# MAVEN: Video Retrieval System using A Multi-Agent Visual Exploration Network

Ngo Quang Duc[1] *, Nguyen Hai Long[1], Tran Le Huy[2], and Vu Tung Linh[1]

[1] Ha Noi University of Science and Technology, Ha Noi, Vietnam
[2] University of Economics Ho Chi Minh City, Ho Chi Minh City, Vietnam
Email: duc.nq213697@sis.hust.edu.vn, long.nh24911@sis.hust.edu.vn, linh.vt210523@sis.hust.edu.vn, tranhuy2000com@gmail.com

**Abstract** Effective video retrieval systems are essential as video data grows across various fields. Traditionally, these systems rely on OCR, object detection, color extraction, and audio analysis. Current approaches like CLIP bridge the text-image embeddings gap for search, but they often lack contextual depth for complex, multi-frame searches. We propose a solution that integrates traditional methods and CLIP with advanced language models and prompting techniques for image captioning, extracting rich information from individual frames. Our system includes an Agent that automates searches, classifies queries, generates prompts, and verifies results, improving search accuracy while reducing user effort. Additional features like temporal search, video previews, and frame filtering further enhance the user experience. This comprehensive approach provides a powerful toolkit for achieving more accurate and efficient video search results, addressing the growing complexity of video data retrieval across various domains.

**Keywords:** CLIP · LLMs · Image Captioning · Video Retrieval · Agent

## 1 Introduction

The exponential expansion of multimedia data, especially video makes effective video retrieval systems indispensable for users to rapidly find pertinent materials. Content-based video retrieval has attracted a lot of interest as users expect to see particular frames from extensive collections of videos based on textual input. The demand for faster and more exact query results rises, especially with recent developments in multimodal models that span the gap between textual and visual data.

Among these developments, Contrastive Language-Image Pretraining [10] has been shown to be a strong model for precisely matching images and text [8]. While CLIP enables more efficient retrieval tasks by incorporating both modalities into a shared vector space, it also suffers from capturing detailed

* Corresponding author

information for challenging searches. Our approach is to extract rich semantic information that CLIP by itself cannot detect by using Large Language Models (LLMs) to create captions for video frames.

We also enhance the search process using an agent that automates and improves the retrieval flow. Providing users with a simple and quick retrieval experience, the Agent identifies user searches, runs searches utilizing CLIP and captioning techniques, and evaluates the obtained results for relevance and accuracy. This totally automated system increases accuracy and drastically lowers hand work. Further improving the user experience is our system's integration of sophisticated features such temporal search, video previews, and the capacity to eliminate extraneous frames.

In Ho Chi Minh City AI Challenge 2024, with a similar format to the Lifelog Search Challenge (LSC) [5] [6] and Video Browser Showdown (VBS), about 200 teams and nearly 1,000 participants were tasked with retrieving data from a large dataset of news videos. The performance of our system shows that it can efficiently and with accuracy manage big-scale video retrieval chores. In this paper, we will introduce our solution and how our system works.

## 2 Related Work

In our everyday lives, data such as images, events, and video information play a crucial role in helping us recall past knowledge. The increasing dependence on data retrieval has promoted the development of information retrieval systems globally.

Several previous approaches have been developed to create efficient retrieval systems. SIFT and SURF in Image Search [2] uses feature extraction for visual comparison but struggles with abstract semantics and large datasets. Image Recognition with HSV Color Space [9] focuses on color-based searches but faces reliability issues in varying lighting. Methods like [1][3] address texture and text extraction but are limited by complex backgrounds or lack of text. While effective in certain contexts, these approaches generally fall short in handling detailed, high-level semantic information.

The release of the CLIP [8] [10] model by OpenAI introduced a powerful method for embedding images and text into a shared vector space, enabling highly efficient retrieval tasks. Its handling of abstract, high-level semantics overcomes previous retrieval methods' limitations. We improve its power by augmenting it with FAISS-based search functionality [7]. Despite its high performance, it still faces limitations in complex scenes with detailed information. In this paper, we improve the search process by leveraging Large Language Models to generate captions for images, which are then embedded into a vector space for more effective retrieval. This approach dramatically enhances retrieval performance in complex scenes, especially those containing multiple objects, text, or contextual elements.

Despite improvements, manual search systems still require users to validate and filter results themselves, making them less efficient for complex queries and

large numbers of displayed images. To overcome these drawbacks, we implement an Agent that automates the search process by classifying user queries, generating optimized prompts, and executing searches with minimal user intervention. The agent also validates the relevance of the retrieved results and explains each image's relevance based on the query, streamlining the process and improving accuracy while reducing manual effort.

## 3 Video Preprocessing

In this section, we present our strategy for processing the provided raw videos into usable data for dataset construction. The TransNet V2 [11] model is a powerful deep-learning model designed for video boundary detection, capable of accurately identifying transition points between scenes. We use it to transform long videos into a list of short scenes in the initial stage of our retrieval system pipeline. After processing, we have a series of segments with the common format [a, b] where a is the start frame and b is the end frame. We extract two frames for each segment using the formula below:

$$k_i \in \left\{ a + (b-a) \times \left( \frac{2i-2}{3} \right) \mid i \in [1,2] \right\} \tag{1}$$

## 4 Feature Extraction

### 4.1 Visual Embedding via CLIP

Previously, before the release of multimodal feature extraction, many retrieval systems face a lot of limitations, such as a reliance on low-level features that fail to capture high-level semantics and suffer from reduced effectiveness in dealing with complex queries. At the same time, computational complexity and limited scalability hinder their performance on larger datasets. The CLIP model's abilities allow it to encode visual and textual information into a shared vector embedding space for more intuitive and accurate retrieval, enabling it to handle diverse content types and complex queries with greater efficiency and scalability. In our retrieval system, we employ the CLIP model for extracting visual embeddings with a specific version of ViT/B16 with vector embedding of 512 dimensions.

To complement the CLIP model in our retrieval system, we use FAISS, designed for efficient similarity search, especially in large-scale datasets. FAISS is optimized for fast nearest neighbor search, enabling it to efficiently handle high-dimensional vector embeddings like those generated by CLIP's ViT/B16 model. With CLIP and FAISS, we can easily and quickly retrieve the most relevant images by comparing them in a scalable way.

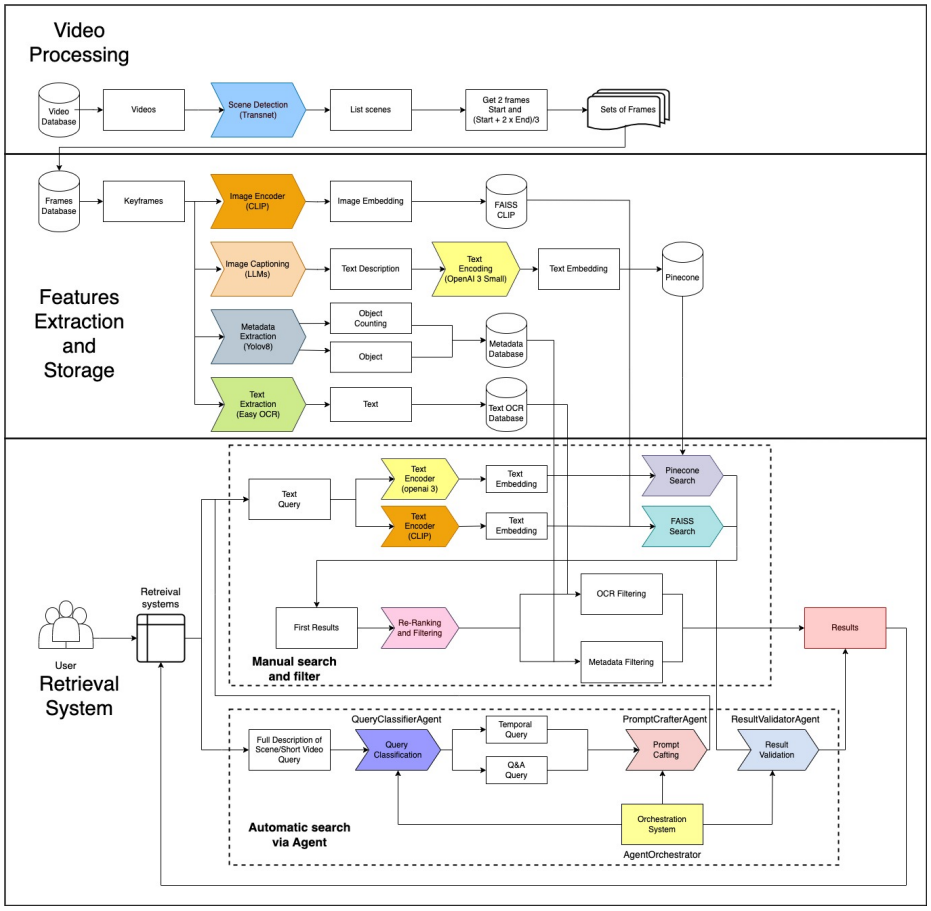


**Figure 1.** Our video retrieval pipeline consists of three stages: Video Processing, Feature Extraction and Storage, and Retrieval System. In the first stage, scenes are detected by TransnetV2. Then, we extract key frames, pass them through LLMs to generate captions for each image. Those frames are embedded using CLIP and indexed using FAISS while captions are embedded using OpenAI's text-embedding-3-small and stored on a Pinecone index. Information such as Text, Object, and object count are also extracted using specialized models (EasyOCR, YOLOv8) for filtering purposes. In addition, we also optimize, enhance, and automate the search process by deploying an agent that can perform the following functions: query classification, optimal prompt generation, and result validation.

### 4.2 Captioning and Semantic Embedding via LLM

To address CLIP's limitations, we implement an advanced image captioning system using LLMs to generate rich, detailed descriptions of each frame. This approach ensures robust performance and high flexibility.

We employ a carefully crafted prompt instructing the LLM to analyze the image and provide a structured description. The prompt is designed with a specific JSON structure, incorporating multiple fields that build upon each other. This design simulates a Chain of Thought effect, allowing the LLM to analyze the image carefully before providing the final caption.

This granular approach serves two purposes: it encourages the LLM to "think" about each aspect of the image and allows for a pre-filtering process. Given that our dataset includes media from news channels, the prompt design helps identify and separate consistent but irrelevant elements like ticker text, time displays, and television logos from the main content of the scene. The resulting caption is produced in a JSON format, reflecting the structure of the prompt and allowing for easy parsing and indexing of specific information.

After caption generation, we use OpenAI's text-embedding-3-small model to convert the Vietnamese captions into 1536-dimensional vector embeddings. These embeddings are then indexed in a Pinecone serverless vector database for efficient similarity search.

### 4.3 Metadata Information

Finally, to create a method to improve query functionality through filter search, we extract metadata information, including text, object, and color, by using multi-models.

**Text and Object Metadata Extraction** We use EasyOCR, which combines CRAFT [4] and a deep learning model, to extract text from frames. Additionally, YOLOv8 [12] powers object and object count filters for efficient detection, classification, and counting. The extracted text, object types, and counts are used as metadata, enhancing search filters and improving retrieval accuracy, particularly for queries involving text or object presence and quantity.

## 5 Retrieval System Overview

### 5.1 Manual Search System

Our system provides two powerful search methods: CLIP and Caption Search. While the CLIP model excels in scene retrieval for frames with simple descriptions and distinct objects. With Caption Search, each frame is first captioned using a captioning model. These detailed captions are then embedded and used for similarity search. This framework significantly improves retrieval performance for scenes with intricate content, including those with multiple objects, text, or context that the captioning model's OCR capabilities can capture.

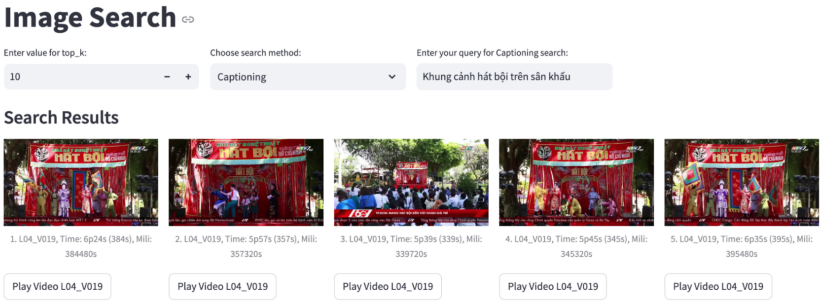


**Figure 2.** Caption Search demonstrates effectiveness when querying scene with textual details. Users can easily find scenes about Hat Boi - a traditional art form in Vietnam.

The query process works as follows. When accessing the system, users first set up the top-k relevant images, select a search method (either CAPTION or CLIP), and enter a text query. If they choose the CAPTION method, the text is embedded using OpenAI 3 text embedding Small, converting it into a vector that represents its features. Alternatively, if the CLIP method is chosen, the text is embedded into a vector embedding space using CLIP ViT/B16. This text embedding is then compared to the stored frame embedding vectors using a cosine similarity to find relevant images. The system retrieves the top-K results based on the highest scores and displays the images on the user interface.

### 5.2 Agent Augmented and Automated Search

**Overview** Manual searching, though user-controlled, is inefficient for complex or large-scale queries, highlighting the need for agent-driven automation. Advances in Large Language Models (LLMs) with enhanced reasoning capabilities enable automated search processes that streamline tasks and expand search capabilities.

Our agent-based system features four components: QueryClassifierAgent, PromptCrafterAgent, ResultValidatorAgent, and AgentOrchestrator. These agents, guided by refined prompts and JSON-formatted outputs, work sequentially to analyze user queries and deliver precise results. This structured approach ensures consistency and efficient task execution.

A key element is a well-crafted prompt instructing the AI assistant to analyze and categorize user queries for advanced image retrieval, breaking them into scenes or identifying time-series aspects as needed. The use of structured outputs ensures seamless communication between system components, exemplifying the clarity and focus of our methodology.

**Workflow** The agent-based workflow (Fig. 3) is managed by the Agent Orchestrator, a programmatic component coordinating interactions among LLM-based agents through an asynchronous pipeline. Key steps include:

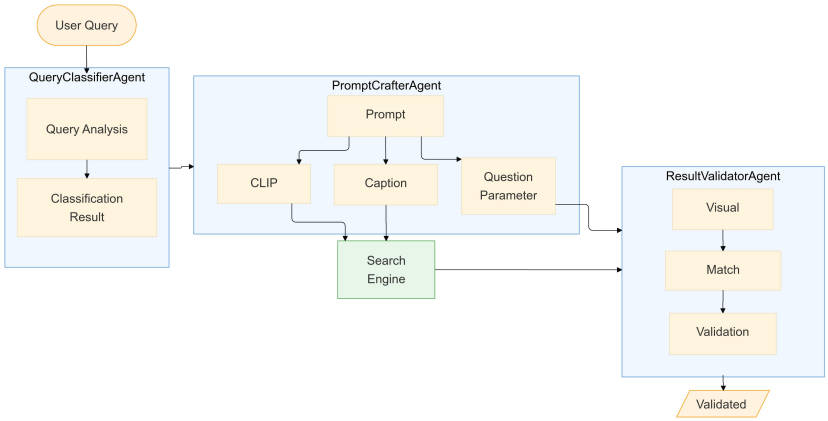


**Figure 3.** Our agent workflow involves analyzing user queries, generating prompts, and validating results before providing the final output.

The **QueryClassifierAgent** segments user input into temporal queries, questions, or scenes. For example, "Show me a car stopping at a red light, then turning right" is segmented into two scenes for precise processing. The **PromptCrafterAgent** generates tailored prompts. For CLIP searches, it focuses on visual elements, e.g., "car stopping at traffic signal". For caption searches, it integrates textual elements, e.g., "Xe ô tô dừng lại tại đèn đỏ có chữ 'STOP'. "The **ResultValidatorAgent** evaluates retrieved images with a vision-enabled LLM, categorizing them as "Exact Match", "Near Match", "Weak Match", or "No Match" and assigning confidence scores (0.0–1.0).

For temporal queries, the system processes scenes sequentially, narrowing subsequent searches to a limited frame range for efficiency. For example, a query on a math lecture slide (Fig. 4) returned an exact match (confidence 1.00) where the agent inferred "OA = 10" from visible formulas, demonstrating the system's ability to process complex, multi-step queries.

### 5.3 Graphical User Interface

**Re-ranking using filters** After the system retrieves and displays the top-k images most similar to the user's search query, a range of filtering options are available to refine further the results based on specific characteristics.

*Filtering Images Using OCR, Object Detection, and Object Counting* After the top-k images are shown on the user interface (UI), users can apply various filters to refine results quickly. The OCR Filter allows users to keep only images containing text they believe will be present, focusing on specific phrases or keywords, which is particularly useful for text-heavy images such as signs or documents (Fig. 6). Additionally, Object Detection and Object Counting filters

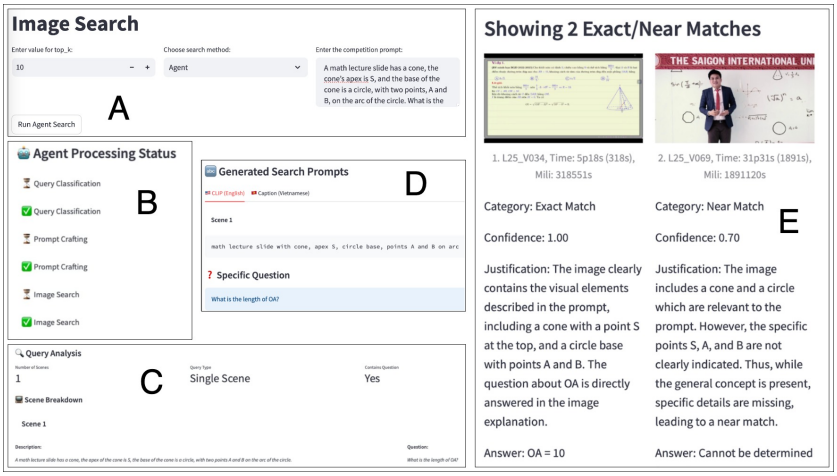


**Figure 4.** The agent search workflow consists of five key parts: (A) the user sets up the top-k relevant images, selects Agent Search, and enters query details. (B) The agent processing status panel provides feedback through visual indicators. (C) The QueryClassifierAgent's analysis is shown, including scene detection and question recognition, while (D) the PromptCrafterAgent's output is displayed in English and Vietnamese. Finally, (E) the result visualization area presents matched images with information.

enable users to refine search results based on the presence or quantity of specific objects, such as "3 people" or "2 cars." These features streamline the search process by narrowing results to only relevant images, saving time and effort.

**Temporal Search** During the retrieval process, users need tools to confirm the accuracy of their chosen answer (Fig. 7). Sometimes, they might question the validity of their selection, especially when dealing with queries involving multiple complex consecutive scenes. In our system, the Temporal Search function allows users to view five frames before and five after the selected frames. This feature provides an overview of what is happening immediately around the chosen scene, helping users confirm the accuracy of the selection by comparing scenes before and after the selected frame.

**Short Video Preview** Similar to the Temporal Search feature, it enables users to have more information about the context of a selected frame through a short video clip instead of surrounding static frames (Fig. 7). By playing a video segment that allows users to watch 60 seconds before and after the selected frame, users can easily capture the context around their point of interest. Short Video Preview provides a dynamic view of the action within a longer timespan. This feature enhances confidence in their decisions, making evaluating detailed scenarios directly from the surrounding video content easier.

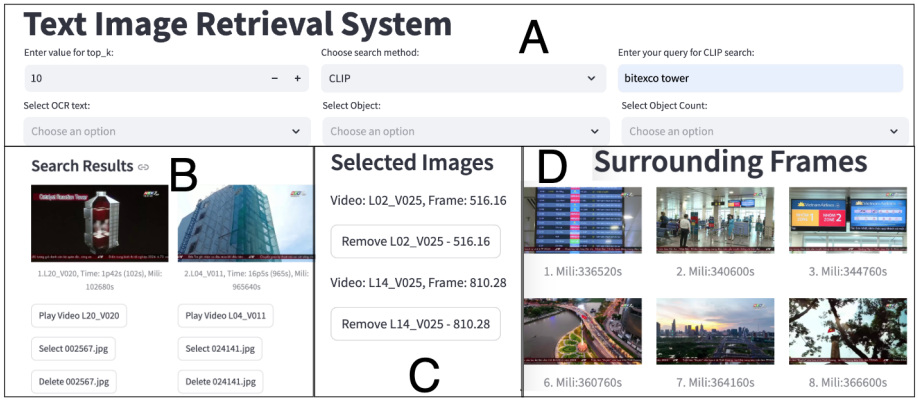


**Figure 5.** Our user interface (UI) system has 4 main components: Part A allows users to set up top-k relevant images, select search methods, and apply filters for faster searching; Part B displays the top-k images with options to watch a short video preview and select or delete frames; Part C shows user-selected images with removal options; and Part D provides surrounding frames for context, showing 5 frames before and after the selected image to clarify the sequence of events.

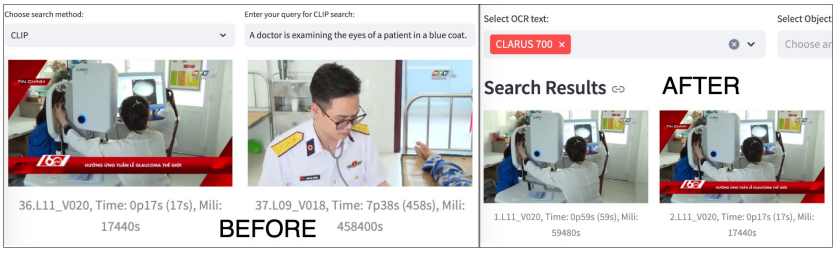


**Figure 6.** In the "Before" section, the correct result appears at rank 36 among irrelevant images, while in the "After" section, applying OCR filters promotes the correct result to ranks 1 and 2 by focusing on the "CLARUS 700" text.

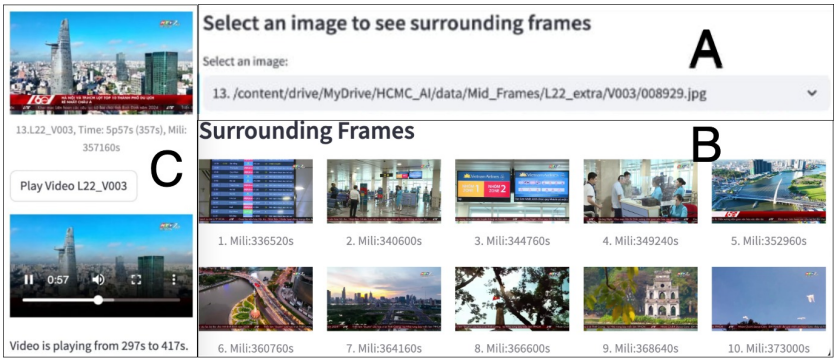


**Figure 7.** Our system provides many functions to support users in searching. In part A, users can select a photo, then the system will display 5 before and after photos of that photo (Part B). In part C, under each photo there will be a Play Video button, allowing users to watch a short video around that photo.

**Dynamic Image Management: Delete and Return** On the user interface, our system provides the Delete and Return Function, which allow users to manage the display of images dynamically while retrieving videos. If the retrieved images are not helpful or relevant to the text query, users can delete them temporarily in order to have more UI space for new images. These deleted images will not appear on the user's UI until they click the Return All button.

## 6 Conclusion

In summary, we have presented a highly efficient video retrieval system integrating CLIP and LLMs. Our system handles complex queries, providing detailed, highly contextually aware search results through the combination of LLMs and user interface support functions. In addition, the integration of Agent automates the entire process. Furthermore, the system's ability to retrieve information based on complex scenes, objects, and texts highlights its versatility in a variety of applications.

However, our system faces challenges in handling non-linear or reverse-time queries and occasional 'hallucinations'. A notable challenge occurs in question-based queries where the retrieved frames don't contain the exact information needed. In such cases, the system may hallucinate answers, attempting to extract information that isn't actually present in the frame. Developing more advanced authentication methods will help ensuring more accurate and reliable responses.

Our current implementation limits the agent to checking exactly 5 frames in the future for time queries due to a hard-coded parameter, which can cause the agent to miss relevant content outside this range. Future versions could allow for more adaptive time navigation by allowing the agent to adjust its window size, which would allow for checking frames in both directions. These improvements in the future would extend the system's capability.

## 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*, 12(7), 2021.
2. Nouman Ali, Khalid Bashir Bajwa, Robert Sablatnig, Savvas A Chatzichristofis, Zeshan Iqbal, Muhammad Rashid, and Hafiz Adnan Habib. A novel image retrieval based on visual words integration of sift and surf. *PloS one*, 11(6):e0157428, 2016.
3. MM AlyanNezhadi, H Qazanfari, A Ajam, and Z Amiri. Content-based image retrieval considering colour difference histogram of image texture and edge orientation. *International journal of Engineering*, 33(5):949–958, 2020.
4. Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374, 2019.
5. Cathal Gurrin, Björn ór Jónsson, Klaus Schöffmann, Duc-Tien Dang-Nguyen, Jakub Lokoč, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Graham Healy. Introduction to the fourth annual lifelog search challenge, lsc'21. ICMR '21, page 690–691, New York, NY, USA, 2021. Association for Computing Machinery.
6. Cathal Gurrin, Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn ór Jónsson, Jakub Lokoč, Wolfgang Hürst, Minh-Triet Tran, and Klaus Schöffmann. Introduction to the third annual lifelog search challenge (lsc'20). ICMR '20, page 584–585, New York, NY, USA, 2020. Association for Computing Machinery.
7. Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.
8. Jakub Lokoč, Zuzana Vopálková, Patrik Dokoupil, and Ladislav Peška. Video search with clip and interactive text query reformulation. In *International Conference on Multimedia Modeling*, pages 628–633. Springer, 2023.
9. Hamed Qazanfari, Hamid Hassanpour, and Kazem Qazanfari. Content-based image retrieval using hsv color space features. *International Journal of Computer and Information Engineering*, 13(10):533–541, 2019.
10. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
11. Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020.
12. Rejin Varghese and M Sambath. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6. IEEE, 2024.