MÔ TẢ GIẢI PHÁP

I. Dữ liệu được sử dụng

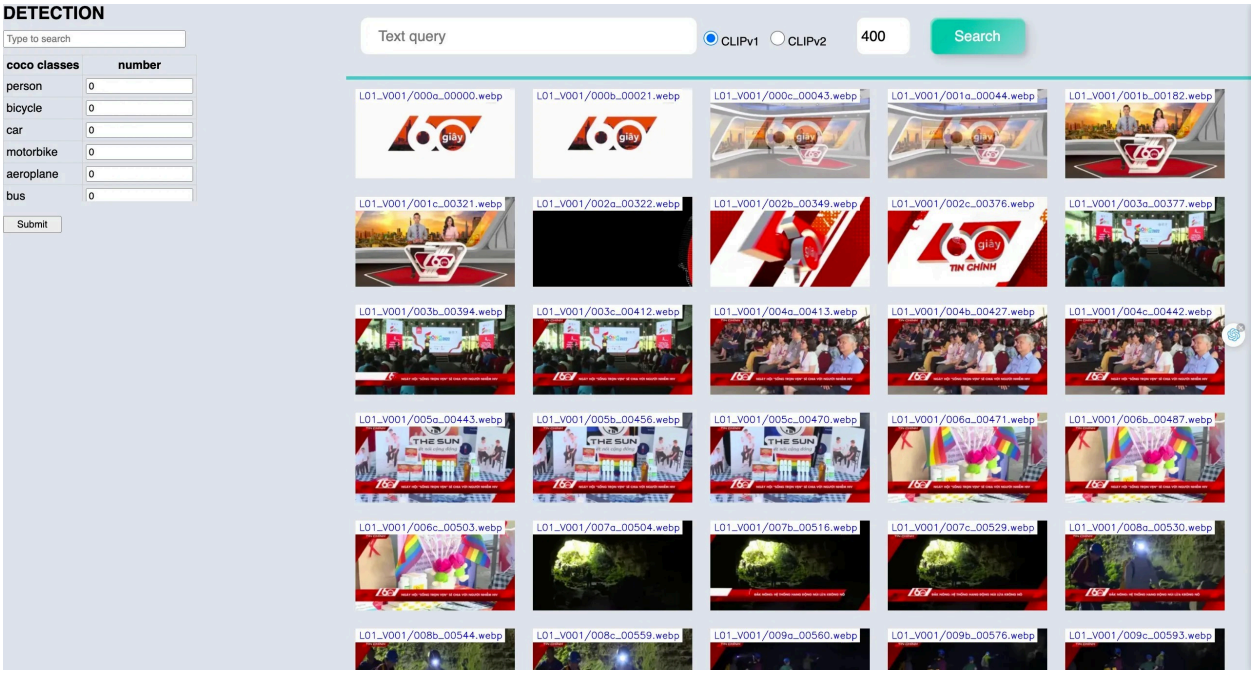
Dữ liệu được sử dụng bao gồm:

1. 36 List Video mà ban tổ chức cung cấp, là dữ liệu chính.
2. Các tập tin metadata mà ban tổ chức cung cấp, được sử dụng để thẩm định lại kết quả trước khi nộp.

II. Các công nghệ được sử dụng để xây dựng hệ thống

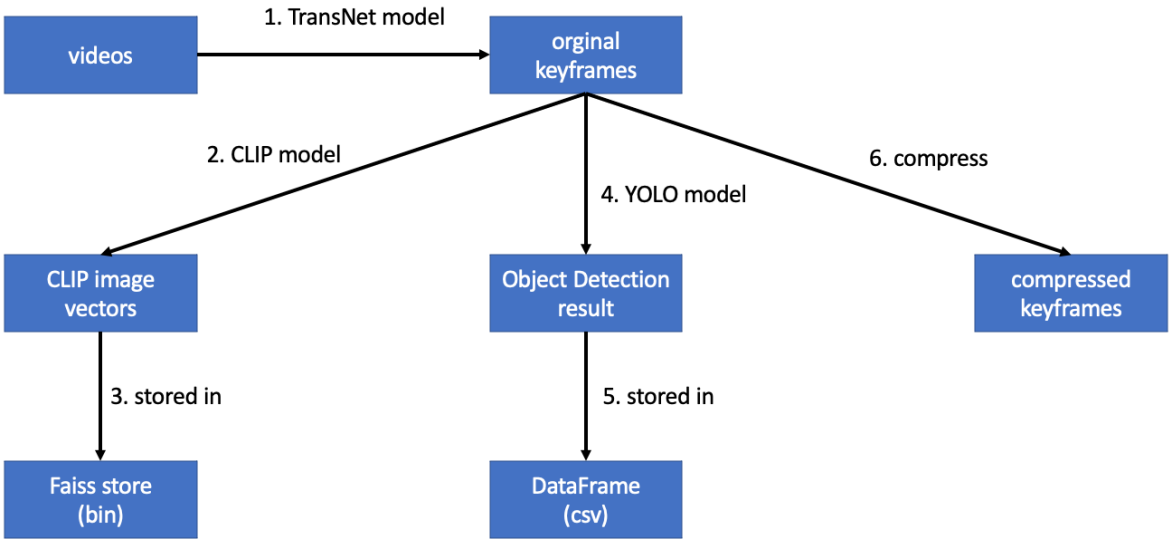
Hệ thống và giao diện tương tác được xây dựng trên nền tảng Web.

1. Backend sử dụng framework Flask Python.
2. Frontend được xây dựng bằng các ngôn ngữ HTML, CSS, JS.



III. Quá trình xử lý dữ liệu và xây dựng kho lưu trữ

Sơ đồ mô tả luồng xử lý dữ liệu và cách dữ liệu được lưu trữ



Các mô hình được sử dụng:

1. TransNet: https://github.com/soCzech/TransNetV2/
2. Official CLIP OpenAI: https://github.com/openai/CLIP
3. CLIPv2: https://github.com/mlfoundations/open_clip
4. YOLOv8: https://github.com/ultralytics/ultralytics

Ở bước 1, mỗi một video sẽ được sử dụng làm đầu vào cho mô hình TransNet, để trích xuất ra các keyframes quan trọng của video. Mô hình TransNet nhận đầu vào là 1 video, có tác dụng phân chia video thành các cảnh (có thể được gọi là scene hoặc segment), mỗi cảnh quay được đánh dấu bằng frame đầu frame cuối của cảnh đó. Các frame trong một cảnh quay sẽ có nội dung khá giống nhau. Vậy nên, nhóm lựa chọn lấy ra 3 frames ở vị trí lần lượt là đầu cảnh, giữa cảnh, cuối cảnh, vừa để tránh phải lưu trữ quá nhiều ảnh giống nhau, tiết kiệm tài nguyên lưu trữ, vừa đảm bảo độ chính xác, không bỏ lỡ bất kỳ chi tiết quan trọng nào trong video. Các ảnh thu được theo cách như vậy được gọi là original keyframes.

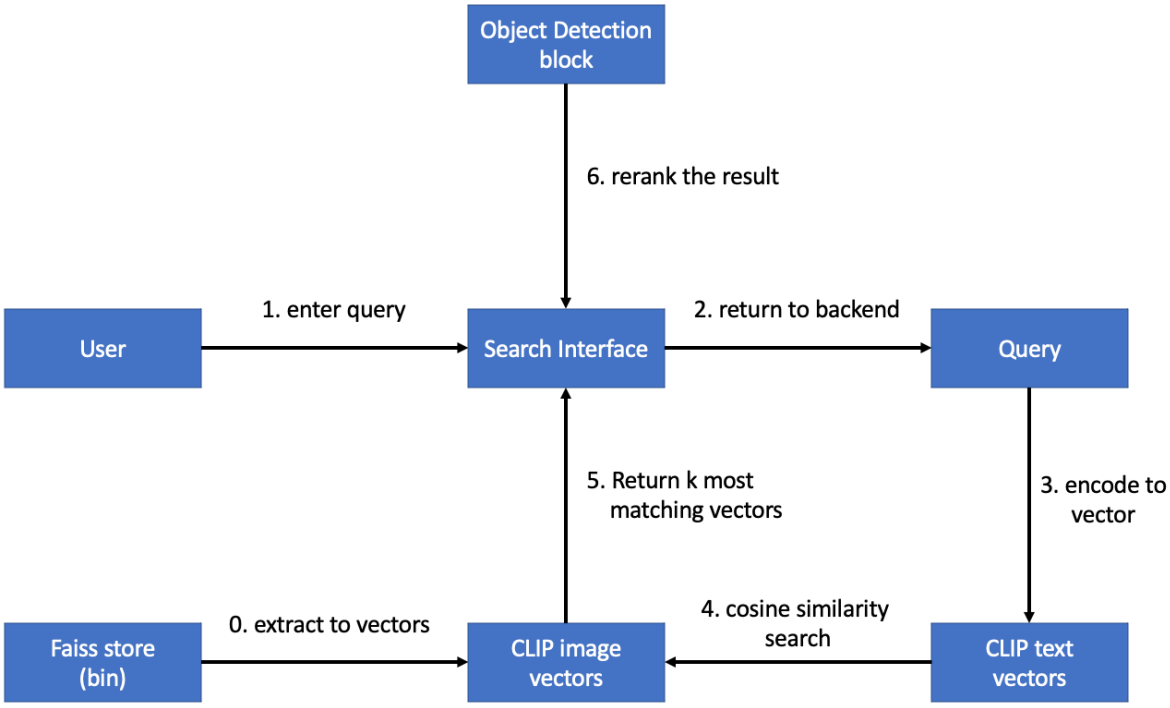
Sau đó, tại bước 2, original keyframes được sử dụng để làm đầu vào cho mô hình CLIP. Mô hình này có tác dụng mã hóa các ảnh hoặc text về dạng vector trong cùng một không gian vector. Kết quả thu được sau quá trình CLIP inference là tập hợp các vectors 512 chiều, với mỗi vector đại diện cho một original keyframe. Các vector được lưu trữ trong kho lưu trữ Faiss dưới dạng file bin (bước 3). Về bản chất, Faiss là một thư viện cho phép các nhà phát triển tìm kiếm một số lượng k vectors có độ tương đồng cao nhất với một vector được chọn trước đó.

Ngoài ra, tại bước 4, original keyframes cũng được sử dụng để làm đầu vào cho mô hình YOLOv8. Mô hình này trả về các vị trí và class của các đối tượng có trong ảnh. Được sử dụng như một mô hình phụ trợ để tăng hiệu quả tìm kiếm cho mô hình chính. Kết quả của quá trình YOLO inference sau đó được hậu xử lý, thống kê số lượng mỗi class của mỗi ảnh và lưu trong một DataFrame dưới dạng file csv (bước 5).

Cuối cùng, ở bước 6, để nâng cao hiệu quả lưu trữ, original keyframes sẽ được tiến hành nén lại. Mỗi ảnh ban đầu có kích thước 1280x720, được lưu trữ ở dạng file jpg, sẽ được nén thành một ảnh với kích thước 255x144, được lưu trữ ở dạng file webp. Việc lưu trữ như vậy vẫn đảm bảo không khiến ảnh bị mất quá nhiều nội dung.

IV. Hệ thống hoàn chỉnh và luồng xử lý

Sơ đồ mô tả hệ thống hoàn chỉnh và luồng xử lý dữ liệu



Khi khởi chạy hệ thống, Faiss Store giải nén ra các CLIP image vectors được lưu trong nó để tiện xử lý (bước 0).

Ở bước 1, người dùng nhập vào câu truy vấn trên giao diện tìm kiếm. Sau đó, câu truy vấn sẽ được chuyển về Backend của hệ thống để tiến hành xử lý (bước 2). Tại Backend, câu truy vấn sẽ được mã hóa hóa thành dạng vector nằm trên cùng một không gian vector của mô hình CLIP (bước 3).

Ở bước 4, CLIP text vectors sẽ được tiến hành so sánh với các CLIP image vectors của hệ thống. Và từ đó, trả về màn hình giao diện k CLIP image vectors có độ tương đồng cao nhất với câu query đã nhập (bước 5).

Ngoài ra, ở bước 6, khối Object Detection cũng có thể được sử dụng để sắp xếp lại các kết quả và hiển thị bổ sung trên giao diện tìm kiếm. Giúp quá trình tìm kiếm diễn ra nhanh và hiệu quả hơn