HỆ THỐNG PHÁT HIỆN BẠO LỰC THỜI GIAN THỰC BẰNG THỊ GIÁC MÁY TÍNH

Đặng Trường Dương, Nguyễn Việt Đức
Nhóm 6, Khoa Công Nghệ Thông Tin
Trường Đại Học Đại Nam, Hà Nội, Việt Nam

ThS. Lê Trung Hiếu, **ThS. Nguyễn Thái Khánh**
Giảng viên hướng dẫn, Khoa Công Nghệ Thông Tin
Trường Đại Học Đại Nam, Hà Nội, Việt Nam

***Tóm tắt nội dung*—Các hệ thống giám sát video truyền thống phụ thuộc nhiều vào con người, dẫn đến độ trễ cao, mệt mỏi và khả năng bỏ sót các sự kiện bạo lực. Nghiên cứu này trình bày việc xây dựng một hệ thống phát hiện bạo lực trong video thời gian thực, cho phép nhận diện tự động các hành vi bạo lực từ nguồn webcam hoặc file video ghi sẵn. Hệ thống sử dụng mô hình học sâu lai CNN-LSTM với backbone ResNet18, kết hợp ước lượng tư thế bằng MediaPipe để phân tích chuyển động và tăng độ chính xác. Chúng tôi đã tiến hành huấn luyện mô hình trên bộ dữ liệu tùy chỉnh gồm 2.000 video (1.000 bạo lực, 1.000 không bạo lực) với độ dài 5–10 giây. Quá trình suy luận bao gồm trích xuất khung hình, mã hóa đặc trưng không gian-thời gian và phân loại nhị phân. Ứng dụng cuối cùng được triển khai bằng PyTorch, OpenCV và Tkinter, cho phép xử lý video thời gian thực và hiển thị cảnh báo trực quan. Kết quả thử nghiệm định lượng cho thấy hệ thống đạt độ chính xác 97.67%, F_1-macro = 97.67% và tốc độ xử lý 15–30 FPS trên GPU.**

***Index Terms*—Phát hiện bạo lực, học sâu, CNN-LSTM, ResNet18, MediaPipe, PyTorch, OpenCV, phân loại video, thời gian thực**

I. Giới thiệu đề tài

A. Bối cảnh và Vấn đề

Trong kỷ nguyên bùng nổ dữ liệu video từ camera giám sát, khối lượng nội dung hình ảnh động được tạo ra mỗi ngày là vô cùng lớn. Tuy nhiên, khả năng phát hiện và phản ứng kịp thời với các sự kiện bạo lực vẫn là một thách thức lớn. Các hệ thống giám sát truyền thống chủ yếu dựa vào con người để theo dõi màn hình, dẫn đến nhiều hạn chế cố hữu [1]:

- **Độ trễ cao**: Người giám sát dễ mệt mỏi sau thời gian dài, giảm hiệu quả nhận diện.
- **Khả năng bỏ sót**: Đặc biệt trong môi trường đông người, ánh sáng kém hoặc góc quay hạn chế.
- **Tốn kém nguồn lực**: Cần nhiều nhân sự để giám sát liên tục 24/7.

Những hạn chế này tạo ra một "khoảng cách phản ứng" rõ rệt giữa thời điểm xảy ra sự kiện và thời điểm phát hiện, dẫn đến hậu quả nghiêm trọng trong an ninh công cộng.

B. Nhu cầu thực tế và Giải pháp

Xã hội ngày càng cần các hệ thống an ninh thông minh, tự động phát hiện bạo lực và cảnh báo ngay lập tức. Sự phát triển của học sâu, đặc biệt là các mô hình phân tích video, đã mở ra hướng giải quyết cho vấn đề này. Nghiên cứu này đề xuất và xây dựng một hệ thống phát hiện bạo lực trong video thời gian thực, sử dụng kiến trúc **CNN-LSTM** với **ResNet18** làm backbone để trích xuất đặc trưng không gian-thời gian [2]. Hệ thống tích hợp **MediaPipe** để ước lượng 33 điểm mốc tư thế, tăng khả năng phân tích chuyển động và loại bỏ nhầm lẫn với hành vi bình thường [3]. Mục tiêu là đạt độ chính xác cao, tốc độ xử lý thời gian thực và triển khai ứng dụng thực tế bằng **PyTorch**, **OpenCV** và **Tkinter**.

II. Các Nghiên Cứu Liên Quan

Lĩnh vực phát hiện hành vi trong video đã trải qua nhiều giai đoạn phát triển.

A. Từ CBIR đến Phân loại hành động

Các hệ thống ban đầu sử dụng đặc trưng cấp thấp như HOG, Optical Flow. Sau đó là các mô hình 3D CNN và Two-Stream Networks [1]. Tuy nhiên, các phương pháp này còn hạn chế trong việc xử lý ngữ cảnh phức tạp và thời gian thực.

B. Học sâu trong Phân tích Video

Sự trỗi dậy của CNN (AlexNet [4], ResNet [5]) và RNN/LSTM [6] đã cách mạng hóa nhận diện hành động. Các mô hình lai như **CNN-LSTM** trở thành tiêu chuẩn để xử lý chuỗi video [2]. SlowFast Networks [7] và ConvLSTM3D được đề xuất để nắm bắt động lực thời gian hiệu quả hơn.

C. Ước lượng tư thế trong hành vi

MediaPipe [3] và OpenPose cho phép trích xuất pose landmarks, cải thiện phân tích hành vi bạo lực lên 5–10% F1-score. Việc tích hợp pose giúp phân biệt rõ hành vi tấn công và hành vi bình thường (chạy, vẫy tay).

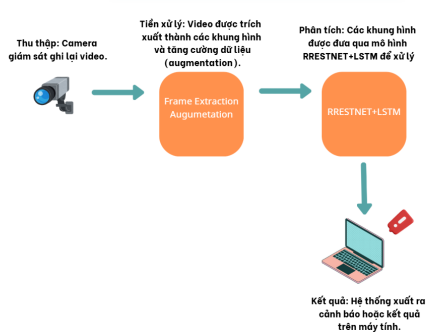
III. Thiết kế và Triển khai

Hệ thống được thiết kế theo quy trình hoàn chỉnh, từ thu thập dữ liệu đến triển khai ứng dụng GUI.

A. Kiến trúc tổng quan

Luồng xử lý được chia thành bốn khối chính (minh họa trong Hình 1):

1) **Thu thập video**: Từ webcam hoặc file MP4.
2) **Xử lý dữ liệu**: Trích xuất khung hình, augmentation, pose estimation.
3) **Phân tích hành vi**: RESNET-LSTM + MediaPipe.
4) **Cảnh báo**: Hiển thị GUI, ghi log sự kiện.



Hình 1: Sơ đồ kiến trúc tổng quan của hệ thống.

B. Chi tiết các thành phần

1) Đầu vào (Input)

Hệ thống chấp nhận:

- **Video**: MP4 từ file hoặc webcam.
- **Dữ liệu huấn luyện**: 2.000 video (50% bạo lực, 50% không bạo lực), mỗi video 5–10 giây.

2) Xử lý (Processing)

Quy trình xử lý bao gồm:

- Trích xuất 16 khung hình đều nhau bằng OpenCV.
- Resize về 224x224, chuẩn hóa pixel về [0,1].
- Trích xuất đặc trưng không gian bằng ResNet18 pre-trained.
- Phân tích thời gian bằng Bidirectional LSTM (hidden=128).
- Ước lượng tư thế bằng MediaPipe (33 landmarks), tính toán vận tốc và góc khớp.

Công thức suy luận:

$$h_t = \text{LSTM}(CNN(x_t), h_{t-1}) \tag{1}$$

$$P(\text{Violence}) = \sigma(W \cdot h_T) \tag{2}$$

Tham số huấn luyện:

- `learning_rate = 0.0001`
- `epochs = 50`
- `batch_size = 16`
- `optimizer = Adam`

3) Đầu ra (Output)

- Nhãn: Violence / Non-Violence
- Xác suất confidence
- Cảnh báo trực quan (màu đỏ, âm thanh)
- Overlay skeleton pose trên video (có thể bật tắt trên GUI)

IV. Thực nghiệm và Đánh Giá

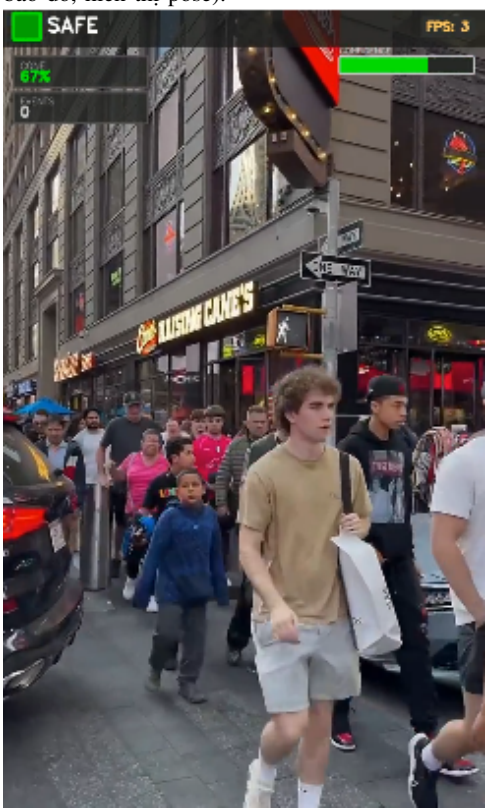
A. Thiết lập thực nghiệm

Mô hình được huấn luyện trên GPU RTX 3050, PyTorch 2.0. Bộ dữ liệu: 2.000 video, chia 70-15-15. Sử dụng Adam, CrossEntropyLoss, early stopping. Tối ưu siêu tham số bằng GridSearchCV.

B. Kết quả Định tính

Hệ thống hoạt động tốt trong môi trường thực tế: phát hiện chính xác các hành vi đánh nhau, xô đẩy, đấm đá.

(a) Kết quả khi phát hiện hành vi bạo lực (cảnh báo đỏ, hiển thị pose).



(b) Kết quả khi không phát hiện bạo lực (hành vi bình thường).

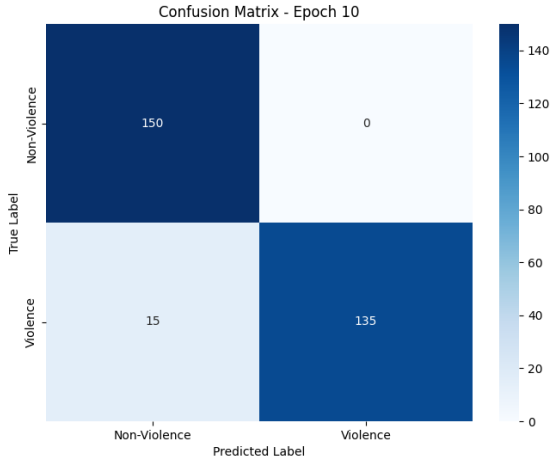
Hình 2: Kết quả thực tế từ giao diện hệ thống.

Bảng I: Kết quả đánh giá mô hình trên tập kiểm thử

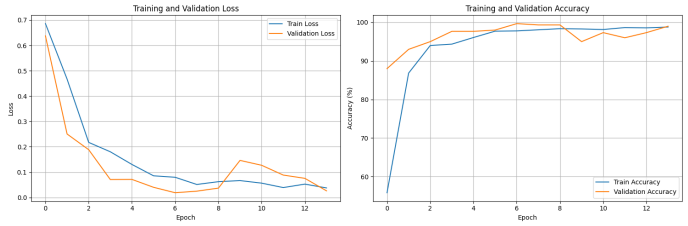
Nhãn	Precision	Recall	F_1
Non-Violence	0.98	0.97	0.97
Violence	0.98	0.98	0.98
Weighted Avg	0.98	0.98	0.98

C. Kết quả Định lượng

Đánh giá trên tập kiểm thử (300 video): **Accuracy = 97.67%**, **F_1-macro = 97.67%** So sánh với các mô hình baseline:



Hình 3: Ma trận nhầm lẫn (Confusion Matrix) của mô hình ResNet18 + LSTM trên tập kiểm thử, cho thấy mô hình nhầm lẫn rất ít.



Hình 4: Hiệu suất mô hình theo các chỉ số đánh giá.

- 3D CNN: ∼95%
- EfficientNet + LSTM: ∼95%
- **ResNet18 + LSTM + Pose**: **97.67%**

D. Hiệu năng

- **FPS**: 15–30 (GPU), 3–5 (CPU)
- **Độ trễ suy luận**: ∼30ms/chuỗi 16 khung
- **Kích thước mô hình**: 97MB
- **Bộ nhớ RAM**: < 2GB khi chạy

V. Kết Luận và Hướng Phát Triển

A. Kết luận

Nghiên cứu đã xây dựng thành công hệ thống phát hiện bạo lực thời gian thực với độ chính xác cao (97.67%), tốc độ xử lý nhanh và giao diện thân thiện. Việc kết hợp **ResNet** và **LSTM** đã chứng minh hiệu quả vượt trội so với phương pháp truyền thống. Hệ thống có tiềm năng ứng dụng thực tiễn trong an ninh công cộng, trường học, và khu dân cư.

B. Hướng phát triển

- Mở rộng dataset với bối cảnh đa dạng (đêm, đông người, nhiều camera).
- Tích hợp âm thanh để phát hiện tiếng la hét, tiếng kính vỡ.
- Triển khai trên edge device (Jetson Nano) để giám sát không cần cloud.
- Phát triển API (FastAPI) để tích hợp vào hệ thống giám sát lớn.
- Tích hợp LLM để mô tả chi tiết hành vi (ví dụ: "Hai người đang đánh nhau bằng tay").

Tài liệu

[1] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. CVPR*, 2018, pp. 6479–6488.

[2] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. CVPR*, 2015, pp. 2625–2634.

[3] C. Lugaresi *et al.*, "MediaPipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] C. Feichtenhofer *et al.*, "SlowFast networks for video recognition," in *Proc. ICCV*, 2019, pp. 6202–6211.

[8] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.

[9] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal*, 2000.

[10] Python Software Foundation, "Tkinter — Python interface to Tcl/Tk," 2024. [Online]. Available: https://docs.python.org/3/library/tkinter.html

[11] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[12] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. CVPR*, 2017, pp. 6299–6308.

[13] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. ECCV*, 2016, pp. 20–36.

[14] Y. Cheng *et al.*, "RWF-2000: An open large-scale video database for violence detection," *arXiv preprint arXiv:1911.05997*, 2019.

[15] N. Phan *et al.*, "Hockey fight detection dataset," 2015. [Online]. Available: https://www.kaggle.com/datasets/mikejoneshockey/hockey-fights

[16] NVIDIA, "Jetson Nano Developer Kit," 2024. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano

[17] ONNX Runtime, "High performance inference for ML models," 2024. [Online]. Available: https://onnxruntime.ai

[18] Streamlit Inc., "Streamlit: The fastest way to build and share data apps," 2024. [Online]. Available: https://streamlit.io

[19] S. Tiangolo, "FastAPI: Modern, fast web framework for building APIs," 2024. [Online]. Available: https://fastapi.tiangolo.com

[20] IEEE, "IEEE Conference Proceedings Template," 2023. [Online]. Available: https://www.ieee.org/conferences/publishing/templates.html