

Bài Thực Hành: Giấu Tin Trong Video: So sánh một số phương pháp phát hiện thay đổi khung cảnh

1. Mục đích

- Hiểu về kỹ thuật giấu tin trong video sử dụng phát hiện thay đổi khung cảnh.
- Áp dụng kỹ thuật DCT để nhúng và trích xuất thông điệp bí mật.
- Thực hành giấu tin và truyền video giữa hai container (Alice và Bob).

2. Yêu cầu thực hành

- Có kiến thức cơ bản về ngôn ngữ lập trình Python.
- Hiểu khái niệm Steganography và các phương pháp phát hiện scene change.
- Sử dụng terminal với các lệnh cơ bản và công cụ: ffmpeg, scp, Python, ...
- Hiểu và nhận biết các điểm chuyển cảnh và phương pháp phát hiện.

3. Nội dung thực hành

3.1. Khởi động bài lab:

- Chạy lệnh trên terminal:

`labtainer stego_code_scenechange_compare`

Tùy chọn -r đảm bảo môi trường được làm mới (reset) nếu đã chạy trước đó, cung cấp một môi trường sạch để thực hành.

(Chú ý: Sinh viên sử dụng của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm.)

Môi trường lab được khởi động. Để minh họa nguyên tắc giấu tin trong video HEVC, lab sử dụng một video mẫu và các công cụ như FFmpeg và Python.

3.2. Xem video và xác định các điểm chuyển cảnh

Trên terminal đã chuẩn bị 5 file code python phục vụ cho 5 phương pháp xác định các điểm chuyển cảnh.

- Gõ lệnh: `ffplay <video>` để xem qua 1 lần video gốc, bạn có thể nhận biết được bao nhiêu điểm chuyển cảnh ?

Chạy lần lượt 5 phương pháp phát hiện chuyển cảnh:

○ **Chênh lệch hệ số trong ma trận DCT**

Phương pháp này hoạt động bằng cách chia mỗi khung hình (frame) của video thành các khối nhỏ, chẳng hạn như các khối 8x8 pixel. Sau đó, biến đổi Cosine Rời rạc (Discrete Cosine Transform - DCT) được áp dụng để chuyển đổi dữ liệu pixel thành các hệ số tần số. Các hệ số này phản ánh cấu trúc tần số của hình ảnh, giúp dễ dàng phát hiện sự thay đổi giữa các frame. Sự khác biệt giữa các hệ số DCT của hai frame liên tiếp được đo lường, thường bằng khoảng cách Euclidean hoặc một phép đo tương tự. Nếu giá trị chênh lệch vượt qua một ngưỡng xác định trước, frame đó được đánh dấu là điểm chuyển cảnh. Phương pháp này có ưu điểm là khả năng chống nhiễu tốt và hiệu quả trong việc phát hiện cả các chuyển cảnh đột ngột lẫn phức tạp, nhưng nhược điểm là nó đòi hỏi nhiều tài nguyên tính toán, đặc biệt với video có độ phân giải cao hoặc dài.

○ **Tổng của chênh lệch thống kê**

Phương pháp này sử dụng các đặc trưng thống kê của frame để phát hiện chuyển cảnh, chẳng hạn như giá trị trung bình (mean), độ lệch chuẩn (standard deviation), hoặc độ lệch (skewness) của cường độ pixel trong frame. Đầu tiên, các đặc trưng này được tính toán cho từng frame, sau đó so sánh với frame trước đó bằng cách lấy tổng các chênh lệch tuyệt đối hoặc sử dụng một hàm

khoảng cách khác. Nếu tổng chênh lệch vượt quá một ngưỡng cụ thể, frame đó được coi là chuyển cảnh. Phương pháp này có thể nhận diện được cả những thay đổi nhỏ và tinh tế trong nội dung video, nhưng để đạt hiệu quả cao, ngưỡng cần được điều chỉnh cẩn thận. Ngoài ra, nó có thể trở nên chậm khi xử lý các video dài hoặc có nhiều frame.

- **Chênh lệch pixel**

Đây là phương pháp đơn giản và trực quan nhất trong số các kỹ thuật phát hiện chuyển cảnh. Nó so sánh trực tiếp giá trị của từng pixel giữa hai frame liên tiếp và tính tổng các chênh lệch tuyệt đối của chúng. Nếu tổng này vượt quá một ngưỡng nhất định, frame đó được xác định là chuyển cảnh. Phương pháp này rất nhanh và dễ triển khai trong thực tế, phù hợp với các hệ thống có tài nguyên hạn chế. Tuy nhiên, nó lại cực kỳ nhạy cảm với nhiễu (noise) và các thay đổi nhỏ không mong muốn trong video, đồng thời gặp khó khăn khi phát hiện các chuyển cảnh từ từ như hiệu ứng fade-in (mờ dần vào) hoặc fade-out (mờ dần ra).

- **Chênh lệch histogram**

Phương pháp này dựa trên việc xây dựng histogram của cường độ pixel (hoặc giá trị màu sắc) cho mỗi frame, sau đó so sánh histogram của hai frame liên tiếp. Các phép đo như khoảng cách Euclidean, Chi-square, hoặc khoảng cách Bhattacharyya được sử dụng để định lượng sự khác biệt giữa các histogram. Nếu sự khác biệt vượt quá ngưỡng, frame đó được xem là chuyển cảnh. So với phương pháp chênh lệch pixel, cách này chính xác hơn và ít bị ảnh hưởng bởi nhiễu, vì histogram phản ánh phân bố tổng thể của pixel thay vì giá trị riêng lẻ. Tuy nhiên, nó có thể không hiệu quả với các chuyển cảnh từ từ, nơi sự thay đổi histogram diễn ra không rõ rệt.

- **Thư viện SceneDetect**

SceneDetect là một công cụ mã nguồn mở được viết bằng Python, tích hợp nhiều thuật toán phát hiện chuyển cảnh, nổi bật nhất là thuật toán ContentDetector. Thuật toán này phân tích sự thay đổi nội dung giữa các frame dựa trên các đặc trưng hình ảnh và xác định chuyển cảnh khi sự thay đổi vượt quá ngưỡng do người dùng thiết lập. SceneDetect rất dễ sử dụng, cho phép tùy chỉnh ngưỡng và các tham số khác để phù hợp với từng loại video, từ video ngắn đơn giản đến các bộ phim phức tạp. Nó cũng linh hoạt nhờ khả năng kết hợp nhiều phương pháp khác nhau nhằm nâng cao độ chính xác. Tuy nhiên, việc cấu hình tối ưu có thể hơi phức tạp đối với người mới, và hiệu suất phụ thuộc vào các thư viện bên ngoài mà nó sử dụng.

Chạy lần lượt 5 phương pháp, mỗi phương pháp sẽ tạo ra 1 file txt chứa danh sách các điểm chuyển cảnh cùng với thời gian thực thi

Hãy đọc file .txt thật kỹ và tính toán để trả lời các câu hỏi trong file answer.txt

4. Kết quả cần đạt

- Chạy được tất cả các bước như yêu cầu và trả lời các câu hỏi.
- Cần nộp 1 file: trong thư mục: /home/student/labtainer_xfer/TÊN_BÀI_LAB (tên tài khoản.TÊN_BÀI_LAB.lab)
- Kết thúc bài lab: Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab: `stoplab stego_code_scenechange_compar`
- Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

- Sinh viên cần nộp file .lab để chấm điểm.
- Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh: `checkwork`
- Khởi động lại bài lab: Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh: `labtainer -r stego_code_scenechange_compare`