

HOG - Histogram of Oriented Gradient

Van-Thinh Vo

May 2025

Mục lục

1	Giới thiệu	2
2	Triển khai thuật toán	2
2.1	Tiền xử lý	2
2.2	Tính gradient của ảnh	2
2.3	Tìm histogram cho từng ô (cell)	3
2.4	Chuẩn hóa khối (block)	5
2.5	Tính toán vector đặc trưng	5
3	HOG-SVM cho bài toán phân loại trên tập CIFAR-10	5

Nếu có bất kỳ sai sót nào, mọi người có thể phản hồi qua email thinh.vovan@hcmut.edu.vn để hỗ trợ mình với nha :v

1 Giới thiệu

Histogram of Oriented-Gradient (HOG) là một "feature descriptor", là một thuật toán trong xử lý ảnh và thị giác máy tính, giúp trích xuất các đặc trưng của ảnh và thường được sử dụng trong các bài toán về nhận diện đối tượng (Object Recognition) hay xác định đối tượng (Object Detection). Kỹ thuật này được công bố lần đầu năm 2005 tại hội nghị hàng đầu thế giới *Computer Vision and Pattern Recognition* của *Navneet Dalal, Bill Triggs* [1].

Thay vì dựa vào giá trị của các pixel, HOG tính toán trên phân phối về cường độ gradient (intensity gradient) hay hướng cạnh (edge direction) để mô tả thông tin về cấu trúc, góc, cạnh của đối tượng - những phần chứa nhiều đặc trưng đáng giá hơn là phần phẳng. Ví dụ, trong trường hợp ta cần nhận diện một chiếc cốc áo, rõ ràng thông tin về cấu trúc như hình tròn, có các lỗ nhỏ để luồn chỉ,... là hữu ích hơn thông tin về màu sắc.

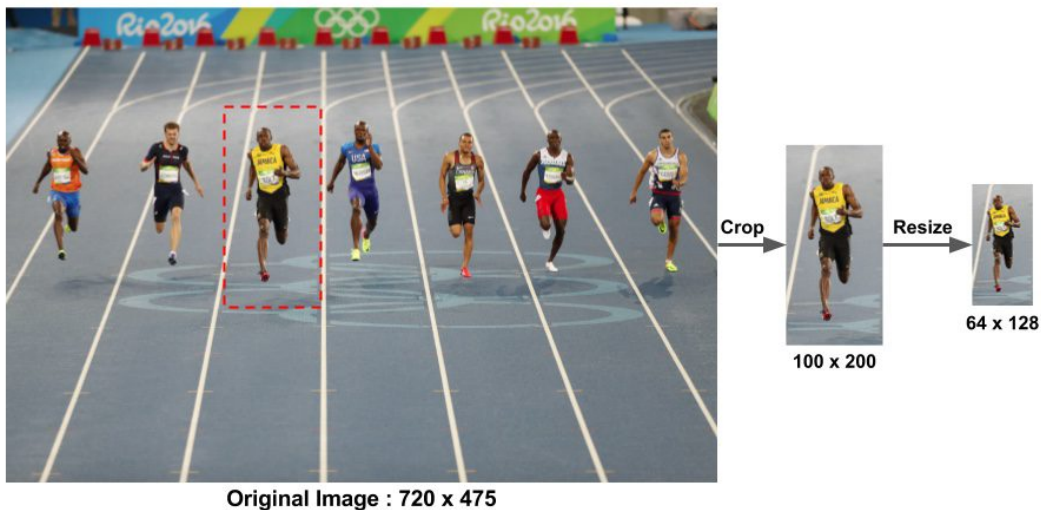
Các pixel cạnh nhau sẽ tạo thành một ô (cell) và tính toán histogram cho từng ô. Với các histogram này lại ta sẽ có thông tin đặc trưng của ảnh. Bên cạnh đó, để tăng độ chính xác, ta sẽ chuẩn hóa độ tương phản các histogram cục bộ qua một vùng rộng hơn, gọi là khối (block) để tăng tính bất biến về thay đổi ánh sáng hay bóng.

Sử dụng HOG để trích xuất đặc trưng hữu ích trước khi đưa vào huấn luyện các mô hình Machine Learning hay Deep Learning làm cải thiện cả về kết quả và thời gian. Việc chỉ giữ lại các thông tin có ích cho mô hình hội tụ nhanh và chính xác hơn.

2 Triển khai thuật toán

2.1 Tiền xử lý

Một bức ảnh có thể có kích thước tùy ý, tuy nhiên ta nên resize lại toàn bộ ảnh cùng chung kích thước để dễ dàng cho việc chia ô (cell) và khối (block). Ở bài viết này, ta sẽ tính toán với ảnh có tỉ lệ là 1:2. Chuyển ảnh về thang xám cũng giúp tính toán dễ dàng hơn.



2.2 Tính gradient của ảnh

Gradient của ảnh sẽ được tính bằng độ thay đổi cường độ của hai pixel gần kề nó theo chiều tính gradient (zero padding cho các pixel ở cạnh). Ta thường coi chiều rộng của ảnh là trục x .

$$\begin{cases} I_x(x, y) = I(x, y + 1) - I(x, y - 1) \text{ theo chiều } x \\ I_y(x, y) = I(x + 1, y) - I(x - 1, y) \text{ theo chiều } y \end{cases}$$

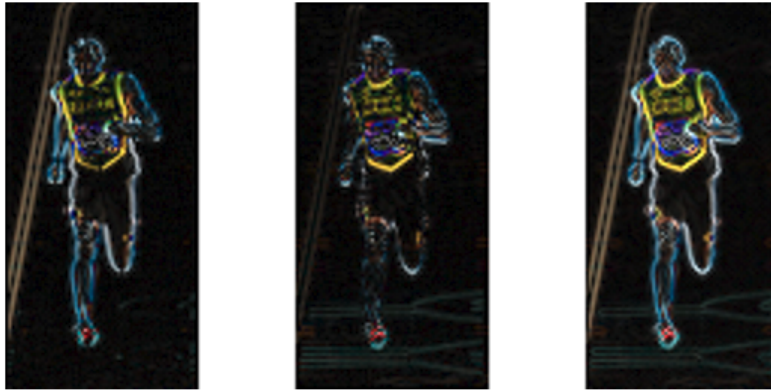
Phép tính trên tương đương với phép nhân tích chập ma trận ảnh với 2 kernel (filter) theo 2 chiều x và y sau. Có rất nhiều lựa chọn cho kernel, tuy nhiên theo bài báo thì đây là lựa chọn đơn giản và hiệu quả nhất:

$$\mathbf{D}_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \mathbf{D}_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Sau khi đã có được hai đạo hàm riêng, ta sẽ tính được cường độ đạo hàm (Gradient Magnitude) và hướng (Gradient Direction). Nếu ảnh có nhiều kênh màu thì lấy gradient lớn nhất trong các kênh:

$$\begin{cases} \mathbf{G} = \sqrt{\mathbf{I}_x^2 + \mathbf{I}_y^2} \\ \theta = \arctan \frac{\mathbf{I}_y}{\mathbf{I}_x} \end{cases}$$

Sau khi có được ma trận gradient theo các chiều, ta có thể trực quan hóa bằng các hình sau:



Left : Absolute value of x-gradient. Center : Absolute value of y-gradient.
Right : Magnitude of gradient.

Nhận xét :

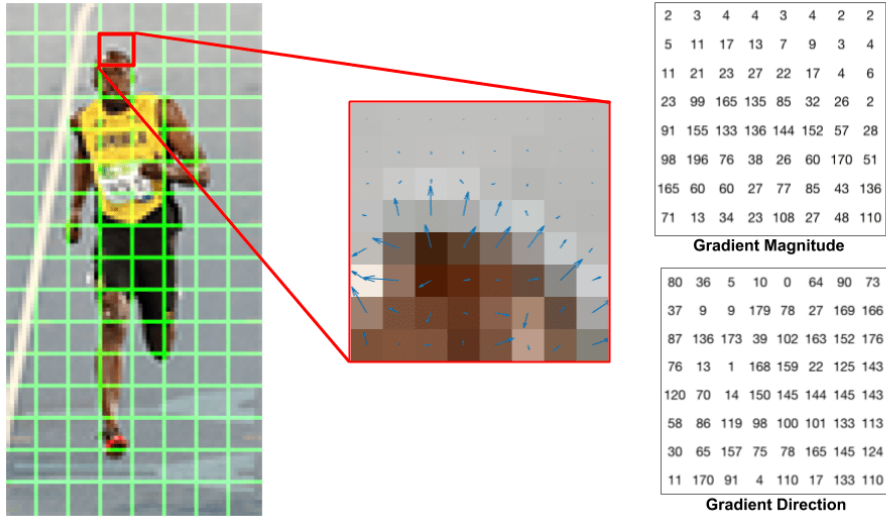
- Gradient theo trục x cho ta thông tin về sự thay đổi cường độ theo phương ngang, do đó các cạnh dọc sẽ được phát hiện.
- Gradient theo trục y làm rõ thay đổi theo chiều dọc, qua đó nhấn mạnh các cạnh có chiều ngang.
- Gradient Magnitude là tổng hợp của hai đạo hàm riêng thành phần cho thông tin chi tiết nhất.

2.3 Tìm histogram cho từng ô (cell)

Theo thực nghiệm của bài báo, với ảnh kích thước 64×128 thì ta có các khối tạo như sau: mỗi ô (cell) gồm 8×8 pixels, hướng gradient không dấu từ 0^0 đến 180^0 (Unsigned Gradient Direction). Tức là hai gradient ngược hướng sẽ có cùng giá trị góc θ .

Mỗi histogram được chia thành 9 ngăn (bin) dựa trên hướng gradient với mỗi ngăn (bin) rộng 20^0 (gọi "bin x^0 " là ngăn bắt đầu từ x^0 đến $x^0 + 20^0$). Ta sẽ "vote" Gradient Magnitude của từng pixel vào histogram với cơ chế như sau:

- Một pixel có θ là bội của 20 thì ta cộng thêm giá trị Gradient của pixel này vào ngăn (bin) đó. Ví dụ $\theta = 20$ thì bin 20^0 sẽ tăng thêm giá trị là gradient của pixel.

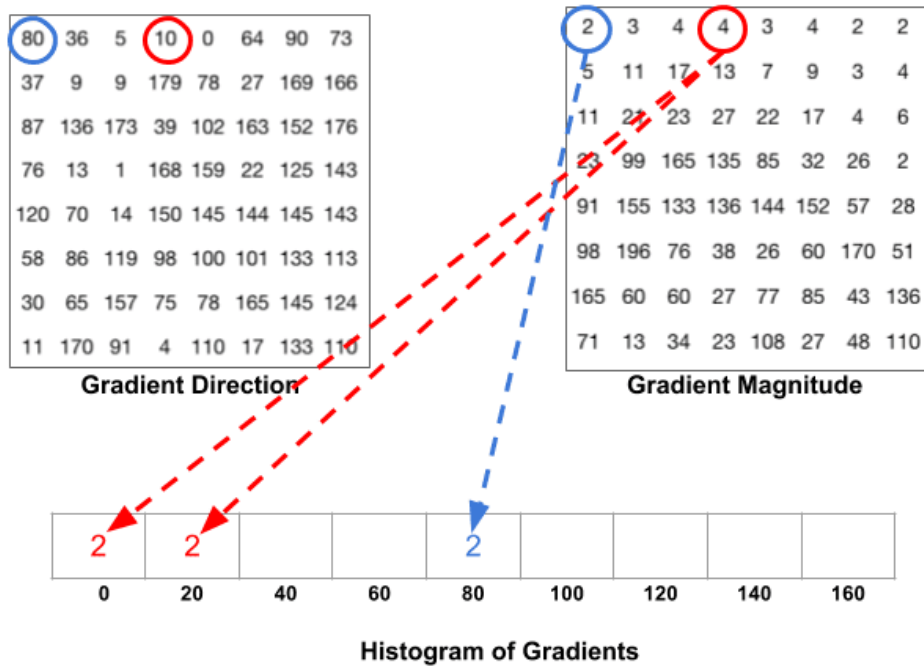


Hình 1: Mũi tên đại diện cho hướng và độ lớn của gradient

- Nếu θ nằm trong khoảng giữa của một ngăn (bin), thì ta sẽ chia giá trị gradient thành hai phần cho bin này và bin kế tiếp (interpolated bilinearly), nếu bin hiện tại là bin 160^0 thì bin tiếp theo là bin 0^0 . Đặt hai bin này là bin x và bin y:

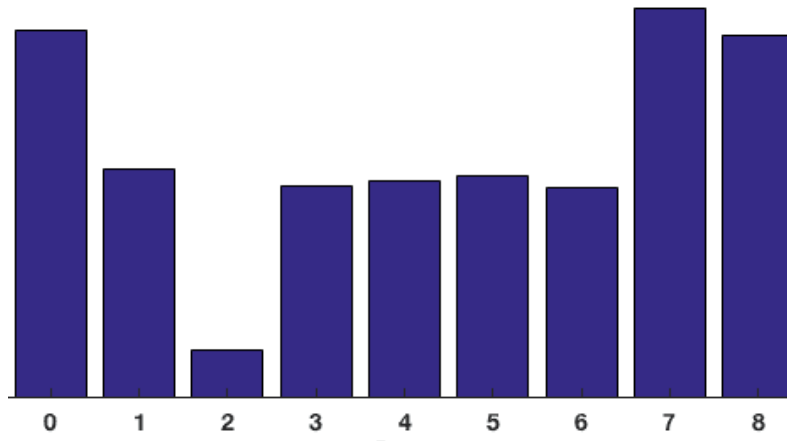
$$histogram[\text{bin } x] += \frac{x + 20 - \theta}{20} \times \text{Gradient Magnitude}$$

$$histogram[\text{bin } y] += \frac{\theta - x}{20} \times \text{Gradient Magnitude}$$



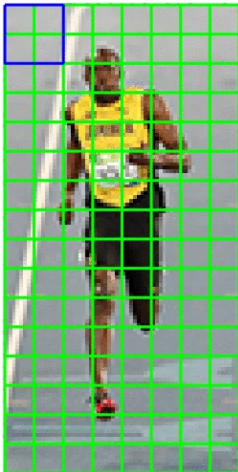
Hình 2: Ta thấy pixel được khoanh vòng tròn màu xanh sẽ vào bin 80^0 và pixel khoanh đỏ sẽ được chia vào hai bin theo công thức như trên.

Tổng hợp tất cả các pixel trong ô (cell), ta được biểu đồ histogram như sau :



Hình 3: Quan sát thấy có các bin 0 và bin 8 (tức bin 0^0 và bin 160^0) lớn chứng tỏ đa số các gradient là hướng lên hoặc xuống.

2.4 Chuẩn hóa khối (block)



Gradient thường nhạy cảm với sự thay đổi độ sáng. Khi ta thay đổi giá trị của tất cả pixel cùng một lượng, gradient vẫn sẽ giống ban đầu. Tuy nhiên nếu nhân hay chia các pixel cho một lượng, các pixel có giá trị lớn sẽ tăng mạnh hơn pixel có giá trị nhỏ và ngược lại, dẫn đến thay đổi phân phối gradient. Giả sử ta giảm độ sáng của ảnh bằng cách chia 2 tất cả các pixel, dẫn đến gradient magnitude cũng giảm đi một nửa và kích thước histogram vì thế cũng giảm một nửa.

Để tăng tính bất biến về thay đổi độ sáng của histogram, ta sẽ tiến hành chuẩn hóa bằng cách gom các ô lân cận thành 1 khối (tối ưu là 1 khối gồm 4 ô) và chuẩn hóa theo khối. Ta có thể dùng nhiều phương pháp như $L2-norm$, $L2-Hys$, $L1-norm$ hay $L1-sqrt$. Các khối có thể chồng lên nhau, trong trường hợp của bài viết này, khối sẽ dịch chuyển 8 pixel mỗi lần trượt, vì thế sẽ có 7×15 khối.

2.5 Tính toán vector đặc trưng

Ta sẽ ghép các histogram sau khi chuẩn hóa của 7×15 khối lại với nhau. Mỗi khối sẽ là một vector có kích thước 36×1 (mỗi khối gồm 4 ô và 9 bin mỗi ô). Vậy kích thước vector đặc trưng của ảnh sẽ là $7 \times 15 \times 36 = 3780$.

3 HOG-SVM cho bài toán phân loại trên tập CIFAR-10

Coming soon...

Tài liệu

- [1] Navneet Dalal, Bill Triggs, *Histograms of Oriented Gradients for Human Detection*, International Conference on Computer Vision & Pattern Recognition (CVPR), 2005.
- [2] Wikipedia contributors, *Histogram of oriented gradients*, Wikipedia, The Free Encyclopedia. Available at: https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients (accessed May 21, 2025).
- [3] Hải Hà, *Tìm hiểu về phương pháp mô tả đặc trưng HOG (Histogram of Oriented Gradients)*, Viblo, ngày 20 tháng 02 năm 2019. Truy cập ngày 21 tháng 5 năm 2025, từ <https://viblo.asia/p/tim-hieu-ve-phuong-phap-mo-ta-dac-trung-hog-histogram-of-oriented-gradients-V3m5WAwz07>
- [4] Satya Mallick, *Histogram of Oriented Gradients*, LearnOpenCV, [Online]. Available: <https://learnopencv.com/histogram-of-oriented-gradients/>. [Accessed: 21-May-2025].