An Interactive System For Visual Data Retrieval Through Multimodal Intelligence

Tu Van Nguyen[1,2,***], Nghia Trung Duong[1,2,**], Nhan Thanh Pham[1,2,**], Thanh Xuan Luong[1,2], and Dang Duy Bui[1,2]

[1] Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
{22127434,22127293,22127307,22127387}@student.hcmus.edu.vn
bddang@fit.hcmus.edu.vn
** Equal contribution

Abstract. The widespread sharing of visual data online has created challenges in querying and extracting relevant information efficiently. This led to the establishment of the AI Challenge 2024, which aims to develop systems capable of processing queries and accurately returning event images from a large dataset of multigenre videos. Our team addressed this by creating an interactive system leveraging models such as CLIP, GPT-4o, PaddleOCR, and Whisper to generate precise embeddings and enable efficient data retrieval from diverse visual formats. The system is supported by an GUI with features like semantic search, OCR-based and voice-based queries, generative AI image queries, query enhancement, and advanced video preprocessing. Through this approach, our system reduced unimportant data by 21% and achieved an accuracy of 81.54% for correct answers among the top 10 responses [3]. This result showcases the system's potential for improving query response accuracy and efficiency in large-scale visual data processing.

Keywords: Multimodal retrieval · Query enhancement · Generative image-based query · Voice-based query

1 Introduction

Images and videos are used to record activities in daily life, from personal moments to significant events. These images and videos can be stored on social media platforms, video-sharing services, and news applications, resulting in a massive and diverse dataset in terms of size, genre, and subject matter. Because of this richness, retrieving and searching for precise images presents a significant challenge. This issue is the central focus of the 2024 Vietnam AI Challenge[4] whose format is similar to the Lifelog Search Challenge (LSC)[5] and

* corresponding author
[3] https://github.com/NghiaZun/AIthena_Statistic
[4] https://aichallenge.hochiminhcity.gov.vn/
[5] https://klausschoeffmann.com/lifelog-search-challenge/

Video Browser Showdown (VBS)[6]. The challenge provides a dataset containing Vietnamese videos across various genres and topics, with several tasks requiring the output of a specific frame (image) indexed from the videos in the dataset based on given inputs (e.g., text query). The goal of such tasks focuses on two factors: accuracy and speed, which is the objective that our tool paper aims to address.

This paper provides a tool that assists humans in solving tasks in the competition, as well as using several approaches to retrieve information. To do that, our research contains a two-stage system: **Data Preprocessing** and **Retrieval Processing**. In the first stage, videos from the input dataset are extracted into frames (images) and meta data (e.g., name entities), which become a database used for the **Retrieval Processing** stage. The latter stage is divided into two main activities: listing image candidates and confirming such images. For the first activity, the tool allows users to input natural language format queries as text, voice, or images, and the system returns relevant frame candidates (from the database) ranked by semantic similarity and context. The latter activity aims to confirm such candidates based on further reranking algorithms and humans. Note, for the competition, our team constructed the query and configured our tool instead of putting the full query from the challenge. To demonstrate the usefulness of our tool, some case studies in the competition are analyzed, which helps us to get the first rank in the 3rd preliminary round, and some statistical results are shown.

The following sections provides the related work, tasks' information in the competition, our tool architecture, along with experimental results. Our paper will conclude with key findings and future research directions.

2 Related Work

In the recent LSC'23 competition, LifeXplore [6] won first place by leveraging Open-CLIP (ViT-H/14) for powerful free-text search capabilities. Additionally, two other strong systems, Momento 3.0 [9] and Voxento 4.0 [2], also used CLIP as their primary method to tackle the challenge. Their approaches typically utilized CLIP to establish semantic compatibility between text and images by mapping both into a shared feature space for prediction. While this model achieved relatively high performance, it still fell short of fully realizing its potential for precise information retrieval. To maximize CLIP's capabilities and enhance its performance, these teams often integrated it with search functions based on FAISS [5], a technique that narrows the search space for feature vectors, especially for small or hard-to-access images.

In recent research, querying images based on their content has emerged as a popular approach. This method allows users to leverage key details in the image context, such as street names, license plates, and so on. For this method to work effectively, the images in the dataset need to be high resolution, and the feature matrices must be clearly defined. In this study [1], the authors propose an

[6] https://videobrowsershowdown.org/

OCR-based query method using Tesseract OCR and the Levenshtein algorithm. Tesseract OCR is a robust tool with multilingual text recognition capabilities. Meanwhile, the Levenshtein algorithm measures differences between character strings, enabling accurate identification of key phrases even when the text contains typos or variations, thereby improving query effectiveness and precision.

Another approach for image querying is based on video content, with an Automatic Speech Recognition (ASR) model playing a key role. ASR converts video data into speech, extracting relevant phrases to create a text search tool from voice data, optimizing language data, and supporting image querying. In the LSC'23 competition, some systems used speech-to-text technology, such as [8] with DeepSpeech. However, there has been limited discussion on systems that retrieve images in videos based on voice, making our system a unique example. Previous studies, like [7], have used the Speech API in their systems, but the Whisper from OpenAI has yet to be widely adopted. Our team has chosen to leverage the Whisper in our system to establish a new standard of accuracy and efficiency in voice-based information retrieval.

3 Preliminary

The Vietnam AI Challenge 2024, with the theme "Event Retrieval from Visual Data", is organized to discover optimal, accurate, and fast solutions for querying images based on prompts provided by the competition organizers. For each round, each team is given a large dataset consisting of videos covering diverse topics and genres and video frames extracted by the organizers. Throughout the competition, we receive a set of questions, and our task is to return the video identifier along with the frame index of the image we believe is the correct answer.

We have three tasks to complete, each with its own challenges. For the **Text Kis**[7] part, it is described in quite a bit of detail the retrieval scene in natural language format, however it often includes some noisy elements. In the **Video Kis** section, a video will be played, and we are required to find the corresponding segment of that video; however, the dataset contains many duplicate videos that are only slightly different, so finding the right frames will require high accuracy. The **Question and Answer Kis** section will combine the Text Kis part with answering questions that relate to the event, which brings a big challenge: answering questions accurately in difficult situations like blurry or noisy images...

According to the competition rules, each query has a maximum retrieval time of 4 minutes for the Video KIS format and 5 minutes for the Textual KIS and Q&A formats, with a maximum score of 1000 points, consisting of 500 points for accuracy and 500 points for retrieval time. Each incorrect submission loses 100 points of the question. Retrieval time points are calculated using a linear reduction from 500 to 0, depending on the time the team takes to find the answer. For each question given by the contest organizers, the answer is required in these fields:

[7] **Kis:** Known-item search

– The name of the video file.
– How many milliseconds does it take for the frame to appear?
– The answer result (for Q&A KIS questions).

4 Proposed Method

4.1 Overview

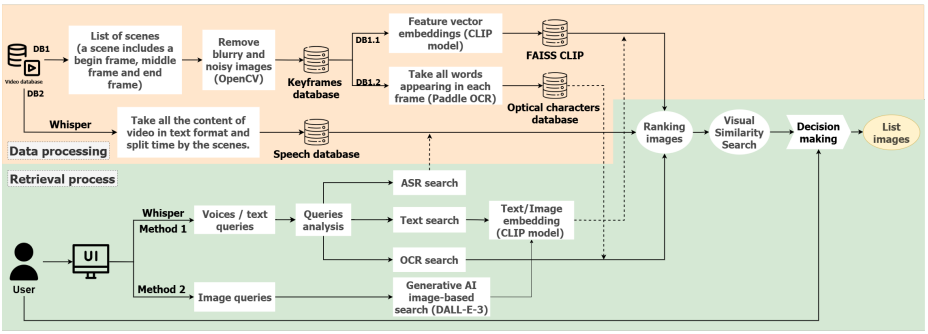

Fig. 1: Workflow for video content-based image retrieval: The system processes video with shot detection, keyframe extraction, and embedding generation. It combines text, audio, and image queries, using indexing to improve ranking, and supports ASR, OCR, and image generation for better multimodal capabilities.

The diagram in Fig. 1 illustrates our system architecture, which retrieves images from the dataset using various AI models, including CLIP, Whisper, and PaddleOCR. The process is divided into two main parts: **Data Preprocessing** and **Retrieval Process**.

In the **Data Preprocessing** stage, the system extracts key images (keyframes) from the video, ensuring that blurry or unclear frames are filtered out. Next, the system employs the CLIP model to generate a digital representation (embedding) of each image, capturing its semantic and visual information. For text extraction, PaddleOCR is utilized to detect and store any textual content present in the images. Simultaneously, the system processes the audio using Whisper, which converts spoken language into text, enabling comprehensive multimodal data analysis.

In the **Retrieval Processing** stage, users can search for images in different ways: by typing in a query, speaking the query (speech is converted to text), or even uploading an image. These query methods are classified into two groups: Language (including text and voice) and Image. For language queries, the user should analyze the queries, and the system compares them with previously processed data. The system uses several methods, such as audio, text, or text-from-image (OCR) search. For image queries, the system can match the user's image

with similar images in the database using the CLIP model. There is also an object outline feature, which uses DALL-E 3 to help retrieve relevant images.

To rank and sort the results, the system uses FAISS [10] to find the best matches, then decides which images to show to the user. This makes it easier for users to find the right image based on their query.

4.2 Data preprocessing

Keyframe extraction: We utilize the TransnetV2 [15] model to analyze video content and detect scene transitions effectively. This model computes the transition ratio at each frame in the video. If the calculated transition ratio exceeds a predefined threshold, it is classified as a scene change. For every identified scene, we strategically extract three keyframes: the first frame, the middle frame, and the last frame of the scene.
The set of keyframes KF is given by:

$$KF = \{KF_{\text{first}}, KF_{\text{middle}}, KF_{\text{last}}\} = \{S_1, S_{\lfloor \frac{n}{2} \rfloor}, S_{n-1}\} \tag{1}$$

- S be the set of frames in the identified scene.
- n be the total number of frames in the scene, where $n = |S|$.
- S_i be the i-th frame in the scene.

By selecting only three frames to represent each scene, rather than capturing every frame, we achieve a more efficient yet effective summarization. Focusing on these specific frames enhances both the accuracy of content retrieval and the quality of the video summary.

Remove blurry images: To effectively eliminate blurry images from a dataset, we employ a technique that involves calculating the Laplacian variance [11] of each image. When we compute the Laplacian of an image, we measure how much the intensity of pixels changes across the image. A blurry image tends to have a more uniform intensity distribution, resulting in a low variance value. This technique removed a large amount of blurry images. Approximately 78,000 blurry images, constituting 17.33% of the dataset, were detected using this method.

Remove redundant images: For further optimizing the dataset, we propose a method using color histograms and edge detection with OpenCV. This approach identifies and removes redundant frames based on predefined thresholds, comparing each frame against a set of classified redundant frames. Frames deemed similar are excluded, eliminating about 30,000 redundant images (6.67% of the dataset) and improving retrieval accuracy.

Indexing and name entity extraction: We extract the 768-dimensional embedding of each frame using the CLIP ViT-L/14 model. This model has a larger architecture and significantly more parameters compared to the CLIP ViT-B/32,

which was recommended by the competition organizers. The increased number of parameters allows the CLIP ViT-L/14 model to capture more semantic features. Afterward, all the vector embeddings are indexed using FAISS, which builds an efficient vector database for the system. This database enables fast and accurate retrieval of similar frames during the search process. Further, the PaddleOCR [13] and the Whisper model are integrated. PaddleOCR is employed to extract text directly from image frames, enabling rapid identification of characters and words within the visual content. While that, the Whisper model processes audio streams to transcribe spoken language from videos.

4.3 Voice-based and text-based retrieval:

In our tool, users can input queries in a natural language format, supporting both Vietnamese and English. Additionally, the tool offers voice-based input, utilizing the Whisper [4] to accurately convert spoken queries into text. After that, based on the input query, GPT-4o will be used to extract information and generate an enhanced query that better captures the user's intent and provides more relevant results. This approach enables a convenient experience and improve accuracy

Sematic search: To implement the semantic search function, we leverage the power of the CLIP [3] model, specifically the CLIP ViT-L/14 model, along with FAISS [5]. This process is carried out in a step-by-step manner, as outlined below:

1. When a natural language query is provided, the CLIP model is used to encode the query into a vector embedding.
2. Then FAISS compares the query embedding with the dataset's indexed embeddings by computing the similarity between the query vector and the image vector using L2 distance.
3. Received an array of indexes sorted based on their similarity to the query.
4. Mapping the index back to the corresponding image
5. The result is a list of ranked images that have similar semantic features with the query

Characters and named entity search: To match extracted text and audio data with user queries, the Jaro-Winkler algorithm [12] is employed to calculate the similarity between text strings. This algorithm is particularly effective for handling minor variations in spelling or pronunciation. By ranking text based on similarity scores, the approach ensures that the most relevant result appears at the top of the list. The Jaro-Winkler algorithm assigns higher similarity scores to strings that share a common prefix, making it well-suited for comparing named entities or keywords. For example, it can effectively match variations like "Jonathon" and "Jonathan" or "color" and "colour." To improve search precision, a similarity threshold (e.g., 0.85) is applied. Only text with a similarity score above this threshold is retained. This filtering process enhances the accuracy of query results for both OCR and transcribed audio content, ensuring that only closely matching entities are considered relevant.

4.4 Visual similarity search

As mentioned, some "noisy" images share similar characteristics with the target image, making accurate identification challenging. To address this issue, we designed a function called Image Retrieval (IR). This function re-ranks the set of images previously ranked by other methods based on the image selected by the user for verification. For example, the user selects an image i that they believe is the correct match. The system then compares the embedding vector of the selected image i with the dataset embeddings using FAISS, as described in the Semantic Search section. After re-ranking, a new set of images with high similarity to the selected one is generated, making it easier to review and confirm the accuracy of the results.

4.5 Generative AI image-based query

During our recent competition, we encountered a critical obstacle: the video-KIS queries involved a highly abstract concept that couldn't be processed using standard tools. This query required a level of semantic understanding and visual creativity that went beyond conventional information retrieval methods. Faced with this limitation, we leveraged the DALL-E 3 model, known for its robust capability to generate images based on textual prompts. By converting our initial prompt ideas into corresponding images, we created a visual representation that facilitated further processing and comparison in our Imaged Retrival (IR) function. These AI-generated images were transformed into vectors, enabling precise matching with similar images in the system's database. Upon evaluation, this approach demonstrates superior accuracy compared to the traditional method. However, during further testing, it showed a longer response time. The above strategy is shown in Alg. 11.

Algorithm 1 Generative AI image-based query

Require: Textual prompt p representing the abstract concept
Ensure: List of relevant images B that match the concept in query p
Initialize an empty list V to store vectors of AI-generated images
Generate an image I using the DALL-E 3 model based on prompt p
Convert the generated image I into a vector representation v_I using feature extraction CLIP model
Add v_I to list V
for each vector embeddings v_i in the system's database **do**
    Compute similarity score s between v_I and v_i using FAISS
    if similarity score s meets the threshold t **then**
        Add image i to list B
    end if
end for
return list B containing relevant images that match the query

5 Experiment

To visualize all the above-mentioned sections, our system employed FLASK framework for optimizing the backend retrieval API traffic, while ReactJS and Tailwind CSS created an intuitive graphical user interface consisting of multiple sections with distinct roles, as can be seen in fig Fig. 2.

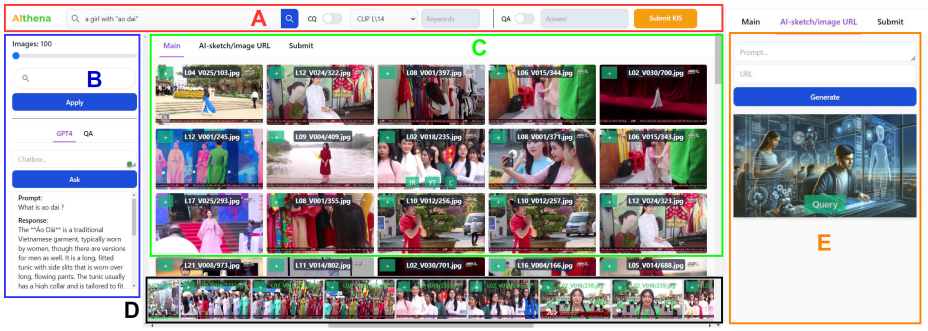

Fig. 2: The Graphical User Interface of the system with a simple CLIP-based query ranked result

The Graphical User Interface (see Fig. 2) includes five sections:

- Section A Toolbar Section: this section contains the text input field for entering the query, along with options for selecting query modes.
- Section B allows users to adjust the number of images retrieved and includes tools like a GPT-powered chatbox for query refinement and a QA module for image-related answers.
- After the query process, section C displays the images ranked according to relevance after a query is processed.
- Section D displays images that come before and after the selected image.
- Section E, This section of the system allows users to create new images based on a prompt; this images is then used for querying similar images

5.1 Case studies

As discussed, the tool mainly helps users to address tasks mentioned in Sect. 3. In addition to traditional semantic search methods that typically yield a ranked list, there are certain cases where alternative methods achieve higher rankings. In this section, we will provide three examples.
In the video-KIS format, queries often include specific named entities, such as “SG93711TS” shown in Fig. 3, We use the OCR-based querying function to retrieve images that contain this phrase. PaddleOCR detects and extracts the code even if it is partially damaged or obscured, such as by chipped paint.

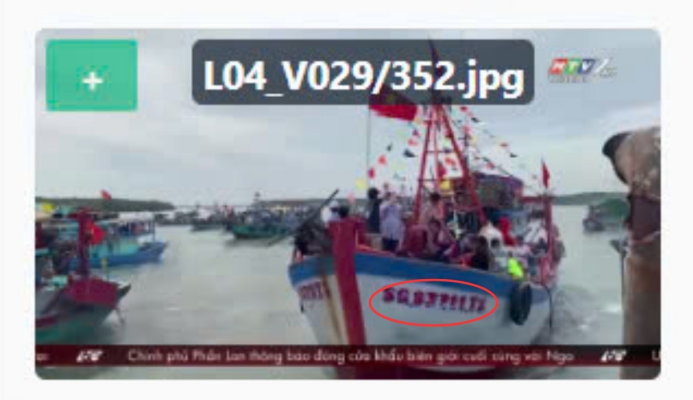

Fig. 3: The text "SG93711TS" was successfully recognized from the body of the boat in the image, which matches the search keyword "sg93" entered in the search bar.

Additionally, in the text-KIS task, to make it able to return the best candidates, our tool allows for users to search for images based on text prompts. For instance, the input query "Three Olympic medals of Paris 2024" (shown in Fig. 4) which ranks and returns results based on relevance. Note that the correct result is shown in the fifth example.

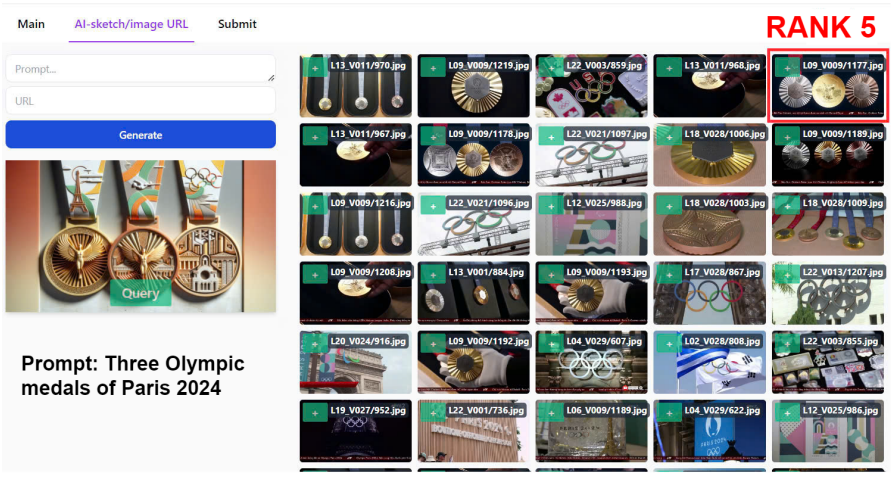

Fig. 4: Prompt-based search generates ranked image results. In this example, the query prompt specifies "Three Olympic medals of Paris 2024."

In the Q&A task, thanks to the power of visual question answering in the GPT-4o model, our tool automatically shows the answer of the question that related to the output image instead of using human eyes as usual. As can be seen in **Fig.** 5, with the question, "How many green rectangles have only the number one?", the chatbox shows the result is "one" in a short response time.

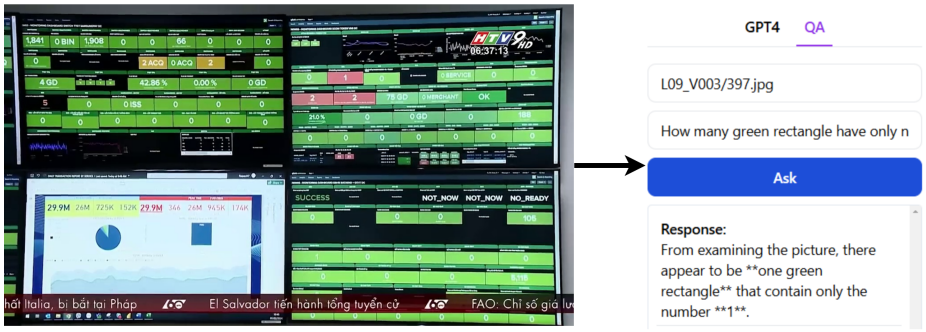

Fig. 5: Answering question: "How many green rectangles have only the number one?" from given image using GPT-4o

5.2 Performance statistics

This section shows some statistical results refer to two contexts, where one (i) is to use direct inputs from the competition and one (ii) is to use GPT-o to refine such inputs. We focus on the dataset with 65 questions from the preliminary round and 15 questions from the final round, each having 4 hints. Tab. 1 shows the results For the 65 preliminary questions based on the ranking range[8] where "Unenhanced query" and "Enhanced queries" refer to the contexts (i) and (ii), respectively. In the context (i), most rankings fall within rank 1 to 5 or above 50. However, in the context (ii), the prevalence within ranks 1 to 10 increases significantly, and there is a notable reduction in rankings from 20 onwards. It indicates that using LLMs (such as GPT-4o) can reduce noise from the original queries.

Tab. 2 shows results for the 15 final round questions based on their rankings for each hint level. In the final round, with each ranking level (@1, @5, @10, @20, @50, @100)[9], we observed a gradual increase in the correct return rate as more information is provided, particularly with Hint 4. As a result, the more enriching the information provided through each hint, the better the results are. The full result is shown in `https://github.com/NghiaZun/AIthena_Statistic`.

[8] R Ranking: @$R1$: Rank $\in \{1\}$; @$R5$: Rank $\in [2,5]$; @$R10$: Rank $\in [6,10]$; @$R20$: Rank $\in [11,20]$; @$R50$: Rank $\in [21,50]$; @$R100$: Rank $\in [51,100]$, where Rank is an integer.

[9] @k: Rank $\in [1,k]$, where $k \in 1,5,10,20,50,100$ and Rank is an integer.

Table 1: Comparative evaluation of correct answer rates between unenhanced and enhanced query tested on 65 know-item search and question and answer know-item search queries

	@R1	**@R5**	**@R10**	**@R20**	**@R50**	**@R100**
Unenhanced queries	27.69	27.69	9.24	10.77	9.24	15.38
Enhanced queries	36.92	33.85	10.77	9.23	3.08	6.15

Table 2: Comparative evaluation of system performance for 15 known-item searches in the final of the competition, when the number of hints increases.

Hint	**@1**	**@5**	**@10**	**@20**	**@50**	**@100**
H1	22.13	52.45	54.76	69.34	73.12	85.47
H2	31.84	64.29	75.62	82.36	87.48	90.20
H3	46.75	62.88	72.65	80.01	88.15	92.45
H4	51.45	69.78	78.12	86.21	91.60	94.32

6 Conclusion and Future work

In this paper, we presented an interactive video content retrieval system designed to efficiently handle multimodal queries. By leveraging large models such as CLIP [3], GPT-4o [14], Whisper [4], and PaddleOCR [13], our system provides robust search capabilities through features like semantic image search, OCR-based queries, voice-based search, AI-powered query sketching, and query standardization. The compact web interface ensures a seamless user experience, making complex queries more accessible.

Despite these achievements, several areas for improvement remain. In future work, we will focus on further optimizing query response times, ensuring minimal latency during high-load scenarios. Fine-tuning models to handle ambiguous and noisy inputs more effectively, especially with complex audio and reversed text, will also be a priority. Moreover, we aim to enhance query standardization techniques to improve consistency across diverse input formats. Lastly, expanding the system's scalability through distributed cloud deployment is another direction to accommodate even larger datasets and simultaneous queries.

Acknowledgement

This research is supported by research funding from Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh City.

References

1. Charles Adjetey and Kofi Sarpong Adu-Manu. Content-based image retrieval using tesseract ocr engine and levenshtein algorithm. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 2021.
2. Ahmed Alateeq et al. Voxento 4.0: A more flexible visualisation and control for lifelogs. *Proceedings of the 6th Annual ACM Lifelog Search Challenge (LSC '23)*, 2023.
3. Alec Radford et al. Learning transferable visual models from natural language supervision. *International conference on machine learning*, 2021.
4. Alec Radford et al. Robust speech recognition via large-scale weak supervision. 2022.
5. Jeff Johnson et al. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.
6. Klaus Schoeffmann et al. lifexplore at the lifelog search challenge 2023. *Proceedings of the 6th Annual ACM Lifelog Search Challenge (LSC '23)*, 2023.
7. Ly-Duyen Tran et al. E-myscéal: Embedding-based interactive lifelog retrieval system for lsc'22. *Proceedings of the 5th Annual ACM Lifelog Search Challenge (LSC'22)*, 2022.
8. MinhTriet Tran et al. Lifeinsight: An interactive lifelog retrieval system with comprehensive spatial insights and query assistance. *Proceedings of the 6th Annual ACM Lifelog Search Challenge (LSC '23)*, 2023.
9. Naushad Alam et al. Memento 3.0: An enhanced lifelog search engine for lsc'23. *Proceedings of the 6th Annual ACM Lifelog Search Challenge (LSC '23)*, 2023.
10. Premanand P. Ghadekar et al. Sentence meaning similarity detector using faiss. *7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, 2023.
11. Raghav Bansal et al. Blur image detection using laplacian operator and opencv. *International Conference System Modeling & Advancement in Research Trends (SMART)*, 2016.
12. Yaoshu Wang et al. Efficient approximate entity matching using jaro-winkler distance. *Web Information Systems Engineering*, 2017.
13. Yuning Du et al. Pp-ocr: A practical ultra lightweight ocr system. *arxiv*, 2020.
14. OpenAI. Gpt-4 technical report. 2023.
15. JTomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. 2020.