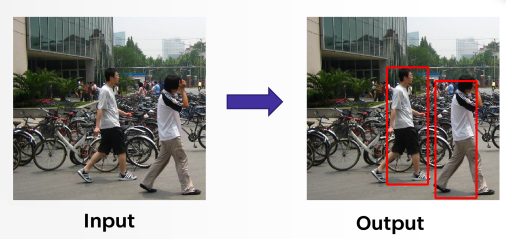
SỬ DỤNG HOG VÀ SVM TRONG NHẬN DIỆN NGƯỜI ĐI BỘ

*

1st Nguyễn Đỗ Quỳnh Như
Khoa Khoa học Máy tính, UIT
HCMC, Viet Nam
21521243@gm.uit.edu.vn

2nd Phạm Thị Trâm Anh
Khoa Khoa học Máy tính, UIT
HCMC, Viet Nam
21520146@gm.uit.edu.vn

***Tóm tắt nội dung*—Nhận diện đối tượng là một trong những nhiệm vụ quan trọng trong Thị giác máy tính. Và nhận dạng người đi bộ là bài toán phổ biến nhưng đầy thiết thức vì tính ứng dụng cao, ví dụ: hệ thống hỗ trợ lái xe nâng cao (ADAS) và video nâng cao hệ thống giám sát (AVSS). Mặc dù có nhiều mô hình học sâu và tiến tiến khác nhau trong lĩnh vực nhận diện ảnh, như CNN (Convolutional Neural Networks) và RNN (Recurrent Neural Networks), chúng em quyết định tập trung vào sử dụng HOG và SVM để giải quyết bài toán nhận diện người đi bộ vì đây là mô hình đơn giản và có hiệu suất tốt. Hệ thống của bọn em chạy trên Google Colab bản thường.**



Hình 1. Mô tả bài toán.

I. Giới thiệu

Nhận diện người đi bộ trong ảnh là một bài toán có ý nghĩa quan trọng trong lĩnh vực thị giác máy tính nói riêng và trí tuệ nhân tạo nói chung. Ứng dụng của bài toán này rộng rãi và có thể được áp dụng trong nhiều lĩnh vực như giám sát an ninh, quản lý giao thông, và tự động hóa. Tuy có sự phát triển mạnh mẽ của các mô hình học sâu và tiên tiến, nhưng trong báo cáo này, chúng em chọn sử dụng phương pháp HOG và SVM. Phương pháp HOG có khả năng trích xuất đặc trưng mạnh mẽ từ ảnh, trong khi SVM là một công cụ phân loại có hiệu suất tốt. Việc kết hợp HOG và SVM rất phổ biến và mang lại kết quả tốt cho bài toán. Trong quá trình thực hiện, chúng em đã điều chỉnh các tham số của HOG và SVM để đạt được hiệu suất tối ưu trong việc phân loại "người đi bộ" và "không phải người đi bộ". Sau đó, chúng em kết hợp các kỹ thuật xử lý ảnh cơ bản Sliding window, Image Pyramid và Non Max Suppression để nhận diện đối tượng với các bouding box cụ thể.

II. Các công trình liên quan

Có nhiều nghiên cứu đã sử dụng phương pháp HOG và SVM trong việc nhận diện đối tượng, bao gồm cả nhận diện người đi bộ. Một số công trình tiêu biểu trong lĩnh vực này bao gồm:

- N. Dalal và B. Triggs, "Histograms of Oriented Gradients for Human Detection" (2005): Đây là bài báo gốc đầu tiên giới thiệu phương pháp HOG cho việc nhận diện người.
- P. Felzenszwalb, D. McAllester, và D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model" (2008): Bài báo này mở rộng phương pháp HOG và đề xuất mô hình phân loại dựa trên nó. Chúng em đã xem xét và tham khảo các công trình trên để cải thiện và áp dụng phương pháp HOG và SVM một cách hiệu quả trong bài toán nhận diện người đi bộ.

III. Phương pháp

A. HOG (Histogram of Oriented Gradients):

1) Khái niệm: HOG (Histogram of Oriented Gradients) là một phương pháp mô tả hình ảnh trong lĩnh vực thị giác máy tính và nhận dạng đối tượng. Phương pháp này đã được giới thiệu bởi Navneet Dalal và Bill Triggs vào năm 2005 cho ứng dụng nhận dạng người đi bộ trong hình ảnh. HOG dựa trên việc tính toán các đặc trưng chủ yếu từ các cạnh và hướng gradient trong ảnh.

2) Tiền xử lý dữ liệu: : Trước khi tính toán HOG, hình ảnh đầu vào cần được tiền xử lý để chuẩn hóa và giảm nhiễu. Các bước tiền xử lý bao gồm[1]:

- Chuyển đổi hình ảnh sang không gian màu xám.
- Resize lại hình ảnh với kích thước 64x128.

3) Tính toán đặc trưng HOG: : Bước này bao gồm tính toán đặc trưng HOG cho từng ô vuông của hình ảnh (cell). Cụ thể:

- Chia hình ảnh thành các ô vuông không chồng nhau có kích thước nhỏ (6x6 pixel).
- Tại mỗi ô vuông, tính gradient theo chiều ngang (G_x) và chiều dọc (G_y) của các điểm ảnh.
- Tính toán magnitude (độ lớn) và hướng của gradient tại từng điểm ảnh trong mỗi ô vuông:
  Magnitude (M) của gradient tại mỗi điểm (i, j) được tính bằng công thức:
  $$M(i,j) = \sqrt{{G_x}^2(i,j) + {G_y}^2(i,j)}$$
  Hướng Θ của gradient tại mỗi điểm (i, j) được tính bằng công thức:
  $$\Theta(i,j) = \arctan G_y(i,j), G_x(i,j)$$
- Chuẩn bị histogram các hướng gradient cho mỗi ô vuông. Ví dụ: phân chia không gian góc (0 đến 180 độ) thành các bin (thùng) và đếm số lượng gradient rơi vào từng bin. Histogram được tính toán như sau:
  $$\text{h}(\Theta_{LB}) = \Theta_{LB} + m(x,y)(\frac{\Theta_{UB} - \Theta(x,y)}{d_\Theta}))$$
  $$\text{h}(\Theta_{UB}) = \Theta_{UB} + m(x,y)(\frac{\Theta(x,y) - \Theta_{LB}}{d_\Theta}))$$
- Gộp các histogram của các ô vuông liền kề thành một vector đặc trưng cuối cùng cho hình ảnh.

4) Chuẩn hóa HOG bằng L2: Sau khi tính toán vector đặc trưng HOG, chúng ta thực hiện chuẩn hóa bằng L2 để đảm bảo rằng các đặc trưng có cùng mức độ ảnh hưởng khi huấn luyện mô hình SVM. Chuẩn hóa bằng L2 thực chất chia tất cả các giá trị trong vector đặc trưng cho căn bậc hai của tổng bình phương các giá trị.

B. SVM

1) Khái niệm: SVM là một thuật toán học có giám sát, được sử dụng cho bài toán phân loại. Mục tiêu của SVM là tìm ra một siêu phẳng (hyperplane) trong không gian đặc trưng, tối đa hóa khoảng cách (margin) giữa hai lớp dữ liệu (ví dụ: người đi bộ và không phải người đi bộ). Khoảng cách từ các điểm dữ liệu gần nhất đến siêu phẳng được gọi là độ rộng của margin. Trong bài báo cáo này, chúng em sử dụng linear SVM để phân loại người đi bộ hoặc không phải người đi bộ.

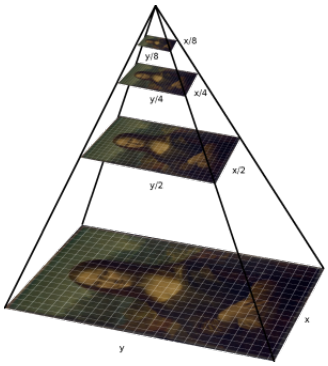
2) Công thức toán học: Đối với bài toán phân loại nhị phân, SVM tìm siêu phẳng có phương trình:
$$f(x) = w^T x + b$$
Trong đó:

- x là vector đặc trưng của một điểm dữ liệu
- w là vector trọng số (hướng của siêu phẳng)
- b là độ lệch (bias)
- $f(x)$ là giá trị dự đoán của điểm dữ liệu x (nếu $f(x) \geq 0$, điểm dữ liệu thuộc lớp 1, ngược lại thuộc lớp -1)

C. Sliding windows

1) Khái niệm: Trong thị giác máy tính, cửa sổ trượt (sliding window) [2] là một vùng hình chữ nhật có chiều rộng và chiều cao cố định "trượt" qua một hình ảnh. Mỗi cửa sổ trượt qua, chúng ta áp dụng bộ phân loại hình ảnh để xác định xem cửa sổ có đối tượng mà chúng ta quan tâm hay không — trong trường hợp này là người đi bộ. Kết hợp với image pyramids, chúng ta có thể nhận ra các đối tượng trong ảnh ở nhiều tỷ lệ và vị trí khác nhau. Kỹ thuật này, mặc dù đơn giản, nhưng đóng một vai trò cực kỳ quan trọng trong việc phát hiện đối tượng và phân loại hình ảnh.

D. Image pyramids



Hình 2. Một ví dụ về kim tự tháp hình ảnh. Tại mỗi lớp của kim tự tháp, hình ảnh được thu nhỏ lại và (tùy chọn) làm mịn

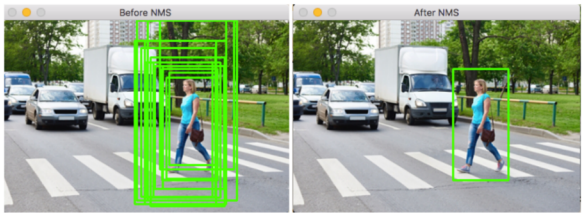
Image pyramid[3] biểu diễn hình ảnh ở nhiều tỉ lệ khác nhau. Việc sử dụng kỹ thuật này cho phép chúng ta tìm các đối tượng ở các tỷ lệ khác nhau của hình ảnh. Và khi kết hợp với cửa sổ trượt (sliding window), đối tượng sẽ được phát hiện ở vị trí khác nhau với kích thước lớn nhỏ. Phía đáy của kim tự tháp, chúng ta có hình ảnh với kích thước nguyên bản (về chiều rộng và chiều cao). Và ở mỗi lớp tiếp theo, nó được thay đổi kích thước (lấy mẫu phụ) và làm mịn tùy chọn (thường thông qua làm mờ Gaussian). Quy trình lấy mẫu phụ sẽ dừng cho đến khi đáp ứng một số tiêu chí dừng, thường là đã đạt đến kích thước tối thiểu

E. Non Maximum Suppression (NMS)

1) Khái niệm: Non Maximum Suppression (NMS)[4] là một kỹ thuật được sử dụng trong Thị giác máy tính. Việc sử dụng NMS giúp giảm bớt các bounding boxes xuống còn một số ít, hay thậm chí là phù hợp nhất. Chúng ta sử dụng các tiêu chí như xác suất hay IoU (Intersection over Union) để để chọn ra kết quả mong muốn.

2) Các bước thực hiện: Chúng ta nhận được một danh sách P gồm các bounding boxes dự đoán có dạng (x_1, y_1, x_2, y_2, c). Trong đó (x_1, y_1) và (x_2, y_2) là các góc của thực thể và c là đỉ lệ dự đoán (confidence score).

- Bước 1: Chọn dự đoán S có điểm tin cậy cao nhất và xóa nó khỏi P và thêm nó vào danh sách dự đoán.
- Bước 2: So sánh dự đoán S này với tất cả các dự đoán có trong P. Tính IoU của dự đoán S này với mọi dự đoán



Hình 3. Áp dụng NMS

khác trong P. Nếu IoU lớn hơn ngưỡng thresh cho bất kỳ dự đoán T nào có trong P, hãy xóa dự đoán T khỏi P.

- Bước 3: Nếu vẫn còn các dự đoán trong P, hãy quay lại Bước 1, nếu không, hãy trả về danh sách tiếp tục chứa các dự đoán đã lọc.

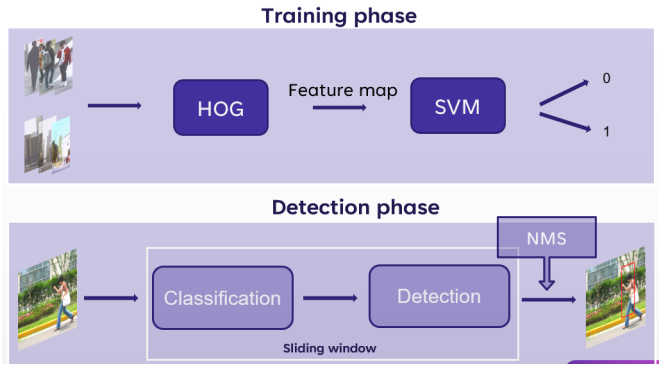
IV. Thực nghiệm

A. Bộ dữ liệu

Hình 4. Penn-Fundan Dataset.

Trong bài này, chúng em sử dụng bộ dữ liệu Penn-Fundan. Có 170 hình ảnh với 345 người đi bộ được gắn nhãn, trong đó 96 hình ảnh được chụp từ xung quanh Đại học Pennsylvania và 74 hình ảnh khác được chụp từ xung quanh Đại học Fudan. Để làm tăng sự đa dạng về dáng người đi bộ khi huấn, chúng em đã trộn lẫn Penn-Fundan với bộ dữ liệu khác[5].

B. Chạy thực nghiệm



Hình 5. Giai đoạn 1: Áp dụng HOG để trích xuất đặc trưng và mô hình phân loại SVM. Giai đoạn 2: Áp dụng các kỹ thuật xử lý ảnh cơ bản và mô hình huấn luyện HOG SVM để nhận diện người đi bộ trong ảnh.

1) Giai đoạn 1: Trong quá trình huấn luyện (Giai đoạn 1), kích thước của ảnh đầu vào là 64x128 với nhãn cụ thể (positive - người đi bộ hay negative - không phải người đi bộ). Sau đó, chúng em tiến hành rút trích đặc trưng bằng HOG với tham số:

- orientations = 8
- pixels per cell = (6, 6)
- cells per block = (2, 2)
- threshold = 0.3

Và chúng em đã sử dụng các mô hình phân loại ảnh khác nhau: SVM, Logistic Regression và KNN và chọn mô hình phù hợp nhất thông qua việc đánh giá độ chính xác (Accuracy).

	SVM	Logistic Regression	KNN
Accuracy	0.97	0.97	0.83

Hình 6. Bảng so sánh mô hình.

2) Giai đoạn 2: Trong giai đoạn 2, kích thước của ảnh đầu vào là bất kỳ và gồm có một hoặc nhiều người đi bộ. Chúng em cài đặt ô cửa sổ trượt (sliding window) có kích thước là 64x128 và image pyramid với downscale = 1.5 để phát hiện đối tượng trong ảnh. Và chúng em chỉ chấp nhận những bounding boxes có confidence score >= 0.5. Sau đó, áp dụng NMS để lọc ra ô cửa sổ (bounding box) phù hợp nhất.

C. Đô đo đánh giá

Khi thực hiện đánh giá bài toán nhận diện đối tượng, trước tiên chúng ta nên tìm hiểu các trường hợp dự đoán có thể xảy ra:

- True Positive (TP): Đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng).
- False Positive (FP): Đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai).
- False Negative (FN): Đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai).

Sau đây, nhóm chúng em sẽ trình bày cụ thể các độ đo được sử dụng trong bài.

1) Độ chính xác (Accuracy): Tỉ lệ các trường hợp được dự báo đúng trên tổng số các trường hợp được dự đoán. Accuracy giúp ta đánh giá hiệu quả dự báo của mô hình trên một bộ dữ liệu. Độ chính xác càng cao thì mô hình phân loại của chúng ta càng chuẩn xác. Công thức tính accuracy là

2) Độ chính xác Precision: Ý nghĩa của độ chính xác precision: Cho biết trong các trường hợp được dự báo là positive thì có bao nhiêu trường hợp là đúng. Và tất nhiên precision càng cao thì mô hình của chúng ta càng tốt. Công thức tính Precision

3) Recall: Recall cũng là một metric quan trọng, nó đo lường tỷ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive. Công thức tính của recall như sau:

V. Kết luận

Nhìn chung, sau khi áp dụng những kỹ thuật xử lý ảnh và mô hình máy học cơ bản, kết quả cho ra tốt trong các trường hợp:

- Đối tượng chiếm phần lớn diện tích ảnh.

```
precision = TP / float(TP + FP)
recall = TP / float(TP + FN)
```

```
print("Precision:", precision)
print("Recall:", recall)
```

```
precision 0.614406779661017
recall 0.34278959810874704
```

Hình 7. Kết quả thực nghiệm.

- Dáng đi nằm trong bộ dữ liệu huấn luyện.
- Chất lượng ảnh rõ nét

Vì thế, trong các trường hợp thực tế, mô hình nhận diện sai và thiếu đối tượng. Tương lai, nếu có thời gian, sẽ áp dụng nhiều mô hình học sâu khác nhau, mô hình hiện đại hơn và áp dụng trên bộ dữ liệu lớn hơn và đa dạng dáng người hơn để có thể hướng tới mục tiêu bao quát nhất có thể.

TÀI LIỆU

[1] Real-time HOG+SVM based object detection using SoC FPGA for a UHD video stream by Mateusz Wasala and Tomasz Kryjak.
[2] Rosebrock, A. (2015). Sliding Windows for Object Detection with Python and OpenCV - PyImageSearch.
[3] Rosebrock, A. (2015). Image Pyramids with Python and OpenCV - PyImageSearch.
[4] Non Maximum Suppression: Theory and Implementation in PyTorch. (2021).
[5] github.com/kbpranav/Pedestrian-Detection-using-CNN