

PHÁT HIỆN CẢNH BẠO LỰC TRONG VIDEO

Trịnh Quang Trường - 18520393- CS114.K21

Tăng Năng Chung - 18520536- CS114.K21

Phan Lê Phú - 18521247- CS114.K21

Link GitHub: [Nhấn vào đây](#)

Tóm Tắt

Sử dụng TVL1 trích xuất chuyển động -> dùng Canny để phát hiện cạnh -> dùng closing để nối cạnh -> dùng drawcontours để lấp đầy lỗ trong vùng chuyển động -> dùng opening để xóa đường nét còn sót lại và thu được ảnh nhị phân

Trích xuất HOG từ ảnh vừa tìm được, ta thu được vector đặc trưng

Training trên nhiều model cuối cùng chọn SVM để nhận diện vì cho kết quả tốt nhất.

Bài toán

Input: một video bất kì

Output: phát hiện những cảnh đánh nhau (nếu có) trong video



Bình thường



Có bạo lực

Mục tiêu

Phát hiện những video chứa nội dung bạo lực (cảnh đánh nhau) .

Phát hiện và thông báo các vụ đánh nhau trên đường phố.

Là cơ sở để phát triển các ứng dụng cao hơn.

Ứng dụng

Phát hiện những video chứa cảnh đánh nhau để phân loại và cảnh báo cho người dùng trước khi xem video.

Sử dụng video thu thập được từ các camera đường phố để nhận diện, trong quá trình nhận diện nếu phát hiện bạo lực sẽ đưa ra thông báo cho chính quyền để kịp thời ngăn chặn và xử lý.

Mô tả dữ liệu

Dữ liệu nhóm tự thu thập bằng việc download các video có cảnh đánh nhau được camera quan sát ghi lại trên mạng xã hội như facebook, youtube.... Với tổng số khoảng gần 200 video.

Tiến hành chia video thành các đoạn clip nhỏ với 90 khung hình trên thời lượng 3s.

Sàng lọc, lựa ra những clip chỉ chứa cảnh đánh nhau và clip không chứa cảnh đánh nhau.

Mô tả dữ liệu

Tổng số clip sau khi sàng lọc từ việc cắt video là 817 clip trong đó:

- + 405 clip chỉ chứa cảnh đánh nhau.

- + 412 clip không chứa cảnh đánh nhau.

Gán nhãn 0 với clip đánh nhau và 1 với clip không đánh nhau.

Rút trích đặc trưng

Mỗi clip lấy 15 frame ảnh (cách 6 frame lấy một frame). Trích xuất Optical Flow giữa 2 frame cạnh nhau.



Rút trích đặc trưng

Tiến hành phát hiện cạnh bằng thuật toán Canny trên image Optical Flow. Sau đó nối liền những cạnh phát hiện được, lấp đầy những vùng có cạnh bao quanh thu được vùng chuyển động



Rút trích đặc trưng

Áp dụng vùng chuyển động như một vùng quan tâm (ROI). Tiến hành gắn vào ảnh xám.

Trích xuất HOG trên ảnh vừa gắn ROI.



Train model

Chia tập dữ liệu làm hai phần train và test theo tỷ lệ (8:2).

Vì đây là bài toán binary classification nên sử dụng các thuật toán sau để train model : `logisticRegression`, `GaussianNB`, `RandomForestClassifier`, `SVM`.

Lựa chọn model tối ưu nhất dựa trên kết quả thu được của từng model.

Kết quả, đánh giá

Kết quả trên tập dữ liệu 1

	precision	recall	fscore	support
LogisticRegression	0.886810	0.887131	0.886419	None
GaussianNB	0.759103	0.759381	0.758957	None
RandomForestClassifier	0.734528	0.727175	0.726810	None
SVM	0.876573	0.876918	0.876663	None

Kết quả trên tập dữ liệu 2

	precision	recall	fscore	support
LogisticRegression	0.872105	0.873515	0.872603	None
GaussianNB	0.675747	0.674303	0.674724	None
RandomForestClassifier	0.701454	0.695463	0.696201	None
SVM	0.884854	0.885173	0.885006	None

Kết quả trên cả hai tập dữ liệu 1 và 2

	precision	recall	fscore	support
LogisticRegression	0.754144	0.753318	0.752505	None
GaussianNB	0.610604	0.608937	0.608099	None
RandomForestClassifier	0.631559	0.627982	0.626364	None
SVM	0.786569	0.786585	0.786577	None

Nhận xét : Model SVM luôn cho kết quả cao nhất (fscore) ở từng tập dữ liệu. Do đó model SVM được sử dụng để tiến hành bước demo phía sau.

Khó khăn

Mô hình bị overfitting:

+ Nguyên nhân:

- Data thu thập qua internet nên có chất lượng thấp
- Nhiều cảnh quay hỗn loạn, đông người, khung hình quay bị giật liên tục...
- Phương pháp rút trích chưa liên kết được các frame trong cùng một clip

+ Hướng khắc phục:

- Lọc bỏ những video chất lượng thấp
- Sử dụng phương pháp rút trích Local HOG ([tham khảo thêm](#))
- Sử dụng mô hình Deep Learning

Demo

Link video demo: [Tai đây](#)

Link file colab chứa ứng dụng: [Tai đây](#)