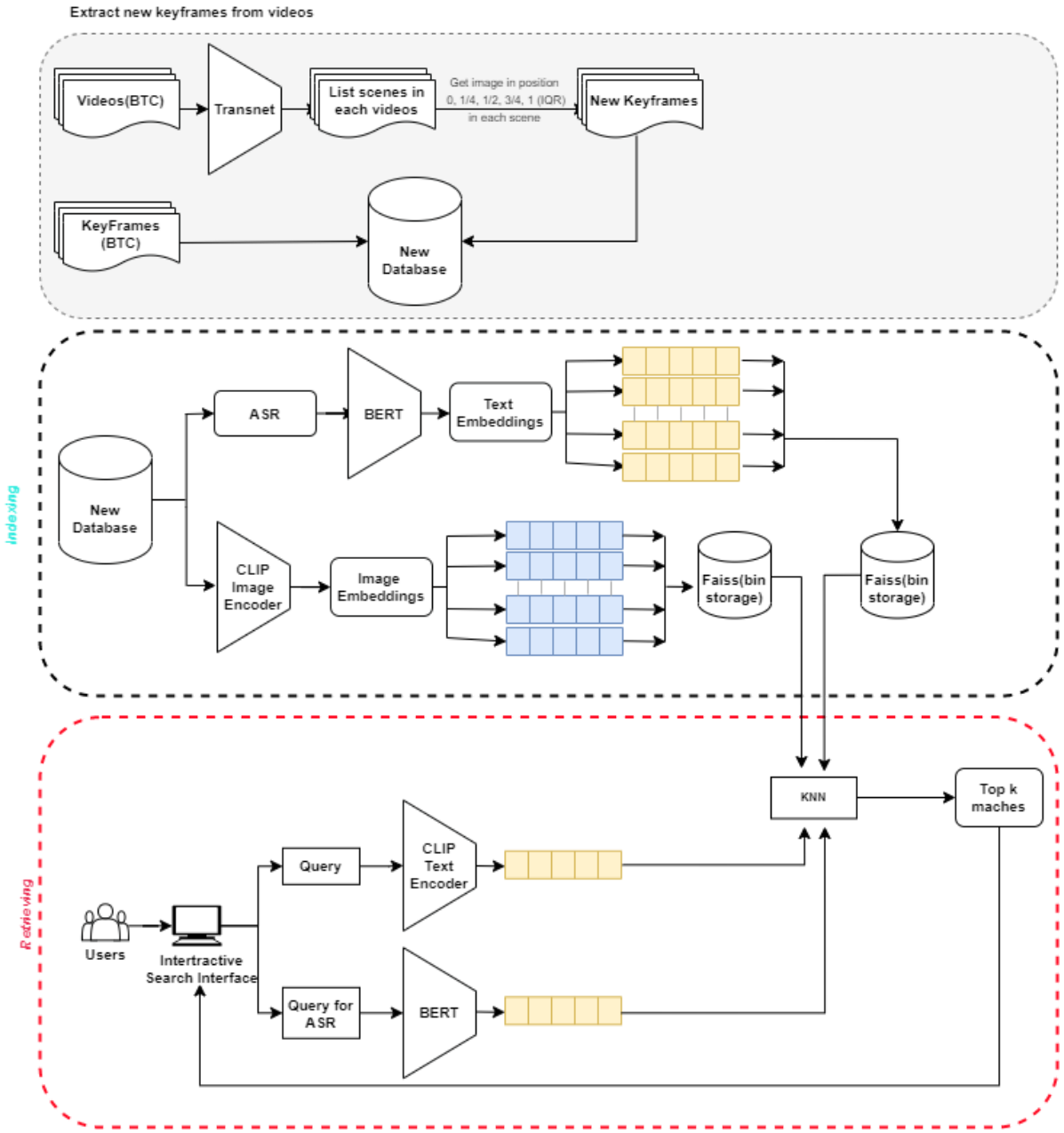
HCM AI Challenge 2022
Team AIO-ISC

- **Mô tả giải pháp**

  Giải pháp của nhóm gồm 3 bước:
  - **Preprocessing data**
  - **Indexing**
  - **Retrieving**



Hình 1: Pipeline giải pháp

- ***Giai đoạn* Preprocessing data:**

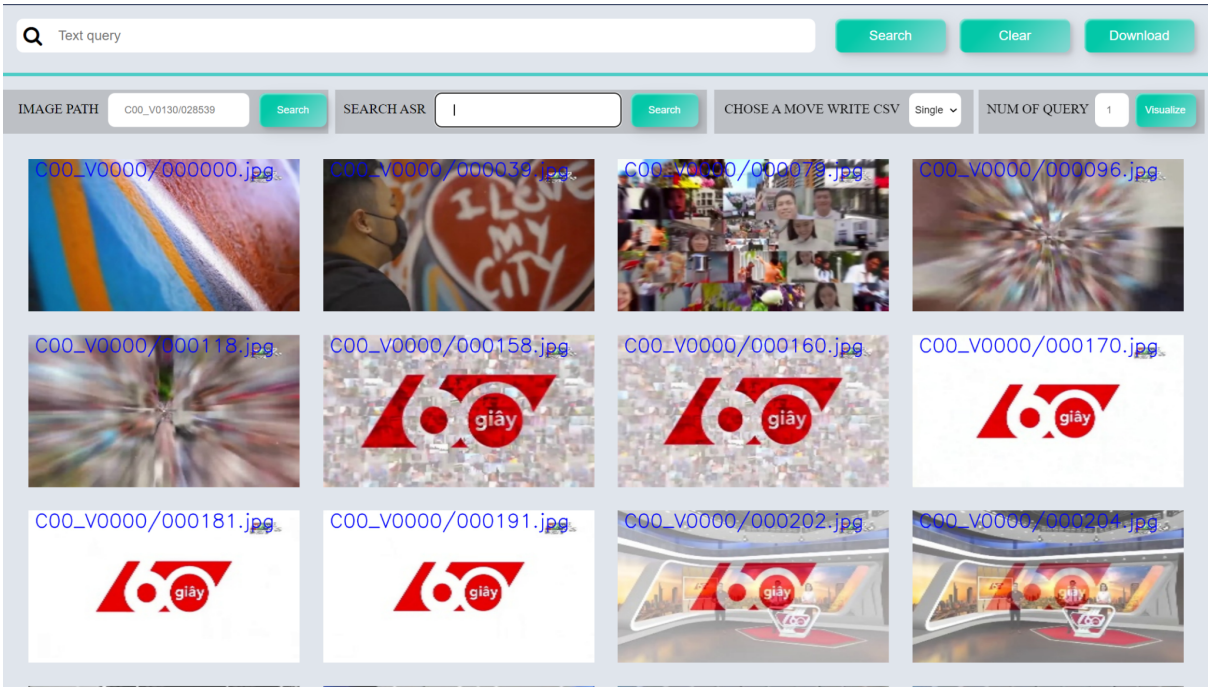
- Ngoài các Feature Vector được Ban tổ chức cung cấp, nhóm đã tận dụng thêm các videos để trích xuất thêm các keyframes của từng khung cảnh (scene) bên trong từng video. Để thực hiện yêu cầu này, nhóm đã ứng dụng model Transnet (https://github.com/soCzech/TransNet) để tìm được các cảnh trong mỗi video.
- Transnet là một model dùng để nhận biết được các cảnh chuyển giao của 1 video.
    - Đầu vào: 1 video
    - Đầu ra: Danh sách các cảnh của 1 video. Mỗi cảnh sẽ có thông tin của các frame thuộc về cảnh đó (Vị trí frame đầu(s) và vị trí frame cuối(e) thuộc về cảnh đó trong video).
- Tiếp theo áp dụng model này để có thể tìm ra được các cảnh trong một video. Để từ đó, lấy ra được thêm các keyframes thuộc về cảnh đó. Trong mỗi cảnh, nhóm quyết định lấy thêm 6 keyframe nằm ở vị trí e+(e-s)*0, e+(e-s)*0.25, e+(e-s)*0.5, e+(e-s)*0.75 và e+(e-s)*1
- Sau khi lấy được 6 keyframes ở mỗi cảnh của từng video, kết hợp các keyframes mới này với keyframes mà BTC cung cấp để tạo ra 1 danh sách các keyframes mới (New dataset) để phục vụ cho bài toán.
- Đây cũng là giai đoạn đầu tiên trong pipeline của nhóm. Pipeline sẽ được trình bày rõ hơn ở phía dưới.

- ***Giai đoạn Indexing:***
- Sau khi đã có được bộ Keyframes mới (New Database), nhóm tiến hành encode các Keyframes thông qua CLIP model (Với pretrain Vit-B/16).
    - CLIP model: là một mạng nơ-ron được đào tạo trên nhiều cặp (hình ảnh, văn bản) khác nhau. Nó có thể được hướng dẫn bằng ngôn ngữ tự nhiên để dự đoán đoạn văn bản có liên quan nhất, được cung cấp bằng hình ảnh, mà không cần tối ưu hóa .
        - Đầu vào: Có thể là 1 đoạn text hoặc 1 hình ảnh
        - Đầu ra: Một vector embedding có kích thước (512,)
- Sau khi đã encode tất cả Keyframe có trong "New Database" thành danh sách các vector embedding, sử dụng các vector embeddings này để tính cosine similarities giữa các keyframes.
- Để có thể tính được độ tương đồng giữa các KeyFrame, nhóm đã áp dụng Faiss library.
    - Faiss: Là 1 thư viện sử dụng similiarity search cùng với clustering các vector. Faiss xài thêm kỹ thuật **Hashing** (phương pháp giảm chiều vector -> việc này đánh đổi giữa tốc độ và độ chính xác) cùng với kỹ thuật tối ưu trên C++ và GPU**.** Bộ thư viện bao gồm các thuật toán tìm kiếm vector đa chiều trong similarity search
        - Đầu vào: Một danh sách các vector (Giá trị bên trong vector phải có kiểu dữ liệu là float32), một query vector, topk
        - Đầu ra: Hai danh sách images và indexes. Images chứa topk các vector thuộc danh sách vector đầu vào mà tương đồng với query vector nhất. Indexes là danh sách chứa topk vị trí tướng ứng với các vector của Images.
- Trong bài toán này, nhóm sẽ sử dụng phép đo cosine similarity vì nó sẽ giúp mình truy vấn được các keyframes chuyển cảnh (mức contrast thấp)
- Trước khi sử dụng Faiss để tính cosine similarities, khởi tạo Faiss trước để lưu danh sách Vector embeddings . Danh sách vector embeddings sẽ được lưu dưới dạng 1 file cosine.bin. Việc lưu thành file cosine.bin này sẽ giúp cho việc tính cosine giữa các keyframes với query tiết kiệm thời gian hơn rất là nhiều.
- Bên cạnh việc sử dụng CLIP để lấy các vector embedding thì nhóm còn sử dụng thêm 1 thông tin khác trong Video mà BTC cung cấp, đó là Speech(Giọng nói).
- Nhóm sử dụng Automatic Speech Recognition model (ASR) để chuyển các audio (âm thanh) bên trong của từng scene thành text. Sau đó sử dụng BERT model để embedding đoạn text đó thành 1 vector (Text Embedding). Cuối cùng thu được danh sách các text embedding của các scene của từng video.
- Tiếp theo nhóm cũng đưa các text embeddings này vào Faiss và lưu lại dưới dạng 1 file cosine_bert.bin

- ***Giai đoạn Retrieving:***

- Trong giai đoạn này, người dùng sẽ có 2 lựa chọn:
    - Tìm kiếm sự kiện bằng 1 câu text (Text Query).
    - Tìm kiếm sự kiện bằng 1 câu text đã được nói trong video (Text ASR)
- Khi người dùng tìm kiếm bằng Text Query. Câu text sẽ được encode thành 1 vector bằng CLIP. Sau đó sẽ được đưa vào Faiss để similarity search với các vector embeddings(Các vector embeddings này được đọc từ file cosine.bin). Kết quả trả về sẽ là topk các frames với độ tương đồng cao nhất so với Query.
- Khi người dùng tìm kiếm bằng Text ASR. Câu text sẽ được encode thành 1 vector thông qua model BERT. Sau đó sẽ được đưa vào Faiss để similarity search với các vector embeddings(Các vector embeddings này được đọc từ file cosine_bert.bin). Kết quả trả về sẽ là topk các frames với độ tương đồng cao nhất so với Query .
- Top k các frames sẽ được hiển thị lên hệ thống để người dùng lựa chọn.

Xây dựng hệ thống



Hình 2: Giao diện hệ thống

- Trang web hệ thống gồm các chức năng như sau:
    - Tìm kiếm thông qua Text Query
    - Tìm kiếm thông qua Text ASR
    - Tìm kiếm thông qua tên của ảnh
    - Hiển thị các ảnh có trong file .csv để check lại các file submit
    - TÌm kiếm ảnh thông qua 1 ảnh khác (KNN search)
    - Select ảnh để lưu vào file submit
    - Download file csv

+ *Tìm kiếm thông qua Text Query*:
Người dùng, nhập thông tin đoạn text vào Textbox "Text Query" sau đó nhấn nút Search thì hệ thống sẽ trả về các KeyFrames tương ứng.

+ *Tìm kiếm thông qua Text ASR:*

Người dùng nhập thông tin ASR vào Textbox “Search ASR” sau đó nhấn nút search thì hệ thống sẽ trả về các KeyFrames tương ứng.

+ *Tìm kiếm thông qua tên của ảnh:*
Khi người dùng nhập thông tin tên frame vào Textbox “Image Path” sau đó nhấn nút search. Hệ thống sẽ trả về frame tương ứng với đường dẫn đó và các Frames khác cùng nằm trong video đó.

+ *Tìm kiếm ảnh thông qua 1 ảnh khác.*
Khi người dùng rê chuột vào hình ảnh bất kỳ, sẽ xuất hiện 2 button là KNN và select. Khi nhấn vào button KNN, hệ thống sẽ trả về danh sách các frames similarity với hình ảnh đã chọn, Việc này được thực hiện bằng cách encode frame đã chọn bằng CLIP sau đó đưa vào faiss để tính cosine similarity với tất cả frames trong New Database.

+ *Select ảnh để lưu vào file submit:*
Nếu người dùng nhấn vào nút select thì hệ thống sẽ kiểm tra chế độ lưu mà người dùng đã chọn. Có 2 chế độ lưu cho người dùng chọn ở ComboBox “Choose a mode write csv” là Single, và List Shot.
    - Nếu người dùng chọn *Single*. Thì hệ thống chỉ lưu lại frame mà người dùng đã chọn
    - Nếu người dùng chọn *List Shot*. Thì hệ thống sẽ lưu frame mà người dùng chọn lên hàng đầu của file csv và lưu thêm các frame nằm trong cùng 1 scene (6 frame cùng scene).

+ *Download file csv*:
Sau khi đã chọn xong các frames như ý, người dùng có thể nhấn nút Download để tải file csv chứa các frames đã chọn về máy.

+ *Hiển thị các ảnh có trong file .csv để check lại các file submit:*
Đây là chức năng giúp cho người dùng dễ dàng check lại các file csv đã chọn nhằm đảm bảo các frames mà người dùng chọn đã chính xác hay chưa? Khi người dùng chọn 1 file csv chứa các frames đã chọn, thì hệ thống sẽ hiển thị các frames đó lên giao diện để người dùng dễ dàng thao tác.

Note: Hiện hệ thống vẫn còn được cập nhật thêm tính năng và dần cải thiện theo đề bài vòng chung kết.