

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Báo cáo bài tập lớn môn học:
Xử lý ảnh

EDGE DETECTION & HOG

Nhóm sinh viên thực hiện:

Tạ Quang Tùng

Phạm Minh Tâm

Lớp: KSTN-CNTT-K60

Giáo viên hướng dẫn:

PGS.TS Nguyễn Thị Hoàng Lan

Hà Nội, Ngày 26 tháng 12 năm 2018

Mục lục

1	Edge Detection (Xác định biên)	1
1.1	Định nghĩa và khái niệm	1
1.2	Các kỹ thuật phát hiện biên	1
1.3	Quy trình phát hiện biên	2
1.4	Phương pháp Gradient	2
1.5	Nhận xét các phương pháp Gradient	3
2	Histogram of Oriented Gradients (HOG)	4
2.1	Giới thiệu về HOG	4
2.2	Các bước thực hiện tính HOG	4
3	Phát hiện người sử dụng HOG và SVM	7
3.1	Sơ lược về hệ thống phát hiện người	7
3.2	Ứng dụng phát hiện người	7
3.3	Huấn luyện phát hiện đồng hồ	9

Tài liệu tham khảo	11
---------------------------	-----------

1 Egde Detection (Xác định biên)

1.1 Định nghĩa và khái niệm

Các khái niệm:

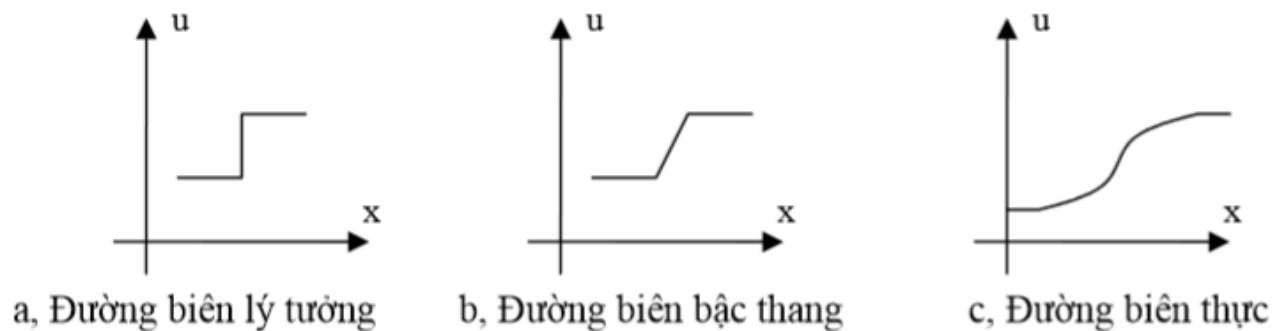
- Điểm biên: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu). Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng. [?] [?]
- Đường biên (đường bao - boundary): Tập hợp các điểm biên liên tiếp tạo thành một đường biên hay đường bao.

Ý nghĩa của đường biên trong xử lý:

- Đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh.
- Được sử dụng làm phân cách các vùng xám (màu) cách biệt.

Tuy nhiên ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách.

Mô hình biểu diễn đường biên theo toán học: Điểm ảnh có sự biến đổi mức xám $u(x)$ một cách đột ngột theo hình dưới.



Hình 1: Đường bao của ảnh

1.2 Các kỹ thuật phát hiện biên

Từ định nghĩa toán học của biên người ta sử dụng hai phương pháp phát hiện biên như sau

- Phương pháp phát hiện biên trực tiếp: Phương pháp này chủ yếu dựa vào sự biến thiên độ sáng của điểm ảnh để làm nổi biên bằng kỹ thuật đạo hàm.
 - Nếu lấy đạo hàm bậc nhất của ảnh: Ta có phương pháp Gradient
 - Nếu lấy đạo hàm bậc hai của ảnh: Ta có phương pháp Laplace.

Hai phương pháp này được gọi chung là phương pháp dò biên cục bộ. Ngoài ra, người ta còn sử dụng phương pháp “đi theo đường bao” dựa vào công cụ toán học là nguyên lý quy hoạch động và được gọi là phương pháp dò biên tổng thể. Phương pháp dò biên trực tiếp có hiệu quả và ít bị tác động của nhiễu.

- Phương pháp phát hiện biên gián tiếp: Nếu bằng cách nào đấy, chúng ta thu được các vùng ảnh khác nhau thì đường phân cách giữa các vùng đó chính là biên. Nói cách khác, việc xác định đường bao của ảnh được thực hiện từ ảnh đã được phân vùng. Phương pháp dò biên gián tiếp khó cài đặt nhưng áp dụng tốt khi sự biến thiên độ sáng nhỏ.

Chú ý: Kỹ thuật dò biên và phân vùng ảnh là hai bài toán đối ngẫu của nhau.

1.3 Quy trình phát hiện biên

- B1: Do ảnh ghi được thường có nhiễu, bước một là phải lọc nhiễu theo các phương pháp đã tìm hiểu ở các phần trước.
- B2: Làm nổi biên sử dụng các toán tử phát hiện biên.
- B3: Định vị biên. Chú ý rằng kỹ thuật nổi biên gây tác dụng phụ là gây nhiễu làm một số biên giả xuất hiện do vậy cần loại bỏ biên giả.
- B4: Liên kết và trích chọn biên.

1.4 Phương pháp Gradient

Theo định nghĩa về Gradient, nếu áp dụng nó vào xử lý ảnh, việc tính toán sẽ rất phức tạp. Để đơn giản mà không mất tính chất của phương pháp Gradient, người ta sử dụng kỹ thuật Gradient dùng cặp mặt nạ H_1 , H_2 trực giao (theo 2 hướng vuông góc). Nếu định nghĩa g_1 , g_2 là Gradient theo hai hướng x , y tương ứng thì biên độ $g(m, n)$ tại điểm (m, n) được tính:

$$g(m, n) = \sqrt{g_1^2(m, n) + g_2^2(m, n)} = A_0$$
$$\theta_r(m, n) = \arctan\left(\frac{g_2(m, n)}{g_1(m, n)}\right)$$

Để giảm độ phức tạp tính toán, A_0 được tính gần đúng như sau:

$$A_0 = |g_1(m, n)| + |g_2(m, n)|$$

1.4.1 Toán tử Robert (1965)

Với mỗi điểm ảnh $I(x, y)$ đạo hàm theo x , y được các giá trị tương ứng g_x , g_y :

$$g_x = I(x+1, y) - I(x, y+1)$$
$$g_y = I(x+1, y+1) - I(x, y)$$

Các công thức kể trên được cụ thể hóa bằng các mặt nạ theo chiều x và y tương ứng như sau:

$$H_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \downarrow$$

Hướng ngang (x) Hướng dọc (y)

1.4.2 Toán tử Sobel

Toán tử Sobel được Duda và Hart đặt ra năm 1973 với các mặt nạ tương tự như của Robert nhưng cấu hình khác như sau:

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Hướng ngang (x)

$$H_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Hướng dọc (y)

1.4.3 Toán tử Prewitt

Toán tử được Prewitt đưa ra vào năm 1970 có dạng:

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Hướng ngang (x)

$$H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Hướng dọc (y)

1.5 Nhận xét các phương pháp Gradient

- Toán tử Prewitt có thể tách sườn đứng tốt hơn toán tử Sobel, trong khi đó toán tử Sobel tách các sườn trên các điểm ở đường chéo tốt hơn.
- Toán tử Robert nhược điểm là nhạy với nhiễu.
- Để đạt được kết quả mong muốn các toán tử Gradient thường được dùng trước để làm sạch nhiễu.
- Các mặt nạ của các toán tử trên có kích thước 2×2 hoặc 3×3 chiều. Các mặt nạ có số chiều lớn hơn cũng được sử dụng. Ví dụ trong kỹ thuật phát hiện biên người ta dùng mặt nạ 5×5 cho toán tử Sobel:

$$H_x = \begin{bmatrix} 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \end{bmatrix} \quad H_y = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

Hình 4.7 Toán tử Sobel 5x5.

- Các toán tử kể trên đều sử dụng các mặt nạ theo hai chiều (x, y) tức là bốn hướng $(-x, y; -y, y)$.

2 Histogram of Oriented Gradients (HOG)

2.1 Giới thiệu về HOG

Histogram of Oriented Gradients (HOG) là một loại đặc trưng được dùng trong lĩnh vực thị giác máy tính và xử lý hình ảnh với mục đích phát hiện đối tượng [?]. Các khái niệm về HOG được nêu ra từ năm 1986 tuy nhiên cho đến năm 2005 HOG mới được sử dụng rộng rãi sau khi Navneet Dalal và Bill Triggs công bố những bổ sung về HOG. HOG tương tự như các biểu đồ edge orientation, scale-invariant feature transform descriptors (như sift, surf,...), shape contexts nhưng HOG được tính toán trên một lưới dày đặc các cell và chuẩn hóa sự tương phản giữa các block để nâng cao độ chính xác.

Ý tưởng chính trong đặc trưng HOG là hình dạng và trạng thái của vật thể được đặc trưng bằng sự phân bố của gradient và hướng của cạnh.

Đặc trưng HOG được tính trên một vùng, do sự biến thiên về màu sắc của các vùng là khác nhau, kết quả là mỗi vùng sẽ cho ta một vector đặc trưng của nó.

2.2 Các bước thực hiện tính HOG

Thuật toán tính toán đặc trưng HOG gồm 5 bước chính:

- B1: Chuẩn hóa hình ảnh trước khi xử lý.
- B2: Tính độ lớn và góc gradient của các thành phần ảnh.
- B3: Phân chia ảnh thành các cell và block. Lấy phiếu bầu để sinh ra vector đặc trưng cho từng cell.
- B4: Tính toán vector đặc trưng cho mỗi block.
- B5: Tổng hợp vector đặc trưng của block để tạo ra vector cuối cùng là đặc trưng HOG của ảnh

2.2.1 Chuẩn hóa hình ảnh

Đầu tiên cắt lấy phần ảnh muốn chọn sao cho các ảnh có kích thước cố định (VD: 64×128 pixels)

Tiếp theo là chuẩn hóa giá trị các pixel. Bước chuẩn hóa này hoàn toàn không bắt buộc, nhưng trong một số trường hợp, bước này có thể cải thiện hiệu suất của bộ mô tả HOG. Có ba phương pháp chuẩn hóa chính mà chúng ta có thể xem xét:

- Chuẩn hóa Gamma/lũy thừa: lấy $\log(p)$ của mỗi pixel p trong ảnh đầu vào.
- Chuẩn hóa căn bậc hai: lấy \sqrt{p} của mỗi pixel p trong ảnh đầu vào.
- Chuẩn hóa phương sai (Variance normalization): Đưa các giá trị pixel trong ảnh về dạng phân bố chuẩn $(0, 1)$.

2.2.2 Tính độ lớn và góc gradient của các thành phần ảnh

Có thể sử dụng bộ lọc $D_x = [-1 \ 0 \ 1]$, $D_y = [-1 \ 0 \ 1]^T$. Hoặc phổ biến hơn là bộ lọc Sobel.

Xác định góc của vector gradient, có thể xác định theo kiểu “có dấu” hoặc “không dấu”.

Với ảnh màu:

- Tính gradient theo chiều x , y cho mỗi kênh màu (RGB có 3 kênh màu).
- Với mỗi kênh màu, chọn gradient lớn nhất, lấy độ lớn của vector đó và góc là góc của vector lớn nhất đó.
- Mỗi pixel khi đó chỉ còn biểu diễn bởi 2 giá trị.

2.2.3 Phân chia ảnh thành các cell và block. Lấy phiếu bầu để sinh ra vector đặc trưng cho từng cell

Mỗi hình ảnh được phân chia thành nhiều khối, các khối này có thể đặt chồng lên nhau, khoảng cách giữa hai khối liên tiếp là bằng nhau.

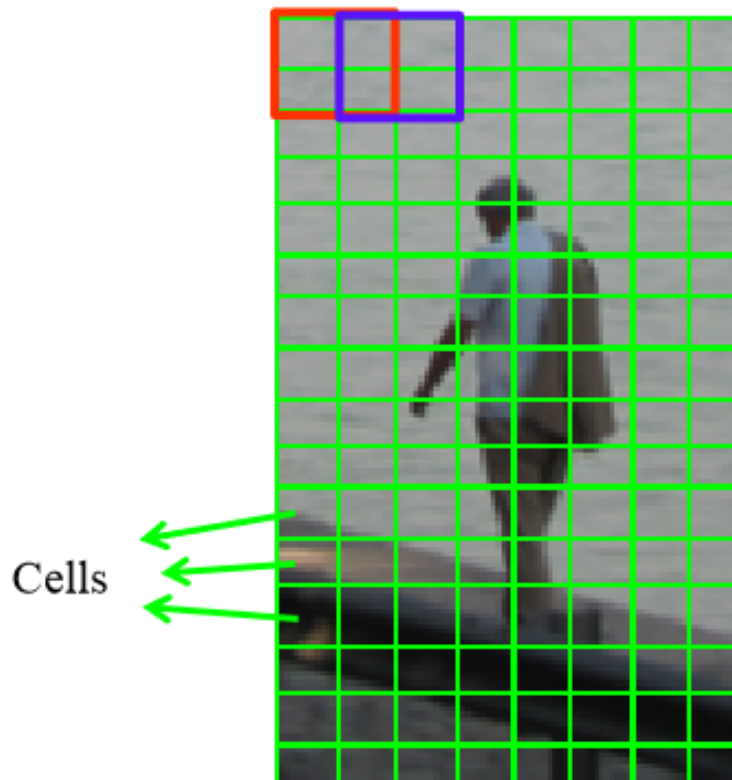
Mỗi khối gồm nhiều ô. Mỗi ô có kích thước bằng nhau, số lượng ô trong các khối là bằng nhau.

Ví dụ:

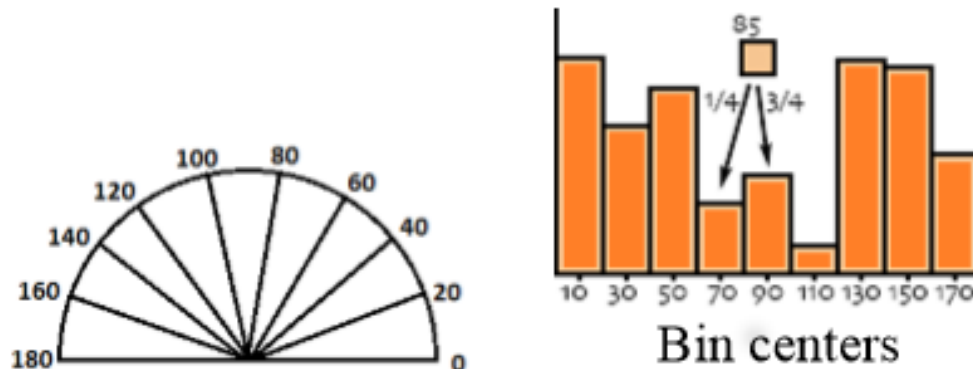
Nếu ta có một hình ảnh 128 x 128 với

pixel_per_cell = 4 x 4 thì sẽ có 32 x 32 = 1024 cell,

pixel_per_cell = 32 x 32, sẽ có 4 x 4 = 16 cell.



Với mỗi ô (cell), ta chia không gian góc thành các bin (thường là 9 ứng với mỗi bin là góc 20 độ).
Mỗi pixel trong ô (cell) được vote vào biểu đồ, trọng số của mỗi vote chính là cường độ gradient tại pixel đó.



2.2.4 Tính toán vector đặc trưng cho mỗi block

Ghép vector đặc trưng của từng cell lại với nhau và chuẩn hóa sẽ cho ta vector đặc trưng của block.

Chuẩn hóa có thể theo chuẩn L1 hoặc L2.

Ví dụ một block có kích thước 16x16 pixel:

Cell kích thước 8 × 8 pixel, 9 bin gradient.

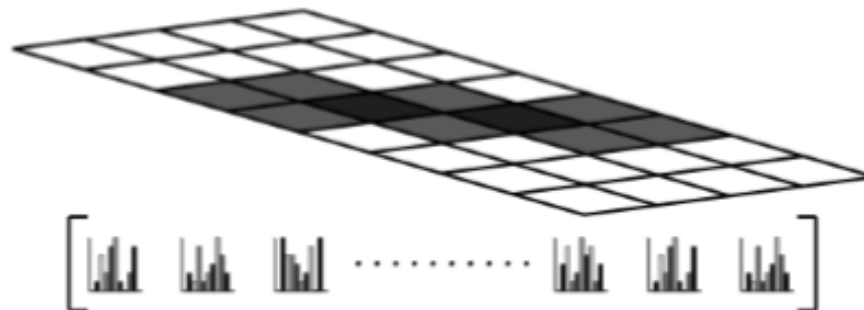
Mỗi block gồm 4 cell.

Vector đặc trưng của mỗi block gồm $9 \times 4 = 36$ thành phần.

2.2.5 Tính toán vector đặc trưng của toàn bộ ảnh

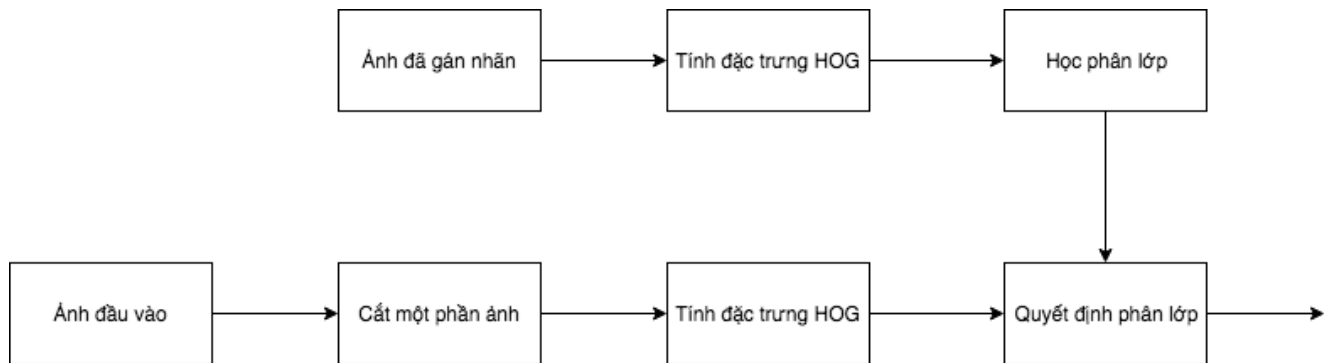
Ghép vector đặc trưng của từng block lại với nhau và chuẩn hóa sẽ cho ta vector đặc trưng của ảnh.

VD: Ảnh 64×128 , cell 8×8 , block 16×16 , stride 8×8 thì vector đặc trưng của toàn ảnh có $7 \times 15 \times 36 = 3780$ chiều.



3 Phát hiện người sử dụng HOG và SVM

3.1 Sơ lược về hệ thống phát hiện người



Để phát hiện một vật thể trong ảnh, đầu tiên cần phải huấn luyện cho hệ thống. Quy trình huấn luyện [?]:

- Dữ liệu huấn luyện là một tập các ảnh của vật thể và ảnh không chứa vật thể đã được gán nhãn với nhãn 1 là ảnh vật thể, nhãn 0 là ảnh không chứa vật thể.
- Tính đặc trưng HOG của tập ảnh. Sử dụng hàm **HOGDescriptor()** trong thư viện **opencv**, với ngôn ngữ **python**.
- Khi đã có được tập đặc trưng HOG của các ảnh và nhãn tương ứng, tiến hành học phân lớp SVM với tập dữ liệu trên. Sử dụng hàm **svm.LinearSVC()** của thư viện **sklearn**. Đầu vào của hàm là mảng các đặc trưng HOG và nhãn tương ứng của chúng. Đầu ra là một vector phân lớp SVM.

Đầu vào của quá trình huấn luyện là tập các ảnh và nhãn tương ứng của mỗi ảnh. Đầu ra là một vector phân lớp SVM (hay vector tham số của mặt siêu phẳng).

Quy trình phát hiện ảnh:

- Đọc ảnh cần xử lý.
- Đưa vector phân lớp SVM vào hệ thống bằng hàm **setSVMDetector()**.
- Tiến hành phát hiện đối tượng trong ảnh bằng hàm **detectMultiScale()**.

Đầu vào của quá trình phát hiện ảnh là một ảnh. Đầu ra là tập vị trí các hình chữ nhật bao quanh đối tượng cần phát hiện.

3.2 Ứng dụng phát hiện người

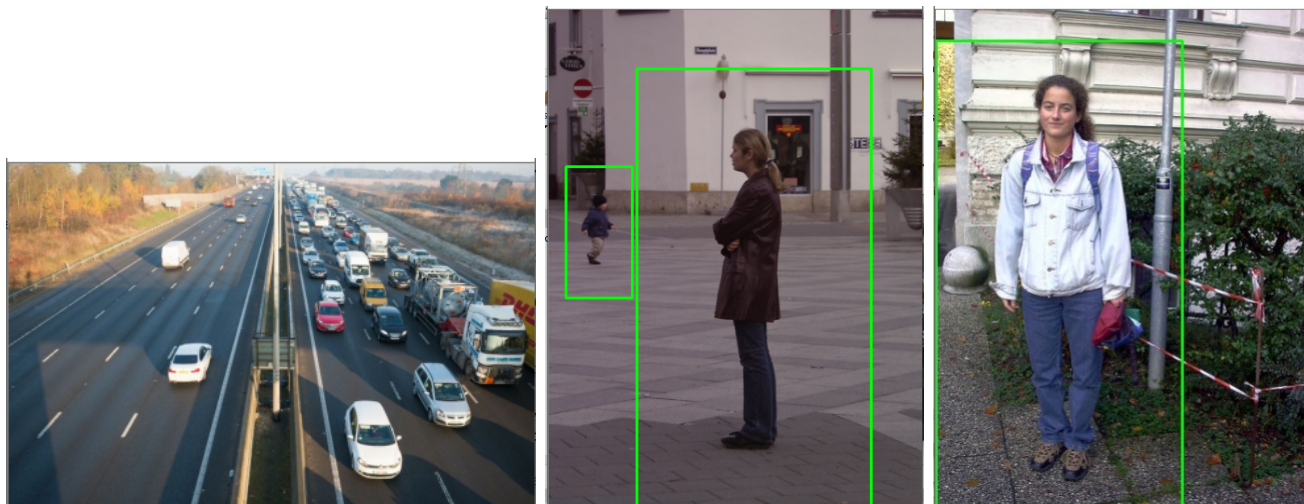
Phát hiện người sử dụng vector được huấn luyện sẵn của thư viện **OpenCV** được lấy từ hàm **cv2.HOGDescriptor_getDefaultPeopleDetector()** [?]. Với đặc trưng HOG được tính theo bộ tham số:

<code>blockSize = (16, 16)</code>	-Kích thước mỗi khối.
<code>blockStride = (8, 8)</code>	-Khoảng cách giữa hai khối liền nhau.
<code>cellSize = (8, 8)</code>	-Kích thước mỗi ô.
<code>nbins = 9</code>	-Số lượng tập các góc được chia.

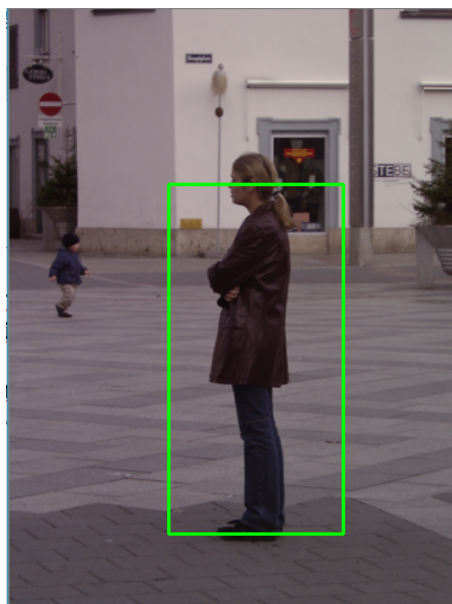
Dùng hàm **detectMultiScale()** để xác định người trong ảnh [?].
 Với bộ tham số:

<code>winStride = (8, 8)</code>	-Mức độ dịch của cửa sổ trượt.
<code>scale = 1.05</code>	-Tỉ lệ thu nhỏ ảnh.

Kết quả như hình:



Nếu tăng `winStride = (16, 16)` hoặc tăng `scale` lên 1.5, thu được kết quả có sự thay đổi như hình:



Nhận xét:

- Nếu winStride hoặc scale quá lớn, một vài đối tượng sẽ bị bỏ qua – không được phát hiện.
- Nếu winStride hoặc scale quá nhỏ, thời gian chạy thuật toán tăng lên.

3.3 Huấn luyện phát hiện đồng hồ

3.3.1 Huấn luyện hệ thống

Dữ liệu huấn luyện là tập các ảnh đồng hồ và không phải đồng hồ.

Ảnh đồng hồ:



Ảnh không phải đồng hồ:



Ảnh đồng hồ có 55 ảnh, ảnh không chứa đồng hồ có 258 ảnh.

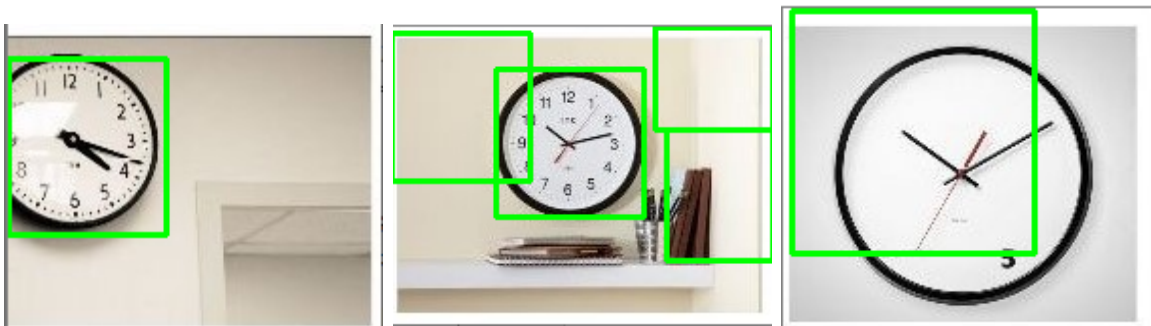
Các ảnh được đưa về cùng kích thước (64,64) trước khi tính đặc trưng HOG và tiến hành huấn luyện mô hình.

3.3.2 Kết quả

Kích thước ảnh được tính HOG là 64×64 . Với tham số tính HOG:

blockSize = (16, 16)	-Kích thước mỗi khối.
blockStride = (8, 8)	-Khoảng cách giữa hai khối liền nhau.
cellSize = (8, 8)	-Kích thước mỗi ô.
nbins = 9	-Số lượng tập các góc được chia.

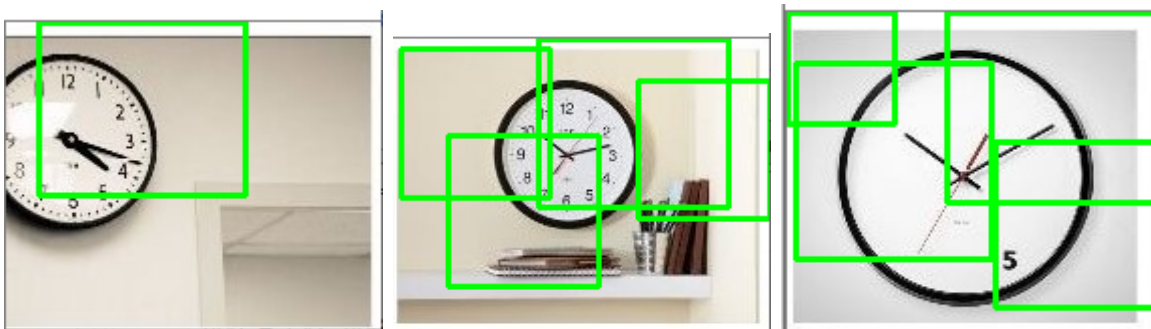
Kết quả thu được:



Với tham số tính HOG:

blockSize = (32, 32)	-Kích thước mỗi khối.
blockStride = (8, 8)	-Khoảng cách giữa hai khối liền nhau.
cellSize = (8, 8)	-Kích thước mỗi ô.
nbins = 9	-Số lượng tập các góc được chia.

Kết quả thu được:



Nhận thấy với blockSize lớn hơn, chương trình đưa ra nhiều ô không đúng vị trí đối tượng.

Nhận xét: Chương trình chạy chưa tốt, còn phát hiện sai nhiều. Nguyên nhân do bộ dữ liệu huấn luyện quá nhỏ, không tổng quát hết được các trường hợp.

Tài liệu tham khảo

- [1] Histogram of Oriented Gradients and Object Detection - PyImageSearch. <https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection>.
- [2] HOG detectMultiScale parameters explained - PyImageSearch. <https://www.pyimagesearch.com/2015/11/16/hog-detectmultiscale-parameters-explained>.
- [3] Pedestrian Detection OpenCV - PyImageSearch. <https://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv>.
- [4] Slide Xử lý ảnh, PGS.TS Nguyễn Thị Hoàng Lan.
- [5] Tìm hiểu về hog(histogram of oriented gradients). <https://viblo.asia/p/tim-hieu-ve-hoghistogram-of-oriented-gradients-m68Z0wL6KkG>.
- [6] Xử lý ảnh, Học viện công nghệ bưu chính viễn thông, 2006.