



fit@hcmus

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG - HCM
KHOA CÔNG NGHỆ THÔNG TIN

Báo Cáo về Các hệ thống truy vấn video bằng câu truy vấn sử dụng FiftyOne tích hợp Milvus hoặc Pinecone

THỰC HIỆN BỞI:

Nguyễn Thị Thu Duyên – 22C11005
Đặng Hoàng Minh Triết – 22C11048

Dưới sự hướng dẫn của:
PGS. TS. Trần Minh Triết
TS. Lê Trung Nghĩa
ThS. Đỗ Trọng Lẽ

LỜI CẢM ƠN

Chúng em xin bày tỏ lòng biết ơn chân thành đến PGS. TS. Trần Minh Triết, TS. Lê Trung Nghĩa và ThS. Đỗ Trọng Lễ đã dành thời gian, kiến thức và sự hỗ trợ để chúng em hoàn thành đồ án môn học này.

Trong suốt quá trình thực hiện đồ án, các thầy đã cung cấp cho chúng em những chỉ dẫn quan trọng và những gợi ý giá trị để chúng em có thể tiến hành nghiên cứu và phân tích một cách hiệu quả, đồng thời, cũng đã luôn sẵn sàng lắng nghe và giải đáp những thắc mắc của chúng em, giúp chúng em vượt qua các khó khăn và tiến bộ trong quá trình làm việc.

Chúng em cũng xin bày tỏ lòng biết ơn đến các thành viên khác trong lớp học đã chia sẻ ý kiến, góp ý và đóng góp xây dựng vào đồ án của chúng em.

Cuối cùng, chúng em xin chân thành cảm ơn gia đình, bạn bè và những người thân yêu đã luôn ủng hộ, động viên và tạo điều kiện thuận lợi để chúng em hoàn thành đồ án này.

Rất cảm kích vì sự hỗ trợ và đóng góp của mọi người, chúng em hy vọng rằng đồ án môn học này sẽ đáp ứng được mong đợi và mang lại giá trị thực tế.

Xin chân thành cảm ơn!

MỞ ĐẦU

Bài toán truy vấn dữ liệu video bằng văn bản, đặc biệt là mô tả sự kiện, đã trở thành một thách thức đáng kể trong lĩnh vực trí tuệ nhân tạo và xử lý dữ liệu đa phương tiện. Với sự bùng nổ của nội dung video trên Internet và nguồn dữ liệu ngày càng lớn, khả năng tìm kiếm và truy xuất thông tin từ video dựa trên mô tả văn bản đã trở nên cần thiết để tận dụng tri thức chứa đựng trong dữ liệu đa phương tiện này.

Mô tả sự kiện trong video là một yếu tố quan trọng để hiểu nội dung của nó. Tuy nhiên, việc tự động trích xuất thông tin từ video và hiểu mô tả văn bản về sự kiện trong đó là một thách thức phức tạp. Đây có thể là các sự kiện thể thao, diễn biến trong video an ninh, hoặc bất kỳ sự kiện nào có thể được mô tả bằng từ ngữ. Việc tự động truy vấn và trích xuất dữ liệu từ video dựa trên mô tả văn bản sẽ giúp cải thiện khả năng tìm kiếm, sắp xếp, và tận dụng nguồn dữ liệu video một cách hiệu quả hơn.

Trong đồ án này, chúng tôi giới thiệu một cách tiếp cận độc đáo để giải quyết bài toán này. Chúng tôi sử dụng FiftyOne, một công cụ mạnh mẽ cho việc xử lý và quản lý dữ liệu hình ảnh và video, để tính toán mức độ tương đồng giữa mô tả văn bản và hình ảnh trong video. Bằng cách này, chúng tôi có thể biểu diễn mô tả văn bản và hình ảnh dưới dạng vector, từ đó tạo cơ hội để thực hiện truy vấn nhanh chóng và hiệu quả.

Để hỗ trợ việc truy vấn và truy xuất dữ liệu dựa trên các vector này, chúng tôi tiến hành thử nghiệm tích hợp với Milvus và Pinecone như các cơ sở dữ liệu bên dưới nhằm so sánh những ưu khuyết điểm của hai hệ cơ sở dữ liệu này khi thực hiện các truy vấn. Sự kết hợp giữa FiftyOne và Milvus hoặc Fifty One và Pinecone giúp chúng tôi xây dựng hệ thống truy vấn video mạnh mẽ và hiệu quả, cho phép người dùng truy vấn dữ liệu video bằng văn bản mô tả sự kiện và tìm kiếm thông tin tương tự trong nguồn dữ liệu video lớn mà không cần tiêu tốn nhiều thời gian và công sức.

Chúng tôi hy vọng rằng đồ án này sẽ đóng góp vào việc giải quyết thách thức quan trọng trong việc tận dụng tri thức từ dữ liệu video và cung cấp một phương pháp tiếp cận cơ bản để truy vấn dữ liệu video bằng văn bản mô tả sự kiện.

Mục lục

LỜI CÁM ƠN	2
MỞ ĐẦU.....	3
MỤC LỤC	4
CHƯƠNG 1: GIỚI THIỆU	5
1.1. Mục tiêu	5
1.2. Công cụ tích hợp	6
CHƯƠNG 2: PHÂN TÍCH BÀI TOÁN	10
2.1. Mô tả bài toán.....	10
2.2. Các công trình liên quan.....	10
2.3. Cách tiếp cận	12
CHƯƠNG 3: TRIỂN KHAI	15
3.1 Dữ liệu triển khai	15
3.2 Tiền xử lý dữ liệu video	15
3.3 Trích xuất đặc trưng văn bản và hình ảnh dựa vào CLIP-ViT	17
3.4 Lưu trữ và cập nhật cơ sở dữ liệu.....	19
3.5 Truy vấn dữ liệu.....	20
CHƯƠNG 4: KẾT QUẢ THỬ NGHIỆM	22
4.1 Truy vấn và hiển thị kết quả trên FiftyOne_Milvus.....	22
4.2 Truy vấn và hiển thị kết quả trên FiftyOne_Pinecone	24
CHƯƠNG 5: KẾT LUẬN.....	26
CHƯƠNG 6: HƯỚNG PHÁT TRIỂN	27
TÀI LIỆU THAM KHẢO.....	28

Chương 1: GIỚI THIỆU

1.1. Mục tiêu

Mục tiêu chính của đồ án này là giải quyết bài toán truy vấn dữ liệu video bằng văn bản thông qua việc kết hợp ba yếu tố chính: FiftyOne, cơ sở dữ liệu (ở đây có thể là Milvus hoặc Pinecone), và mô hình CLIP-ViT. Sự hợp nhất này nhằm mang lại những giải pháp sáng tạo và hiệu quả cho việc quản lý và tìm kiếm dữ liệu video, làm cho việc truy xuất thông tin từ video trở nên dễ dàng và chính xác hơn. Dưới đây là một số mục tiêu cụ thể mà chúng tôi đặt ra cho sự kiện này:

Tích hợp hiệu quả giữa FiftyOne và Pinecone: Chúng tôi muốn đảm bảo tích hợp giữa FiftyOne, một công cụ mạnh mẽ cho việc quản lý và khám phá dữ liệu hình ảnh và video, và Pinecone là một nền tảng tìm kiếm vector cho phép tìm kiếm và gợi ý dựa trên nội dung. Sự tích hợp này sẽ giúp cho việc lưu trữ và quản lý dữ liệu video trở nên đơn giản và hiệu quả hơn.

Tối ưu hóa khả năng tích hợp giữa FiftyOne và Milvus: Để cải thiện hiệu suất và khả năng tìm kiếm dữ liệu đa phương tiện, FiftyOne cung cấp khả năng quản lý dữ liệu hình ảnh và video một cách dễ dàng, trong khi Milvus là một hệ thống tìm kiếm vector hiệu suất cao. Chúng tôi muốn thử nghiệm và đánh giá hiệu quả tích hợp giữa hai nền tảng này trong việc tối ưu hóa quá trình xử lý, lưu trữ và truy vấn dữ liệu dạng vector.

Hiểu biết dữ liệu video thông qua CLIP-ViT: Sử dụng mô hình CLIP-ViT, chúng tôi đặt ra mục tiêu xây dựng khả năng hiểu biết dữ liệu video dựa trên văn bản. Điều này có nghĩa là người dùng có thể truy vấn dữ liệu video bằng cách sử dụng văn bản mô tả sự kiện, mô hình CLIP-ViT sẽ giúp tìm kiếm và trả về các đoạn video phù hợp với mô tả đó.

Tích hợp và triển khai tiện ích thực tế: Mục tiêu của sự kiện là không chỉ nghiên cứu và phát triển các khái niệm và giải pháp mới, mà còn triển khai chúng vào các ứng dụng thực tế. Chúng tôi muốn đảm bảo rằng giải pháp được xây dựng từ sự kiện này có thể áp dụng trong nhiều ngữ cảnh khác nhau, từ quản lý thư viện video đến tìm kiếm nội dung đa phương tiện trực quan.

Đánh giá và chia sẻ kiến thức: Chúng tôi hy vọng rằng sự kiện này sẽ cung cấp một cơ hội để đánh giá hiệu suất của giải pháp và chia sẻ kiến thức về cách tích hợp các công cụ mạnh mẽ như FiftyOne, Milvus, Pinecone và CLIP-ViT để giải quyết bài toán truy vấn dữ liệu video bằng văn bản mô tả sự kiện.

1.2. Công cụ tích hợp

1.2.1 FiftyOne

FiftyOne [1] là một công cụ mạnh mẽ trong lĩnh vực quản lý và khám phá dữ liệu hình ảnh và video. Với mục tiêu cung cấp giải pháp toàn diện cho việc làm việc với dữ liệu thị giác máy tính, FiftyOne đã trở thành một công cụ không thể thiếu trong nhiều ứng dụng, từ nghiên cứu AI đến phát triển ứng dụng thực tế.

Quản lý dữ liệu linh hoạt: Một trong những điểm mạnh của FiftyOne là khả năng quản lý dữ liệu hình ảnh và video một cách linh hoạt. Người dùng có thể dễ dàng nhập dữ liệu từ nhiều nguồn khác nhau và tổ chức chúng vào các bộ dữ liệu có cấu trúc. Điều này giúp đảm bảo tính đầy đủ và chất lượng của dữ liệu trước khi tiến hành các tác vụ khám phá hoặc đào tạo mô hình.

Khám phá và biểu đồ hóa: FiftyOne cung cấp nhiều công cụ để khám phá và hiểu dữ liệu hình ảnh và video. Người dùng có thể duyệt qua dữ liệu, xem hình ảnh và video, và thậm chí vẽ các biểu đồ thống kê để phân tích phân bố của các lớp hoặc thuộc tính.

Kiểm tra dữ liệu và đánh giá mô hình: Công cụ này cho phép người dùng kiểm tra dữ liệu và đánh giá mô hình một cách toàn diện. Bằng cách tích hợp với các framework deep learning phổ biến như PyTorch và TensorFlow, FiftyOne cho phép bạn đánh giá kết quả của mô hình và tạo các ghi chú, nhận xét trực quan cho từng mẫu dữ liệu.

Tích hợp và mở rộng: FiftyOne được thiết kế với tính mở rộng cao, cho phép tích hợp với nhiều công cụ khác nhau và tùy chỉnh theo nhu cầu cụ thể của dự án. Điều này làm cho nó trở thành một công cụ phổ quát cho cả những nghiên cứu viên AI và những nhà phát triển ứng dụng.

Cộng đồng và hỗ trợ: FiftyOne có một cộng đồng người dùng đang phát triển mạnh mẽ và sẵn sàng hỗ trợ. Người dùng có thể tìm kiếm thông tin hữu ích, chia sẻ kiến thức, và đặt câu hỏi trên các diễn đàn trực tuyến và nhận được sự giúp đỡ từ cộng đồng.

Tóm lại, FiftyOne là một công cụ quan trọng cho quản lý và khám phá dữ liệu hình ảnh và video trong các dự án thị giác máy tính. Sự linh hoạt, tích hợp dễ dàng, và khả năng đánh giá mô hình là những đặc điểm quan trọng làm cho FiftyOne trở thành một công cụ ưa thích trong cộng đồng AI và phát triển ứng dụng.

1.2.2 Milvus

Milvus [2] là một hệ thống tìm kiếm vector mã nguồn mở, được thiết kế để hỗ trợ việc xử lý và tìm kiếm dữ liệu dạng vector một cách hiệu quả. Được phát triển bởi Zilliz, một công ty chuyên về trí tuệ nhân tạo và học máy, Milvus đã nhanh chóng trở thành một trong những lựa chọn hàng đầu cho các dự án và ứng dụng có yêu cầu cao về tìm kiếm và gợi ý dựa trên vector.

Một trong những điểm mạnh của Milvus là khả năng quản lý và tìm kiếm hàng triệu hoặc thậm chí hàng tỷ vector một cách nhanh chóng và hiệu quả. Hệ thống này được xây dựng để hỗ trợ nhiều loại dữ liệu, từ hình ảnh, âm thanh, văn bản đến dữ liệu số học. Có khả năng tích hợp với nhiều ngôn ngữ lập trình và các framework phổ biến như Python, Java, và C++, Milvus là lựa chọn lý tưởng cho các nhà phát triển ứng dụng đa dạng.

Milvus cung cấp nhiều tính năng mạnh mẽ bao gồm tìm kiếm vector gần giống, gợi ý dựa trên nội dung, tối ưu hóa hiệu suất truy vấn, và quản lý dữ liệu dạng vector theo cách linh hoạt và tiện lợi. Hệ thống này có thể được triển khai trên các môi trường on-premises hoặc đám mây, đáp ứng nhu cầu đa dạng của các doanh nghiệp và dự án.

Milvus đang là một công cụ quan trọng trong lĩnh vực trí tuệ nhân tạo, khoa học dữ liệu, và các ứng dụng liên quan đến tìm kiếm và gợi ý. Sự linh hoạt, hiệu suất và khả năng mở rộng của nó đã giúp cho nhiều dự án thành công trong việc xây dựng các ứng dụng dựa trên vector và tìm kiếm nâng cao.

1.2.3 Pinecone

Pinecone [2] là một nền tảng tìm kiếm vector hàng đầu, được xây dựng để giúp các doanh nghiệp và tổ chức tận dụng sức mạnh của tìm kiếm vector và gợi ý dựa trên nội dung. Nền tảng này đã nhanh chóng trở thành một công cụ quan trọng trong lĩnh vực trí tuệ nhân tạo và khoa học dữ liệu, giúp đẩy mạnh việc phân tích dữ liệu và cải thiện trải nghiệm người dùng.

Pinecone thừa hưởng những ưu điểm của tìm kiếm vector, một phương pháp mạnh mẽ để biểu diễn và tìm kiếm dữ liệu dưới dạng vector. Hệ thống này giúp đo lường độ tương đồng giữa các điểm dữ liệu một cách hiệu quả, cho phép tạo ra các ứng dụng như tìm kiếm hình ảnh dựa trên nội dung, gợi ý sản phẩm, và phân loại dữ liệu dựa trên nội dung.

Pinecone được thiết kế với hiệu suất cao và tính mở rộng là ưu tiên hàng đầu. Hệ thống này có khả năng xử lý hàng triệu truy vấn mỗi giây và hỗ trợ việc triển khai trên nhiều môi trường, bao gồm đám mây và on-premises. Điều này làm cho Pinecone trở thành lựa chọn lý tưởng cho các doanh nghiệp có nhu cầu về tìm kiếm vector và gợi ý dựa trên nội dung,

đặc biệt là trong các ứng dụng thương mại điện tử, dịch vụ streaming, và nhiều lĩnh vực khác.

Ngoài ra, Pinecone cung cấp các công cụ và API đáng tin cậy để quản lý dữ liệu vector, tối ưu hóa truy vấn, và tích hợp dễ dàng với các ứng dụng và dự án sử dụng các ngôn ngữ lập trình phổ biến. Pinecone đã giúp cho nhiều tổ chức nhanh chóng phát triển các ứng dụng sáng tạo và tối ưu hóa trải nghiệm người dùng thông qua tìm kiếm và gợi ý dựa trên nội dung đỉnh cao.

1.2.4 Mô hình CLIP-ViT

Mô hình CLIP-ViT [4] là một sáng tạo trong lĩnh vực học máy và thị giác máy tính, kết hợp giữa hai mô hình nổi tiếng là CLIP và Vision Transformer (ViT), để cải thiện khả năng hiểu và tương tác giữa văn bản và hình ảnh. Được phát triển bởi các nhà nghiên cứu hàng đầu tại OpenAI, mô hình này đã đạt được sự quan tâm lớn từ cộng đồng nghiên cứu và đã được triển khai trong nhiều ứng dụng thực tế.

Kết hợp văn bản và hình ảnh: Một trong những đặc điểm quan trọng của CLIP-ViT là khả năng kết hợp thông tin từ cả văn bản và hình ảnh. Điều này có nghĩa rằng mô hình có khả năng hiểu được mối liên hệ giữa hình ảnh và mô tả văn bản một cách tự nhiên. Điều này làm cho CLIP-ViT trở thành một công cụ mạnh mẽ cho các ứng dụng liên quan đến truy vấn và tìm kiếm dựa trên nội dung đa phương tiện.

Sử dụng kiến thức từ mô hình ViT: ViT là một trong những mô hình tiên tiến nhất cho việc xử lý hình ảnh. CLIP-ViT sử dụng kiến thức từ mô hình ViT để biểu diễn hình ảnh. Điều này cho phép mô hình học được các đặc trưng phức tạp từ hình ảnh và cải thiện khả năng phân loại và tìm kiếm.

Ứng dụng đa dạng: CLIP-ViT đã được triển khai trong nhiều ứng dụng đa dạng, bao gồm tìm kiếm hình ảnh và video, phân loại đa lớp, và trích xuất thông tin từ dữ liệu đa phương tiện. Mô hình này cung cấp một cơ hội để nghiên cứu và phát triển các ứng dụng mới trong lĩnh vực AI và thị giác máy tính.

Đào tạo và fine-tuning: CLIP-ViT cho phép đào tạo và tinh chỉnh trên nhiều nhiệm vụ khác nhau. Điều này làm cho nó trở thành một công cụ linh hoạt cho nghiên cứu và phát triển. Bạn có thể tùy chỉnh mô hình cho nhiệm vụ cụ thể của bạn và cải thiện hiệu suất của nó.

Tóm lại, CLIP-ViT là một sự kết hợp đầy triển vọng giữa các mô hình học máy hàng đầu để định nghĩa một tiêu chuẩn mới cho việc hiểu và tương tác giữa văn bản và hình ảnh.

**KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

Sự tích hợp thông tin đa dạng và khả năng mở rộng làm cho mô hình này trở thành một trong những công cụ quan trọng trong lĩnh vực trí tuệ nhân tạo và thị giác máy tính.

Chương 2: PHÂN TÍCH BÀI TOÁN

2.1. Mô tả bài toán

Bài toán truy vấn dữ liệu video bằng văn bản là một trong những thách thức quan trọng trong lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên. Đây là một ứng dụng quan trọng có nhiều ứng dụng thực tế, từ giám sát an ninh đến tìm kiếm video trên mạng.

Khi ta nói về truy vấn dữ liệu video bằng văn bản, ta đang ám chỉ việc tìm kiếm và trích xuất thông tin từ các đoạn video dựa trên mô tả văn bản của sự kiện trong video. Điều này bao gồm việc sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên để hiểu ý nghĩa của câu mô tả và sử dụng thị giác máy tính để phát hiện và trích xuất hình ảnh hoặc phân đoạn video liên quan.

Một ví dụ cụ thể về bài toán này là khi chúng ta muốn tìm kiếm trong cơ sở dữ liệu video để xác định tất cả các cảnh trong đó có sự kiện xác định. Để làm điều này, hệ thống trước hết phải hiểu được nghĩa của sự kiện đó trong ngôn ngữ tự nhiên. Sau đó, nó sẽ quét qua các video để xác định những phần nào chứa sự kiện này, có thể thông qua việc phát hiện hoặc hình ảnh đặc trưng của sự kiện.

Bài toán này đòi hỏi sự kết hợp giữa công nghệ xử lý ngôn ngữ tự nhiên, công nghệ thị giác máy tính, và khả năng trích xuất dữ liệu từ video. Điều này có thể giúp tổ chức dễ dàng tìm kiếm và truy xuất thông tin từ các nguồn video đồng thời giảm thời gian và công sức cần thiết so với việc thủ công xem từng đoạn video một.

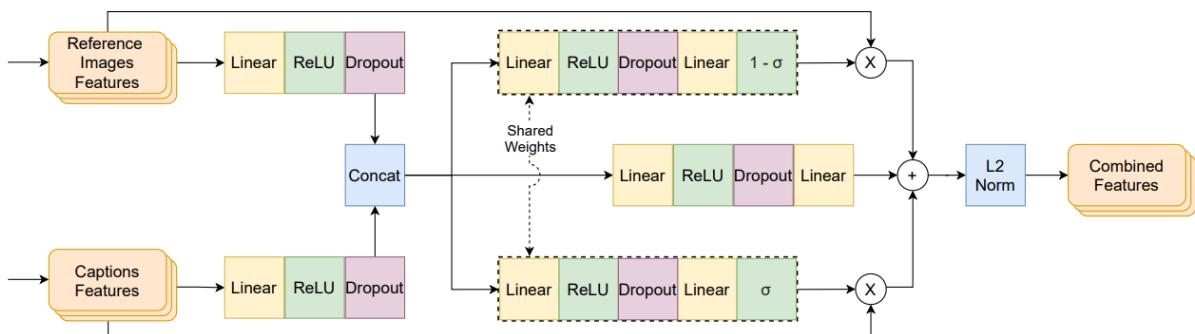
2.2. Các công trình liên quan

2.2.1. Effective conditioned and composed image retrieval combining CLIP-based features [3]

Công trình này dựa trên CLIP, tận dụng tiềm năng của nó cho việc truy vấn hình ảnh được điều kiện. Các phương pháp khác để học sự cân bằng giữa hình ảnh và văn bản đã được đề xuất trong các công trình nghiên cứu trước, sử dụng một kiến trúc hai bộ mã hóa và được đào tạo trên một bộ dữ liệu lớn với 1 tỷ cặp hình ảnh-văn bản. Một cách khác, phương pháp này tận dụng việc nén thông tin trái ngược, dẫn đến một quá trình tiết kiệm dữ liệu nhiều hơn, yêu cầu một tập dữ liệu đào tạo nhỏ hơn 133 lần so với CLIP.

Phương pháp này sử dụng tính năng CLIP để mã hóa cả hình ảnh và văn bản thành các tính năng trong không gian chung. Sau đó, nhiệm vụ là học cách chuyển đổi từ tính năng hình ảnh tham chiếu và văn bản đầu vào thành một tính năng kết hợp bao gồm thông tin đa

dạng và gần nhất có thể với hình ảnh mục tiêu trong không gian chung. Phần chuyển đổi này được gọi là hàm Kết hợp và được thiết kế bằng kiến trúc mạng nơ-ron được đào tạo để học chính xác hàm này. Mặc dù thiết kế mạng đơn giản, hệ thống đạt được kết quả đầu ngành trên hai bộ dữ liệu tiêu chuẩn thường được sử dụng, bao gồm bộ dữ liệu FashionIQ cho lĩnh vực thời trang và bộ dữ liệu CIRR cho nội dung tổng quát



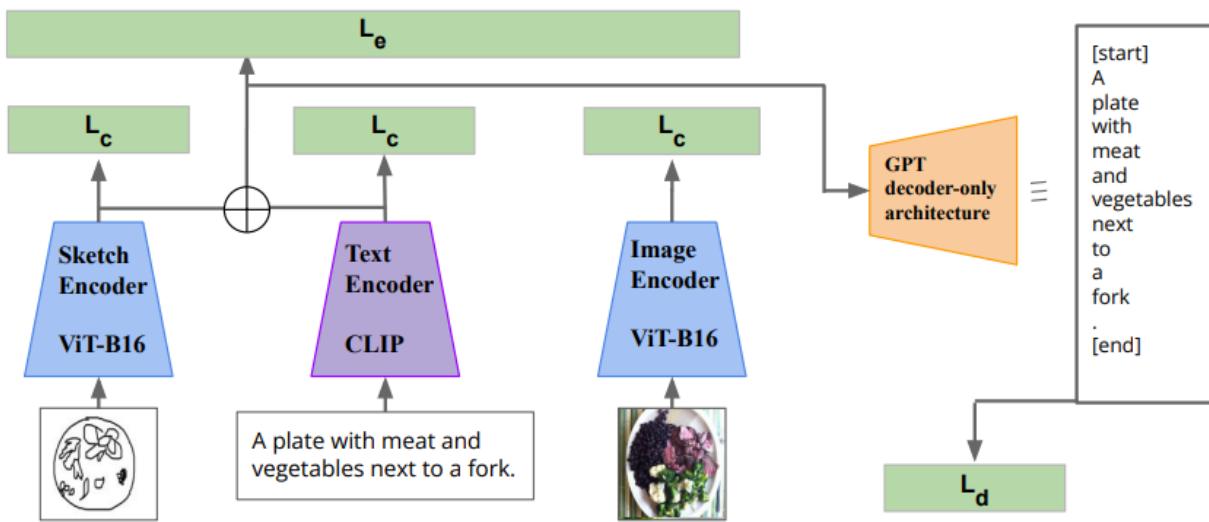
2.2.2. TASK-former (Text And SKetch transformer) [4]

Trong công trình nghiên cứu này giải quyết vấn đề về truy vấn hình ảnh bằng câu truy vấn kết hợp với bản phác thảo với bản phác họa (tùy chọn) được coi là bổ sung cho truy vấn văn bản để cung cấp thông tin thêm mà có thể khó biểu đạt bằng văn bản (ví dụ: vị trí của nhiều đối tượng, hình dạng đối tượng). Ở phương pháp này, nhấn mạnh việc không cần bắt buộc phải có bản phác thảo nhưng nếu có (kể cả khi bản vẽ có xấu) cũng sẽ thu hẹp phạm vi tìm kiếm của mô hình đáng kể

Quy trình được tóm tắt trong hình dưới đây. Mỗi truy vấn đầu vào bao gồm:

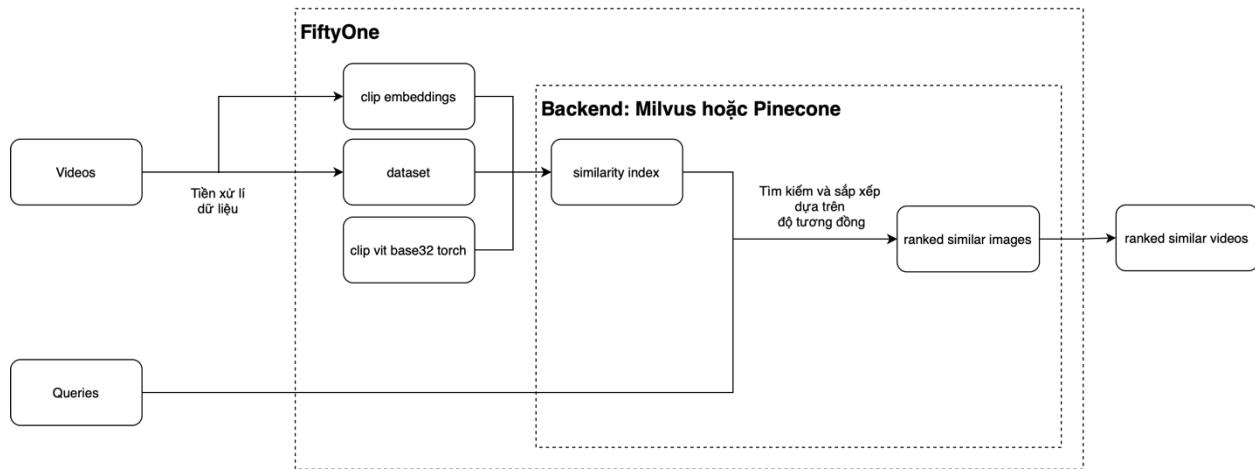
1. một bản phác họa do con người vẽ
2. mô tả văn bản về hình ảnh mục tiêu mong muốn.

Phương pháp sử dụng kiến trúc mã hóa hình ảnh và văn bản giống như mô tả trong CLIP để tận dụng các mạng đã được đào tạo trước với dữ liệu đào tạo quy mô lớn. Sự lựa chọn này tuân theo thực hành phổ biến của việc sử dụng, ví dụ, mạng được đào tạo trước trên ImageNet cho các nhiệm vụ thị giác khác nhau. Vì hình ảnh và bản phác họa nằm trong cùng một lĩnh vực (thị giác) nên mô hình đã sử dụng cùng một kiến trúc cho chúng (ViT-B/16 được đào tạo trên CLIP), gọi là ViT-B/16, trong đó bộ mã hóa hình ảnh dựa trên Vision Transformer, mà bài nghiên cứu thấy là tốt nhất cho nhiệm vụ này.



2.3. Cách tiếp cận

2.3.1 Sơ đồ chung của hệ thống



Hình 2.3.1 Sơ đồ chung của hệ thống

Hệ thống tích hợp giữa FiftyOne và Milvus là một cấu trúc mạng hoạt động hiệu quả để lưu trữ, xử lý và truy vấn dữ liệu video dựa trên câu truy vấn. Dưới đây là mô tả sơ đồ chung của cách hệ thống này hoạt động:

Dữ liệu đầu vào: Hệ thống nhận đầu vào là các video và các câu truy vấn của người dùng. Các dữ liệu này sẽ là nguồn gốc cho quá trình trích xuất đặc trưng và tìm kiếm.

Tiền xử lý video: Trước khi được thêm vào hệ thống, các video trải qua quá trình tiền xử lý. Trong quá trình này, các video được phân tích để thu được dữ liệu gọi là "clip embeddings." Các clip embeddings là các vectơ đặc trưng biểu diễn nội dung của từng video. Dữ liệu này cùng với thông tin về video được tổ chức thành một đối tượng "Dataset."

Mô hình CLIP-ViT: Một mô hình deep learning CLIP-ViT (trong đồ án này, chúng tôi sử dụng mô hình "clip-vit-base32-torch" được cung cấp bởi FiftyOne Model Zoo) được sử dụng để tạo ra các vectơ đặc trưng cho cả video và câu truy vấn. Mô hình này chịu trách nhiệm biểu diễn nội dung của video và các mô tả văn bản tương ứng.

Milvus: Dữ liệu và các clip embeddings của video được lưu trữ trong hệ thống cơ sở dữ liệu Milvus. Milvus cung cấp một backend mạnh mẽ cho việc tính similarity index giữa các vectơ đặc trưng của video và câu truy vấn.

Tìm kiếm và sắp xếp: Khi người dùng nhập một câu truy vấn, FiftyOne sẽ sử dụng mô hình CLIP-ViT để tính toán vectơ đặc trưng của câu truy vấn. Sau đó, hệ thống sử dụng Milvus để tìm kiếm và sắp xếp các video trong cơ sở dữ liệu dựa trên tương đồng về vectơ đặc trưng giữa câu truy vấn và video.

Kết quả đầu ra: Kết quả của quá trình tìm kiếm và sắp xếp là danh sách các video có độ tương đồng cao nhất với câu truy vấn. Người dùng sẽ nhận được danh sách video kết quả dựa trên sự phù hợp về nội dung và mô tả văn bản.

Việc tích hợp Milvus làm backend nhằm cung cấp một cơ sở dữ liệu vector có khả năng tính toán similarity index giữa các vectơ đặc trưng một cách nhanh chóng và hiệu quả giúp tìm kiếm dữ liệu video với số lượng lớn và đảm bảo thời gian đáp ứng ngắn. Trong khi đó, **hệ thống tích hợp giữa FiftyOne và Pinecone** có cách hoạt động khá tương tự. Tuy nhiên với khả năng thực hiện các truy vấn phức tạp, bao gồm việc thực hiện các tìm kiếm đa mức độ tương đồng và kết hợp nhiều vectơ đặc trưng giúp người dùng tạo ra các tìm kiếm thông minh và chính xác. Hệ thống này cũng mang đến nhiều tiềm năng trong việc triển khai và phát triển trong các dự án và doanh nghiệp.

Tóm lại, tuỳ thuộc vào nhu cầu của người dùng và hệ thống, ta có thể chọn lựa Milvus hoặc Pinecone làm backend hỗ trợ cho hệ thống truy vấn video.

2.3.2 Giới thiệu mô hình clip-vit-base32-torch được cung cấp bởi FiftyOne Model Zoo

FiftyOne Model Zoo, một nguồn tài nguyên chứa các mô hình được huấn luyện trước, mà bạn có thể tải về và sử dụng để thực hiện suy luận trên các tập dữ liệu FiftyOne thông qua một vài lệnh đơn giản. FiftyOne Model Zoo chứa hơn 70 mô hình được huấn luyện

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

trước cho việc phân tích dữ liệu. Danh sách các Zoo Model khả dụng có thể được tìm thấy tại: https://docs.voxel51.com/user_guide/model_zoo/models.html

Các mô hình có thể yêu cầu các gói phần mềm bổ sung như TensorFlow hoặc PyTorch để có thể sử dụng tùy vào mô hình được sử dụng. Nếu thiếu các gói này, thông báo lỗi về việc cài đặt chúng sẽ được hiển thị giúp người dùng nhanh chóng chỉnh sửa cho phù hợp.

Để sử dụng FiftyOne Model Zoo, các mô hình có thể được tải về bằng cách sử dụng lệnh `foz.load_zoo_model("<tên-mô-hình>")`. Sau đó, một số mẫu từ tập dữ liệu có thể được tải về và áp dụng mô hình đã tải lên chúng bằng cách sử dụng các phương thức như `apply_model()` và `compute_embeddings()`. Các mô hình này còn có thể được dùng để lưu trữ các logits và dự đoán kết quả tùy thuộc vào bài toán của người dùng. Ngoài ra, nhiều mô hình được cung cấp còn có khả năng tính toán embeddings cho dự đoán, điều này mở ra các cơ hội cho việc phân tích dữ liệu dựa trên các vector embeddings. Trong đồ án này, chúng tôi dựa vào khả năng này để có thể tính toán và áp dụng các embeddings cho việc xử lý và truy vấn dữ liệu. Thêm nữa, FiftyOne còn cung cấp lớp TorchImageModel để có thể tích hợp mô hình PyTorch tùy chỉnh và sử dụng chúng trong các phương thức tích hợp.

Đối với bài toán truy vấn video này, chúng tôi sử dụng mô hình clip-vit-base32-torch trong FiftyOne Model Zoo vì những ưu điểm của chúng đối với tác vụ liên kết ngữ nghĩa giữa hình ảnh và văn bản đã được đề cập ở phần trước.Thêm nữa, mô hình clip-vit-base32-torch là bộ mã hóa văn bản/hình ảnh CLIP, được đào tạo trên 400 triệu cặp văn bản-hình ảnh. Dung lượng của mô hình là 337.58 MB và có khả năng tạo embeddings. Nó được phát triển từ nguồn mở tại: <https://github.com/openai/CLIP>, và yêu cầu các gói phần mềm như torch và torchvision. Mô hình này hỗ trợ cả CPU và GPU. Nó được sử dụng trong các nhiệm vụ liên quan đến phân loại, logits, embeddings và zero-shot learning.

Chương 3: TRIỂN KHAI

3.1 Dữ liệu triển khai

Chúng tôi sử dụng tập dữ liệu videos cùng nhiều thông tin đi kèm sau:

Videos: Dữ liệu chứa hơn 100 giờ video với tổng cộng 299 video.

Keyframes: Dữ liệu này bao gồm tất cả các keyframe được trích xuất từ các video nêu trên. Mỗi keyframe được lưu trong một thư mục riêng, tên thư mục dựa trên tên của video gốc. Ví dụ, các keyframe từ video L01_V001.mp4 sẽ được lưu trong thư mục L01_V001. Tên file keyframe được đánh số thứ tự tăng dần và vị trí của keyframe được xác định trong file metadata.

Objects: Dữ liệu này chứa thông tin về các vật thể (objects) được phát hiện từ mô hình Faster RCNN, được huấn luyện trên tập dữ liệu OpenImagesV4. Kết quả từ mô hình phát hiện vật thể được lưu dưới dạng tệp JSON. Mỗi tệp JSON chứa danh sách các objects tương ứng với các keyframe. Ví dụ, keyframe L01_V001/0000.jpg sẽ có một tệp JSON chứa thông tin về các object và có tên L01_V001/0000.json.

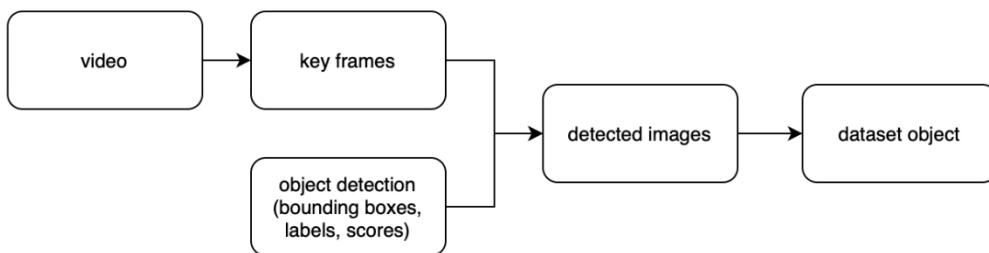
CLIP Features: Thư mục này chứa thông tin về CLIP features của tất cả các frames trong thư mục Keyframes. Tất cả các CLIP features của các keyframe được lưu vào một tệp npy duy nhất. Các vector features được sắp xếp theo thứ tự tăng dần ứng với chỉ số của keyframes.

Metadata: Dữ liệu metadata bao gồm thông tin từ YouTube của kênh cung cấp dữ liệu. Mỗi video có một tệp JSON metadata tương ứng với tên video gốc. Tên tệp metadata là tên video với phần mở rộng là .json. Tuy nhiên, có thể có một số video không có tệp metadata đi kèm.

3.2 Tiền xử lí dữ liệu video

Để có được nguồn dữ liệu như mô tả ở phần trên từ các video, quá trình tiền xử lí dữ liệu là một bước quan trọng đảm bảo tính chính xác và hiệu quả của việc xử lí sau này. Quá trình này bao gồm nhiều bước, bắt đầu từ việc chia video thành các khung ảnh (frames), sau đó loại bỏ các khung ảnh trùng lặp để giảm kích thước dữ liệu. Dưới đây là mô tả chi tiết về các bước tiền xử lí dữ liệu dựa trên các phương pháp đã đề cập:

Tiền xử lý dữ liệu



Hình 2.2.1 Sơ đồ mô tả quá trình tiền xử lý dữ liệu. Các video sẽ được chia thành các khung ảnh và loại bỏ các khung ảnh trùng lặp và lưu lại thành các key frames. Các key frame này sẽ được áp dụng mô hình mô hình Faster RCNN, được huấn luyện trên tập dữ liệu OpenImagesV4 nhằm phát hiện các đối tượng có trong khung ảnh. Các kết quả này sẽ được lưu lại thành đối tượng dataset.

1. Chia video thành các khung ảnh (frames): Đầu tiên, video đầu vào sẽ được chia thành các khung ảnh. Mỗi khung ảnh đại diện cho một trạng thái của video tại thời điểm cụ thể. Chia video thành các khung ảnh là bước cơ bản để tiếp tục xử lý dữ liệu.

2. Loại bỏ khung ảnh trùng lặp: Loại bỏ khung ảnh trùng lặp trong một đoạn video là một vấn đề quan trọng trong xử lý ảnh và video nhằm giảm độ lớn dữ liệu, đồng thời tăng hiệu suất đối với hệ thống trong quá trình lưu trữ, xử lý và truy vấn dữ liệu. Việc loại bỏ này có thể áp dụng các phương pháp sau:

Phát hiện và loại bỏ dựa trên sự tương đồng: Sử dụng thuật toán so sánh hình ảnh, ví dụ như phép toán diff, để xác định sự khác biệt giữa các khung ảnh. Các khung ảnh trùng lặp có thể được xác định bằng việc thiết lập một ngưỡng cho độ tương đồng. Các khung ảnh vượt qua ngưỡng này sẽ được loại bỏ.

Phân đoạn ảnh (Image Segmentation): Sử dụng kỹ thuật phân đoạn hình ảnh để phát hiện và loại bỏ các vùng trùng lặp. Thuật toán Watershed và phân đoạn dựa trên đối tượng là một số ví dụ.

Sử dụng Optical Flow: Optical flow giúp theo dõi chuyển động giữa các khung ảnh. Các khung ảnh có sự di chuyển rất ít hoặc không có sự di chuyển có thể được loại bỏ.

Phát hiện đối tượng (Object Detection): Sử dụng mô hình phát hiện đối tượng như YOLO hoặc Faster R-CNN để xác định và loại bỏ các đối tượng trùng lặp trong các khung ảnh.

Sử dụng phân đoạn thời gian (Temporal Segmentation): Dựa vào thông tin về thời gian để xác định các khung ảnh trùng lặp dựa trên sự xuất hiện liên tiếp của cùng một nội dung trong khoảng thời gian ngắn.

Sử dụng kỹ thuật học máy: Sử dụng mô hình học máy đã được đào tạo để phát hiện và loại bỏ khung ảnh trùng lặp.

3. Lưu các khung ảnh sau khi loại bỏ trùng lặp: Các khung ảnh sau khi loại bỏ trùng lặp sẽ được lưu vào các thư mục video tương ứng. Mỗi khung ảnh được lưu với một tên tệp riêng cùng với một id tương ứng. Điều này giúp duy trì tính nguyên vẹn của dữ liệu và dễ dàng quản lý.

4. Tạo đối tượng dataset: Sau khi đã tiền xử lí dữ liệu và loại bỏ khung ảnh trùng lặp, một đối tượng dataset có thể được tạo từ các khung ảnh trong thư mục "Keyframes". Điều này giúp tổ chức và quản lý dữ liệu một cách hiệu quả hơn.

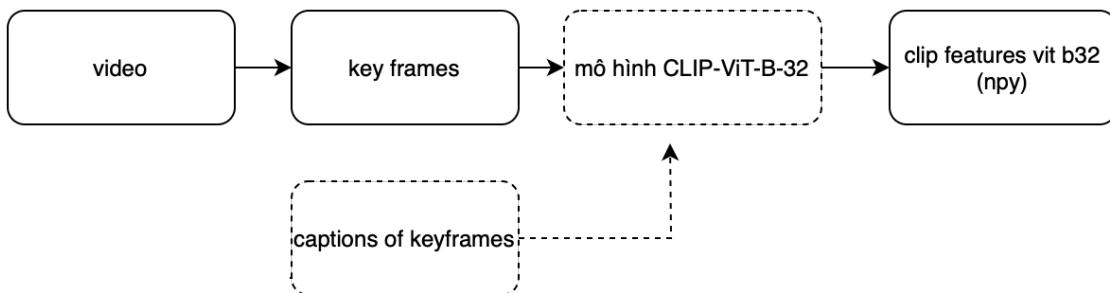
5. Lưu lại đối tượng dataset: Đối tượng dataset đã được tạo có thể được lưu lại để sử dụng cho các lần xử lí dữ liệu sau này mà không cần thực hiện lại các bước tiền xử lí ban đầu đối với các dữ liệu đã xử lí. Điều này giúp tiết kiệm thời gian và tài nguyên khi cần xử lí dữ liệu mới hoặc cập nhật dữ liệu đã có.

Như vậy, quá trình tiền xử lí dữ liệu này là một phần quan trọng trong việc xử lí video và hình ảnh, giúp làm giảm kích thước dữ liệu, tăng hiệu suất và dễ dàng quản lý dữ liệu.

3.3 Trích xuất đặc trưng văn bản và hình ảnh dựa vào CLIP-ViT

Đối với việc trích xuất đặc trưng từ dữ liệu gồm keyframes và captions tương ứng sử dụng CLIP-ViT là quá trình kết hợp khả năng của CLIP-ViT để hiểu và biểu diễn thông tin từ cả hình ảnh và văn bản trong một không gian biểu diễn chung. Quá trình này bao gồm các bước:

Trích xuất đặc trưng văn bản và hình ảnh sử dụng CLIP-ViT-B-32



Hình 2.3.1 Sơ đồ trích xuất đặc trưng văn bản và hình ảnh sử dụng mô hình CLIP-ViT-B-32 trên tập dữ liệu triển khai.

Thu thập dữ liệu: Quá trình bắt đầu bằng việc thu thập dữ liệu keyframes và captions tương ứng. Keyframes là các hình ảnh đại diện cho nội dung video hoặc tài liệu hình ảnh. Captions là mô tả văn bản liên quan đến mỗi keyframe, chúng mô tả nội dung hoặc sự kiện trong hình ảnh.

Trích xuất đặc trưng hình ảnh: Đối với keyframes, sử dụng mô hình ViT của CLIP-ViT để biến đổi hình ảnh thành các vectơ đặc trưng. ViT tách hình ảnh thành các patch và biểu diễn chúng dưới dạng các vectơ. Điều này giúp mô hình hiểu cấu trúc hình ảnh và các đặc điểm quan trọng của nó.

Trích xuất đặc trưng văn bản: Đối với các captions, sử dụng mô hình học sâu để biểu diễn thông tin văn bản. Mô hình này hiểu nội dung của mô tả và biểu diễn nó dưới dạng các vectơ đặc trưng. Điều này giúp mô hình hiểu ý nghĩa và mối tương quan giữa các từ trong mô tả.

Xây dựng không gian biểu diễn chung: Khả năng trích xuất đặc trưng từ cả hai nguồn dữ liệu cho phép CLIP-ViT xây dựng một không gian biểu diễn chung cho văn bản và hình ảnh. Điều này giúp mô hình biểu diễn thông tin từ cả hai nguồn dữ liệu dưới dạng các vectơ có khả năng so sánh và tương tác.

Ứng dụng trong các tác vụ: Sau khi có được không gian biểu diễn chung, CLIP-ViT có thể được sử dụng trong nhiều ứng dụng khác nhau. Ví dụ, mô hình có thể được sử dụng để tìm kiếm keyframes dựa trên mô tả văn bản hoặc ngược lại, trích xuất thông tin từ keyframes và captions, hoặc tạo ra mô hình dự đoán hình ảnh và văn bản dựa trên các mô tả cụ thể.

Quá trình triển khai này là một sự kết hợp độc đáo của khả năng trích xuất đặc trưng từ văn bản và hình ảnh của CLIP-ViT, mở ra nhiều cơ hội trong việc hiểu và tương tác với thông tin từ cả hai nguồn dữ liệu quan trọng này.

3.4 Lưu trữ và cập nhật cơ sở dữ liệu

Việc lưu trữ và cập nhật cơ sở dữ liệu được triển khai như sau:

Lưu trữ dữ liệu keyframe trong dataset: Các keyframe, cụ thể là các mẫu (samples), được tổ chức và lưu trữ trong đối tượng dataset. Mỗi sample đại diện cho một keyframe của video và bao gồm thông tin về video gốc, độ tương tự, và các thuộc tính khác cần thiết.

Lưu trữ thông tin đối tượng detect: Thông tin về các đối tượng đã được detect trong mỗi keyframe được lưu trữ trong các tệp JSON riêng biệt trong thư mục (folder) "object." Mỗi tệp JSON chứa các thông tin quan trọng như tọa độ của bounding-box của các đối tượng, nhãn (label), và điểm tin (confidence score) của từng đối tượng trong keyframe. Điều này giúp xác định các đối tượng trong keyframe đó cũng như trong video.

Mapping Sample với các đối tượng detect: Thực hiện ánh xạ (mapping) từng sample với các đối tượng đã detect và có confidence score vượt qua ngưỡng được xác định trước. Điều này tạo ra một danh sách các đối tượng có trong từng keyframe.

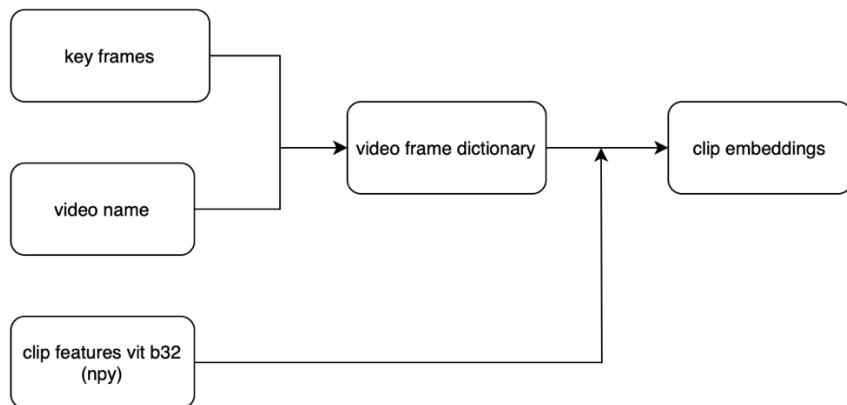
Tạo dictionary "video_keyframe_dict": Một dictionary được tạo ra với tên "video_keyframe_dict." Đây là một danh sách chứa thông tin về các keyframe của mỗi video. Trong đó, mỗi video có danh sách các keyframe (Sample) tương ứng.

Sắp xếp lại danh sách keyframe: Để đảm bảo rằng thứ tự đọc video và vector embedding của keyframe đúng, danh sách các keyframe trong từng video được sắp xếp lại. Điều này đặc biệt quan trọng khi cần so sánh và tìm kiếm dựa trên nội dung video.

Tạo dictionary "embedding_dict": Một dictionary khác có tên "embedding_dict" được tạo ra để lưu trữ thông tin về vector CLIP embedding của từng keyframe trong mỗi video. Điều này giúp tạo một liên kết giữa keyframe và các vector đặc trưng tương ứng.

Tạo danh sách "clip_embedding": Cuối cùng, danh sách "clip_embedding" được tạo ra để lưu trữ các vector embedding CLIP của từng mẫu (sample) trong dataset. Điều này giúp xác định sự tương đồng và tìm kiếm dựa trên nội dung video khi câu truy vấn được thực hiện.

Tạo CLIP Embeddings



Hình 3.3.1 Sơ đồ mô tả quá trình tạo CLIP Embeddings

3.5 Truy vấn dữ liệu

3.5.1 Truy vấn dữ liệu với hệ thống tích hợp giữa FiftyOne và Milvus

Milvus là một trong những cơ sở dữ liệu vector phổ biến nhất hiện có, vì vậy, việc sử dụng khả năng tìm kiếm vector của Milvus trên dữ liệu video từ FiftyOne trở nên rất đáng mong đợi. Để tạo các bộ sưu tập Milvus, chúng tôi tải lên các vector và thực hiện các truy vấn dựa trên độ tương đồng, cả bằng cách lập trình Python và thông qua giao diện người dùng.

Quá trình cơ bản sử dụng Milvus để tạo chỉ mục tương tự (similarity index) trên tập dữ liệu FiftyOne và sử dụng nó để truy vấn dữ liệu như sau:

1. Tải một tập dữ liệu bất kì vào FiftyOne.
2. Tính toán các vector nhúng cho các mẫu hoặc các phần trong tập dữ liệu của bạn hoặc chọn một mô hình để tạo ra các vector nhúng.
3. Sử dụng phương thức `compute_similarity()` để tạo chỉ mục tương tự Milvus cho các mẫu hoặc các phần đối tượng trong tập dữ liệu bằng cách thiết lập tham số `backend="milvus"` và chỉ định một `brain_key` tùy chọn.
4. Sử dụng chỉ mục tương tự Milvus này để truy vấn dữ liệu với `sort_by_similarity()`.
5. Nếu cần, ta có thể thêm – xoá dữ liệu, hoặc thêm xoá cả chỉ mục.

Các tham số cấu hình Milvus: Milvus hỗ trợ nhiều tham số truy vấn có thể được sử dụng để tùy chỉnh các truy vấn. Các tham số này bao gồm:

collection_name (None): tên của bộ sưu tập Milvus để sử dụng hoặc tạo. Nếu không có, một bộ sưu tập mới sẽ được tạo

metric ("dotproduct"): phép đo khoảng cách nhúng sẽ sử dụng khi tạo chỉ mục mới. Các giá trị được hỗ trợ là ("dotproduct", "euclidean")

consistency_level ("Session"): cấp độ nhất quán sẽ sử dụng. Các giá trị được hỗ trợ là ("Strong", "Session", "Bounded", "Eventually")

3.5.2 Truy vấn dữ liệu với hệ thống tích hợp giữa FiftyOne và Pinecone

Cấu hình Pinecone: Người dùng cần cấu hình thông tin đăng nhập Pinecone và tạo chỉ mục tương tự Pinecone trong FiftyOne bằng cách sử dụng API.

Quá trình cơ bản sử dụng Pinecone để tạo chỉ mục tương tự (similarity index) trên tập dữ liệu FiftyOne và sử dụng nó để truy vấn dữ liệu như sau:

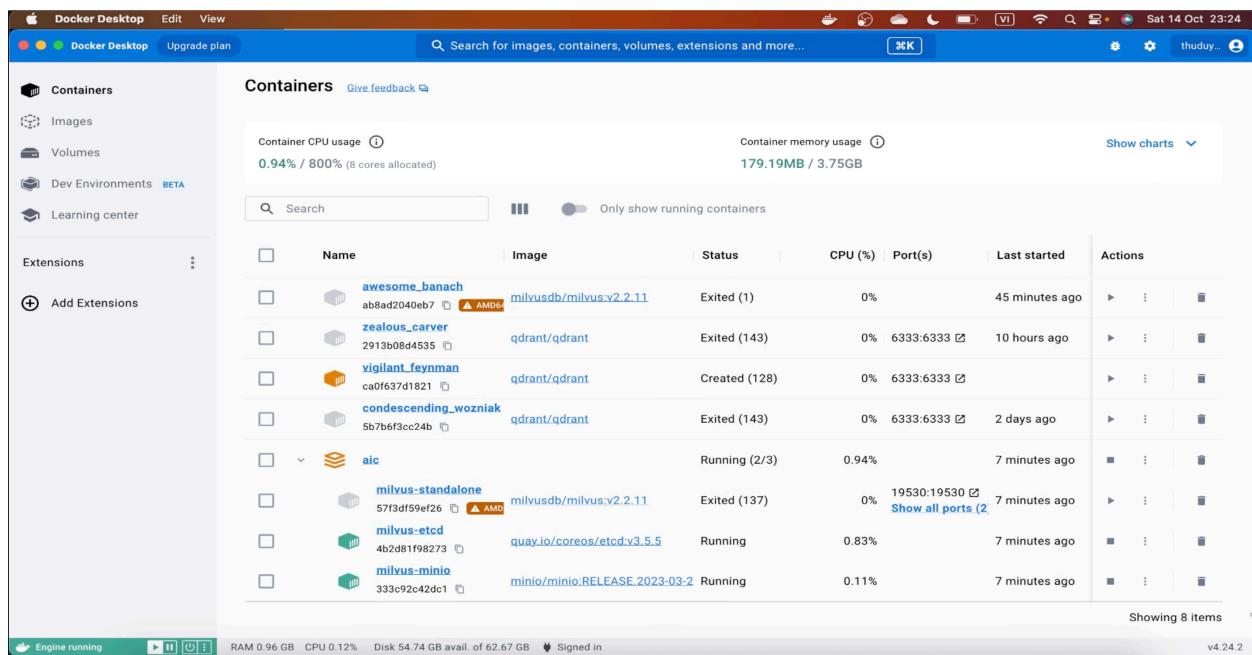
1. Tải một bộ dữ liệu video bất kì vào FiftyOne.
2. Tính toán vectors nhúng cho các mẫu hoặc vùng mẫu trong bộ dữ liệu hoặc chọn một mô hình để tạo vectors nhúng.
3. Sử dụng phương thức `compute_similarity()` để tạo chỉ mục tương tự Pinecone cho các mẫu hoặc vùng mẫu trong bộ dữ liệu bằng cách đặt tham số `backend="pinecone"` và chỉ định `brain_key` theo ý muốn.
4. Sử dụng chỉ mục tương tự Pinecone này để truy vấn dữ liệu bằng cách sử dụng `sort_by_similarity()`.
5. Nếu cần, người dùng có thể thêm – xoá dữ liệu, hoặc xóa cả chỉ mục.

Người dùng có thể truy cập trực tiếp đối tượng Pinecone để sử dụng các phương pháp tùy chỉnh bên trong Pinecone.

Ngoài ra, FiftyOne còn cung cấp các phương pháp để quản lý các brain chạy, bao gồm liệt kê chúng, lấy thông tin chi tiết, đổi tên và xóa chúng.

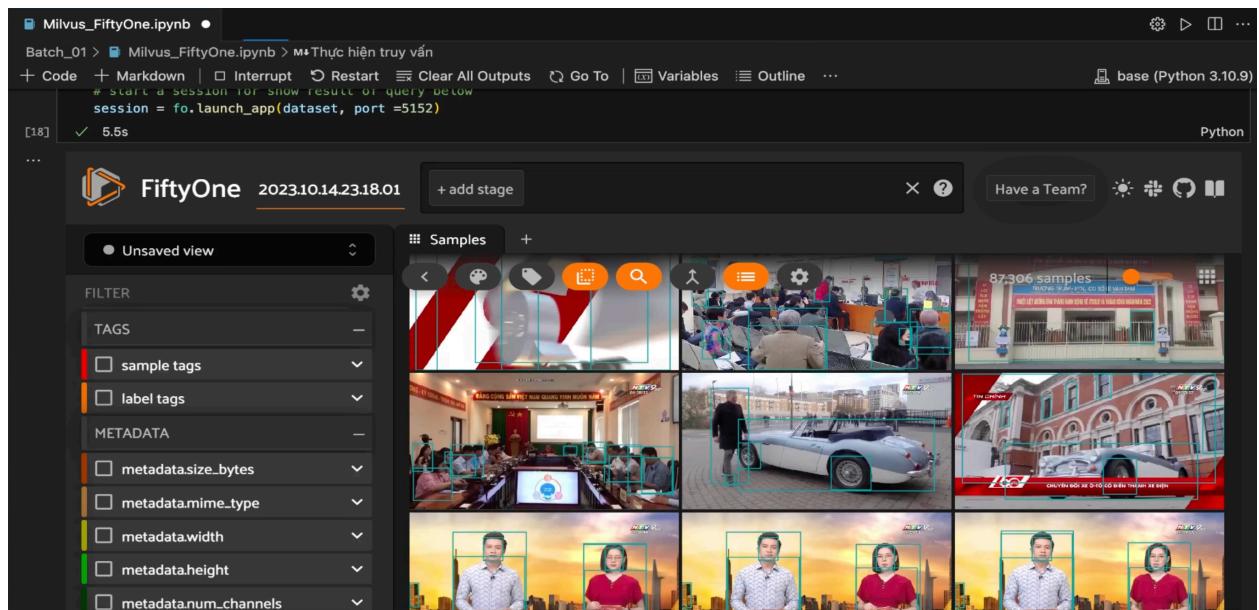
Chương 4: KẾT QUẢ THỬ NGHIỆM

4.1 Truy vấn và hiển thị kết quả trên FiftyOne_Milvus

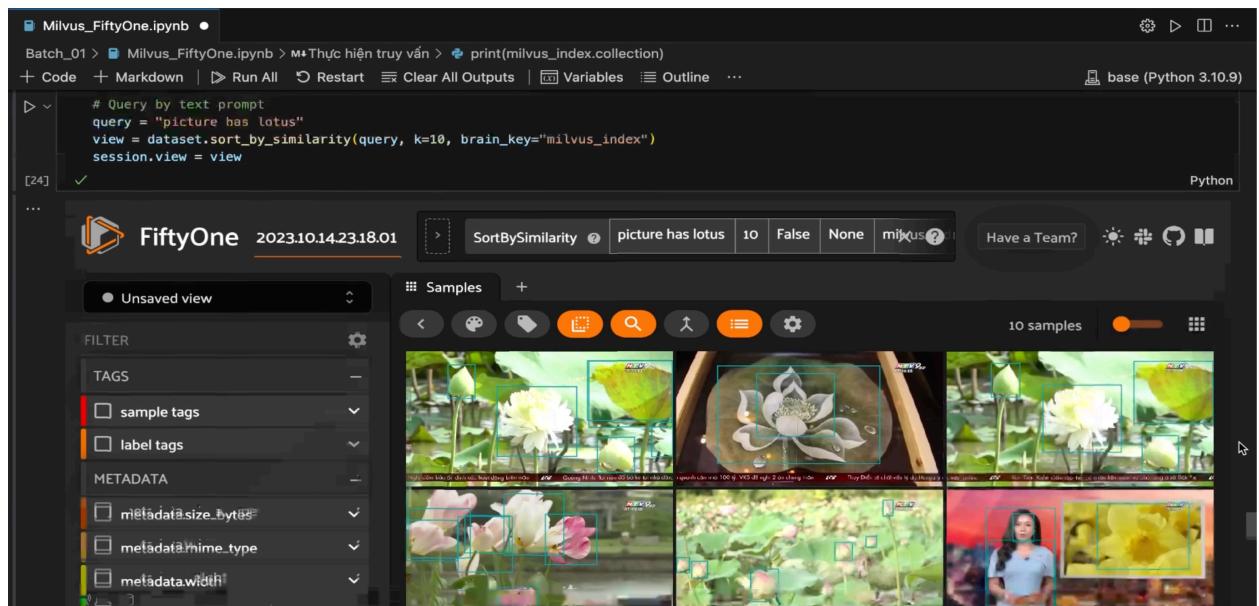


Hình 4.1.1 Kết nối và khởi chạy Milvus thông qua Docker Desktop

**KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

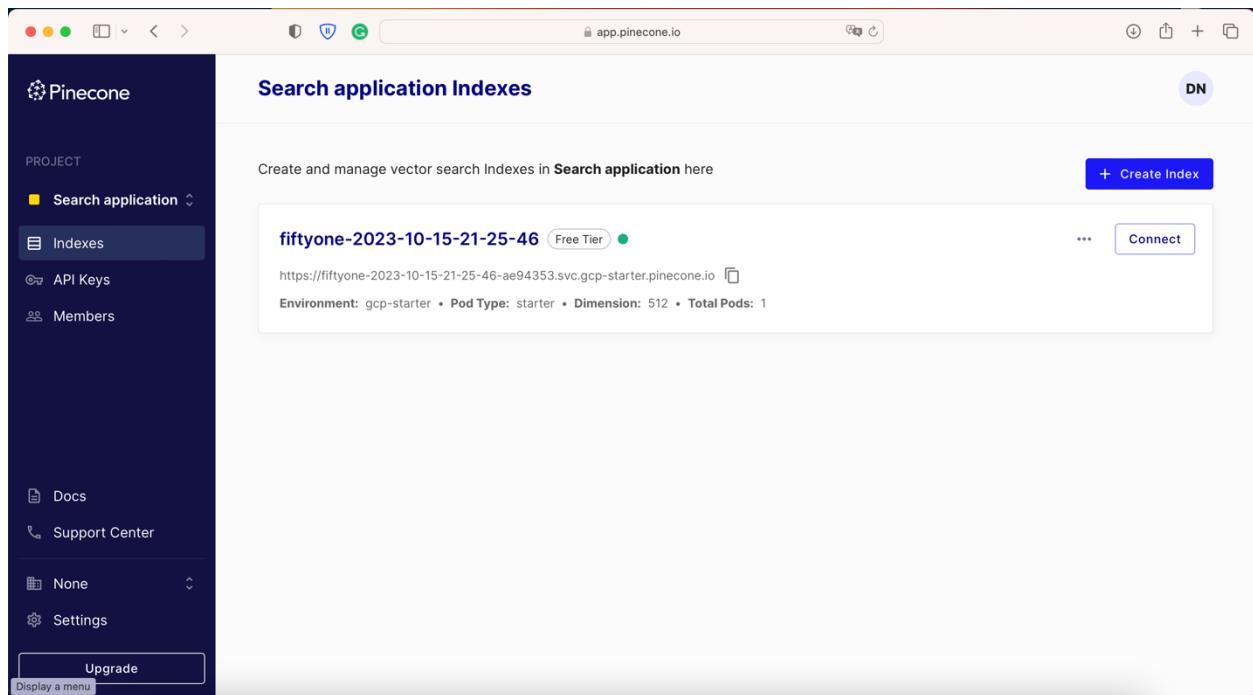


Hình 4.1.2 Khởi chạy FiftyOne

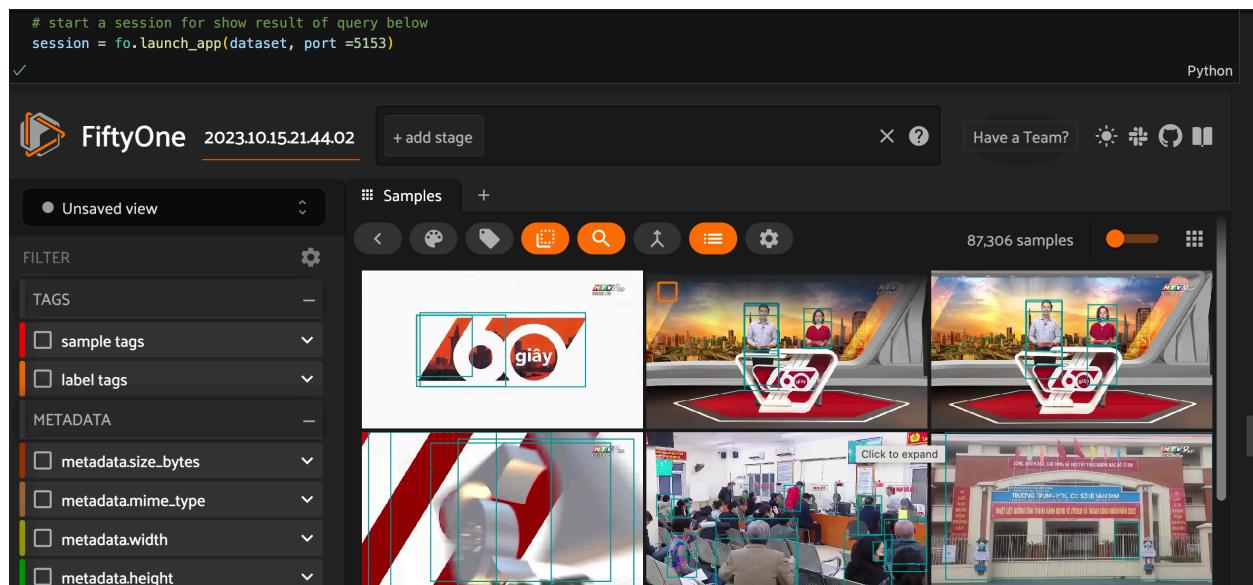


Hình 4.1.3 Thực hiện tìm kiếm video bằng câu truy vấn và kết quả truy vấn

4.2 Truy vấn và hiển thị kết quả trên FiftyOne_Pinecone



Hình 4.2.1 Dữ liệu sau khi được tiền xử lí và tính độ tương đồng được lưu thành index trong hệ cơ sở dữ liệu Pinecone



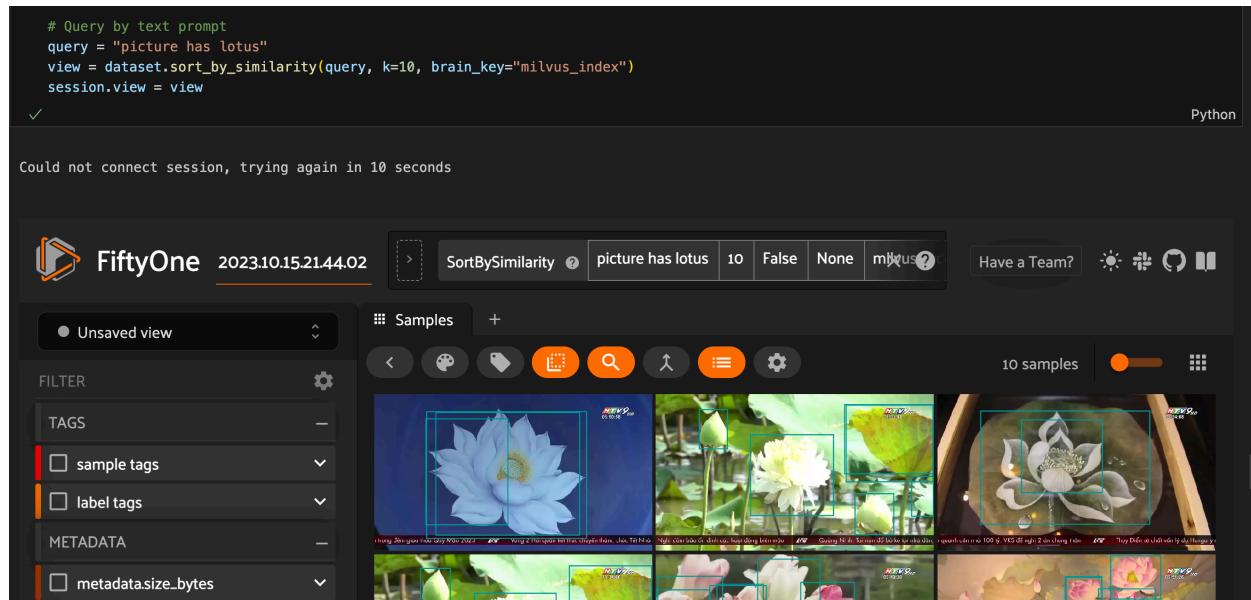
Hình 4.2.2 Khởi chạy FiftyOne

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

```
# Query by text prompt
query = "picture has lotus"
view = dataset.sort_by_similarity(query, k=10, brain_key="milvus_index")
session.view = view
✓
```

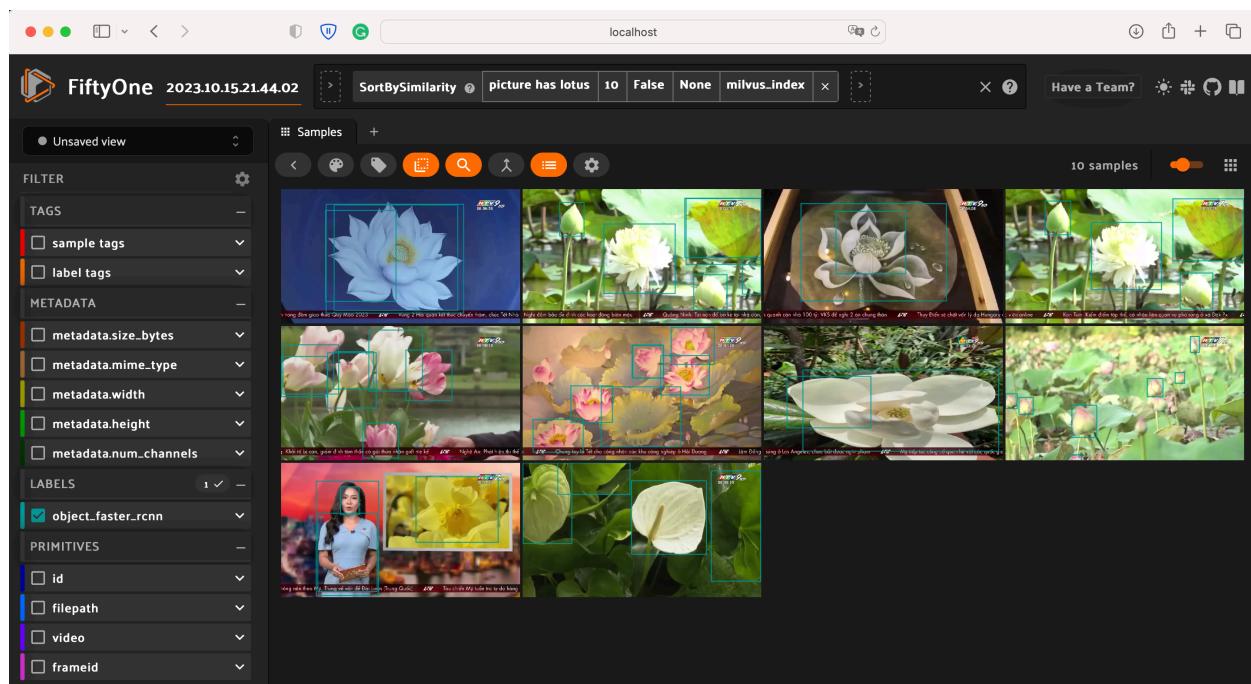
Python

Could not connect session, trying again in 10 seconds



The screenshot shows the FiftyOne application interface. At the top, there is a Python code block that runs a query to find images containing a lotus. Below the code, a message says "Could not connect session, trying again in 10 seconds". The main window displays a grid of 10 image samples related to lotus flowers. On the left, there is a sidebar with a "FILTER" section containing "TAGS" (sample tags, label tags) and "METADATA" (metadata.size.bytes). The top navigation bar includes tabs like "SortBySimilarity", "picture has lotus", "10", "False", "None", and "milvus".

Hình 4.2.3 Thực hiện truy vấn video bằng câu truy vấn



Hình 4.2.4 Kết quả truy vấn video

Chương 5: KẾT LUẬN

Trong bài báo cáo này, chúng em đã thực hiện một nghiên cứu về truy vấn video sử dụng câu truy vấn mô tả, sử dụng FiftyOne và tích hợp Pinecone, Milvus. Kết quả của nghiên cứu cho thấy việc kết hợp các công nghệ này đã đem lại những kết quả khá ấn tượng.

Việc sử dụng FiftyOne cho phân tích video và trích xuất các câu mô tả đã giúp chúng em tạo ra một bộ dữ liệu mô tả phong phú và đa dạng. Điều này đã cung cấp cho chúng em cơ sở dữ liệu đủ lớn để thực hiện các thử nghiệm truy vấn.

Pinecone với khả năng xử lý truy vấn nhanh chóng và hiệu quả, cho phép tìm kiếm nội dung video dựa trên các câu truy vấn mô tả. Công nghệ này giúp tối ưu hóa thời gian truy vấn và cung cấp kết quả chính xác. Ngoài ra, Milvus với thiết kế để hỗ trợ lưu trữ và truy vấn các vectơ biểu đồ hoặc biểu đồ nhúng, và cho phép tìm kiếm dữ liệu dựa trên khoảng cách giữa các vectơ, thường được sử dụng trong các ứng dụng tìm kiếm, phân loại, và gợi ý dựa trên dữ liệu vector.

Tóm lại, việc tích hợp FiftyOne, Pinecone và Milvus trong quá trình truy vấn video bằng câu truy vấn mô tả đã đem lại kết quả khá tốt. Hệ thống này giúp cải thiện khả năng tìm kiếm và truy vấn nội dung video, mang lại giá trị cho các ứng dụng có liên quan đến trí tuệ nhân tạo, thương mại điện tử và hơn thế nữa.

Chương 6: Hướng phát triển

Trong phần này, chúng tôi đề xuất một số hướng phát triển quan trọng để cải thiện hiệu suất và độ chính xác của hệ thống truy vấn video bằng câu truy vấn mô tả. Các hướng phát triển sau đây có thể giúp giải quyết các hạn chế hiện tại và nâng cao khả năng truy vấn.

1. Detaching Đặc Điểm Mô Tả (Feature Detachment): Nghiên cứu và phát triển các kỹ thuật để cho phép detach đặc điểm mô tả từ video, tạo điều kiện cho việc truy vấn dựa trên đặc điểm cụ thể.
2. Tiếp Cận Nâng Cao Ranking: Sử dụng học máy và deep learning để cải thiện ranking của kết quả truy vấn, tối ưu hóa mối quan hệ giữa câu truy vấn mô tả và video.
3. Mở Rộng Dữ Liệu Đào Tạo (Data Augmentation): Tăng cường dữ liệu đào tạo bằng cách thêm nhiều câu mô tả và video, giúp mô hình hiểu rõ nhiều ngữ cảnh khác nhau.
4. Kết Hợp Nhiều Mô Hình (Ensemble Models): Sử dụng kỹ thuật kết hợp nhiều mô hình truy vấn để tạo sự đa dạng trong việc tìm kiếm và ranking kết quả.
5. Tối Ưu Hóa Tài Nguyên (Resource Optimization): Nghiên cứu và triển khai các chiến lược tối ưu hóa tài nguyên để đảm bảo hệ thống hoạt động một cách hiệu quả và nhanh chóng.
6. Phản Hồi Từ Người Dùng (User Feedback): Thu thập phản hồi từ người dùng về kết quả truy vấn và sử dụng nó để cải thiện hệ thống.
7. Liên Kết Với Các Nghiên Cứu Tiên Tiến Khác: Hợp tác với các nghiên cứu và dự án tiên tiến khác để sử dụng các kỹ thuật và công nghệ mới nhất trong lĩnh vực truy vấn video.

Những hướng phát triển này sẽ cung cấp cơ hội quan trọng để cải thiện đáng kể hiệu suất và độ chính xác của hệ thống truy vấn video bằng câu truy vấn mô tả và giải quyết những thách thức hiện tại.

Tài liệu tham khảo

References

- [1] Moore, B. E. and Corso, J. J., "FiftyOne," *GitHub*. Note: <https://github.com/voxel51/fiftyone>, 2020.
- [2] "Milvus," [Online]. Available: <https://milvus.io/>.
- [3] Pinecone. [Online]. Available: <https://www.pinecone.io/>.
- [4] Alec Radford, "Learning Transferable Visual Models From Natural Language Supervision," *CoRR*, vol. abs/2103.00020, 2021.
- [5] Baldrati, Alberto and Bertini, Marco and Uricchio, Tiberio and Del Bimbo, Alberto, "Effective conditioned and composed image retrieval combining clip-based features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21466--21474.
- [6] Sangkloy, Patsorn and Jitkrittum, Wittawat and Yang, Diyi and Hays, James, "A sketch is worth a thousand words: Image retrieval with text and sketch," in *European Conference on Computer Vision*, Springer, 2022, pp. 251--267.