ReViMM: Enhanced Video Retrieval with Reweighting Mechanism for Multi-Modal Queries

To Anh Phat[1,2†B], Truong Thanh Minh[1,2†B], Doan Nguyen Tran Hoan[1,2†], and Khanh-Duy Nguyen[1,2*]

[1] University of Information Technology, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
{21520085, 21520064, 21520239}@gm.uit.edu.vn, khanhnd@uit.edu.vn

Abstract. As the volume of video data grows, efficient retrieval has become crucial for applications in security and multimedia management. Our system, ReViMM, is designed to address this challenge by employing a novel reweighting mechanism that enhances query accuracy. This mechanism dynamically prioritizes essential elements within each query, significantly improving relevance and precision in search results. ReViMM integrates FAISS for similarity search, ElasticSearch for optimized indexing, and Whisper for robust speech transcription, allowing it to process complex multimedia inputs, including text, images, and audio. By combining these tools with large language models to generate precise descriptions, our system delivers a comprehensive solution for accurate, context-sensitive video and multimedia retrieval.

Keywords: search engine · image retrieval · lifelog events · LLMs · FAISS · ElasticSearch · video event retrieval · reweighting.

1 Introduction

In the era of video content explosion, the need for efficient visual information retrieval has become more urgent than ever. The primary goal of a video search tool is to identify and present the most relevant video segments based on the queries entered by users. In recent years, video search algorithms have made significant advancements, enabling faster and more accurate information retrieval. Notably, the emergence of multi-query search systems has opened up new possibilities, allowing users to search for videos using various inputs such as images, audio, or even short video clips. However, managing and accessing this vast amount of data remains a major challenge. The complexity and potential applications of this problem have attracted considerable attention from the research community. Inspired by renowned video retrieval competitions like the

† All three authors contributed equally to this research.
B Corresponding author.
* Instructor and supporter.

Lifelog Search Challenge (LSC) [1] and Video Browser Showdown[*] (VBS), the AI Challenge 2024[*] has been launched in Ho Chi Minh City, Vietnam. In this competition, each team is asked to create an interative retrieval tool that can find answers to specific questions from the given data within a time constraint. The organizers have introduced two tasks for this year's challenge: Known-item Search (KIS) and Question Answering Search (QAS). In the KIS task, teams need to find specific keyframes as quickly as possible. The QAS task, newly introduced this year, asks teams to provide text answers to questions. All tasks will be scored based on combining between correctness and solving time of the team's submissions.

In this paper, we present our system ReViMM (Enhanced Video Retrieval with Reweighting Mechanism for Multi-Modal Queries) to participate in the AI Challenge 2024. ReViMM is a multi-modal search engine based on OpenCLIP [2] as the main method for solving image-text retrieval problems. Moreover, it is integrated with multiple techniques to enhance its information retrieval capability, such as FAISS [3] (Facebook AI Similarity Search) for fast similarity search and ElasticSearch to make the search process more efficient. A key strength of our system is the reweighting mechanism, which allows users to dynamically adjust the importance of text or image elements in their query, significantly improving the precision and relevance of the retrieved results by tailoring the search process to individual needs. Additionally, we have also designed a user interface (UI) to support easier user interaction.

2 Related Research

Research on video retrieval can be categorized into three main areas: concept-based search tools, multimodal data fusion approaches, and vision-language joint embedding models.

- **Concept-Based Search Tools:** Concept-based methods focus on transforming images into a collection of annotations, such as objects, text, or additional data. For example, Vitrivr [4] offers multiple search modalities, including concept-based and OCR text retrieval, with added image stabilization to enhance input quality. LifeConcept [7] further enhances retrieval by using concept recommendation to bridge semantic gaps and supports features like people count and color-specific object recognition.
- **Multimodal Data Fusion Approaches:** These methods aim to integrate information from multiple sources, such as images, text, and audio. Myscéal [5] incorporates location and text-based query expansion and introduces a new similarity metric called aTFIDF for improved text matching. MEMORIA [6] integrates keyword search, time-period filtering, and various computer vision techniques to enrich the search database, supporting enhanced keyword-based retrieval.

[*] VBS: `https://videobrowsershowdown.org/`

[*] AIC24: `https://aichallenge.hochiminhcity.gov.vn/`

- **Vision-Language Joint Embedding Models:** This approach bridges the gap between visual and textual data by mapping both into a shared embedding space. OpenAI's CLIP model has been widely adopted for extracting image-text embeddings. For instance, Memento 3.0 [8] uses these embeddings along with cluster-based search to reduce query processing time, while E-LifeSeeker [9] leverages semantic-based search and embedding models to improve query accuracy, even addressing question-answering tasks.

Our Contribution: Unlike previous methods, ReViMM integrates a reweighting mechanism that allows dynamic adjustment of the importance of text and image query elements. This enables more precise retrieval in complex contexts, addressing diverse requirements in video analysis and multimedia management.

3 Overview of ReViMM

The ReViMM system was specifically developed to offer rapid and efficient retrieval functionality for the AIC Dataset from the 2024 challenge.

3.1 AIC 2024 dataset

The dataset used for this year's competition includes news videos from multiple YouTube channels: **HTV9 HD, HTV7 HD, HTV The Thao, etc.** Particularly, the dataset includes:

- **Videos**: 726 videos.
- **Keyframes**: contains all keyframes extracted from the video provided above.
- **Objects**: contains the JSON formats of all objects detected from the Faster-RCNN pre-trained OpenImagesV4 model.
- **CLIP features**: contains OpenAI CLIP ViT B/32 embeddings of all keyframes.
- **Metadata**: video metadata information is taken from YouTube, the channel providing the data. The metadata of each video is a JSON file with a name corresponding to the video file name.

3.2 ReViMM architecture

To deliver a fast, accurate, and intuitive search experience, we propose the ReViMM architecture, which comprises two main components: information extraction and search engine, as illustrated in Figure 1. The information extraction component transforms keyframes, transcripts, and their descriptions into embedding vectors, which are then indexed using FAISS and ElasticSearch. The search engine component consolidates these embeddings, allowing users to search seamlessly and effectively.

4 Information extraction

ReViMM has been designed to take full advantage of the available information in the video, including keyframes, video transcriptions, and image descriptions in order to help users easily find the desired information.

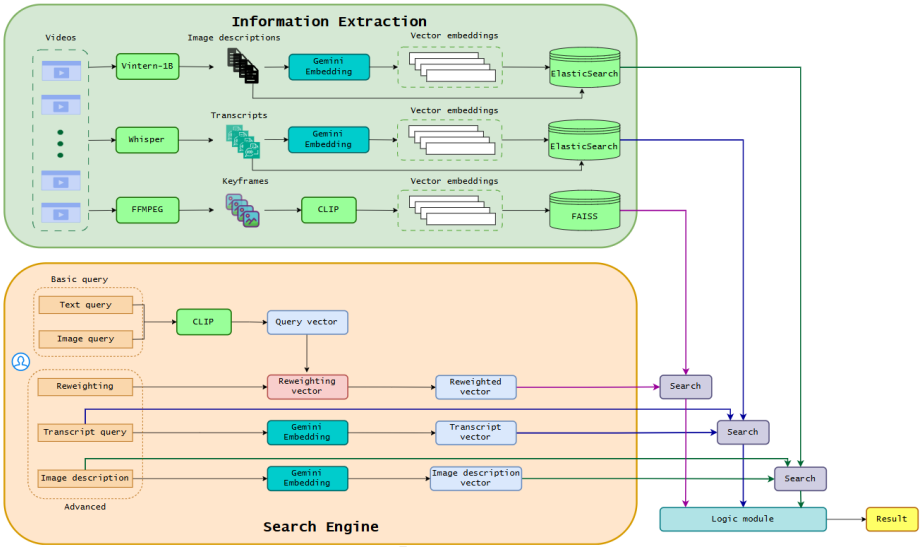

Fig. 1. The system architecture of ReViMM. This diagram illustrates the complete workflow of the system, including information extraction from videos, search engine functionality, and the reweighting mechanism to enhance query customization. Multiple modalities like text, image, transcript, and image description are processed and aggregated to generate the final results.

4.1 Keyframes extraction

We leverage the FFmpeg[*] library to efficiently extract keyframes from videos. These keyframes are then integrated into our search system for visual display. Central to our workflow, the frame encoding process generates searchable frame embeddings. After a thorough evaluation of available models for the image-text retrieval task, we selected OpenCLIP, an open-source variant of OpenAI CLIP. Specifically, we employed the ViT-B/32 model which demonstrated exceptional performance, achieving a 63.3% zero-shot ImageNet-1k [10] validation set accuracy. Once all keyframes have been encoded, we proceed to constructing indexes to facilitate efficient searching of frame embeddings. To accomplish this, we utilize FAISS, a library for efficient similarity search and clustering of dense vectors.

4.2 Speech Information extraction

The transcript of a video is a text document that describes the entire spoken content in the video. It contains everything that is said, converted from audio to written form. Therefore, given a specific query, we can figure out the transcript of the video. With this finding, we have integrated a search function based on the transcript into ReViMM. Particularly, we utilize Whisper [11], an automatic

[*] FFmpeg: `https://ffmpeg.org`

speech recognition (ASR) model developed by OpenAI, to convert the audio from provided video into text automatically and accurately. We utilize Gemini*, a Google DeepMind large language model, to extract information from transcripts. By generating vector embeddings and conducting a similarity search on both the embeddings and the original transcript, we are able to leverage the full content of the transcript. The final similarity score is calculated as the average of these two individual scores.

4.3 Image description extraction

An image description is a written explanation that provides details about the content of an image. We wanted to find specific moments in a video based on what they looked like, so we created a way to search for images within the video. We used a model called Vintern-1B [12] to describe each image in words. Then, just like we did with transcription, we used Gemini to encode these word descriptions into vector embeddings. Finally, we searched for the moments that matched both the word descriptions and the vector embeddings.

5 Search Engine

In this paper, we use cosine similarity as the distance metric to measure the similarity between vectors. Our search engine is designed to handle three types of queries: Basic Query, Transcript Query, and Image Description Query. Each query type is processed using different techniques to maximize retrieval accuracy and efficiency. The final results are aggregated to return the most relevant matches from our system. Additionally, the system introduces an advanced Reweighting Mechanism, which will be discussed separately.

5.1 Basic Query

The basic query is the core component of the search engine and is mandatory for every search. It can be either a text query or an image query. Once the system receives a basic query, the CLIP model converts the input into a vector representation. This query vector is then compared against vectors stored in the FAISS database, where cosine similarity scores are calculated to determine the relevance between the query and stored data.

By processing basic queries with CLIP embeddings and computing cosine similarity scores, the system retrieves results that are semantically relevant to the input, whether it is textual or visual in nature. This forms the foundation for further query refinement through optional reweighting or additional advanced queries.

* Gemini: `https://github.com/google-gemini/generative-ai-python`

5.2 Transcript Query

The transcript query is an optional input that enables users to search for spoken content in videos. Once a transcript query is submitted, the Gemini model generates a vector embedding of the transcript, capturing its semantic meaning. The system then conducts a search in ElasticSearch, combining BM25 with cosine similarity-based vector matching to identify relevant speech segments within the video.

5.3 Image Description Query

The image description query is another optional query type, designed for searching through descriptions generated from keyframes in the video. These descriptions, derived from the Vintern-1B model, capture key visual elements of each frame. When a user submits an image description query, the system embeds the input using the Gemini model, creating a vector representation. This vector is then matched against stored image descriptions in ElasticSearch, using both BM25 and cosine similarity to ensure accurate and contextually relevant retrieval.

5.4 Result Aggregation

The Logic Module aggregates results from the basic query, transcript query, and image description query to produce a final ranking of relevant keyframes. The module computes the average similarity score from all search methods to provide a balanced and comprehensive retrieval.
Let:

- $S_b(i)$ be the similarity score of the basic query for keyframe i.
- $S_t(i)$ be the similarity score of the transcript query for keyframe i.
- $S_d(i)$ be the similarity score of the image description query for keyframe i.

The overall similarity score $S(i)$ for keyframe i is computed by averaging the scores from all available query types:

$$S(i) = \frac{1}{N} \left(S_b(i) + S_t(i) + S_d(i) \right) \tag{1}$$

Where:

- N is the number of queries involved. If all three queries are provided (basic, transcript, and image description), $N = 3$. If one or more queries are missing, N adjusts accordingly based on the available query types.

Finally, the system returns the top k keyframes with the highest similarity scores:

$$\text{Top k results} = argmax_i(S(i)) \tag{2}$$

This formula ensures that the result aggregation is balanced, taking into account the various search methods, and providing results that are consistent across different modalities.

5.5 Reweighting Mechanism

While traditional search engines rely solely on the initial query to retrieve results, the Reweighting Mechanism allows users to refine their search dynamically by assigning different importance levels to additional text or image elements.

How Reweighting Works

1. **User Input:** Users provide a list of additional elements that they wish to reweight. These elements can be in the form of either text or images. Along with each element, the user assigns a **weight** w which can range $[-1, 1]$. The weight indicates how much influence the element should have on the final search vector:
   - A positive weight $w > 0$ means the element should have a **positive influence**, increasing its importance in the search.
   - A negative weight $w < 0$ means the element should have a **negative influence**, effectively acting as a filter or contrast.
   - A weight of $w = 0$ means the element will not affect the query at all.
2. **Vector Embedding:** Each item in the reweighting list is embedded into a vector using the CLIP model. CLIP maps both text and images into a shared embedding space, enabling cross-modal comparisons. These vectors represent the additional features the user wants to emphasize or de-emphasize in their search.
3. **Vector Adjustment**: The core of the reweighting mechanism lies in how the basic query vector is modified. Once the additional elements are embedded into vectors and their corresponding weights are applied, they are combined with the original basic query vector. The formula for computing the final reweighted vector is as follows:

$$q_{reweighted} = norm(q_{basic} + w_1 * v_1 + w_2 * v_2 + \cdots + w_n * v_n) \quad (3)$$

   In this formula:
   - $norm(x)$ denotes the normalization of vector x to have unit length, ensuring alignment with cosine similarity.
   - $q_{reweighted}$ is the final reweighted vector.
   - q_{basic} (basic query vector) is the vector representation of the initial user query (either text or image).
   - $v_1, v_2, \cdots, v_n$ are the vectors corresponding to the elements in the reweighting list.
   - $w_1, w_2, \cdots, w_n$ are the user-assigned weights that control the influence of each vector.

In essence, reweighting transforms a static query into a dynamic, interactive process where the user has the ability to guide the system toward more relevant, personalized results. This significantly improves the overall user experience and increases the effectiveness of the search engine in scenarios where typical queries might not suffice.

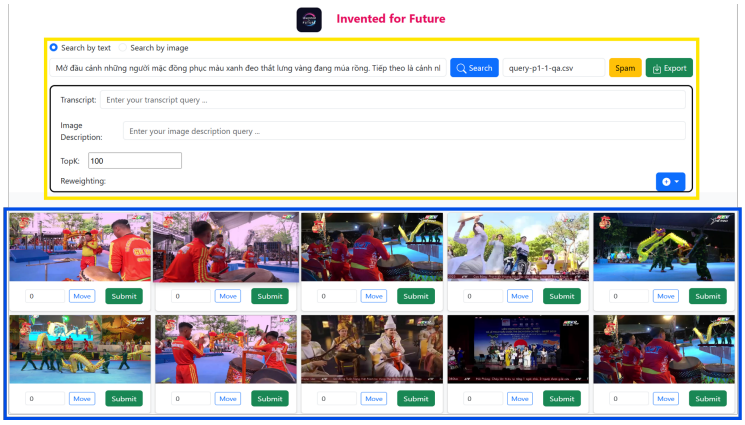

Fig. 2. User Interface. The screen shows the results of a text query without any enhancements: "The opening scene shows people in blue uniforms with yellow belts dancing with dragons. This is followed by a group of people playing drums."

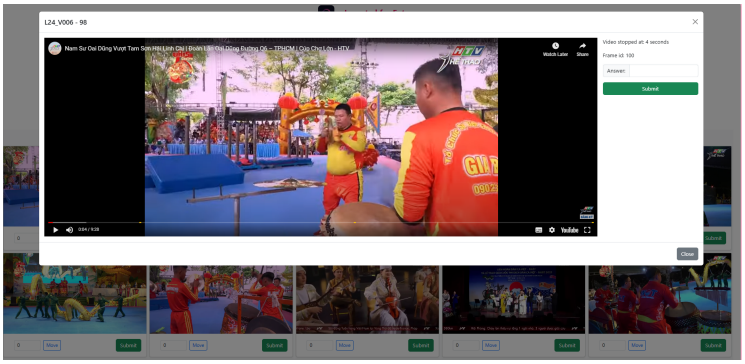

Fig. 3. User interface when clicking on any frame in the returned results

6 User Interface

We have developed a robust video query system that integrates diverse search methods and utilizes a multimodal approach to enhance retrieval performance. As illustrated in Figure 2, the interface consists of two main components: the search modal (yellow boundary area) and the result display area (blue boundary area). Users can easily select between text or image-based search modes and input their queries accordingly. Additionally, the system offers multiple search options such as transcript in video, the number of returned results or image description. These features enhance the user experience by allowing the customization of results to meet their specific needs.

Furthermore, the system supports reweighting search results and delving deep into hundreds of matches to ensure the most accurate results. By combining

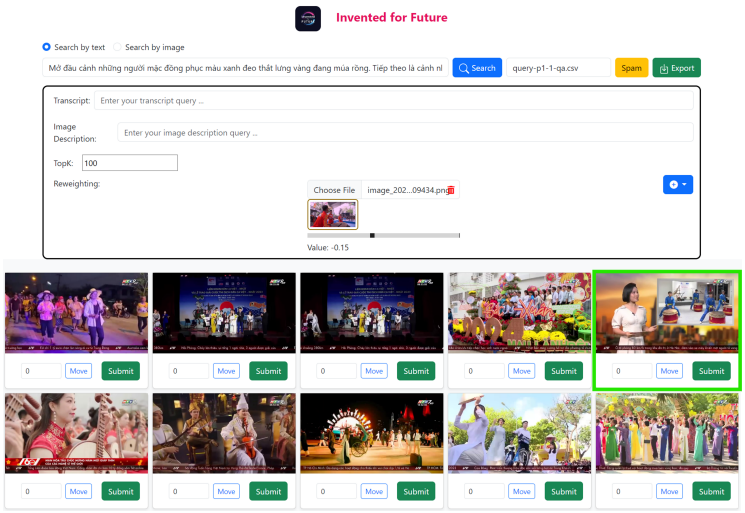

Fig. 4. Results when applying reweighting for the first time. The image outlined in green rectangle is the most similar to the query and will adjust the weight

various types of information, our system ensures that users receive relevant and precise results, making video retrieval more effective for practical applications. Each image in the response should be clickable, allowing users to view the frame ID and video ID, as well as watch the video starting from that specific frame. It is illustrated in Figure 3.

Example To demonstrate the effectiveness of ReViMM, we have the following query: "The opening scene shows people in blue uniforms with yellow belts dancing with dragons. This is followed by a group of people playing drums."

First, we only used the text query to find results. However, this did not return accurate results, as illustrated in Figure 2. In the returned results, we see many images of people in red outfits playing drums, which does not match our description. Therefore, we use reweighting to reduce the importance of similar images. In the example above, we decrease the importance of the second image to -0.15, as shown in Figure 4. After using reweighting for the first time, the returned results still did not meet my expectations. However, this time we noticed that the fifth image slightly resembled the description. Therefore, we decided to increase the weight of the fifth image to 0.2 and phrase "blue uniforms with yellow belts" to 0.8. Finally, we were able to find the desired image, which is in the fifth position in Figure 5.

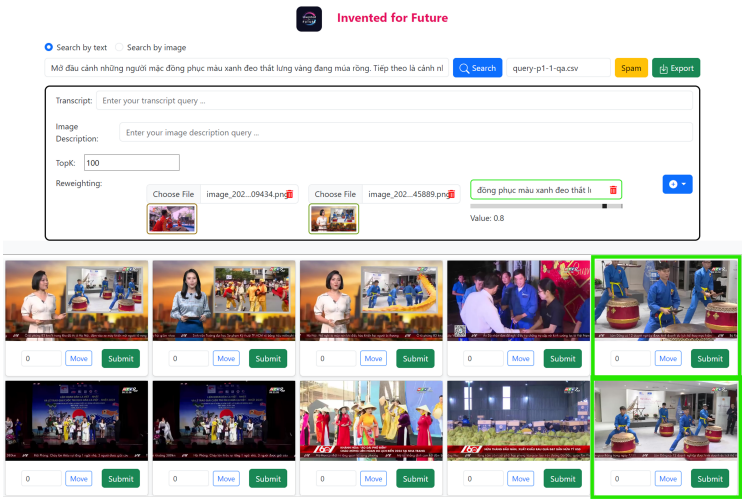

Fig. 5. Final results for the query. The images outlined in green rectangles match the query

7 Conclusion

In conclusion, ReViMM offers a robust and efficient solution for video event retrieval, leveraging advanced multimodal techniques to handle diverse input types, including text, images, and speech. By integrating FAISS for rapid similarity searches and ElasticSearch to improve retrieval results, the system ensures high accuracy and performance. The incorporation of Large Language Models (LLMs) like OpenCLIP and Gemini allows ReViMM to generate precise image descriptions and transcripts, optimizing the search process. Additionally, the innovative reweighting mechanism gives users the flexibility to prioritize or de-emphasize query elements, further enhancing the system's relevance and usability. These features make ReViMM a powerful tool for applications in video analysis, security, and multimedia management, addressing the growing challenges posed by the increasing volume and complexity of video data.

References

1. Cathal Gurrin, Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn Þór Jónsson, Jakub Lokoč, Wolfgang Hürst, Minh-Triet Tran, and Klaus Schoeffmann. 2020. Introduction to the third annual lifelog search challenge (LSC'20). In Proceedings of the 2020 International Conference on Multimedia Retrieval. 584–585
2. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. CoRR abs/2103.00020 (2021). arXiv:2103.00020 https://arxiv.org/abs/2103.00020

3. Matthijs Douze and Alexandr Guzhva and Chengqi Deng and Jeff Johnson and Gergely Szilvasy and Pierre-Emmanuel Mazaré and Maria Lomeli and Lucas Hosseini and Hervé Jégou. 2024. The Faiss library. arXiv:2401.08281 https://arxiv.org/abs/2401.08281
4. Silvan Heller, Ralph Gasser, Mahnaz Parian-Scherb, Sanja Popovic, Luca Rossetto, Loris Sauter, Florian Spiess, and Heiko Schuldt. 2021. Interactive Multimodal Lifelog Retrieval with vitrivr at LSC 2021. In Proceedings of the 4th Annual on Lifelog Search Challenge (LSC '21). Association for Computing Machinery, New York, NY, USA, 35-39. https://doi.org/10.1145/3463948.3469062
5. Ly-Duyen Tran, Manh-Duy Nguyen, Nguyen Thanh Binh, Hyowon Lee, and Cathal Gurrin. 2021. Myscéal 2.0: A Revised Experimental Interactive Lifelog Retrieval System for LSC'21. In Proceedings of the 4th Annual on Lifelog Search Challenge (LSC '21). Association for Computing Machinery, New York, NY, USA, 11–16. https://doi.org/10.1145/3463948.3469064
6. Ricardo Ribiero, Alina Trifan, and Antonio J. R. Neves. 2022. MEMORIA: A Memory Enhancement and MOment RetrIeval Application for LSC 2022. In Proceedings of the 5th Annual on Lifelog Search Challenge (LSC '22). Association for Computing Machinery, New York, NY, USA, 8–13. https://doi.org/10.1145/3512729.3533011
7. Wei-Hong Ang, An-Zi Yen, Tai-Te Chu, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. LifeConcept: An Interactive Approach for Multimodal Lifelog Retrieval through Concept Recommendation. In Proceedings of the 4th Annual on Lifelog Search Challenge (LSC '21). Association for Computing Machinery, New York, NY, USA, 47–51. https://doi.org/10.1145/3463948.3469070
8. Naushad Alam, Yvette Graham, and Cathal Gurrin. 2023. Memento 3.0: An Enhanced Lifelog Search Engine for LSC'23. In Proceedings of the 6th Annual ACM Lifelog Search Challenge (LSC '23). Association for Computing Machinery, New York, NY, USA, 41–46. https://doi.org/10.1145/3592573.3593103
9. Thao-Nhu Nguyen, Tu-Khiem Le, Van-Tu Ninh, Cathal Gurrin, Minh-Triet Tran, Thanh Binh Nguyen, Graham Healy, Annalina Caputo, and Sinead Smyth. 2023. E-LifeSeeker: An Interactive Lifelog Search Engine for LSC'23. In Proceedings of the 6th Annual ACM Lifelog Search Challenge (LSC '23). Association for Computing Machinery, New York, NY, USA, 13–17. https://doi.org/10.1145/3592573.3593098
10. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y
11. Alec Radford and Jong Wook Kim and Tao Xu and Greg Brockman and Christine McLeavey and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356 https://arxiv.org/abs/2212.04356
12. Khang T. Doan and Bao G. Huynh and Dung T. Hoang and Thuc D. Pham and Nhat H. Pham and Quan T. M. Nguyen and Bang Q. Vo and Suong N. Hoang. 2024. Vintern-1B: An Efficient Multimodal Large Language Model for Vietnamese. arXiv: 2408.12480 https://arxiv.org/abs/2408.12480