1. Đặc điểm dữ liệu
2. Hoa cúc



* Màu sắc: Cánh hoa có màu vàng, nhụy có màu cam
* Hình dạng: Cánh hoa thuôn dài,
* Kết cấu:

1. Hoa dã quỳ



* Màu sắc: Cánh hoa màu vàng, nhụy có màu cam
* Hình dạng: Cánh hoa thuôn dài
* Kết cấu: nhụy

1. Hoa đồng tiền



* Màu sắc: Cánh hoa màu vàng, nhụy hoa màu nâu
* Hình dạng: Cánh hoa thuôn dài, số lớp cánh hoa nhiều
* Kết cấu:

1. Hoa hướng dương

A close up of a sunflower

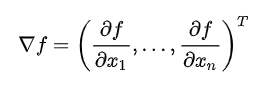
Description automatically generated

* Màu sắc: Cánh hoa màu vàng, nhụy hoa ở vùng trung tâm có màu rêu, ở rìa thì có màu nâu
* Hình dáng: Cánh hoa thuôn dài, nhụy thường chiếm diện tích lớn
* Kết cấu: hình xoắn

1. Các phương pháp trích chọn đặc trưng phổ biến
2. Đặc trưng màu sắc
3. Đặc trưng hình dạng

* Đặc trưng hình dạng là gì?
* Đặc trưng hình dạng trong xử lý ảnh là những thuộc tính liên quan đến hình dạng của các đối tượng trong ảnh, như diện tích, chu vi, độ dài cạnh, góc, đường cong, đối xứng, v.v. Đặc trưng hình dạng có thể giúp phân biệt và nhận dạng các đối tượng trong ảnh.
* Để trích chọn đặc trưng hình dạng, thường cần các bước như phát hiện biên, phân vùng ảnh, rút trích đặc trưng và phân lớp
* Có nhiều phương pháp để rút trích đặc trưng hình dạng, ví dụ như Fourier descriptor, moment invariant, Hough transform, v.v
* Histogram là gì?
* Biểu đồ Histogram là một dạng biểu đồ thể hiện tần suất theo dạng cột. Dữ liệu được biểu thị bằng các cột trên biểu đồ có độ cao khác nhau tùy thuộc vào tần suất (bao nhiêu lần) phạm vi dữ liệu cụ thể xảy ra.
* Mục đích của biểu đồ này là dùng để theo dõi sự phân bổ của các thông số của quá trình hoặc sản phẩm. Từ đây, các nhà quản lý doanh nghiệp có thể đánh giá được hiệu quả của quá trình đó có đáp ứng được yêu cầu của sản xuất hay không hoặc sản phẩm đó có đáp ứng được các tiêu chuẩn về chất lượng hoặc số lượng hay không.
* Biểu đồ Histogram có thể được xây dựng và phân tích với một số kiểu phân bố như sau:
* Biểu đồ mất hai rìa
* Biểu đồ mất một bên
* Biểu đồ mất đỉnh
* Biểu đồ cao ở hai rìa
* Biểu đồ hai đỉnh
* Gradient là gì?
* Gradient là một vector chỉ ra hướng mà tại đó giá trị của hàm thay đổi nhiều nhất và trở thành véc tơ 0 khi hàm đạt giá trị cực đại hoặc cực tiểu địa phương.
* Biểu diễn toán học của Gradient:

Gradient của một hàm f(x1,x2,…,xn) được ký hiệu ∇f là một vector n chiều, mà mỗi thành phần trong vector đó là một đạo hàm riêng phần (partial derivative) theo từng biến của hàm đó:



* Trong xử lý ảnh, độ dốc (tức gradient) đang nói đến ở đây chính là độ dốc về mức sáng. Hay nói cách khác chính là sự thay đổi các giá trị pixel trong ảnh.
* Vùng ảnh trơn (smooth) thì các pixel trong vùng ảnh đó có giá trị xấp xỉ / gần bằng nhau, vì vậy khi tính toán đạo hàm sẽ gần bằng zero. Đạo hàm bằng 0 thể hiện không có biến thiên về giá trị (mức sáng). Điều này có nghĩa là độ dốc của các pixel trong vùng ảnh trơn gần bằng zero. Đạo hàm dương tại một pixel thể hiện rằng biến thiên mức sáng đang ở chiều hướng đi lên, ngược lại đạo hàm âm tại một pixel cho biết biên thiên mức sáng tại đó đang giảm dần. Nói tóm gọn lại gradient của ảnh chính là đạo hàm ảnh.
* Giới thiệu về histograms of oriented gradients (HOG)
* Phương pháp rút trích đặc trưng hình ảnh HOG xuất bản ở hội nghị CVPR 2005 được đề xuất bởi tác giả là Dalal và Triggs.
* Là một mô tả tính năng được sử dụng trong thị giác máy tính và xử lý hình ảnh cho mục đích phát hiện đối tượng. Kỹ thuật này đếm số lần xuất hiện của định hướng gradient trong các phần cục bộ của hình ảnh. Phương pháp này tương tự như biểu đồ định hướng cạnh, mô tả biến đổi tính năng bất biến tỷ lệ và bối cảnh hình dạng, nhưng khác ở chỗ nó được tính toán trên một lưới dày đặc các ô cách đều nhau và sử dụng chuẩn hóa độ tương phản cục bộ chồng chéo để cải thiện độ chính xác
* Các bước trích chọn đặc trưng hình dạng ảnh sử dụng HOG
* Bước 1: Tiền xử lý dữ liệu

Đây là một bước mà hầu hết các bạn sẽ khá quen thuộc. Tiền xử lý dữ liệu là một bước quan trọng trong bất kỳ dự án học máy nào và điều đó cũng không khác gì khi làm việc với hình ảnh.

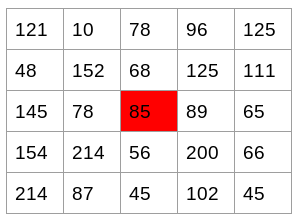
Cần xử lý trước hình ảnh và giảm tỷ lệ chiều rộng trên chiều cao xuống. Kích thước hình ảnh được sử dụng là 128 x 128. Điều này là do ta sẽ chia hình ảnh thành các bản vá 8\*8 ​​và 16\*16 để trích xuất các tính năng. Có kích thước được chỉ định (128 x 128) sẽ làm cho tất cả các tính toán trở nên khá đơn giản.

* Bước 2: Tính gradient

Bước tiếp theo là tính toán độ dốc cho mọi pixel trong ảnh. **Độ dốc là sự thay đổi nhỏ theo hướng x và y.**Phương pháp phổ biến nhất là áp dụng mặt nạ đạo hàm điểm, trung tâm 1-D theo một hoặc cả hai hướng ngang và dọc. Cụ thể, phương pháp này yêu cầu lọc dữ liệu màu sắc hoặc cường độ của hình ảnh bằng các hạt nhân bộ lọc sau:



[-1,0,1] Và [-1,0,1]⊤.**Ta sẽ** lấy một miếng vá nhỏ từ hình ảnh và tính toán độ dốc trên đó. Giả sử ma trận pixel bên dưới cho bản vá nhất định:



Vị trí được đánh dấu là giá trị pixel 85. Bây giờ, để xác định độ dốc (hoặc thay đổi) theo hướng x, chúng ta cần trừ giá trị ở bên trái khỏi giá trị pixel ở bên phải. Tương tự, để tính độ dốc theo hướng y, chúng ta sẽ trừ giá trị pixel bên dưới khỏi giá trị pixel bên trên pixel đã chọn.

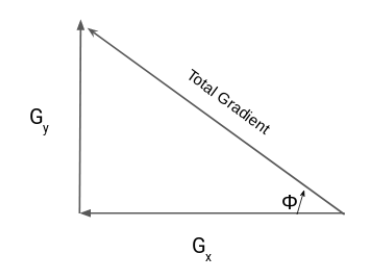
Do đó, độ dốc kết quả theo hướng x và y cho pixel này là:

* Thay đổi theo hướng X(G x ) = 89 – 78 = 11
* Thay đổi theo hướng Y(G y ) = 68 – 56 = 8

Quá trình này sẽ cung cấp cho chúng ta hai ma trận mới – một ma trận lưu trữ độ dốc theo hướng x và ma trận lưu trữ độ dốc khác theo hướng y.

* Tính gradient: áp dụng các toán tử lấy đạo hàm theo hai hướng x và y cho ảnh, sau đó tính cường độ và hướng của gradient tại mỗi điểm ảnh.
* Tính vector đặc trưng cho từng ô (cell): chia ảnh thành các block, mỗi block lại chia thành các cell nhỏ hơn, ví dụ 8x8. Với mỗi cell, tính histogram của các hướng gradient cho các điểm ảnh trong cell đó. Số bin của histogram phụ thuộc vào số lượng hướng gradient được chọn, ví dụ 9 bin cho 9 hướng gradient từ 0 đến 180 độ.
* Chuẩn hóa khối (block): để giảm thiểu ảnh hưởng của ánh sáng, các histogram cục bộ của các cell được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn cell, gọi là khối (block). Một khối có thể gồm nhiều cell, ví dụ 2x2 hoặc 3x3. Các khối có thể chồng lên nhau để tăng tính liên tục của vector đặc trưng.
* Tính toán vector HOG: sau khi chuẩn hóa, các histogram của các cell trong cùng một khối được nối lại với nhau để tạo thành một vector đặc trưng cho khối đó. Sau đó, các vector đặc trưng của các khối được nối lại với nhau để tạo thành vector HOG cho toàn bộ ảnh.
* Bước 3: Tính độ lớn và hướng

Sử dụng độ dốc đã tính toán ở bước 2, bây giờ ta sẽ xác định độ lớn và hướng cho từng giá trị pixel. Đối với bước này, chúng ta sẽ sử dụng định lý Pythagoras.



Ở bước 2, ta có Gx và Gy là 11 và 8. Áp dụng định lý Pythagoras để tính tổng độ lớn của gradient:

Tổng Độ lớn Gradient = 13.6

Tiếp theo, tính toán hướng cho cùng một pixel.

Ta có:

tan(Φ) =

Do đó, giá trị của góc sẽ là:

Φ = arctan() = arctan( = 36

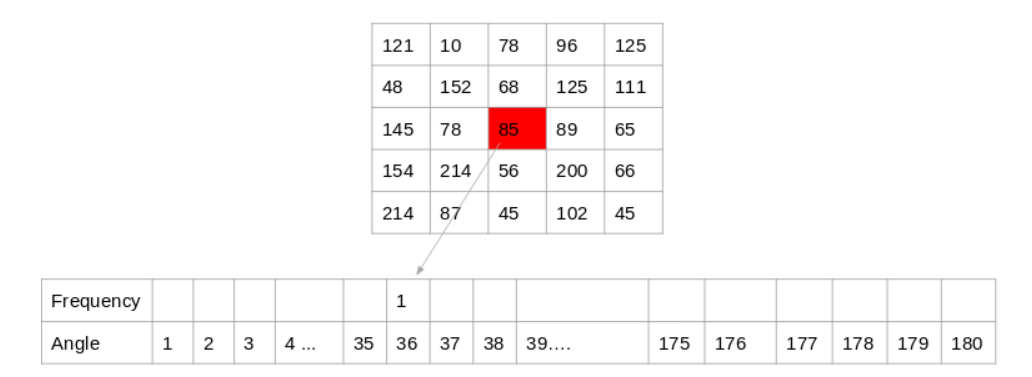
Định hướng xuất hiện là 36. Bây giờ, với mỗi giá trị pixel, chúng ta có tổng độ dốc (độ lớn) và hướng (hướng). Chúng ta cần tạo biểu đồ bằng cách sử dụng các độ dốc và hướng này.

Các phương pháp khác nhau để tạo biểu đồ sử dụng độ dốc và định hướng:

**Biểu đồ tần suất là một đồ thị biểu thị phân bố tần suất của một tập hợp dữ liệu liên tục.**Chúng ta có biến (ở dạng cột) trên trục x và tần số trên trục y. Ở đây, chúng ta sẽ lấy góc hoặc hướng trên trục x và tần số trên trục y.

* Cách 1: Lấy từng giá trị pixel, tìm hướng của pixel và cập nhật bảng tần số

Đây là quy trình cho pixel được đánh dấu (85). Vì hướng của pixel này là 36, ta sẽ thêm một số vào giá trị góc 36, biểu thị tần số:



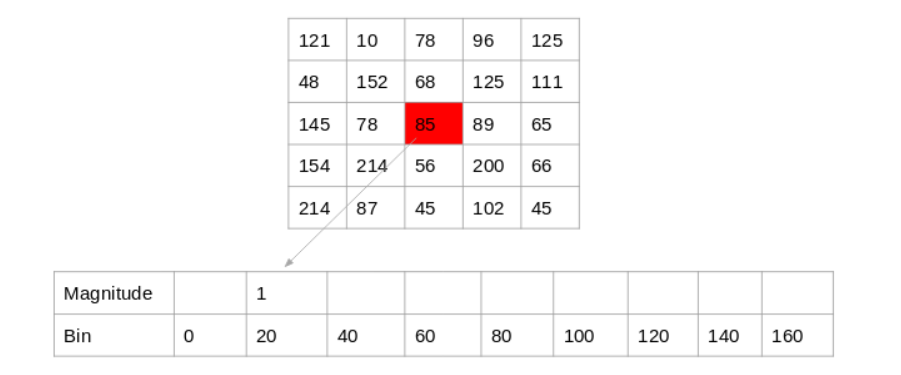
Quá trình tương tự được lặp lại cho tất cả các giá trị pixel và **kết thúc bằng một bảng tần số biểu thị các góc và sự xuất hiện của các góc này trong ảnh.**Bảng tần số này có thể được sử dụng để tạo biểu đồ với các giá trị góc trên trục x và tần số trên trục y.

* Cách 2:

Tạo các tính năng biểu đồ cho các giá trị bin cao hơn

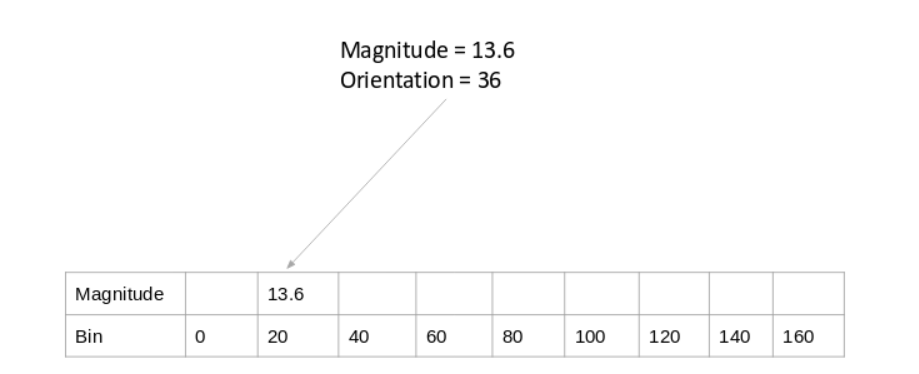
Phương pháp này tương tự như phương pháp trước, ngoại trừ ở đây chúng tôi có kích thước cột là 20. Vì vậy, số lượng cột sẽ nhận được ở đây là 9.

Một lần nữa, đối với mỗi pixel, ta sẽ kiểm tra hướng và lưu trữ tần số của các giá trị hướng ở dạng ma trận 9 x 1. Vẽ sơ đồ này sẽ cho chúng ta biểu đồ:



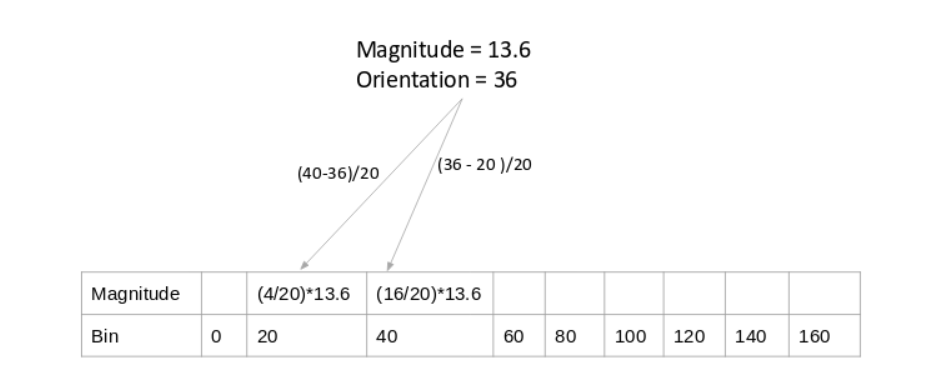
* Cách 3:

Hai phương pháp trên chỉ sử dụng các giá trị định hướng để tạo biểu đồ và không tính đến giá trị độ dốc. Đây là một cách khác để chúng ta có thể tạo biểu đồ **thay vì sử dụng tần số, chúng ta có thể sử dụng cường độ gradient để điền vào các giá trị trong ma trận.**Dưới đây là một ví dụ về điều này:



* Cách 4:

Hãy thực hiện một sửa đổi nhỏ cho phương pháp trên. Ở đây, ta sẽ thêm phần đóng góp của độ dốc của pixel vào các thùng ở hai bên của độ dốc pixel. Hãy nhớ rằng, đóng góp cao hơn phải là giá trị bin gần với định hướng hơn.



Đây chính xác là cách biểu đồ được tạo trong bộ mô tả tính năng HOG.

* Bước 4: Tính toán histogram of gradient trong các ô 8x8 (9x1)

Biểu đồ được tạo trong bộ mô tả tính năng HOG không được tạo cho toàn bộ hình ảnh. Thay vào đó, hình ảnh được chia thành các ô 8x8 và biểu đồ của độ dốc định hướng được tính cho mỗi ô.

Bằng cách đó, ta có được các tính năng (hoặc biểu đồ) cho các bản vá nhỏ hơn, lần lượt đại diện cho toàn bộ hình ảnh. Ta có thể thay đổi giá trị này ở đây từ 8 x 8 thành 16 x 16 hoặc 32 x 32.

Nếu chúng ta chia hình ảnh thành 8×8 ô và tạo biểu đồ, chúng ta sẽ nhận được ma trận 9 x 1 cho mỗi ô. Ma trận này được tạo bằng phương pháp 4 trong bước 3.

* Bước 5: Chuẩn hóa độ dốc trong ô 16x16 (36x1)

Mặc dù đã có các tính năng HOG được tạo cho các ô 8×8 của hình ảnh, độ dốc của hình ảnh rất nhạy cảm với ánh sáng tổng thể. Điều này có nghĩa là đối với một hình ảnh cụ thể, một số phần của hình ảnh sẽ rất sáng so với các phần khác

Ta không thể loại bỏ hoàn toàn điều này khỏi hình ảnh. Nhưng ta có thể giảm sự thay đổi ánh sáng này bằng cách chuẩn hóa độ dốc bằng cách lấy các khối 16 × 16. Dưới đây là một ví dụ có thể giải thích cách các khối 16×16 được tạo:

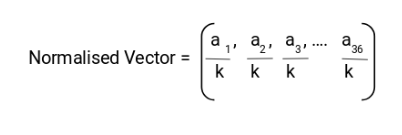
Ở đây, ta sẽ kết hợp bốn ô 8×8 để tạo một khối 16×16. Và ta đã biết rằng mỗi ô 8×8 có một ma trận 9×1 cho một biểu đồ. Vì vậy, ta sẽ có bốn ma trận 9×1 hoặc một ma trận 36×1. Để chuẩn hóa ma trận này, ta sẽ chia từng giá trị này cho căn/ bậc hai của tổng bình phương các giá trị. Về mặt toán học, đối với một vectơ V đã cho:

V = [a1, a2, a3, ….a36]

Ta tính toán gốc của tổng bình phương:

k =

Và chia tất cả giá trị trong vector với giá trị k này:



Kết quả sẽ là một vectơ chuẩn hóa có kích thước 36 × 1.

### Bước 6: Tính toán Biểu đồ của vectơ đặc trưng Định hướng Gradients

Để tính toán véc-tơ đặc trưng cuối cùng cho toàn bộ bản vá hình ảnh, các véc-tơ 36×1 được nối thành một véc-tơ khổng lồ. Kích thước của vectơ này là bao nhiêu?

* Chúng ta có bao nhiêu vị trí của các khối 16×16? Có 128 / 16 \* 2 – 1 = 15 vị trí ngang và 128 / 16 \* 2 – 1 = 15 vị trí dọc, tổng cộng là 15 \* 15 = 225 vị trí.
* Mỗi khối 16×16 được biểu diễn bằng một vectơ 36×1. Vì vậy, khi chúng ta nối tất cả chúng thành một vectơ khuếch đại, chúng ta thu được một vectơ có kích thước 36 \* 225 = 8100 chiều.

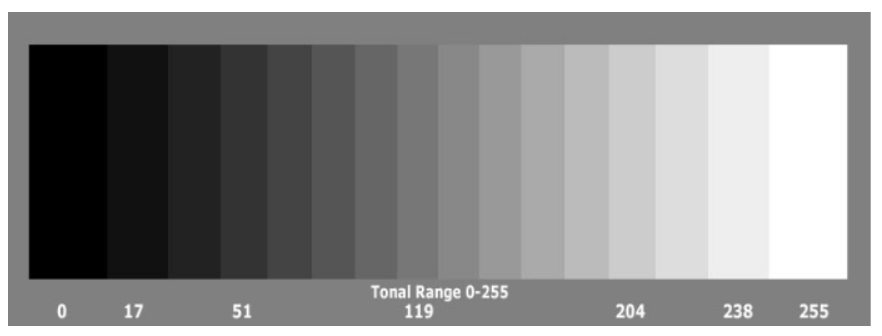
1. Đặc trưng kết cấu

* Đặc trưng kết cấu là gì?
* Đặc trưng kết cấu là một loại đặc trưng thể hiện sự sắp xếp không gian của các giá trị mức xám hoặc màu sắc trong ảnh. Đặc trưng kết cấu có thể được sử dụng để phân biệt các vật liệu, bề mặt, hoa văn hay đối tượng khác nhau trong ảnh.
* Có nhiều phương pháp để rút trích đặc trưng kết cấu trong xử lý ảnh, ví dụ:
* Sử dụng các giá trị moments các cấp: mean, variance, skewness, kurtosis, v.v. để đặc trưng cho phân bố màu sắc trong ảnh.
* Sử dụng ma trận đồng hiện (co-occurrence matrix): là một ma trận thể hiện tần suất xuất hiện của các cặp giá trị mức xám tại các vị trí có quan hệ nhất định trong ảnh. Từ ma trận đồng hiện có thể tính ra các đặc trưng như contrast, energy, homogeneity, entropy, v.v.
* Sử dụng đặc trưng LBP (local binary pattern): là một phương pháp rút trích đặc trưng bằng cách so sánh giá trị mức xám của pixel tâm với các pixel lân cận và mã hóa thành một số nhị phân. Từ số nhị phân này có thể xây dựng histogram để đặc trưng cho kết cấu ảnh.
* Giới thiệu về local binary pattern
* Local binary pattern (LBP) là một phương pháp rút trích đặc trưng kết cấu trong xử lý ảnh. LBP dựa trên việc so sánh giá trị mức xám của pixel trung tâm với các pixel lân cận và mã hóa thành một số nhị phân. Từ số nhị phân này có thể xây dựng histogram để đặc trưng cho kết cấu ảnh.
* LBP có một số ưu điểm như:
* Đơn giản và hiệu quả: LBP chỉ cần thực hiện các phép so sánh và nhị phân hóa, không cần tính toán phức tạp hay sử dụng bộ lọc.
* Bất biến với phép chiếu sáng: LBP không phụ thuộc vào giá trị tuyệt đối của mức xám, mà chỉ quan tâm đến sự chênh lệch giữa pixel trung tâm và pixel lân cận. Do đó, LBP không bị ảnh hưởng nhiều bởi thay đổi độ sáng của ảnh.
* Dễ mở rộng và tùy biến: LBP có thể thay đổi số lượng và vị trí của các pixel lân cận, bán kính của vùng xét, hay cách mã hóa nhị phân để phù hợp với các bài toán khác nhau.
* Các bước trích chọn đặc trưng kết cấu sử dụng LBP
* Bước 1: Chuyển đổi hình ảnh thành không gian thang độ xám

**Thang độ xám là mô hình đơn giản nhất vì nó xác định màu sắc chỉ sử dụng một thành phần là độ sáng.**Mức độ sáng được mô tả bằng giá trị nằm trong khoảng từ 0 (đen) đến 255 (trắng).

Một mặt, hình ảnh thang độ xám truyền tải ít thông tin hơn RGB. **Tuy nhiên, chúng phổ biến trong xử lý ảnh vì sử dụng ảnh thang độ xám cần ít dung lượng khả dụng hơn và nhanh hơn, đặc biệt khi chúng ta xử lý các tính toán phức tạp.**

Dưới đây, chúng ta có thể thấy đầy đủ các màu mà mô hình thang độ xám có thể mô tả:



Các phương pháp chuyển đổi ảnh sang thang độ xám:

* Phương pháp nhẹ nhàng

Một phương pháp rất đơn giản là **lấy giá trị trung bình của các thành phần có giá trị cao nhất và thấp nhất:**

Grayscale =

**Ta có thể dễ dàng thấy rằng phương pháp này có một điểm yếu rất nghiêm trọng do không sử dụng một thành phần RGB.**Đây chắc chắn là một vấn đề vì mức độ sáng mà mắt chúng ta cảm nhận được phụ thuộc vào cả ba màu cơ bản.

* **Phương pháp trung bình**

Một phương pháp khác là lấy **giá trị trung bình của ba thành phần (đỏ, lục và lam) làm giá trị thang độ xám:**

Grayscale =

Mặc dù tính đến tất cả các thành phần, nhưng **phương pháp trung bình cũng có vấn đề vì nó gán trọng số như nhau cho mỗi thành phần.**Dựa trên nghiên cứu về tầm nhìn của con người, chúng ta biết rằng mắt của chúng ta phản ứng với từng màu theo một cách khác nhau. **Cụ thể, mắt chúng ta nhạy cảm hơn với màu xanh lục, sau đó là màu đỏ và cuối cùng là màu xanh lam.**Do đó, các trọng số trong phương trình trên sẽ thay đổi.

* Phương pháp độ sáng

Phương pháp tốt nhất là phương pháp độ sáng giải quyết thành công các vấn đề của các phương pháp trước đó.

Dựa trên các quan sát đã nói ở trên, ta nên lấy trung bình có trọng số của các thành phần. **Phần đóng góp của màu xanh lam vào giá trị cuối cùng sẽ giảm và phần đóng góp của màu xanh lá cây sẽ tăng lên. Sau một số thử nghiệm và phân tích sâu hơn, các nhà nghiên cứu đã kết luận trong phương trình dưới đây:**

Grayscale =

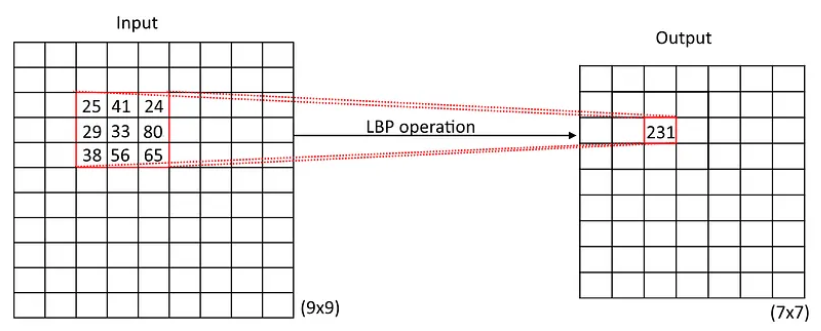
* Bước 2: Đối với mỗi pixel (gp) trong ảnh, hãy chọn các vùng lân cận P bao quanh pixel trung tâm. tọa độ của gp được cho bởi công thức



Chọn số lượng và bán kính của các pixel lân cận (P và R) cho mỗi pixel trung tâm.

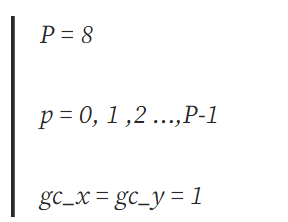
Duyệt qua từng pixel trung tâm trong ảnh và lấy giá trị mức xám của nó và các pixel lân cận theo hình tròn có bán kính R.

Lấy ví dụ, đoạn sau của một hình ảnh thang độ xám:

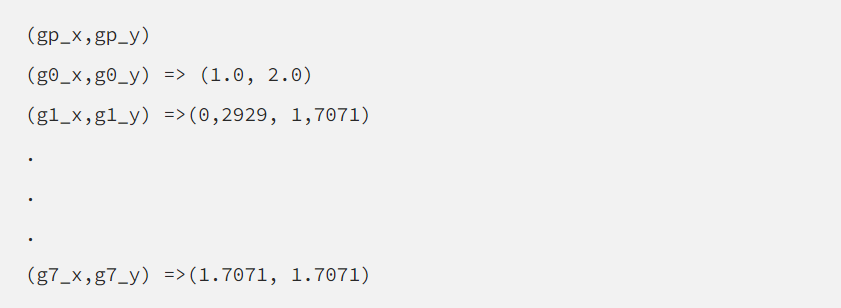
Ta có thể biểu thị kích thước của cửa sổ này (3x3) theo bán kính của hình tròn bằng (2 \* R + 1), nếu bán kính là 1 thì chúng ta có ma trận 3x3.

Tọa độ của pixel trung tâm được ký hiệu là gc(gc\_x,gc\_y) là (1,1) theo trục tọa độ của ma trận(3x3). Giá trị của điểm ảnh này là 33(trung tâm) gc =33. Lấy ví dụ về 8 mẫu hàng xóm (P=8R). Tọa độ của từng điểm mẫu có thể được biểu thị bằng

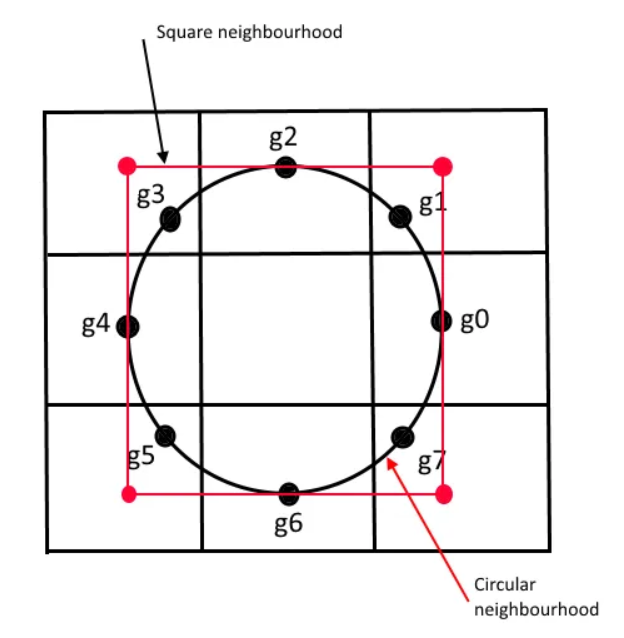




Vì vậy, đối với ma trận trước, ta có các tọa độ sau cho từng mẫu:



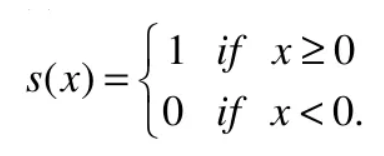
* Bước 3: Lấy pixel trung tâm (gc) và đặt nó làm ngưỡng cho các lân cận P của nó

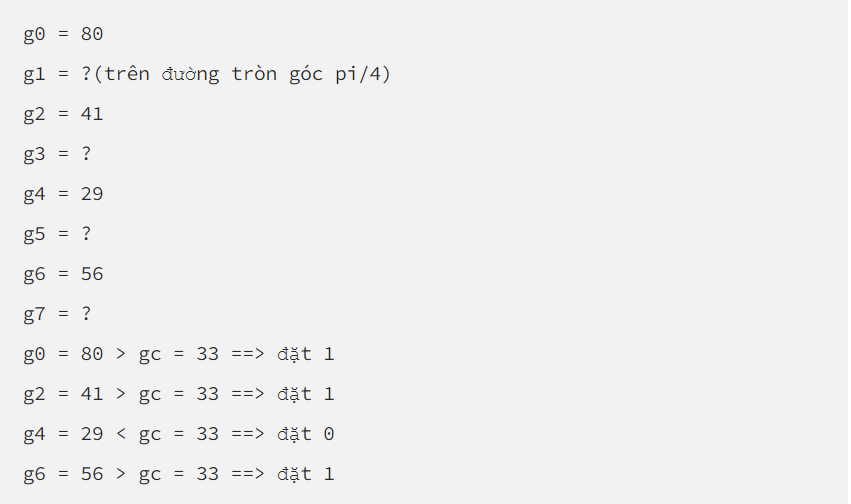


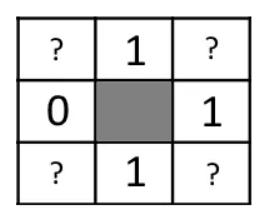
Biểu thị bằng Theta\_i = => Theta = 0, , ,,,,,.

* Bước 4: So sánh giá trị mức xám của pixel trung tâm với các pixel lân cận và gán giá trị 1 nếu pixel lân cận lớn hơn hoặc bằng pixel trung tâm, ngược lại gán giá trị 0.

Hàm S có thể được biểu diễn như sau:







Bây giờ ta cần tìm các giá trị cường độ của g1, g3, g5, g7

Để tìm các giá trị này cần áp dụng một phép nội suy.

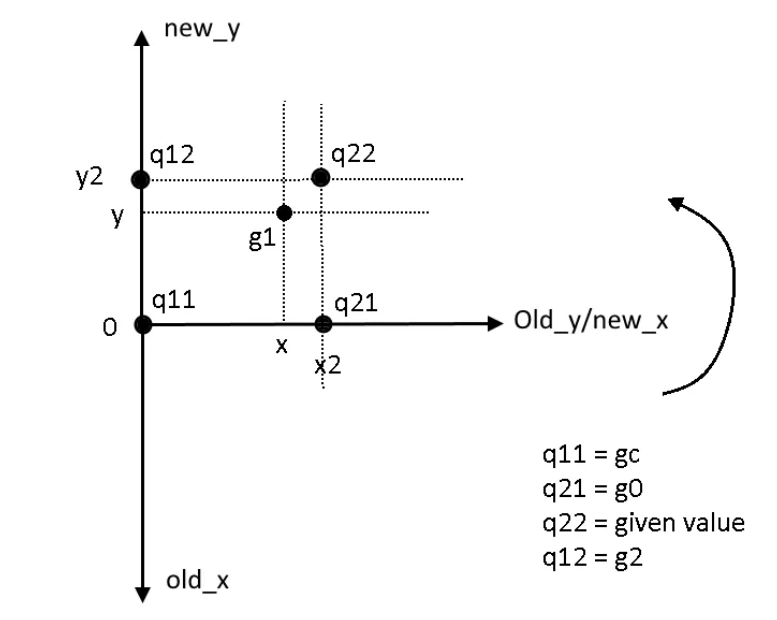
Vì chúng ta có không gian 2d (hình ảnh 2 chiều) nên chúng ta cần một phương pháp nội suy 2d.

Thuật ngữ “**nội suy”** là một cách để tính giá trị trung gian của một hàm từ một số giá trị đã biết của nó.

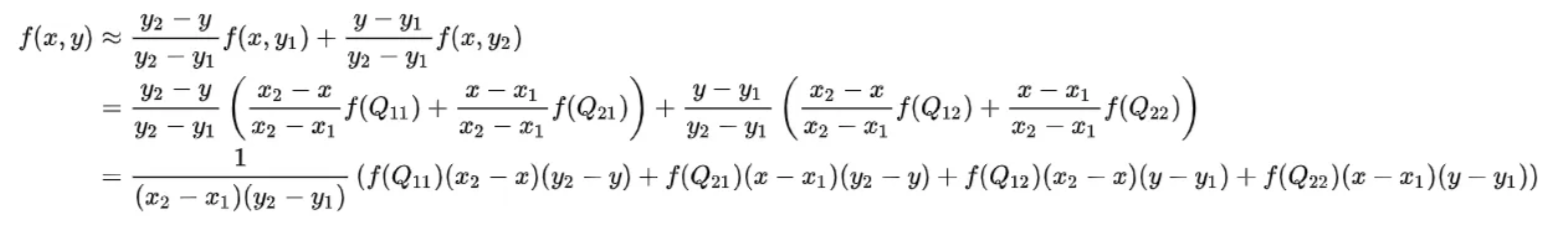
[**Nội suy song tuyến tính**](https://en.wikipedia.org/wiki/Bilinear_interpolation#:~:text=In%20mathematics%2C%20bilinear%20interpolation%20is,again%20in%20the%20other%20direction.) là phép nội suy tuyến tính của hàm hai biến, nghĩa là nội suy bốn điểm. Nếu biết giá trị của hàm số tại các điểm này f(x1,y1),f(x2,y1),f(x1,y2),f(x2,y2).

Để tính cường độ của g1, g3, g5, g7, chúng ta cần tìm tọa độ của hộp bên ngoài chứa giá trị pixel chưa biết.

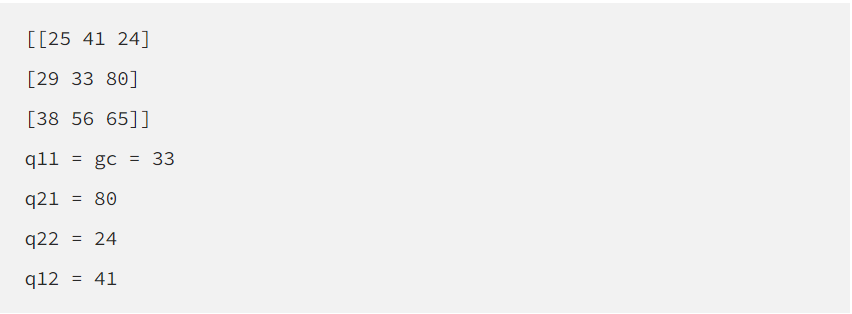
Ví dụ: g1 nằm giữa theta = 0 và theta = pi/2, hình sẽ như sau:



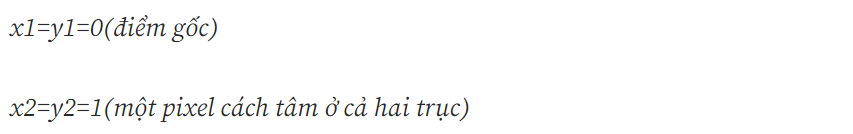
Giá trị pixel của g1 có thể được nội suy bằng công thức :



Quay lại với ví dụ:

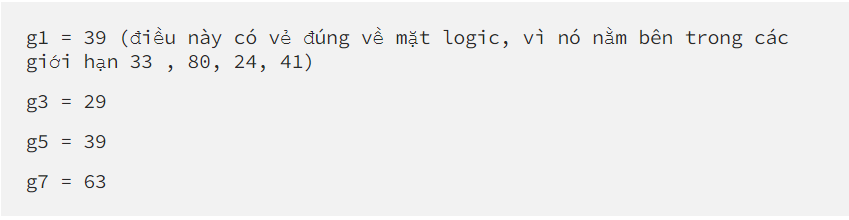


Ta có thể dịch hệ tọa độ về gốc => điều này có nghĩa là

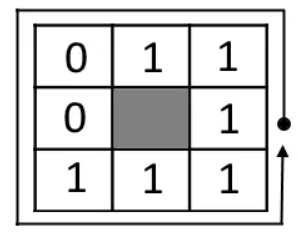


Các giá trị x và y của điểm chưa biết cần được dịch sang hệ tọa độ thông thường (xoay ngược chiều kim đồng hồ 90 độ, có nghĩa là new\_x = old\_y và new\_y = -old\_x )

Áp dụng công thức này trên các mẫu chưa biết, chúng tôi có thể tìm thấy:

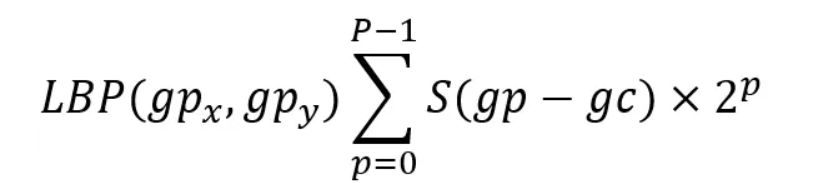


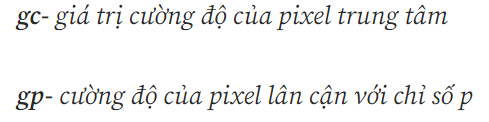
Bây giờ ma trận ngưỡng bằng:

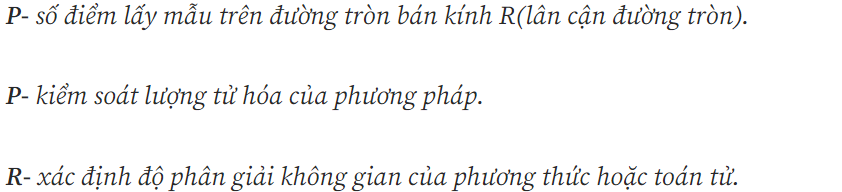


* Bước 5: tính giá trị LBP

Tuần tự ngược chiều kim đồng hồ, viết một số nhị phân bao gồm các chữ số liền kề với pixel trung tâm. Số nhị phân này (hoặc số thập phân tương đương của nó) được gọi là mã pixel **trung tâm LBP** và hơn nữa, được sử dụng làm kết cấu cục bộ được chọn đặc trưng.

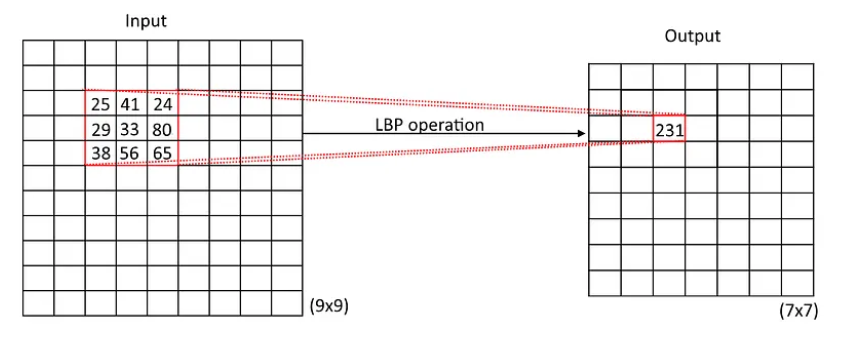






Áp dụng các công thức LBP

LBP =



Quá trình này sẽ lặp lại cho từng khối hình ảnh (dọc theo trục x và y)

Kích thước của hình ảnh được giảm theo hệ số 2\*R dòng và 2\*R cột như trong ví dụ có hình ảnh 9x9 sẽ dẫn đến hình ảnh 7x7 (R=1 cho ví dụ này)

Để biết một pixel có cần nội suy hay không, chúng ta có thể tính phần phân số của gpx và gpy, nếu phần phân số của cả x và y đều bằng 0 (như g0, g2, g4, g6) thì pixel nằm hoàn toàn ở trung tâm của khối, nếu không chúng ta cần thực hiện phép nội suy (g1, g3, g5, g7). Đối với ví dụ này, chúng tôi có 4 pixel cần thực hiện phép nội suy song tuyến tính và bốn điểm còn lại không cần vì giá trị của pixel này đã được cung cấp bởi ma trận.

* Bước 6: Vector hóa