Tạp chí khoa học Trường Đại học Bạc Liêu
BLUNI Journal of Science
ISSN: 2734-973X
Số 6(12), 77-87 (2024)

PHÁT HIỆN HÀNH VI LEO RÀO QUA CAMERA GIÁM SÁT BẰNG GIẢI THUẬT HỌC SÂU

CLIMBING BEHAVIOR DETECTION VIA SURVEILLANCE CAMERA BY USING DEEP LEARNING

Ngô Đức Lưu*, Nguyễn Văn Trọng, Lê Văn Út

Trường Đại học Bạc Liêu
*ndluu@blu.edu.vn

Ngày nhận bài: 18/10/2024
Ngày chấp nhận đăng: 25/11/2024

Keywords: *climbing behavior, deep learning, supervision camera, YOLOv8.*

ABSTRACT

Security supervision systems have played more and more important roles in protecting properties and ensuring security of oganizations and individuals. Using camera for monitoring and recording images, videos in important areas becomes necessary and popular. However, to detect unusual actions with normal camera will take a lot of time and effort, but inefficient. Therefore, developing unusual behavior automatic detection systems via camera by using deep learning algorithms is a potential solution, attracting the attention of machine learning community. In this paper, we do research and develop a climbing behavior automatic detection system via camera by using YOLOv8 algorithm with data sets gathered from the Internet and reality (1000 images and 20 video-clips). Experimental results demonstrate that our surveillance system can automatically detect climbing behaviors with mAP accuracy of 79%.

Từ khóa: *hành vi leo rào, học sâu, camera giám sát, YOLOv8.*

TÓM TẮT:

Hệ thống giám sát an ninh ngày càng đóng vai trò quan trọng trong việc bảo vệ tài sản và đảm bảo an ninh của nhiều tổ chức, cá nhân. Việc sử dụng hệ thống camera giám sát để theo dõi và ghi lại hình ảnh, video ở các khu vực quan trọng đã trở nên cấp thiết và phổ biến. Tuy nhiên, các hệ thống này thông thường không thể tự động phát hiện được những hành vi bất thường của con người bằng từ hình ảnh thu nhận được. Do đó, các phương pháp học sâu đã được đề xuất để xây dựng hệ thống tự động phát hiện các hành vi bất thường của con người thông qua hình ảnh thu được từ camera giám sát. Trong nghiên cứu này, chúng tôi tiến hành nghiên cứu và đề xuất mô hình hệ thống tự động phát hiện hành vi leo rào qua camera giám sát bằng giải thuật học sâu của YOLOv8 với tập dữ liệu được thu thập từ Internet và thực tế (1000 ảnh và 20 phim ngắn). Kết quả thực nghiệm cho thấy hệ thống giám sát của chúng tôi có thể phát hiện hành vi leo rào với độ chính xác mAP là 79%.

1. Giới thiệu

Hệ thống giám sát an ninh ngày càng đóng vai trò quan trọng trong việc bảo vệ tài sản và đảm bảo an ninh của nhiều tổ chức, cá nhân. Việc sử dụng camera giám sát để theo dõi và ghi lại hình ảnh, video ở các khu vực quan trọng đã trở nên cấp thiết và phổ biến. Tuy nhiên, để phát hiện được những hoạt động bất thường bằng

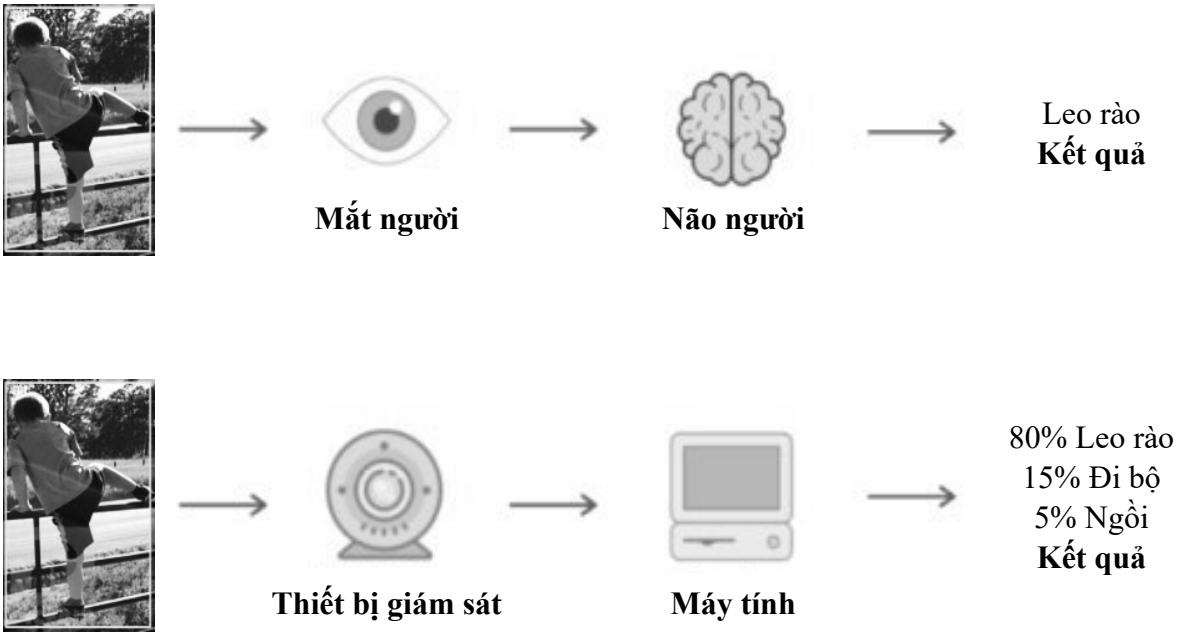
camera giám sát thông thường sẽ mất nhiều thời gian, công sức và đồng thời hiệu quả không cao. Do đó, việc phát triển hệ thống tự động phát hiện các hành vi bất thường thông qua camera giám sát bằng kỹ thuật học sâu là một giải pháp tiềm năng, đang được cộng đồng học máy quan tâm. Trong luận văn này, chúng tôi tiến hành nghiên cứu và xây dựng hệ thống tự động phát hiện hành vi leo rào qua camera giám sát bằng giải thuật học sâu của YOLOv8 (Solawetz, 2023). Kết quả thực nghiệm cho thấy hệ thống giám sát của chúng tôi có thể tự động phát hiện hành vi leo rào với độ chính xác cao.

2. Nghiên cứu liên quan

2.1 Thị giác máy tính

Thị giác máy tính (Stockman & Shapiro,. 2001) là một trong những lĩnh vực quan trọng của khoa học máy tính và trí tuệ nhân tạo. Thị giác máy tính cho phép máy tính và hệ thống lấy thông tin hữu ích từ hình ảnh kỹ thuật số, video và các đầu vào trực quan khác. Hình 1 giải thích quá trình con người ghi lại hình ảnh đối tượng thông qua võng mạc của mắt, sau đó bộ não tiếp nhận và nhận dạng ra đối tượng. Một người có thể dễ dàng nhận biết và phát hiện đối tượng trong bức ảnh một các chính xác vị trí của chúng. Tuy nhiên việc này lại khó khăn với máy tính, hệ thống phải tiếp nhận hình ảnh thông qua thiết bị ghi hình, "đọc" và "hiểu" hình ảnh dưới dạng ma trận số của tập hợp các điểm ảnh, sau đó được mô hình huấn luyện từ trước nhận dạng các đối tượng trong ảnh. Tuy vẫn chưa thể chính xác được như thị giác của con người nhưng đã có rất nhiều ứng dụng hữu ích, điểm hình như điểm danh bằng nhận dạng khuôn mặt, phát hiện các bệnh bằng chẩn đoán hình ảnh, công nghệ xe tự hành,...

Hình 1. Hệ thống thị giác của con người và máy tính



Năm 1966, dự án mang tên "Summer Vision Project" của Seymour Papert và Marvin Minsky (S.Papert,1966) đã mở đầu cho việc nghiên cứu về thị giác máy tính sau khi nỗ lực trong hai tháng để tạo ra một hệ thống máy tính có thể nhận dạng các vật thể trong ảnh. Từ đó đến nay, thị giác máy tính đã phát triển vượt bậc để thực hiện được những tác vụ phổ biến như:

- ***Phân loại ảnh*** **(**Lu & Weng, 2007)***:*** Phân loại hình ảnh cho phép máy tính quan sát và phân loại chính xác một hình ảnh thuộc loại nào. Ví dụ như bài toán phân loại trái cây, phân loại động vật, phân loại bệnh.

- ***Phát hiện đối tượng*** (Amit và cộng sự, 2021)***:*** Xác định và phân loại các đối tượng khác nhau trong hình ảnh hoặc video bằng cách tạo khung bao quanh các đối tượng. Ví dụ phát hiện các đối tượng tham gia giao thông.

- ***Theo dõi đối tượng*** (Yilmaz và cộng sự, 2006)***:*** Theo dõi đối tượng sử dụng mô hình học sâu để xác định và theo dõi các đối tượng. Ví dụ hệ giám sát an ninh tại các địa điểm công cộng để phát hiện và theo dõi các đối tượng quan tâm.

- ***Phân đoạn ảnh*** (Minaee và cộng sự, 2021)***:*** Xác định đối tượng bằng cách chia nhỏ đối tượng thành các vùng khác nhau dựa trên các điểm ảnh quan sát được. Khác với nhận dạng đối tượng, phân đoạn sẽ xác định hình dạng cụ thể của đối tượng.

- ***Truy xuất hình ảnh*** (Datta và cộng sự, 2008)***:*** Đó là khả năng tìm kiếm nhanh một hình ảnh cụ thể từ kho dữ liệu ảnh lớn nhằm phục vụ cho một công việc cụ thể nào đó.

2.2 Phát hiện đối tượng

Phát hiện đối tượng là một kỹ thuật quan trọng trong lĩnh vực Thị giác máy tính. Không chỉ nhận ra một đối tượng, phát hiện đối tượng sẽ vẽ các khung giới hạn xung quanh các đối tượng được phát hiện, từ đó cho phép xác định vị trí và nhãn của chúng. Phát hiện đối tượng được sử dụng rộng rãi trong nhiều lĩnh vực, tiêu biểu như trong công nghệ xe tự hành, tạo lộ trình di chuyển phù hợp bằng cách xác định các vị trí của phương tiện di chuyển, người đi đường, đường xá và các vật cản trong các ảnh được thu về từ video. Hay các hệ thống an ninh cần phát hiện các mục tiêu bất thường, ví dụ như các đối tượng xâm nhập bất hợp pháp.

2.3 Mô hình phát hiện đối tượng bằng YOLO

YOLO - "You only look once" (Redmon,. 2016) là mô hình phát hiện đối tượng một giai đoạn được giới thiệu lần đầu vào năm 2015 và hiện có nhiều phiên bản cải tiến theo thời gian, nổi bật như YOLO, YOLOv2, YOLOv3, YOLOv5, YOLOv7 và gần nhất là YOLOv8. Các mô hình YOLO không phải là thuật toán tốt nhất về độ chính xác nhưng luôn đảm bảo về tốc độ, phù hợp để xử lí các tác vụ thời gian thực. Trong phạm vi nghiên cứu này, chúng tôi sử dụng YOLOv8 để huấn luyện mô hình nhận dạng đối tượng leo rào bởi những tính năng ưu việt của nó. Cụ thể, YOLOv8 có nhiều cải tiến về kỹ thuật và kiến trúc mạng, vẫn đảm bảo mục tiêu tối ưu hóa hiệu suất tính toán và dễ sử dụng. Đặc biệt, cửa sổ trượt (sliding windows) không được sử dụng trong phiên bản YOLOv8. Thay vào đó, YOLOv8 sử dụng một lưới (grid) để phát hiện các đối tượng. Mỗi ô trong lưới *(grid cell)* sẽ dự đoán một hoặc nhiều hộp giới hạn *(bounding boxes)* cùng với xác suất và xác định lớp của đối tượng trong ô đó. Quá trình phát hiện trong YOLOv8 diễn ra như sau:

Chia hình ảnh thành các ô (grid cells): Hình ảnh đầu vào được chia thành một lưới ô có kích thước cố định. Mỗi ô sẽ đại diện cho một phần nhỏ của hình ảnh.

- Dự đoán bounding boxes và xác suất: Mỗi ô trong lưới dự đoán một hoặc nhiều bounding boxes cùng với xác suất và xác định lớp của đối tượng. Mỗi bounding box được biểu diễn bằng tọa độ (x, y, w, h) và xác suất của đối tượng trong ô đó.

- Chọn bounding boxes: Sử dụng các ngưỡng (thresholds) để lọc các bounding boxes có xác suất dự đoán thấp.

- Loại bỏ bounding boxes overlapping: Sử dụng Non-maximum Suppression (NMS) để loại bỏ các bounding boxes trùng lặp và chỉ chọn ra các bounding box tốt nhất.

▪ Quy trình này giúp YOLOv8 hoạt động hiệu quả và nhanh chóng trong việc phát hiện đối tượng trên hình ảnh mà không cần sử dụng sliding window như các phương pháp truyền thống khác.

Bounding box được dự đoán bởi mô hình YOLOv8 chứa các thông tin sau:

▪ Tọa độ của góc trái trên: Đây là tọa độ (x, y) của góc trái trên của bounding box, thường được biểu diễn dưới dạng tỷ lệ so với kích thước của hình ảnh gốc.

▪ Chiều rộng và chiều cao của bounding box: Đây là thông tin về kích thước của bounding box, xác định bởi chiều rộng (width) và chiều cao (height).

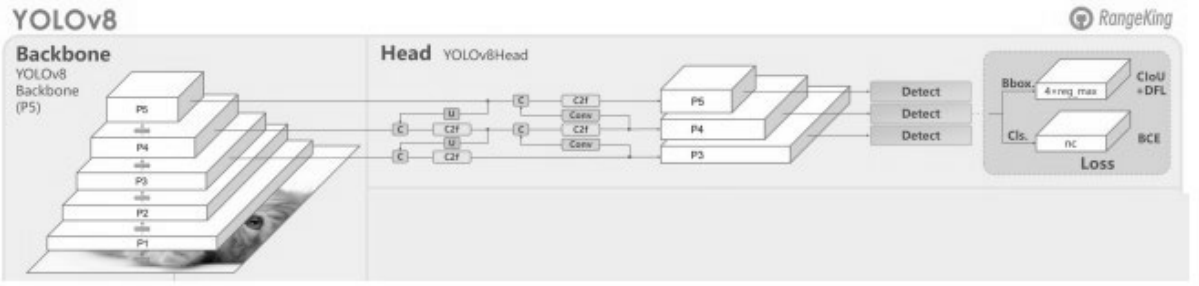
▪ Xác suất dự đoán của lớp đối tượng: Đây là xác suất của lớp đối tượng được dự đoán nằm trong bounding box.

▪ Các điểm mô tả bổ sung (nếu có): Đôi khi, các điểm mô tả (landmarks) hoặc các thuộc tính khác của đối tượng có thể được dự đoán cùng với bounding box.

▪ Các bounding boxes này được sử dụng để định vị và nhận diện các đối tượng trong hình ảnh, giúp mô hình YOLO có khả năng phát hiện nhanh chóng và chính xác các đối tượng mà không cần sử dụng phương pháp trượt cửa sổ như các phương pháp truyền thống

Kiến trúc mạng của YOLOv8 được trình bày như hình 2.

Hình 2. Kiến trúc của YOLOv8 (Solawetz, 2023)



YOLOv8 sử dụng một kiến trúc mạng mới, tối ưu hóa hơn về mặt hiệu suất tính toán và khả năng học sâu. Điều này giúp mô hình có thể học và dự đoán các đối tượng trong hình ảnh một cách hiệu quả hơn. YOLOv8 có độ chính xác cao hơn trong việc phát hiện và phân loại đối tượng, nhờ vào việc cải tiến các thuật toán và kỹ thuật huấn luyện. Điều này dẫn đến tỷ lệ phát hiện chính xác và phân loại chính xác hơn so với các phiên bản trước.

3. Phương pháp nghiên cứu

Trong phạm vi nghiên cứu này, chúng tôi sử dụng mô hình YOLOv8 xây dựng hệ thống phát hiện đối tượng có hành vi leo rào thông qua camera giám sát. Các bước nghiên cứu được tiến hành như sau:

▪ *Bước 1:* Thu thập dữ liệu.

▪ *Bước 2:* Gán nhãn dữ liệu.

▪ *Bước 3:* Huấn luyện mô hình với YOLOv8.

▪ *Bước 4:* Xây dựng hệ thống phát hiện hành vi leo rào với mô hình ở bước 3.

3.1 Thu thập dữ liệu

Chúng tôi tiến hành thu thập dữ liệu gồm 1000 hình ảnh (images) và 20 phim ngắn (video-clips). Trong tất cả ảnh và video ngắn này đều chứa duy nhất 1 khung ảnh đối tượng leo rào. Do đó, chúng tôi xem 1 ảnh hoặc video là một đơn vị dữ liệu để đánh giá kết quả thực nghiệm. Dữ liệu này được thu thập từ 2 nguồn:

▪ Từ Internet: Chúng tôi sử dụng công cụ image google để tìm và tải xuống 800 hình ảnh; sử dụng youtube để tìm và tải xuống 10 đoạn phim ngắn.

▪ Từ thực tế: Chúng tôi dựng hiện trường giả để tiến hành chụp 200 ảnh và quay 10 phim thực tế đối tượng leo rào tại Ký túc xá sinh viên của một trường đại học.

Dữ liệu này được thu thập trong nhiều ngữ cảnh khác nhau về thời gian trong ngày, điều kiện thời tiết, tư thế. Tập dữ liệu thực nghiệm có tất cả 1020 mẫu dữ liệu (samples) tương ứng với 1000 ảnh (được dùng để làm tập train và valid) và 20 phim ngắn (được dùng để làm tập test) và được chia ra thành 3 tập: tập train (800 ảnh), tập valid (200 ảnh) và tập test (gồm 20 phim ngắn) với kích thước như bảng 1.

Bảng 1. Tập dữ liệu thực nghiệm.

	Số lượng mẫu (samples)
Tập train	800
Tập valid	200
Tập test	20

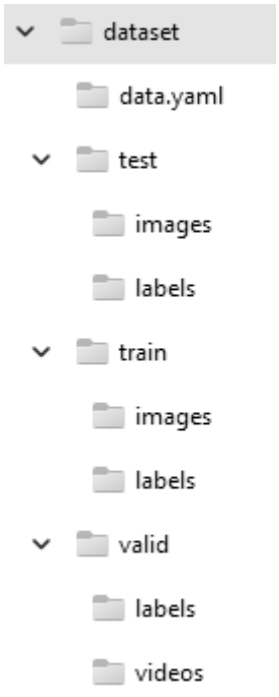
3.2 Gán nhãn dữ liệu

Từ tập dữ liệu hình ảnh thu thập được, chúng tôi sử dụng công cụ Roboflow (Alexandrova và cộng sự, 2015) để gán nhãn các đối tượng leo trèo rào, ví dụ như hình 3. Mỗi hình ảnh cần có thông tin về vị trí và loại đối tượng leo rào. Roboflow là một công cụ mạnh mẽ giúp bạn dễ dàng thu thập, gán nhãn và chuẩn bị dữ liệu cho việc huấn luyện mô hình YOLOv8. Toàn bộ dữ liệu được lưu trữ trên GoogleDrive để đảm bảo cho hệ thống hoạt động nhanh chóng và thông suốt.

Hình 3. Công cụ gán nhãn Roboflow

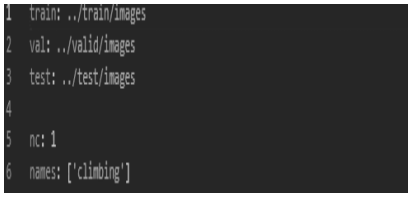
Kết quả của bước này là dữ liệu đã được chuẩn bị và tổ chức lưu trữ theo cấu trúc yêu cầu như hình 4.

Hình 4. Cấu trúc thư mục lưu trữ dữ liệu



Trong đó, tập tin data.yaml chứa thông tin về tập dữ liệu (hình 5).

Hình 5. Thông tin về tập dữ liệu



```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 1
names: ['climbing']
```

3.3 Huấn luyện mô hình với YOLOv8

Ở bước này, chung tôi tiến hành huấn luyện mô hình YOLOv8 trên Google Colab (Bisong, 2019) để tận dụng tối đa tài nguyên tính toán mạnh mẽ và khả năng lưu trữ của Colab. Ngoài ra, chúng tôi còn sử dụng tham số Checkpoint (Luo, 2000) để lưu trữ trạng thái của mô hình trong quá trình huấn luyện. Checkpoint bao gồm các thông tin về các trọng số và tham số hiện tại của mô hình tại một thời điểm cụ thể. Việc sử dụng checkpoint giúp chúng ta dễ dàng quản lý quá trình huấn luyện mô hình và có nhiều lợi ích, bao gồm khả năng phục hồi huấn luyện từ điểm đã lưu, theo dõi hiệu suất và chọn ra mô hình tốt nhất. Dưới đây là các bước chi tiết để bạn có thể thực hiện điều này:

▪ **Bước 1: Cài đặt môi trường**

!pip install ultralytics

▪ **Bước 2: Chuẩn bị dữ liệu**

- Tải dữ liệu của bạn lên Google Drive hoặc sử dụng dữ liệu có sẵn. Đảm bảo rằng dữ liệu được tổ chức theo cấu trúc yêu cầu của YOLOv8 và bạn có tập tin data.yaml chứa thông tin về tập dữ liệu.

- Sử dụng Mount Google Drive để truy cập dữ liệu:

from google.colab import drive

drive.mount('/content/drive')

▪ **Bước 3: Cấu hình và huấn luyện mô hình**

- Chỉnh sửa đường dẫn trong tập tin data.yaml để phù hợp với vị trí của dữ liệu trong Google Drive.

- Tạo và lưu trữ Checkpoint.

Dưới đây là mã để huấn luyện mô hình và lưu checkpoint (hình 6):

Hình 6. Mã huấn luyện checkpoint

```
from ultralytics import YOLO

# Tạo mô hình YOLOv8
model = YOLO('yolov8n.yaml')  # hoặc yolov8s.yaml, yolov8m.yaml, yolov8l.yaml, yolov8x.yaml tùy theo yêu cầu

# Huấn luyện mô hình và lưu checkpoint
model.train(data='/content/drive/MyDrive/path/to/data.yaml', epochs=50, imgsz=640, save_period=5)
```

Trong đó:

data='/content/drive/MyDrive/path/to/data.yaml': Đường dẫn tới tập tin data.yaml.

epochs=81: Số epoch để huấn luyện.

batch=6: Kích thước lô huấn luyện mỗi lần lặp.

lr0=0.01: Tỷ lệ dữ liệu học mỗi lần lặp.

imgsz=640: Kích thước hình ảnh đầu vào.

save_period=5: Lưu checkpoint sau mỗi 5 epoch.

▪ Bước 4: Lưu và quản lý checkpoint

- Các checkpoint sẽ được lưu trong thư mục runs/detect/train/weights/ trong môi trường Colab. Để lưu các checkpoint này vào Google Drive, bạn có thể sao chép chúng vào Drive (hình 7).

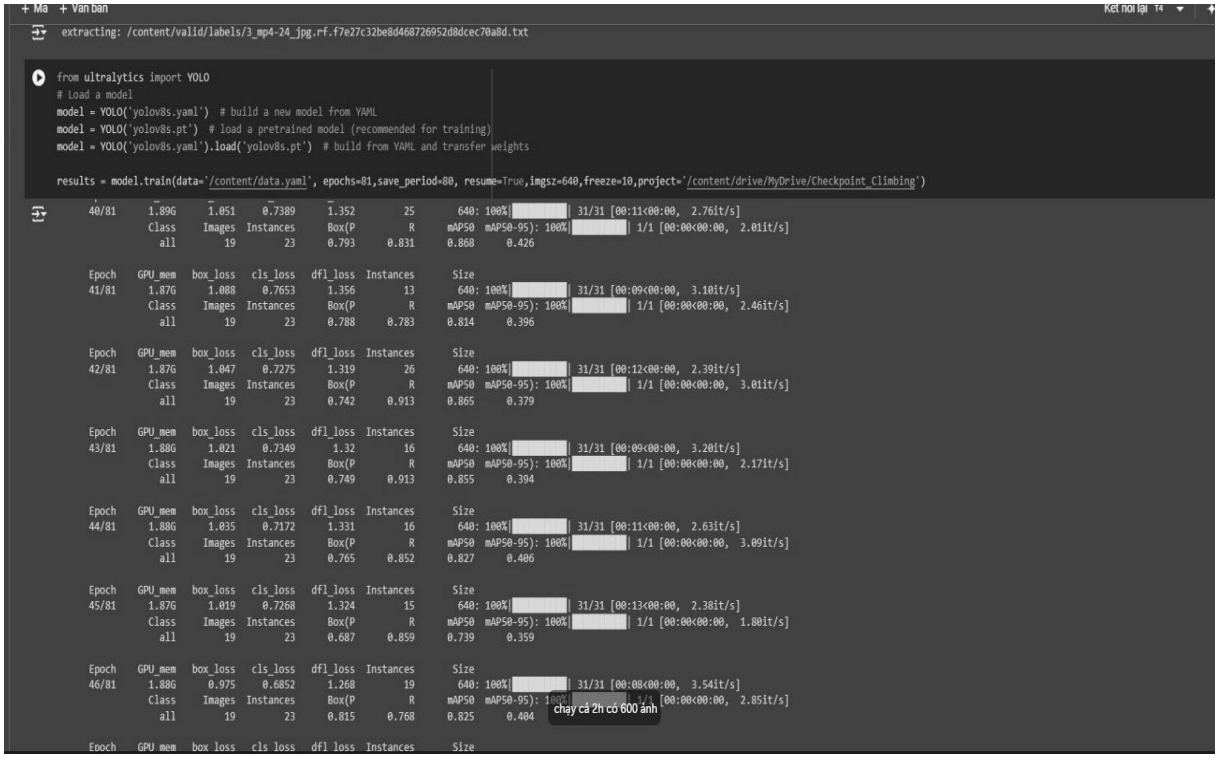
Hình 7. Đường dẫn thư mục chứa checkpoint

```
# Đường dẫn đến thư mục chứa checkpoint
source_path = '/content/runs/detect/train/weights/'
# Đường dẫn đến nơi lưu trữ trên Google Drive
destination_path = '/content/drive/MyDrive/path/to/save/checkpoints/'

# Sao chép toàn bộ thư mục
shutil.copytree(source_path, destination_path)
```

▪ Bước 5: Tải file mô hình tốt nhất sau khi huấn luyện ở Google Colab xong (hình 8).

Hình 8. Tải file mô hình tốt nhất (best.pt)

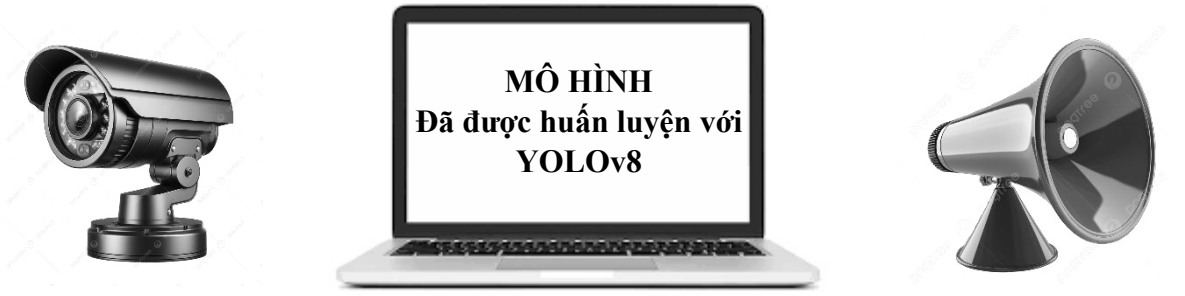


▪ **Bước 6: Sử dụng ngôn ngữ python để thực hiện đọc model và xử lý**

3.4 Xây dựng hệ thống phát hiện hành vi leo rào

Sơ đồ hoạt động của hệ thống phát hiện hành vi leo rào thông qua camera giám sát được mô tả như hình 9.

Hình 9. Sơ đồ hoạt động của hệ thống phát hiện hành vi leo rào



(1). Thu hình ảnh từ hàng rào

(2). Nhận dạng đối tượng leo rào trong video được thu thập từ camera giám sát

(3). Báo động khi phát hiện đối tượng leo rào

(1). Camera giám sát thu hình ảnh trực tiếp tại khu vực hàng rào và truyền dữ liệu này đến máy tính để xử lý.

(2). Máy tính với mô hình YOLOv8 đã được huấn luyện sẵn trên máy tính sẽ kiểm tra xem có đối tượng leo rào xuất hiện trong các hình ảnh do camera giám sát gửi đến hay không?

(3). Nếu mô hình phát hiện có đối tượng leo rào xuất hiện trong dữ liệu thì hệ thống sẽ phát tín hiệu báo động để nhân viên trực biết hoặc có thể gửi tin nhắn đến điện thoại di động của nhân viên, đồng thời lưu trữ các thông tin có liên quan tại thời điểm phát hiện đó.

4. Kết quả thực nghiệm

Hệ thống được thực nghiệm với mô hình YOLOv8 và các tham số epochs khác nhau: *epochs=[61, 71, 81]; batch=6; lr0=0.01; imgsz=640; save_period=5* trên tập dữ liệu Train để chọn lựa mô hình tốt nhất cho bước tiếp theo. Chúng tôi sử dụng các độ đo Precision, Recall và mAP để đánh giá mô hình (xem Bảng 2).

Bảng 2. Kết quả thực nghiệm trên tập Train.

	Precision	**Recall**	**mAP**
epochs =61	0.72	0.68	0.75
epochs = 71	0.78	0.70	0.79
epochs = 81	**0.87**	**0.72**	**0.87**

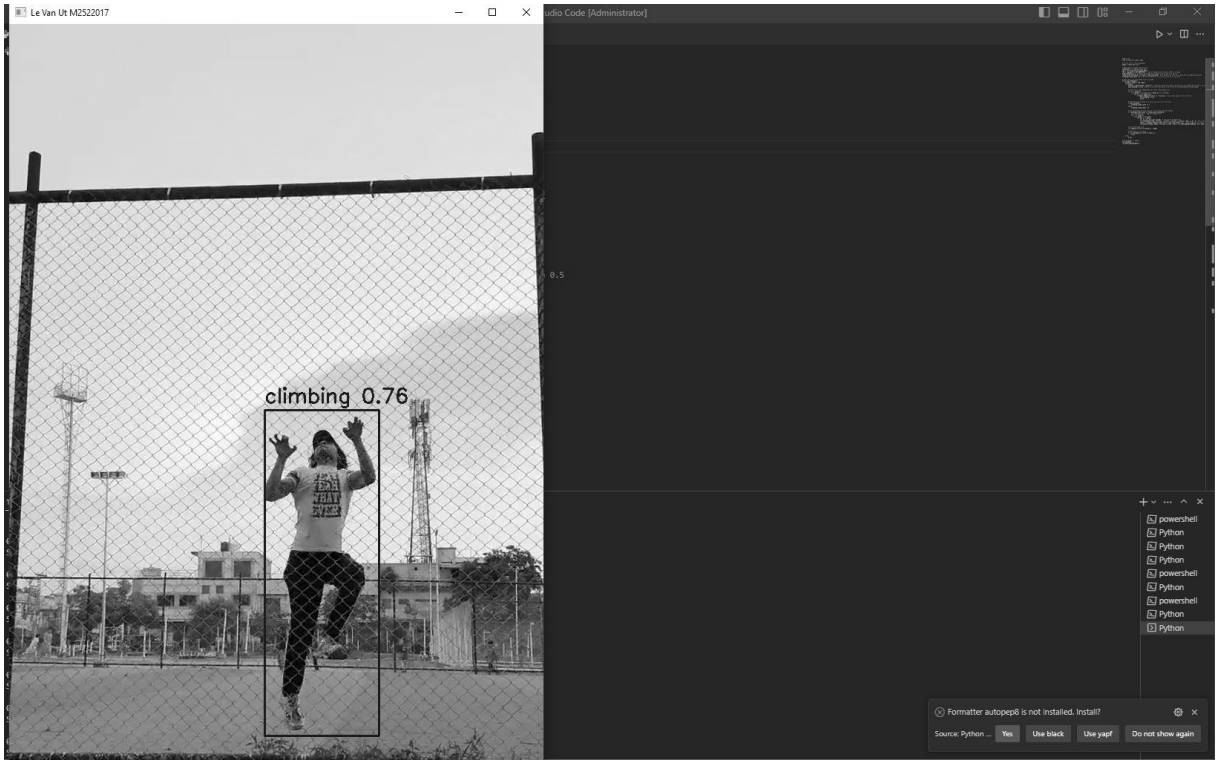
Từ kết quả ở Bảng 1, chúng tôi chọn mô hình với tham số: *epochs=81; batch=6; lr0=0.01; imgsz=640; save_period=5* để thực nghiệm trên 2 tập dữ liệu Valid và Test. Kết quả tổng hợp trên 3 tập dữ liệu được trình bày như ở Bảng 3.

Bảng 3. Tổng hợp kết quả thực nghiệm.

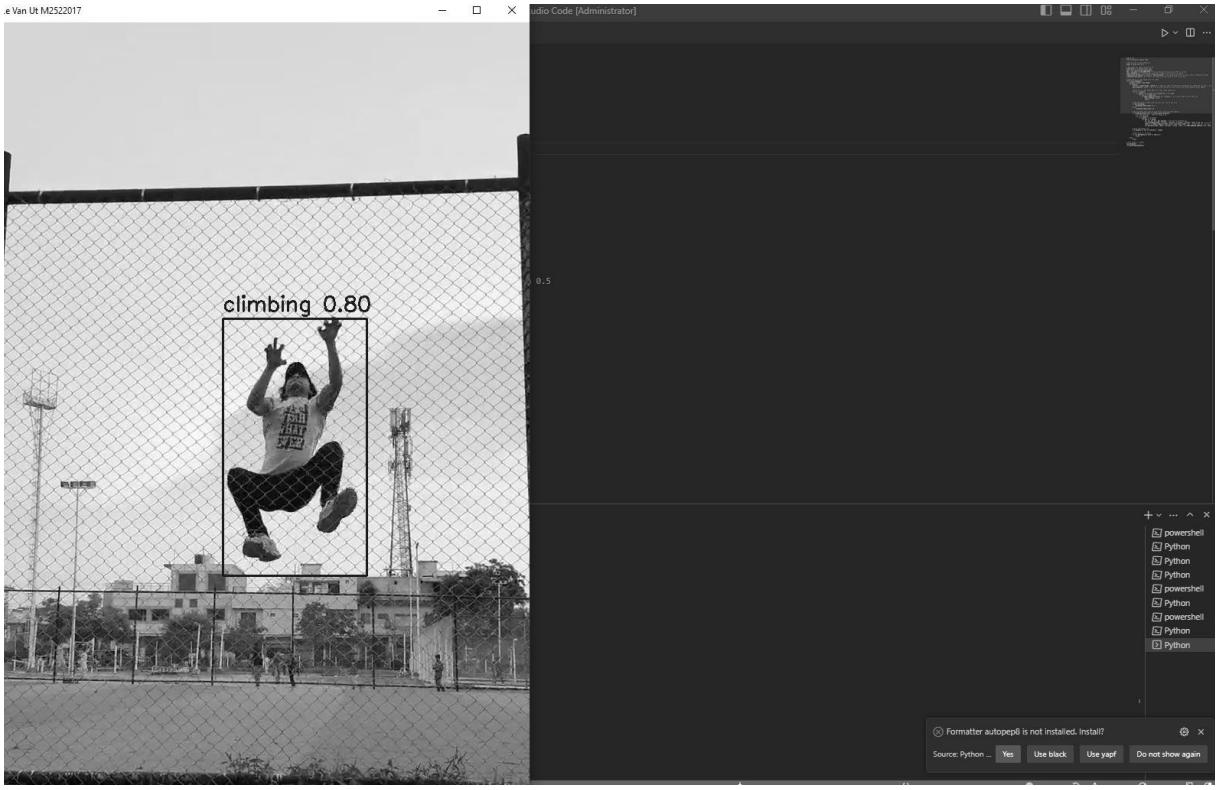
	Precision	**Recall**	**mAP**
Tập Train	0.87	0.72	0.87
Tập Valid	0.82	0.70	0.82
Tập Test	0.79	0.60	0.79

Sau đây là một số hình ảnh mà hệ thống đã phát hiện được hình 10, 11, 12.

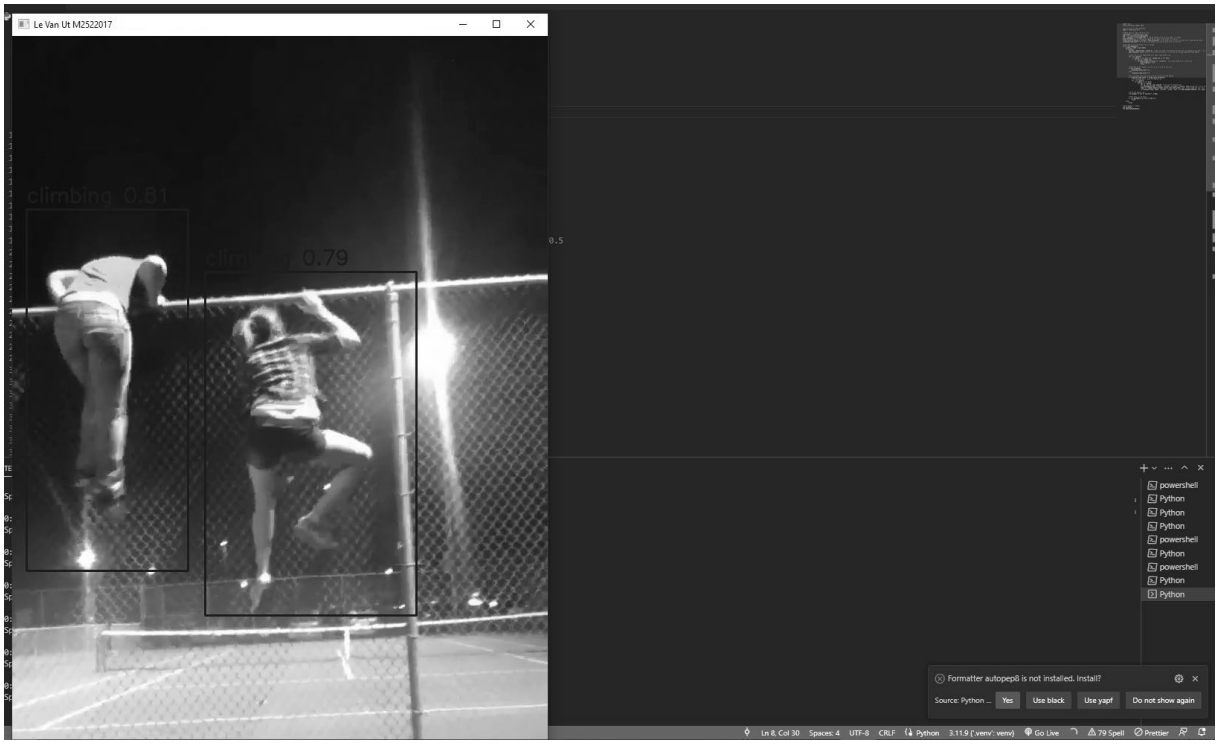
Hình 10. Kết quả chạy video 1 với độ chính xác 76%



Hình 11. Kết quả chạy video 2 với độ chính xác 80%



Hình 12. Kết quả chạy video 3 với độ chính xác 81% và 79%



5. Kết luận và hướng phát triển

Trong bài viết này, chúng tôi đã nghiên cứu và phát triển thành công hệ thống phát hiện hành vi leo rào thông qua camera giám sát bằng cách thu thập dữ liệu có liên quan, tiền xử lý dữ liệu, đưa vào mô hình học sâu YOLOv8 để huấn luyện và chọn mô hình tốt nhất để phát triển hệ thống với khả năng phát hiện hành vi leo rào khá cao. Tuy nhiên, do kích thước dữ liệu thu thập được còn hạn chế nên độ chính xác chưa thật sự tốt, cần phải được tiếp tục nghiên cứu và cải thiện trong tương lai. Ngoài ra, yếu tố môi trường như thời tiết, ánh sáng, tư thế phức tạp còn chưa được xử lý trước khi đưa vào huấn luyện, nên cũng ảnh hưởng đến kết quả thực nghiệm. Trong tương lai, chúng tôi sẽ tiếp tục nghiên cứu và đề xuất giải pháp để khắc phục vấn đề này.

TÀI LIỆU THAM KHẢO

Alexandrova, S., Tatlock, Z., & Cakmak, M. (2015, May). RoboFlow: A flow-based visual programming language for mobile manipulation tasks. In 2015 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5537-5544). IEEE.

Amit, Y., Felzenszwalb, P., & Girshick, R. (2021). Object detection. In Computer Vision: A Reference Guide (pp. 875-883). Cham: Springer International Publishing.

Bisong, E., & Bisong, E. (2019). Google colaboratory. Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners, 59-64.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys (Csur), 40(2), 1-60.

Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. International journal of Remote sensing, 28(5), 823-870.

Luo, Z. (2000). Checkpointing for workflow recovery. Proceedings of the 38th Annual on Southeast

Regional Conference - ACM-SE 38.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(7), 3523-3542.

Redmon, J. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition.

S. Papert (1966). The summer vision project. Artificial Intelligence Group. Version Memo. No. 100.

Solawetz, J. (2023). Francesco,"What is YOLOv8? The Ultimate guide". 2023-1-11.

Stockman, G., & Shapiro, L. G. (2001). Computer vision. Prentice Hall PTR.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. Acm computing surveys (CSUR), 38(4), 13-es.