M = 4.31$, $SD = 0.63$) and YouTube ($M = 3.92$, $SD = 0.95$), video archive search in general ($M = 3.62$, $SD = 1.04$), and somewhat familiar with video categorization ($M = 3.38$, $SD = 1.19$), media archiving ($M = 3.08$, $SD = 1.26$), and professional writing ($M = 2.77$, $SD = 1.24$).

4.3 Study design

We picked 11 video scenes from the V3C1 dataset as retrieval tasks, where the first three were used as simpler practice tasks so that users can familiarize themselves with the system. Further splitting the 8 remaining videos into two groups of videos, we also evaluated two different versions of the UI, one with only basic functionality and one including advanced features (see subsection 4.5). This resulted in a 2x2 factorial within-subjects design with the two video groups and UI variants as independent variables. Consequently, we randomized the order in which participants see the video groups and UI variants, leading to one of four possible outcomes for each participant.

4.4 Video groups

Due to the within-subjects design, every participant saw both systems in a randomized order. Since showing the same videos in both trials would have trivialized parts of the

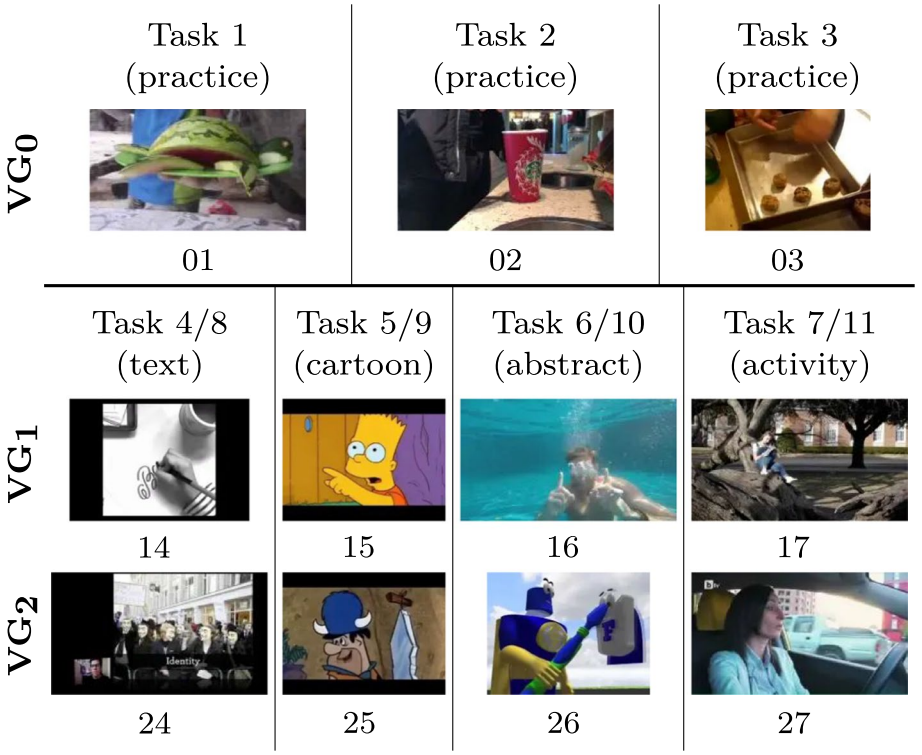

Fig. 3 Overview of the 11 scenes that users were asked to find. The test started with 3 practice tasks, continuing with either VG_1 or VG_2, in which videos had a common theme for each task (on-screen caption, cartoon figures, abstract scene description, activity). The videos are referred to as "video *ij*" based on their group ID *i* and task ID *j*.

challenge in the second trial, we picked a different set of videos for both trials, making an effort to keep the videos similar to each other (compare Fig. 3). Consequently, the scenes from the main video groups VG_1 and VG_2 have been chosen such that the *i*-th task from VG_1 matches the *i*-th task from VG_2 in terms of task difficulty and displayed content. However, the presentation order of the two main video groups was randomized in spite of this to allow us to evaluate whether the video groups were indeed similar enough to not bias the results.

4.5 A/B testing the UI functionality

Since most users were familiar with keyword-based search engines, we hypothesized that CLIP presented a strong deviation in how users could formulate queries and how these were interpreted. Therefore, we aimed to evaluate how well CLIP can be used "as is" for the purpose of video retrieval and how important our additional features are for both user perception and task accuracy. For this purpose, we split the feature set into two different versions of IMPA, a basic (B) version and an advanced (A) version (Table 1). Based on initial testing and feedback, we mainly restricted the basic version to the ability to divide the query into groups and to find the most similar frames for a particular result.

Table 1 Overview of features for the basic vs. the advanced version of IMPA

Features \Version	Basic (B)	Advanced (A)
Grouping	✓	✓
Search history	✓	✓
Video preview	✓	✓
Metadata	✓	✓
Show similar	✓	✓
Show more/less like		✓
Add/subtract groups		✓
Group weighting		✓
Multimodal query		✓
Image query (upload)		✓
Modality weighting		✓
Search filters		✓

Based on the 2x2 factorial within-subjects design (see subsection 4.3), users went through two trials (with 4 tasks each) to see both systems. Randomizing the order in which the two systems are shown in the first vs. the second trial, allowed us to evaluate the results from two different angles: users going from B → A experienced additional features after learning the basics first, while users going from A → B actively experienced the removal of (potentially helpful) advanced features.

4.6 Study procedures

We evaluated the user study based on questionnaires, interviews, and server logs of user interactions. Centered around a trial for each condition, each participant went through the following steps to complete the full user study:

1. Pre-Questionnaire,
2. **Trial 1**: tutorial, 3 practice tasks (VG_0).
   4 tasks with either VG_1 or VG_2 (randomized), either basic or advanced features (randomized),
3. Questionnaire,
4. **Trial 2**: tutorial, 4 tasks with remaining video group and features,
5. Questionnaire,
6. Interview.

After giving informed consent, each participant started the study with a pre-questionnaire of seven questions collecting demographic information and assessing skill level and prior knowledge. To ease users into the usage of the system's features, and to educate users on how CLIP's query interpretation differs from traditional keyword-based search engines, each trial started with a written tutorial as an introduction, for which users needed between 3–14 min ($M = 7.6$, $SD = 3.4$). The tutorial served as an introduction to IMPA's features and illustrated some key differences between conventional keyword-based search and using CLIP.

After the first trial, in which users were either shown the basic or advanced system (and one of the two video groups), users were given the main questionnaire. It consisted of the

Post-Study System Usability Questionnaire (PSSUQ) by Lewis (1995), Net Promoter Score (NPS ®) by Reichheld (2003), Perceived Difficulty Assessment Questionnaire (PDAQ) by Ribeiro and Yarnal (2010), and a custom set of questions inquiring about, e.g., the perceived learning curve, usability, transparency, and credibility of the results. The trial was repeated with the remaining system and video group, concluding with the same questionnaire. On average, users needed $M = 62$ minutes ($SD = 28$) for completion.

Following this practical part of the user study, an experimenter checked the server logs for anomalous behavior for a chance to clarify during the post-study interview. The interview was conducted shortly after via videoconferencing and was scheduled to take 15 min. Besides asking for feedback on features and suggestions, we asked every user on their overall impression, their expectations, whether they "trusted" the results, and whether they would use IMPA at work.

Note that the entire study was designed for remote participation for which we took additional steps to ensure the validity of the results and a smooth and clear user flow. Prior to the experiment, we asked users to set aside a certain amount of time and to participate with a workplace setup that they are personally familiar and comfortable with. In addition, the user study was fully "automated", requiring no intervention: participants received an invitation e-mail with a link and ID token to a survey set up with LimeSurvey (2003) and had no further interaction with experimenters until the final interview. We randomized the conditions within LimeSurvey, dynamically linking to the basic/advanced systems and tasks, opening the tabs in a second browser tab, and asking the user to continue with the survey on the first tab after completing each trial.

To assess *how* users interact with our system, we logged their interactions with the UI. Using these server logs, we were able to evaluate the sequence of events that led to a submission, the popularity and helpfulness of our features, and how much time users spent for certain activities. Naturally, we logged all submitted videos and the time needed for each task (which was limited to 3 min).

5 Results

5.1 Validating the experimental setup

We started our evaluation by validating our experimental setup design, such as our choices for the retrieval videos, the experiment length, and potential social biases reflected by CLIP.

Using the Perceived Difficulty Assessment Questionnaire (PDAQ), we confirmed that on a 7-Likert scale, the perceived difficulty of our retrieval tasks fell, on average, slightly above the neutral midpoint of the scale ($M = 4.27$, $SD = 1.61$), indicating a good balance in difficulty throughout the experiment. Interestingly, users rated the tasks *easier* after using the advanced system ($M = 4.15$, $SD = 1.86$) compared to the basic system ($M = 4.38$, $SD = 1.39$). We also found video group VG_2 to be perceived as slightly more difficult ($M = 4.62$, $SD = 1.76$) than VG_1 ($M = 3.92$, $SD = 1.44$), although looking at the actual task submissions, difficulty and completion times between the two groups even out *on average*.

This can also be seen in Fig. 4, which shows that some tasks took more time than others, somewhat balancing out within the two main video groups. Similarly, while we have found that there were large differences in how much time the participants needed for the entire

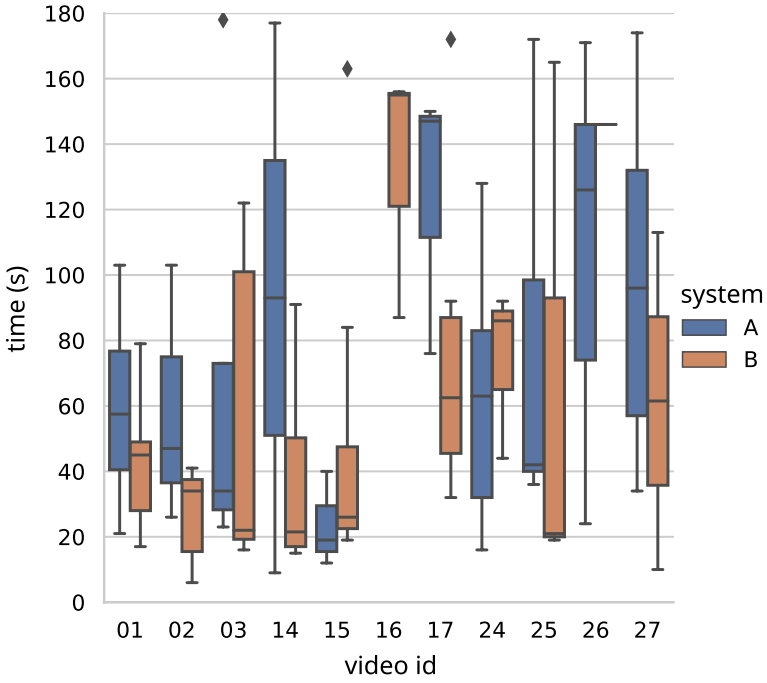

Fig. 4 Time needed per task, advanced (A) vs. basic (B) system. Users had a maximum of 180 s for each task. Users always started with the introductory videos 01, 02, 03 (VG_0) before doing the videos $14-17$ (VG_1) or $24-27$ (VG_2) next.

experiment, the average *perceived* length for the trials was the same ($M = 4.77$), regardless of which system variant was used. Most importantly, we have not found the video groups to correlate to any of the user's system preferences and ratings (subsection 5.2), which validates our experimental setup and thus the foundation of our main conclusions.

As CLIP has been trained on large amounts of open-domain data, it can lead to potentially problematic and off-putting query interpretations, e.g. due to inherited gender and age biases (Agarwal et al., 2021). To monitor this factor, we asked participants in the post-study questionnaire whether they felt like IMPA had a human personality-driven, potentially biased, interpretation of their queries. 70% felt no traces of "personality" or bias, with half of the other 30% attributing this to a misunderstanding caused by their own phrasing. Consequently, social biases did not affect user perception noticeably in our user study.

To ensure a good understanding of IMPA's features, we deliberately provided a tutorial beforehand and asked them to rate its usefulness after the trial. Almost all users gave the tutorial a high rating (compare Fig. 6), many mentioning its usefulness in the interviews. Exploring whether the tutorial had a measurable effect, we find no link between the time participants took to read the tutorial and their enjoyment, how accurately they solved the tasks, how many advanced features they used, or any other relevant metric.

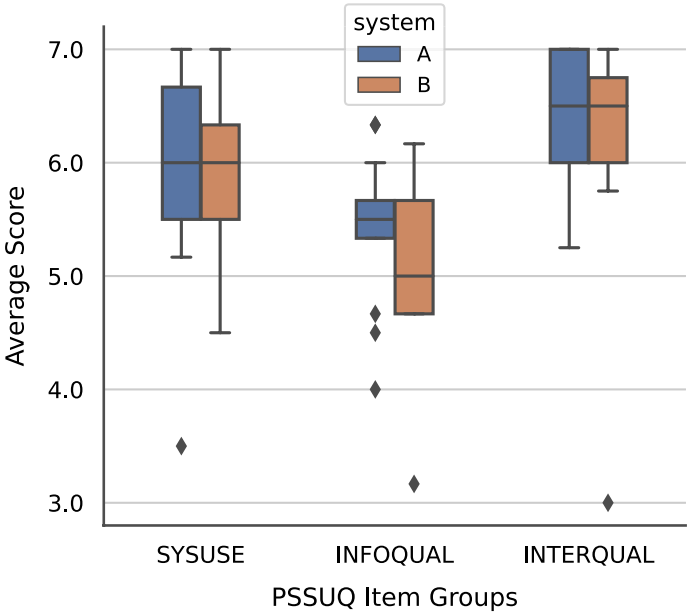

Fig. 5 Average scores for the PSSUQ item groups system usefulness (SYSUSE), information quality (INFOQUAL), and interface quality (INTERQUAL), comparing the advanced (A) to the basic (B) system.

5.2 User perception

We used the 16-item 7-Likert Post-Study System Usability Questionnaire (PSSUQ) to compare the advanced (A) to the basic (B) system w.r.t. its 3 item groups system usefulness (SYSUSE), information quality (INFOQUAL), and interface quality (INTERQUAL). Evaluating the results, we found that both systems had a higher than average rating on all item groups (compare Fig. 5). However, system A was rated higher more consistently, best visible for INFOQUAL ($p = 0.3$). It got particularly more agreement on the statement "The system gave error messages that clearly told me how to fix problems" ($\Delta = 1.23$) and "This system has all the functions and capabilities I expect it to have" ($\Delta = 0.46$). As both systems had the same system messages and user on-screen feedback, we interpret this difference to be primarily caused by the presence of relevance feedback in system A. Overall, the individual answers show that, while system B seems easier to learn, system A is seen as more efficient, providing better feedback, and with more expected functionality.

To calculate the Net Promoter Score (NPS), we asked participants the likelihood that they would recommend IMPA to a friend or colleague. We find that both systems achieved a positive rating, with system A scoring higher (57 vs. 36). During the post-study interviews, we asked participants a similar question, i.e. whether they would use IMPA at work, which 85% affirmed. The same percentage was also self-reported to "trust" in the search results. Further comparing both systems (see Fig. 6), we found that more users prefer system A when asked directly about user satisfaction and system performance. Overall, most of our questions were answered with above-average scale values, multilinguality being an exception due to CLIP's limited multilingual capabilities (we informed our German participants of the possibility to mix languages in the queries).

Fig. 6 Average scores for a set of questions on a 10-point (**a**) and 5-point (**b**) Likert scale, comparing the advanced (A) to the basic (B) system.

We asked the users in the post-study interview which things they liked about IMPA the most and the least. The vector algebra functionality and the GUI design were mentioned the most as positive, whereas two users gave the feedback that it was sometimes difficult to understand why they got the wrong results. A transcript search was suggested the most as additional features for future work. In general, most participants, mentioned that they felt like the system might have a steep learning curve for *other* people, as it works quite different from established keyword-based search systems. Interestingly, we measured the opposite in our evaluation: users found working with system A easier and were measurably more accurate with it. Additionally, we find that participants spend less time with the search bar and gallery as time passes throughout the experiment, indicating a "learning effect" that progresses with each completed task. As such, the perception of a steep learning curve might be caused by the false consensus effect, i.e. participants misjudging the capabilities of their peers. Nevertheless, further research might be necessary to rule out other potential causes such as the novelty effect, the subject-expectancy effect, or other negative manifestations of system complexity that are conflated with learning difficulty.

5.3 User submissions and task accuracy

Throughout the experiment, only one person abandoned a task (twice) and only one person submitted an incorrect video (once). In all other cases, tasks were either submitted correctly or users ran out of time (3 min per task). Overall, participants had an accuracy of 81% with system A and 76% with system B, indicating the success of some of the advanced features. This is also supported by our analysis of how far users had to scroll for their submitted videos. On average, users scrolled less for system A ($M = 7.9$ results) than B ($M = 9.0$ results). As in similar studies (Rossetto et al., 2021), we have found very few submissions beyond the first page of 30 results as users don't seem to scroll beyond the first initial results - despite spending a majority of their time browsing the results. Indeed, the fastest task submissions were for videos that ranked in the initial positions of the results list.

Analyzing potential relationships between a user's individual task performance and how often they used certain features, we found that, while the grouping feature was essential

to find some of the videos, using too many can generally lead to somewhat unforeseeable results. This is particularly true when most of the groups consist of text embeddings. As such, most correct submissions are generally made up of fewer groups, i.e. less than 3 (and ranked in the top 5 search results). At the same time, the more groups were used in correct submissions, the more images were used, whereas the text inputs contained shorter phrases.

To understand how a participant's success impacts their user ratings in the questionnaire, we explored potential relationships between task accuracy and items from the questionnaire. As expected, we found a polynomial trend ($R^2 = 0.158$) between task accuracy and user satisfaction. Somewhat surprisingly, we found a linear correlation between how much people had to scroll in the gallery for a correct solution and their self-reported trust in the system ($R^2 = 0.69$). Possible interpretations for this are that i) people don't trust a system (or the experiment) if "it feels to easy", ii) scrolling more correlates with higher task difficulty, i.e. people trust the system more once they solved a difficult task (we find this to be true for some of the tasks), or iii) the need to scroll more correlates with more complex interactions, i.e. people trust the system more once they have used all the features successfully. The latter explanation is somewhat supported by the fact that using one of the more complex features, the weighting feature, seems to also increase trust ($R^2 = 0.49$). Aggregated over all users, we also report that using more groups (indicative of more complex interactions) correlates strongly with how easy users found to learn the system ($R^2 = 0.93$).

To summarize, using more complex features and interactions seems to have built more trust and familiarity with the system. Conversely, users that did not explore most of the functionality, seem to only have developed limited trust and understanding throughout the experiment.

5.4 Feature usage

Logging all UI interactions with the React *components* and *actions* of the 4 main UI elements, we were able to evaluate how participants interacted with the features and how this correlated to their individual success or failure in the retrieval task.

As expected, the majority of the interactions and time spent was with the search bar (entering or modifying a query) and the gallery (clicking on a shot). In particular, for most users, more than half of their interactions were with the Gallery itself, generally spending around 3–6 s browsing before clicking on a shot or action in the Gallery. Weights and "show less like this" were the most used advanced features, the search filter the least used feature. In general, most participants always started with a simple text search and scrolled through the gallery, before using additional features as a means to refine the previous search (explaining the popularity of some of the features).

Moreover, while we have found large differences between how much time users needed for the entire experiment, most of them spent roughly the same amount of time between clicks for each of the 4 main elements of the UI, positively indicating a similar "flow" despite individual differences in attention and speed. We also found that 30% of users used the mouse to start a query (rather than using the ENTER key), that the image upload feature was used only once (likely due to the time limit), and that over 98% of searches never navigated to the second page, thereby only considering the top 30 search results on the first page.

Figure 7 illustrates heatmaps for all user actions over time, normalized for the individual task times, with the start and end of the timeline marking the beginning/end of a task. We

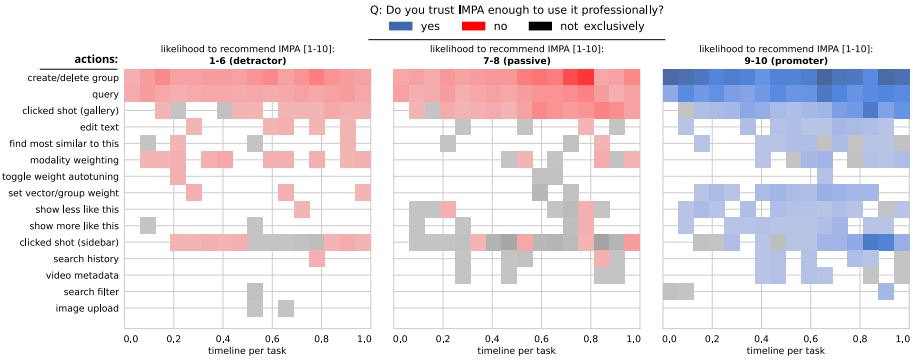

Fig. 7 Heatmap of all user actions over time, task-normalized (0.0: start of task, 1.0: end of task). Separated by columns based on the Net Promoter Score subdivisions grouping the likelihood to recommend IMPA to colleagues (1: least likely, 10: most likely). Heatmap color indicates the self-reported trust in IMPA during the interview (red and green histograms superimposed on black histogram).

grouped the timelines by the NPS subdivisions, which, asked for the likelihood to recommend IMPA from 1 (lowest) to 10 (highest), consisted of "detractors" (rating of 6 or lower), "passives" (7–8), and "promoters" (9–10). As can be seen, users with a high rating, had used significantly more of the advanced features throughout each task, in particular the "show more/less like this" actions, the feature that was voted as most helpful. Moreover, the self-reported trust in IMPA correlated with NPS and feature usage, further reinforcing that a successful trying out of all features led to the best user experience.

6 Discussion

Our evaluation shows that CLIP can be used quite successfully for video retrieval. However, deploying the model "as is" is likely to lead to confusion with users who are more used to enter keywords instead of natural language into a search engine. As such, we identified several key components to a successful user adoption:

First, a guided user introduction is critical for an user to phrase a query that maps to their intent. Our tutorial explained all features with examples, illustrated peculiarities of CLIP, and gave advice on best practices. For the latter, we have found that using descriptive sentences of medium length was often preferable to non-grammatical keyword searches. Similarly, using multiple input groups (embeddings) should only really be done for disambiguation (["white house"] vs ["white" "house"]) as algebraic expressions with a long list of vectors can lead to increasingly arbitrary results. We have found that our heuristics for automatic weight tuning cover most of these issues. In our usability test, relevance feedback, i.e. picking individual results as positive or negative examples to refine the initial query, was the most used and verbally commended feature, leading to collaborative interactions in which query and results were iteratively improved.

Second, features that complement CLIP's capabilities are vital for a successful user adoption. Without the ability to give direct feedback based on the search results, users might quickly become frustrated due to a lack of control. In our usability test, participants rated the advanced system higher (NPS, system quality, information quality, search results, effectiveness), spending more time with it, finding the correct solutions in higher positions,

and scoring a higher task accuracy (81% vs. 75%). Most users tested and utilized the advanced features, leading to an increased self-reported trust in the system's search results.

Another important aspect of the provided features is to package the system's features in a UI that is simple enough to feel intuitive but complex enough to allow more advanced interactions where necessary - especially when the user's intent is misinterpreted. For example, IMPA offers an intuitive abstraction for vector algebra with embeddings that was easily understood and well adopted by our users.

Aside from this, it is equally important to understand the data domain on which CLIP is used, since the model has been shown to exhibit social biases, a side effect of training on large amounts of data scraped from social media websites. While the model poses a general risk of interpreting queries in a backhanded, suggestive, and even sarcastic manner, the likelihood for this to happen (and to be noticed by the user) in a *retrieval* setup is e.g. significantly lower on a dataset of dashcam footage compared to videos crawled from random Twitter profiles. As such, it might be necessary to fine-tune CLIP on some datasets, depending on the associated risk. But while additional training or search filters might have to be used for deployment, this can not replace extensive testing of CLIP on the target domain, as human expert knowledge and large-scale testing currently remains the best practice to mitigate risks associated with large models.

The final ingredient for real-world use is effective and scalable resource management to ensure fast retrieval times, even for multimodal queries. While we have found vector search with FAISS to be fast, reliable and scalable, it requires custom index management if image and text embeddings are kept separate in order to provide modality weighting for the user. Furthermore, FAISS provides no useful option to exclude results at search time based on certain criteria. This makes it difficult to utilize metadata and to filter search results effectively. In many cases, we observed the need to iteratively increase the search space, leading to a degradation of the search speed and, in extreme cases, a noticeably worse user experience. As a future alternative, Elasticsearch[3] provides a REST API to avoid these disadvantages, removing the need for custom metadata filtering and metadata management, as text embeddings, image embeddings and metadata can be stored in and searched from the same index.

7 Conclusion

In this paper, we have introduced IMPA: an open-domain video retrieval system that processes multimodal queries with CLIP. We focus on moment retrieval by embedding a single frame per shot, saving processing time by avoiding explicit temporal modeling. In addition, we unify multimodal inputs in a single UI component to significantly simplify user input despite giving full access to the model's capabilities and even allowing multimodal vector algebra with embeddings.

We evaluated our user interface and CLIP's real-world usability by inviting professionals from the German TV production industry to a specially designed usability test. Our results show that our advanced features based around relevance feedback improve both user accuracy and user acceptance, thereby highlighting the importance of certain features and an intuitive UI to successfully deploy CLIP in a retrieval setup.

[3] https://github.com/elastic/elasticsearch (accessed 18 May 2023).

In the future, we would like to enhance multilingual querying by exploring M-CLIP (Carlsson et al., 2022), improve index management with ElasticSearch, and implement additional features such as temporal querying, transcript search, and audio-based querying. Further research on bias detection (Gezici et al., 2021) and open model variants like OpenCLIP (Ilharco et al., 2021) can also help in introducing better control over data biases, robustness, and domain-specific fine-tuning.

Acknowledgements We thank Jonas Kuntzer and Madalina Stanescu-Bellu for their help in implementing IMPA.

Author contributions T.A. took the lead in writing the manuscript, planned and conducted the user study, and was in charge for the overall planning and direction during development. S.M. supervised the project, conceived the initial idea, and helped in writing the manuscript. T.A. evaluated the user study, while D.S. helped significantly in the evaluation of the UI logs. P.B. carried out the majority of the implementation based on UI mockups and feature specifications by T.A. All authors reviewed the manuscript.

Funding This project has been funded by the Investment and Development Bank of the City of Hamburg and by the European Regional Development Fund (ERDF, CCI 2014DE16RFOP006).

Data availibility Data sharing not applicable to this article.

Declarations

Conflict of interest P.B. is employed by nachtblau GmbH, who intend to use the results for further development of their commercial media asset management system "medialoopster".

Ethical approval The study involving human participants was reviewed and approved by the internal HITeC e.V. ethical review board.

Consent to participate All participants of the presented study provided informed consent, including the participation and collection of pseudonomized data for the purposes of statistical evaluation and research.

Consent to publication All participants of the presented study provided informed consent, including the publication of non-identifying results using anonymized and/or aggregated data from the study.

References

Agarwal, S., Krueger, G., Clark, J. et al. (2021). Evaluating clip: towards characterization of broader capabilities and downstream implications. arXiv preprint arXiv:2108.02818

Akbari, H., Yuan, L., Qian, R., et al. (2021). Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems, 34*, 24206–24221.

Arnab, A., Dehghani, M., Heigold, G., et al. (2021). Vivit: A video vision transformer. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 6836–6846.

Awad, G., Butt, A.A., Curtis, K. et al. (2020). Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. arXiv preprint arXiv:2009.09984

Beaumont, R. (2022). Clip retrieval: Easily compute clip embeddings and build a clip retrieval system with them. https://github.com/rom1504/clip-retrieval.

Bradski, G. (2000). The OpenCV Library. *Dr Dobb's Journal of Software Tools*.

Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems, 33*, 1877–1901.

Bugliarello, E., Cotterell, R., Okazaki, N., et al. (2021). Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language berts. *Transactions of the Association for Computational Linguistics, 9*, 978–994.

Carlsson, F., Eisen, P., Rekathati, F. et al. (2022). Cross-lingual and multilingual clip. In: Proceedings of the Language Resources and Evaluation Conference. European Language Resources Association, pp 6848–6854, https://aclanthology.org/2022.lrec-1.739

Dang-Nguyen, D.T., Gurrin, C., Larson, M., et al. (eds) (2023). MultiMedia Modeling: 29th International Conference, MMM 2023, Bergen, Norway, January 9–12, 2023, Proceedings, Part I, vol 13833. Springer Nature, to appear.

Gabeur, V., Sun, C., Alahari, K. et al. (2020). Multi-modal transformer for video retrieval. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, Springer, pp 214–229.

Gezici, G., Lipani, A., Saygin, Y., et al. (2021). Evaluation metrics for measuring bias in search engine results. *Information Retrieval Journal, 24*, 85–113.

Heller, S., Gasser, R., Illi, C., et al. (2021). Towards explainable interactive multi-modal video retrieval with vitrivr. In I. I. Part (Ed.), *MultiMedia Modeling: 27th International Conference, MMM 2021, Prague, Czech Republic, June 22–24, 2021, Proceedings* (pp. 435–440). Springer.

Huang, Z., Zeng, Z., Liu, B., et al. (2020). *Pixel-bert: Aligning image pixels with text by deep multi-modal transformers*. arXiv preprint arXiv:2004.00849.

Ilharco, G., Wortsman, M., Wightman, R., et al. (2021). *Openclip*.https://doi.org/10.5281/zenodo.5143773

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data, 7*(3), 535–547.

Karpathy, A., Toderici, G., Shetty, S., et al. (2014). Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1725–1732.

Kim, W., Son, B., & Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision. In: International Conference on Machine Learning, PMLR, pp 5583–5594.

Lewis, J. R. (1995). Ibm computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction, 7*(1), 57–78. https://doi.org/10.1080/10447319509526110

Li, G., Duan, N., Fang, Y., et al. (2020a). Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 11,336–11,344.

Li, X., Xu, C., Yang, G., et al. (2019b). W2vv++ fully deep learning for ad-hoc video search. In: Proceedings of the 27th ACM international Conference on Multimedia, pp 1786–1794.

Li, L.H., Yatskar, M., Yin, D., et al. (2019a). Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557.

Li, X., Yin, X., Li, C., et al. (2020b). Oscar: Object-semantics aligned pre-training for vision-language tasks. In: European Conference on Computer Vision, Springer, pp 121–137.

LimeSurvey, (2003). LimeSurvey: An Open Source survey tool. LimeSurvey GmbH, Hamburg, Germany, http://www.limesurvey.org

Lin, J., Yang, A., Zhang, Y., et al. (2020). Interbert: Vision-and-language Interaction for Multi-modal Pretraining. arXiv preprint arXiv:2003.13198.

Liu, Y., Albanie, S., Nagrani, A., et al. (2019). Use what you have: Video retrieval using representations from collaborative experts. arXiv preprint arXiv:1907.13487.

Liu, Z., Ning, J., Cao, Y., et al. (2022). Video swin transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3202–3211.

Lokoč, J., Kovalčík, G., Souček, T., et al. (2019). Viret: A video retrieval tool for interactive known-item search. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval, pp 177–181.

Lokoč, J., Veselỳ, P., Mejzlík, F., et al. (2021). 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):1–26.

Lu, J., Batra, D., Parikh, D., et al. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems, 32*, 456985.
Luo, H., Ji, L., Shi, B., et al. (2020). Univl: A unified video and language pre-training model for multimodal understanding and generation. arXiv preprint arXiv:2002.06353.
Luo, H., Ji, L., Zhong, M., et al. (2021) Clip4clip: An empirical study of clip for end to end video clip retrieval. arXiv preprint arXiv:2104.08860.
Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(4), 824–836. https://doi.org/10.1109/TPAMI.2018.2889473
Miech, A., Zhukov, D., Alayrac, J.B., et al. (2019). Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 2630–2640.
Mithun, N.C., Li, J., Metze, F., et al. (2018). Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, pp 19–27.
OpenAI (2022). ChatGPT. https://openai.com/blog/chatgpt, [Online; accessed 15-Ma-2023].
Ouyang, L., Wu, J., Jiang, X., et al. (2022). Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.
Portillo-Quintero, J.A., Ortiz-Bayliss, J.C., & Terashima-Marín, H. (2021). A straightforward framework for video retrieval using clip. In: Mexican Conference on Pattern Recognition, Springer, pp 3–12.
Radford, A., Kim, J.W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, PMLR, pp 8748–8763.
Reichheld, F. F. (2003). The one number you need to grow. *Harvard Business Review, 81*(12), 46–55.
Ribeiro, N. F., & Yarnal, C. M. (2010). The perceived difficulty assessment questionnaire (pdaq): Methodology and applications for leisure educators and practitioners. *Schole A Journal of Leisure Studies and Recreation Education, 25*(1), 111–115.
Rossetto, L., Schuldt, H., Awad, G., et al. (2019). V3c–a research video collection. In: International Conference on Multimedia Modeling, Springer, pp 349–360.
Rossetto, L., Gasser, R., Lokoč, J., et al. (2021). Interactive Video Retrieval in the Age of Deep Learning - Detailed Evaluation of VBS 2019. *IEEE Transactions on Multimedia, 23*, 243–256. https://doi.org/10.1109/TMM.2020.2980944
Su, W., Zhu, X., Cao, Y., et al. (2019). Vl-bert: Pre-training of generic visual-linguistic representations. arXiv preprint arXiv:1908.08530.
Sun, C., Baradel, F., Murphy, K., et al. (2019a). Learning video representations using contrastive bidirectional transformer. arXiv preprint arXiv:1906.05743.
Sun, C., Myers, A., Vondrick, C., et al. (2019b). Videobert: A joint model for video and language representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 7464–7473.
Tan, H., & Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490.
Veselỳ, P., Mejzlík, F., & Lokoč, J. (2021). Somhunter v2 at video browser showdown 2021. In: MultiMedia Modeling: 27th International Conference, MMM 2021, Prague, Czech Republic, June 22–24, 2021, Proceedings, Part II 27, Springer, pp 461–466.
Vohra, D. (2016). Apache Parquet, Apress, Berkeley, CA, pp 325–335. https://doi.org/10.1007/978-1-4842-2199-0_8
Yu, L., Chen, Y.C., & Li, L. (2020). Self-supervised learning for vision-and-language. https://rohit497.github.io/Recent-Advances-in-Vision-and-Language-Research/slides/tutorial-part5-pretraining.pdf, accessed: 2023-5-15.
Yu, F., Tang, J., Yin, W., et al. (2021). Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 3208–3216.