RESEARCH-ARTICLE

Enhancing Video Retrieval with Robust CLIP-Based Multimodal System

MINH-DUNG LE-QUYNH, Lazada Group, Singapore City, Singapore

ANH-TUAN NGUYEN

ANH-TUAN QUANG-HOANG, Ford Motor Company, Dearborn, MI, United States

VAN-HUY DINH, Ho Chi Minh City University of Technology - HUTECH, Ho Chi Minh City, Vietnam

TIEN-HUY NGUYEN

HOANG-BACH NGO

View all

Open Access Support provided by:

Ford Motor Company

Lazada Group

FPT Corporation

Ho Chi Minh City University of Technology - HUTECH

Published: 07 December 2023

SOICT 2023: The 12th International Symposium on Information and Communication Technology
December 7 - 8, 2023
Ho Chi Minh, Vietnam

https://doi.org/10.1145/3628797.3629011
ISBN: 9798400708916

Enhancing Video Retrieval with Robust CLIP-Based Multimodal System

Minh-Dung Le-Quynh*
Lazada Vietnam
Ho Chi Minh, Viet Nam

Anh-Tuan Nguyen*
University of Science
Ho Chi Minh, Viet Nam

Anh-Tuan Quang-Hoang*
Ford Motor
Los Angeles, United States

Van-Huy Dinh*
HUTECH University
Ho Chi Minh, Viet Nam

Tien-Huy Nguyen*
University of Information Technology
Ho Chi Minh, Viet Nam

Hoang-Bach Ngo
University of Science
Ho Chi Minh, Viet Nam

Minh-Hung An
FPT Telecom
Ho Chi Minh, Viet Nam

ABSTRACT

In the rapidly evolving landscape of multimedia data, the need for efficient content-based video retrieval has become increasingly vital. To tackle this challenge, we introduce an interactive video retrieval system designed to retrieve data from vast online video collections efficiently. Our solution encompasses rich textual to visual descriptions, advanced human detection capabilities, and a novel Sketch-Text retrieval mechanism, rendering the search process comprehensive and precise. At its core, the system leverages the Contrastive Language-Image Pretraining (CLIP) model, renowned for its proficiency in bridging the gap between visual and textual data. Our user-friendly web application allows users to create queries, explore top results, find similar images, preview short video clips, and select and export pertinent data, enhancing the effectiveness and accessibility of content-based video retrieval.

CCS CONCEPTS

• **Information systems** → *Information retrieval diversity*.

KEYWORDS

multimodal retrieval, text-based image retrieval, sketch-based image retrieval, interactive video retrieval

ACM Reference Format:
Minh-Dung Le-Quynh, Anh-Tuan Nguyen, Anh-Tuan Quang-Hoang, Van-Huy Dinh, Tien-Huy Nguyen, Hoang-Bach Ngo, and Minh-Hung An. 2023. Enhancing Video Retrieval with Robust CLIP-Based Multimodal System. In *The 12th International Symposium on Information and Communication Technology (SOICT 2023), December 07–08, 2023, Ho Chi Minh, Vietnam.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3628797.3629011

*All authors contributed equally to this paper.

SOICT 2023, December 07–08, 2023, Ho Chi Minh, Vietnam

ACM ISBN 979-8-4007-0891-6/23/12…$15.00
https://doi.org/10.1145/3628797.3629011

1 INTRODUCTION

The exponential growth of multimedia data, particularly video content on the internet, has ushered in an era where the effective and efficient retrieval of relevant information presents a pressing challenge. Content-based video retrieval, the task of retrieving video frames based on textual queries, has gained significant attention in recent years. The demand from users is steadily increasing, requiring faster query speeds and reduced time to locate a specific frame within a vast collection of videos based on the provided textual query.

In recent times, groundbreaking advancements in multimodal research have emerged in the form of many powerful vision-language models [7, 11, 16], bridging the gap between textual and visual modalities. Among them, Contrastive Language-Image Pretraining (CLIP) [14] has emerged as a powerful pretraining backbone for a broad range of applications in text-image-related tasks. By learning from a vast collection of text-image pairs and minimizing the cosine similarity between the text and image embedding vectors, CLIP has proven to be particularly well-suited for retrieving videos based on textual queries. The advent of CLIP has opened up many possibilities for developing robust and powerful content-based video retrieval pipelines.

In this paper, we introduce our system, which harnesses the power of the CLIP model to extract abstract, content-rich features from the videos in the dataset. To further enhance the capabilities of the CLIP model, we've integrated various supporting models and techniques, including human detection and retrieval from sketches and text. To facilitate rapid and robust retrieval, we efficiently index these features using Faiss, a state-of-the-art similarity search library, in conjunction with our database systems. This strategic integration empowers our system to provide faster and more reliable video retrieval, ensuring users receive accurate and relevant results. We participated in the Ho Chi Minh AI 2023 competition, a formidable challenge that requires the retrieval of pertinent video data from a massive 500+ hour video collection. This undertaking presents a significant challenge for any participating team.

In the following sections, we delve into the architecture and components of our system, shedding light on how it addresses the challenges posed by the burgeoning volume of video data on the internet.

2 RELATED RESEARCH

In our daily lives, this encompasses actions captured through images, events we've participated in, knowledge we've acquired, data related to personal information, and information about various organizations. This data serves to aid our memory—enabling us to search for past knowledge, retrace our paths traveled, and revisit images from specific events [3]. This remarkable concept has significantly contributed to the global increase in information retrieval. People are increasingly inclined to retrieve and revisit data from their past.

In Content-based image retrieval using the Tesseract OCR engine and Levenshtein algorithm [1], the author presents a technical proposal for conducting text-based OCR queries within images or videos. This approach is particularly valuable for large datasets containing small details written in multiple languages worldwide or traditional characters specific to a particular nation. The extracted characters are organized into word groups, resembling a Bag of Words (BOW) [17] structure. This enables users to effectively leverage scene-specific attributes like street names or license plates to determine the context they are searching for. The accuracy of this data is significantly high when the input is a high-resolution image with clearly represented feature matrices.

For users searching for songs but only remembering the lyrics, and sometimes unable to recall the song title or the artist's name, the ASR model [5] transforms video data into speech data of human subjects, extracting them into word clusters or sentences. These are then constructed into a text-based search tool from speech to optimize language data. In the paper by Liu et al. [8], the authors focus on researching transfer learning methods to enhance the efficiency of the automatic speech recognition system in Vietnamese. Specifically, they delve into pre-training and fine-tuning (PT/FT) methods [18, 20, 21], Prognets architecture, and bottleneck features.

The AI Challenge (AIC) has organized a competition focused on data retrieval, emphasizing creativity and implementation among participants. In this paper, we leverage OpenAI's latest state-of-the-art model, CLIP [10, 14], to establish semantic congruence between text and images. CLIP maps text and images into a matrix to make feature predictions. Despite its high performance, this powerful model hasn't yet reached its maximum potential for accurately retrieving required information. To unlock CLIP's capabilities fully and enhance its performance, we augment it with Faiss [4]-based search functionality. Faiss helps narrow the search space for feature vectors, especially for small or less accessible images within our current configuration.

This encompasses our entire preparation process for both text-based and video preview queries. Since over half of the participating teams in the competition are utilizing CLIP as their primary model, we propose a solution: leveraging our memory to sketch details to deliver results. Image search through sketching on modern touch devices has gained traction in recent years, and with Sketch [2, 9, 13, 19], it has become a hot topic. However, this type of search yields a vast dataset, making it challenging to select the desired results. In our system, we combine sketching with text to narrow the search scope, aiming to develop a more compact representation, increase matching speed and enhance the system's search performance [15]. Sketching proves adept at handling image details that cannot be queried with text alone. Developed based on CLIP, it enables users to rediscover past images or forgotten memories. Moreover, our system allows users to search for portions before or after the image in question. The system's architecture is depicted in Figure 1, illustrating the overall process, and will be discussed in detail in the following section.

3 VIDEO PREPROCESSING

The Transnet model is a powerful tool in the field of shot transitions. In the initial stage of our system pipeline (Figure 1), known as video preprocessing, we utilize the Transnet model to transform long video sequences into a list of scenes. Following this, we employ FFmpeg [1], an open-source software framework, to extract three main keyframes, denoted as p_i, for each scene using the formula 1 where s and e represent the first and last frame positions of the scene, respectively.

$$p_i \in \{s + (e - s) \times (0.5i - 0.5) | i \in [1, 2, 3]\} \quad (1)$$

Keyframe extraction from videos is a crucial stage, serving as a cornerstone for the effectiveness of our retrieval system. It enables concise information filtering and minimizes memory usage within the retrieval system. These keyframes will undergo processing in the subsequent indexing stages, as elaborated in Section 4.

4 MULTIMODAL RETRIEVAL

4.1 Text-based Retrieval

The remarkable feature of the CLIP model [14] lies in its exceptional ability to connect text comprehension with image recognition, effectively bridging this gap. This attribute has elevated CLIP into a potent tool for addressing the significant challenges associated with locating images and videos within extensive online multimedia content. Users can input text queries in a natural language format, enabling them to achieve their search objectives through a more intuitive and user-centric approach.

Our system applies the CLIP model for text-based retrieval through the following steps:

- All keyframes collected from videos are embedded into vectors to store features in Faiss before retrieval (indexing part in Figure 1).
- When users input a text description query, it is embedded using the CLIP model to create a text-embedding vector. This step involves converting text into numerical vectors, enabling the collection of query features, and proceeding to the next stage of image retrieval.
- After obtaining the text embedding from the query, we perform a cosine similarity measure between the text embedding vector and all the keyframe embedding vectors stored in Faiss (retrieving part in Figure 1). The returned result consists of the closest vectors in Faiss, limited to a specified number based on the top-K closest vectors with the highest similarity score.
- To improve the performance of the embedding model, we utilize the latest large CLIP model, specifically the Vision Transformer pretrained at a 336-pixel resolution (ViT-L/14@336p).

[1] https://github.com/FFmpeg/FFmpeg

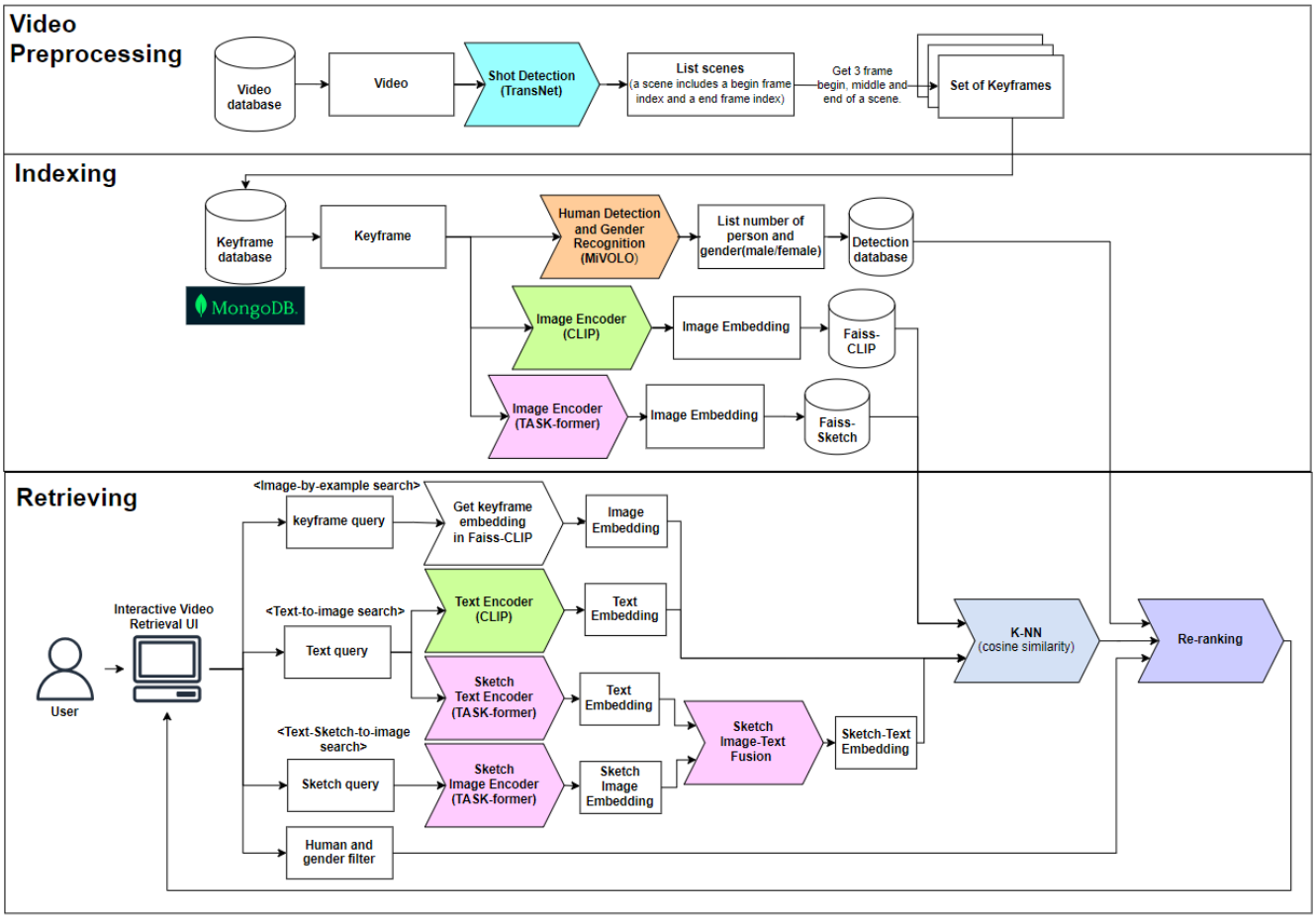

Figure 1: Video retrieval system architecture. Our video retrieval system architecture comprises three distinct phases. The initial phase, known as video preprocessing, involves scene detection using the TransNet model. Subsequently, FFmpeg is utilized to extract keyframes from the video, and these keyframes are stored in a MongoDB database. In the indexing phase, each keyframe in the database undergoes vector embedding. These embeddings are then indexed using the Faiss library. Additionally, we employ a Human Detection and Gender Recognition model to gather crucial information for re-ranking. Lastly, in the retrieval phase, we encode the user's query into an embedding using specialized models. Leveraging the k-nearest neighbors (kNN) algorithm, we identify the most relevant videos. To optimize retrieval accuracy, we implement a simple yet effective re-ranking algorithm based on human detection.

CLIP revolutionizes image retrieval with easy application, cost-efficiency, and high effectiveness. Its natural language query capability simplifies the process, reducing complexity and making it user-friendly for everyone. The low implementation cost makes it accessible, amplifying its utility across various applications. Additionally, CLIP's efficiency in processing natural language queries ensures rapid and accurate image retrieval, underscoring its transformative impact.

4.2 Image Retrieval With Text And Sketch

Retrieving images of events through text-based queries has been a fundamental aspect of image retrieval systems, enabling users to describe events using words and phrases, facilitated by the powerful state-of-the-art CLIP model in vision language processing. Although relying solely on text queries is a common approach in image retrieval systems, it depends heavily on the effectiveness of the pre-trained CLIP model and the characteristics of user query generation. When provided with specific image information about any event, each user may compose the sentence in a written query in various ways. Each type of query can yield a multitude of results with varying levels of effectiveness based on the user. To address this, our system has introduced an innovative retrieval approach by combining text and sketch queries. This integration allows for a richer representation of information, enhancing the accuracy of retrieving images related to the event being searched. In the case of sketch queries, users input a specific shape representing the image they aim to retrieve into the system. These sketched images remain

consistent regardless of the user, providing a more precise means of conveying retrieval intentions.

We employ a pre-trained model called TASK-former [15] to construct this query method. TASK-former is trained on the CLIP model (ViT-B16) and is designed for combined query operations, taking two inputs: a sketch image and text. The model is optimized to handle even poorly drawn sketches, proving to be more effective than traditional text-based image retrieval methods.

To implement this query method within the system, we follow a specific process. Firstly, we extract features for all images in the database, saving them to a numpy file and compiling them into a binary file for storage in Faiss (indexing part in Figure 1). When inputting a query, we encode both the text and sketch queries using the encoding function in the TASK-former model. This yields two corresponding feature representation vectors for each query type. Subsequently, we combine these two vectors into one using an existing function in TASK-former designed for this purpose. Let z_i and z_j represent the embedding vectors obtained from the text query and sketch, respectively. We apply the formula 2 provided by TASK-former.

$$y = \frac{1}{2}(z_i + z_j) \tag{2}$$

The resulting merged vector, representing the queries, is then searched in Faiss to find the closest vector and provide the appropriate result (retrieving part in Figure 1).

This merger signifies a promising paradigm shift in image retrieval, aiming to bridge the gap between linguistic and visual aspects of image search. By harnessing the power of natural language processing for text-based queries and computer vision for sketch-based queries, we strive to pave a new avenue for event image retrieval.

4.3 Human Filter And Re-Ranking The Results

While the CLIP model demonstrates promising performance in video retrieval tasks, it does have certain limitations. Notably, it struggles with object counting and accurately measuring distances between objects in an image. In the AIC contest, there are text queries that provide minimal contextual information, focusing primarily on describing the number of people, their gender, and their appearance in a short video clip.

To mitigate these challenges while still capitalizing on the strengths of the embedding model, we have implemented a human filter to re-rank the results generated by the CLIP model. We employ MiVOLO [2][6], a state-of-the-art model for gender recognition pre-trained on the IMDB-Cleaned dataset, to detect and calculate the number of humans, males, and females appearing in keyframes. This information is stored locally. Subsequently, users can select a filter (Female, Male, or Both) and specify a particular number to re-rank the results of text or sketch-text retrieval accordingly.

Taking the female filter as an example, see Algorithm **??**. The re-ranking algorithm involves comparing the query number with the female count (n) detected by the MiVOLO model and assigning that keyframe to one of three lists: 'equal list' (E), 'large list' (L), or 'small list' (S). These lists are then concatenated in order to create a single list (B) that is displayed on the screen. We have similar filters for males and both genders, employing the same approach as the female filter mentioned earlier. This reordering technique is simple but significantly improves the order of CLIP retrieval results, particularly when users precisely know the number of people (male, female, or both) appearing in the frame.

Algorithm 1 Re-ranking on female filter

Require: List of retrieval keyframe A, query number n
Ensure: List of re-ranking keyframe B

```
Initialize three empty lists: the equal-list E, the larger-list L, and the smaller-list S
for each keyframe a in A do
    if the female number in keyframe a is equal to query number n then
        Add a to E
    else if the female number in keyframe a is larger than query number n then
        Add a to L
    else if the female number in keyframe a is smaller than query number n then
        Add a to S
    end if
end for
Concatenate three lists E, L, and S together into a single list B in the order.
```

5 SYSTEM OVERVIEW

Our content-based video retrieval system is built upon three fundamental building blocks: the system architecture dedicated to data processing and retrieval, a data service block managing data storage and API operations, and a user interface facilitating interactions with the search system. To initiate the retrieval process, videos are initially processed and their frames compressed for optimal performance. We have opted for MongoDB as our preferred data storage solution due to its efficient key-value access to the stored data. Once all video frames are stored in the database, our system leverages this infrastructure to retrieve video frame information using either a sketch or a clip model via a PyMongo connection. These video frames are then converted into a binary format and presented as results on the user interface through API calls.

5.1 System Architecture

The infrastructure serves as the primary component for handling data inputs and outputs. We've selected the Flask framework to optimize performance and API traffic between the system and the user interface within the infrastructure service layer. After the frames are extracted and their information is converted into binary files using sketch and CLIP models, these files are stored within the system to compute and obtain the correct image indices.

Before returning the image indices, we need to process information from the user interface, namely the text input and the binary sketch data input. Depending on the text input, the system will translate it into English for the CLIP model. These information inputs are then vectorized for computation, aiding in the retrieval

[2] https://github.com/WildChlamydia/MiVOLO

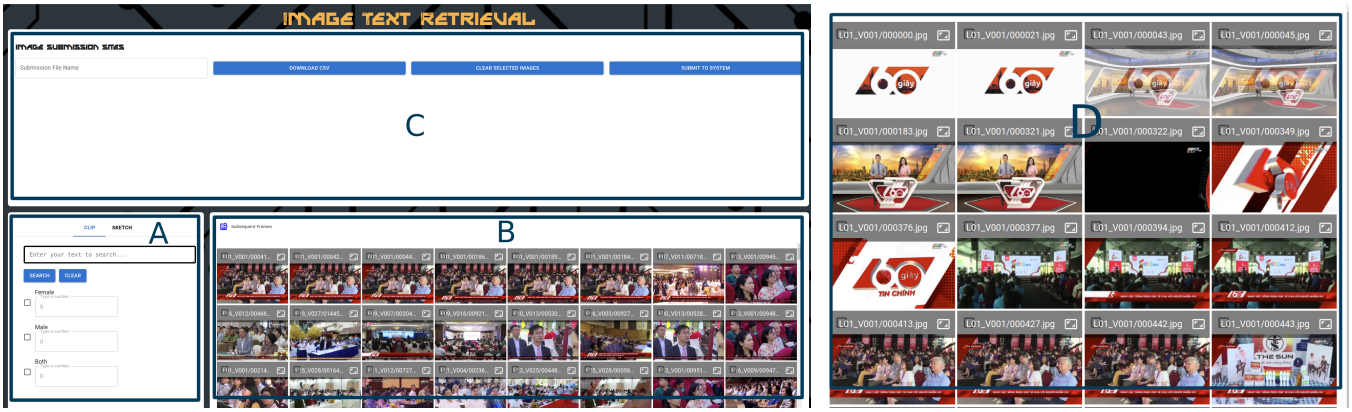

Figure 2: The user interface consists of three major components. Component A allows users to input text, use a filter system, and input sketch data. Once values are provided in Component A, Component B displays the corresponding images based on these inputs. Users can expand the images in Component B for more details and utilize the K-Nearest Neighbors (KNN) algorithm to search for similar images by clicking on a preferred image. Additionally, Component B includes a feature to display subsequent frames (Component D), both preceding and following the selected image when using the KNN search. Component C is designed for selected images, which can be chosen by clicking checkboxes in Component B.

of indices from the binary files (sketch and CLIP). The system architecture utilizes CLIP with backbone ViT-L/14@336px for text and ViT-B-16 for sketches, converting readable information into vectorized numbers for Faiss processing.

Faiss performs the critical task of searching and computing the similarities between the input text, sketch data, and binary files, returning the image indices. These image indices serve as keys to retrieve images from the MongoDB data server. The Pymongo framework, acting as a connection between the MongoDB server and the system architecture, fetches the images from the database based on the input indices. The binary images are retrieved then stored in JSON format and pushed to the user interface for display.

The filter system also referred to as the re-ranking system, filters the number of males, females, or both in the video keyframes. Once the indices are received from Faiss, they pass through this filter layer before the final indices are returned for processing. This filter layer detects the number of males, females, or both based on the values provided by the user interface. In Algorithm **??**, there are two input parameters: a list of retrieval keyframes and the query number n (representing the count of males, females, or both). After passing through various conditions in the filter, a final list of indices B in a specific order is returned. These ordered indices are then used to retrieve images from the MongoDB database.

The API traffic system is structured using the Flask framework, featuring three primary API routes: get-text-search, get-sketch-search, and get-image-search. Each route serves a specific function within the system. The get-text-search route processes input text by encoding it into vectors for further processing, ultimately returning the respective image indices. Similarly, the get-sketch-search route handles both sketch data and input text. In this data processing pipeline, the data undergoes preprocessing, followed by encoding into numerical vectors. These vectors then undergo a normalization step to ensure consistent scaling before leveraging Faiss, a powerful search engine for image indices. Acting as a gateway to the k-Nearest Neighbors Algorithm (KNN) search functionality, the get-image-search route allows users to input an image index, prompting Faiss to execute a search operation and return image indices closely resembling the input image.

5.2 Data Management

All extracted frames and their associated metadata are stored in MongoDB. The 'id' serves as the key for Pymongo to efficiently search and return an array of objects based on the provided input IDs. 'Filename' functions as the key for storing the video name corresponding to the respective frame index value. Similarly, 'Imagedata' is the key for storing the binary format of an image linked to its frame index value. The keyframes are stored consecutively, indexed from 0 to the total length of all keyframes, and extracted using TransNet and FFmpeg. (Table 1)

Type	Key	Value (Example)
Number	id	0
String	FileName	L01_V001/000021
Binary	ImageData	

Table 1: Data format in MongoDB

5.3 Graphical User Interface

The graphical user interface comprises three main components. Part A is designated for inputting queries or sketching images with an integrated face recognition filter system. Part B is responsible for displaying image results based on either text queries or sketches provided in the input query. The number of images displayed is influenced by the 'k' value, as mentioned in the system architecture.

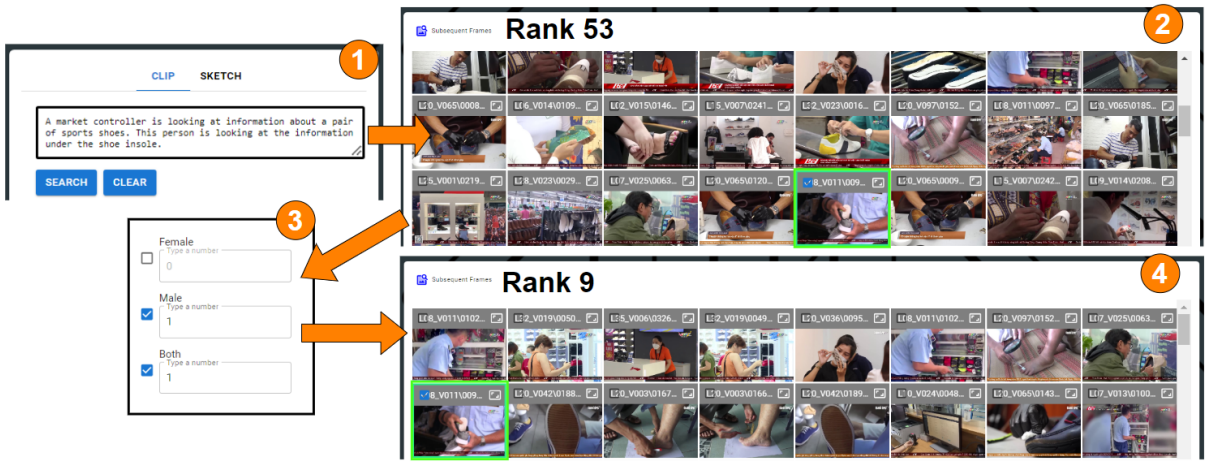

Figure 3: In text-based search with filters, we start by entering a text query in (1) and then clicking the search button to view the retrieval results in (2). Initially, the target scene is ranked 53rd. To improve its ranking, we apply the "Male" and "Both" filters and specify the presence of only one man in the scene by choosing the option "1." Clicking the "Search" button again, we see the results in (4), and now the target scene is ranked ninth.

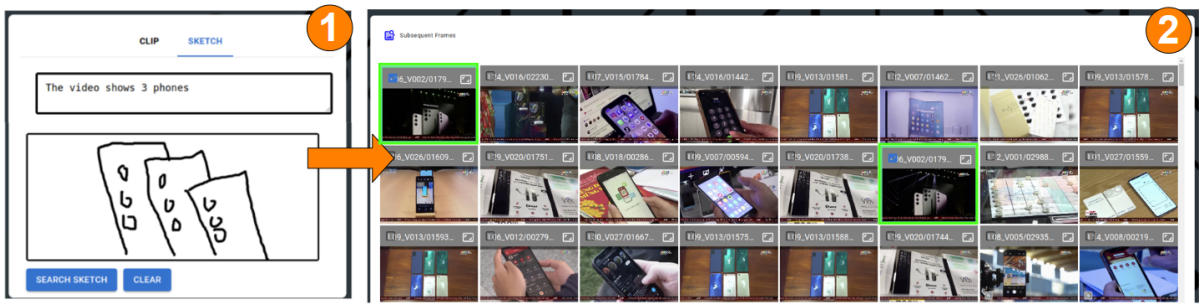

Figure 4: Search based on text and sketch includes 2 steps. Firstly, we input text query and sketch (1). Secondly, the search results are displayed and the target frame is selected (2).

Part C is dedicated to selected keyframes, allowing users to select keyframes by clicking checkboxes in Part B.

In Part A, users have the option to input text queries in either Vietnamese or English for image retrieval using text. Additionally, there is a filter system capable of detecting the number of males, females, or both by utilizing the re-ranking method. Upon clicking the search button, the input text and filter values are transmitted to the system architecture for processing. The system architecture then returns an array of image objects, all of which are displayed in Part B. In Figure 2, the filter system in the user interface accepts values for males, females, or both, with checkboxes provided next to each value to enable or disable them for processing and searching. Regarding sketches, users can toggle the sketch feature on or off, providing additional information for CLIP to re-rank similar images based on the sketch. Image, text, and filter values are sent to the system architecture for processing, and in turn, the system architecture returns an array of image objects for display in Part B. Apart from the aforementioned capabilities, our system offers a seamless experience for users, enhancing their ability to explore and analyze visual data. The 'subsequent frames' feature, as previously mentioned, allows users to effortlessly view not only the current frame but also the surrounding context. By presenting 20 frames before and 20 frames after the selected frame during a KNN search, it offers an extensive chronological perspective. This innovative functionality ensures that the context of the chosen frame is presented with the utmost clarity and precision.

Part C is designated for the selection of frames for submission. After clicking checkboxes to select images in Part B, the chosen frames will be displayed in Part C for review before submission. If users wish to remove frames, the graphical user interface allows them to click and remove frames in Part C. Users can then choose to submit all selected frames to the system or download CSV files to their local storage.

5.4 System Utilization

Text retrieval is the most commonly employed function in Textual KIS (Known-Item Search). Figure 3 provides an example of this

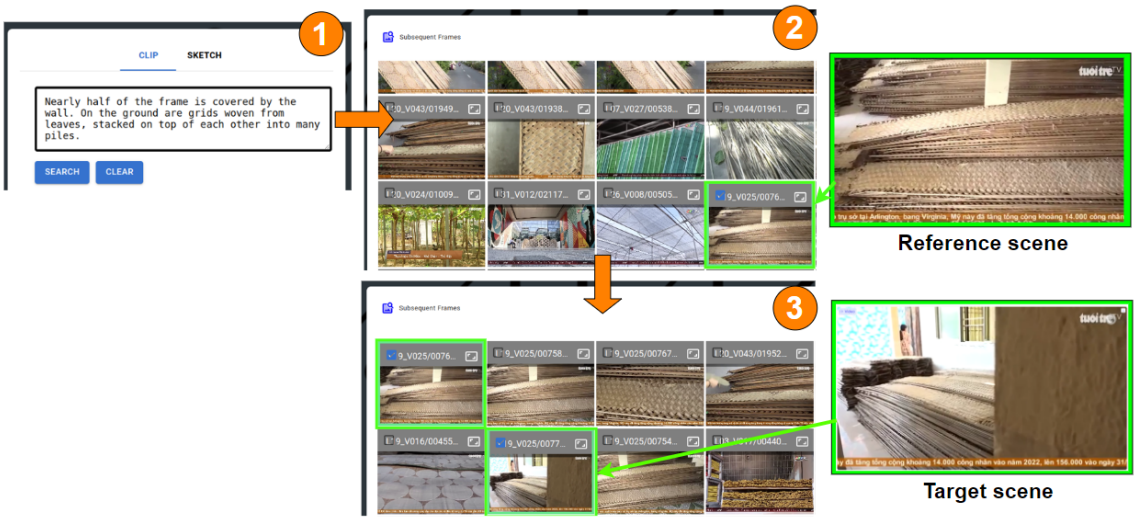

Figure 5: Image-by-example search comprises three steps: Step 1 involves making a text query or a text-sketch query, as shown in (1). In Step 2, we double-click the reference scene in the retrieval results displayed in (2). Step 3 involves checking if the target scene appears on the screen, as depicted in (3). If it does not, we return to Step 1.

function. In the first step, we input the text query: "A market controller is looking at information about a pair of sports shoes. This person is looking at the information under the shoe insole" The target frame is initially ranked 53rd. To achieve better results with higher rankings, we employ a human filter in step (3) to re-rank the results based on the number of men and the total number of people (equal to 1). As a result, the target frame now occupies the ninth position, which is a significantly improved placement.

When retrieving a short video clip from the database to solve video KIS, you can employ the sketch function to describe the video using text and sketches. To illustrate how to use this function, refer to Figure 4. The video features three Samsung phones during a product launch. In step (1), enter the query "The video shows 3 phones," and in step (2), sketch the basic outlines of the three phones, comprising three rectangles stacked on top of each other. Ensure that the ratio of the outline lines on the canvas in step (2) matches the positioning of the three phones according to the video's aspect ratio. After clicking the search button, the results will display images that closely resemble the sketched image and the entered description.

Through the two query methods mentioned above, we propose an additional method of querying through Figure 5. After describing the query "A girl with her hair tied up stands against a yellow pillar. Nearly half of the frame is covered by the wall. On the ground are grids woven from leaves, stacked on top of each other into many piles" in step (1), the results returned in step (2) only provide related objects mentioned in the query but do not actually yield the correct results. To find the exact answer, we select images that are likely to contain the answer and base this on examining Subsequent Frames in Component D at Figure 2. If not found, we will repeat step (1) until we find the correct answer containing the image of the girl standing next to the yellow pillar and the stacked leaf grids as described in step (3).

6 FUTURE WORK

We plan to implement voice queries as a new feature to reduce typing time for text query inputs. This presents an opportunity for users to access information as seamlessly as possible. Additionally, we can apply this feature for convenient mobile searching.

Furthermore, some keyframes contain a substantial amount of text, a challenge known as the Scene-Text problem. Leveraging the information within these texts through optical character recognition techniques [12] enhances query performance.

Finally, there is the concept of a system that suggests classes for the set of results of the current query being displayed [10]. This system allows users to identify missing words and ideas for describing video frames, ultimately facilitating the generation of improved queries. This concept may be a subject for future consideration.

ACKNOWLEDGMENTS

This research is supported by AI VIETNAM.

REFERENCES

[1] Charles Adjetey and Kofi Sarpong Adu-Manu. 2021. Content-based image retrieval using Tesseract OCR engine and levenshtein algorithm. *International Journal of Advanced Computer Science and Applications* 12, 7 (2021).
[2] Ayan Kumar Bhunia, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. 2020. Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9779–9788.
[3] Mariona Carós, Maite Garolera, Petia Radeva, and Xavier Giro-i Nieto. 2020. Automatic reminiscence therapy for dementia. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*. 383–387.
[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.
[5] Peter Kitzing, Andreas Maier, and Viveka Lyberg Åhlander. 2009. Automatic speech recognition (ASR) and its use as a tool for assessment or therapy of voice, speech, and language disorders. *Logopedics Phoniatrics Vocology* 34, 2 (2009), 91–96.
[6] Maksim Kuprashevich and Irina Tolstykh. 2023. MiVOLO: Multi-input Transformer for Age and Gender Estimation. (2023). arXiv:arXiv:2307.04616

[7] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*.

[8] Danyang Liu, Ji Xu, Pengyuan Zhang, and Yonghong Yan. 2019. Investigation of knowledge transfer approaches to improve the acoustic modeling of Vietnamese ASR system. *IEEE/CAA Journal of Automatica Sinica* 6, 5 (2019), 1187–1195.

[9] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. 2017. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2862–2871.

[10] Jakub Lokoč, Zuzana Vopálková, Patrik Dokoupil, and Ladislav Peška. 2023. Video Search with CLIP and Interactive Text Query Reformulation. In *International Conference on Multimedia Modeling*. Springer, 628–633.

[11] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*. 13–23.

[12] Ravina Mithe, Supriya Indalkar, and Nilam Divekar. 2013. Optical character recognition. *International journal of recent technology and engineering (IJRTE)* 2, 1 (2013), 72–75.

[13] Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu. 2016. Sketch-based image retrieval via siamese convolutional neural network. In *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2460–2464.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[15] Patsorn Sangkloy, Wittawat Jitkrittum, Diyi Yang, and James Hays. 2022. A sketch is worth a thousand words: Image retrieval with text and sketch. In *European Conference on Computer Vision*. Springer, 251–267.

[16] Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

[17] Chih-Fong Tsai. 2012. Bag-of-words representation in image annotation: A review. *International Scholarly Research Notices* 2012 (2012).

[18] Keiji Yanai and Yoshiyuki Kawano. 2015. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 1–6.

[19] Sasi Kiran Yelamarthi, Shiva Krishna Reddy, Ashish Mishra, and Anurag Mittal. 2018. A zero-shot framework for sketch based image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 300–317.

[20] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *Advances in neural information processing systems* 27 (2014).

[21] Dong Yu, Li Deng, and George Dahl. 2010. Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition. In *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. sn.