



Học phần: Thị giác máy tính (Computer vision)

Chương 3: Trích chọn đặc trưng và phân lớp



Nguyễn Hữu Tuân: huu-tuan.nguyen@vimaru.edu.vn

Nội dung chương 3

Khái niệm

Một số phương pháp trích chọn đặc trưng cục bộ

Mẫu nhị phân cục bộ LBP

Đặc trưng SIFT

Một số phương pháp phân lớp cơ bản

Phương pháp k-láng giềng gần nhất

Phương pháp SVM

Phương pháp sử dụng mạng nơ ron

Trích chọn đặc trưng (feature extraction)?

Trích chọn đặc trưng (feature extraction): là phương pháp trích rút các đặc điểm có tính chất phân biệt nhất của ảnh hoặc đối tượng cần xử lý và biểu diễn dưới dạng số - 1 vector đa chiều. Các đặc trưng phân biệt chỉ là 1 phần thông tin của đối tượng hay dữ liệu phù hợp nhất với bài toán cần giải quyết. Đặc trưng phân biệt theo 2 khía cạnh: giúp phân biệt các ảnh thuộc về các đối tượng khác nhau, phân biệt được các bức ảnh khác nhau của cùng 1 đối tượng. Vector đặc trưng có số chiều từ vài trăm tới vài trăm nghìn giá trị nhưng vẫn là tương đối bé so với độ phức tạp của dữ liệu gốc.

Trích chọn đặc trưng (feature extraction)?

Sau bước trích chọn đặc trưng, việc xử lý các bức ảnh (so sánh, tính toán ...) được thực hiện với các vector đặc trưng. Trích chọn đặc trưng là bước quan trọng nhất của 1 hệ thống thị giác máy tính vì nó là công cụ duy nhất để biểu diễn dưới dạng số hóa dữ liệu và quyết định độ chính xác của hệ thống.

Có 2 câu hỏi:

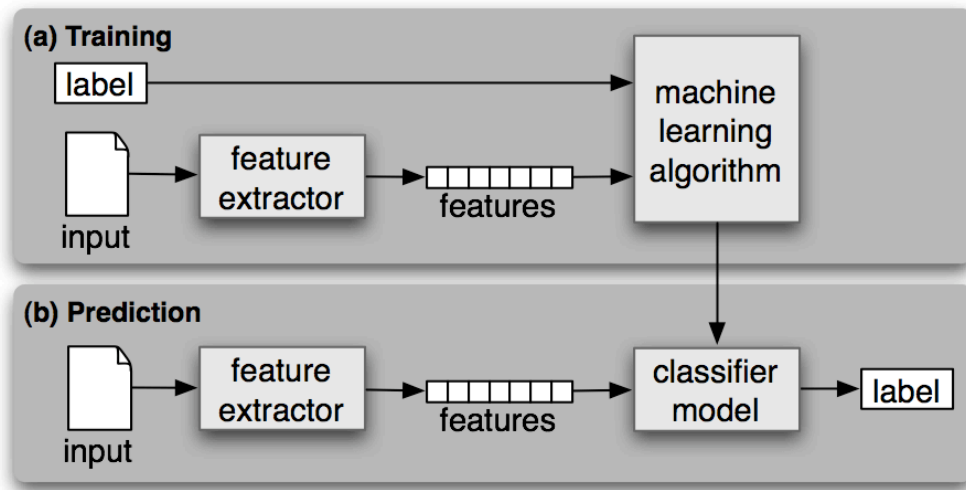
1. Đặc trưng (feature) là gì?
2. Làm thế nào để trích chọn đặc trưng từ dữ liệu ảnh một cách hiệu quả?

Đặc trưng (feature) là gì?

- Đặc trưng (lĩnh vực thị giác máy tính) là các thông tin thích hợp để giải quyết các bài toán ứng dụng của thị giác máy tính.
- Đặc trưng là khái niệm chung, với mỗi bài toán cụ thể, đặc trưng sẽ là một loại thông tin cụ thể khác nhau.
- Có 2 loại đặc trưng cơ bản:
 - Đặc trưng cục bộ (local features): ví dụ như với khuôn mặt, đó là chi tiết về mắt, mũi, miệng, lông mày
 - Đặc trưng toàn cục (global features): ví dụ như đối với khuôn mặt là kiểu khuôn mặt, màu da, độ lớn.
- Dạng số hoá của các đặc trưng đại diện cho ảnh là các vector đặc trưng (số)

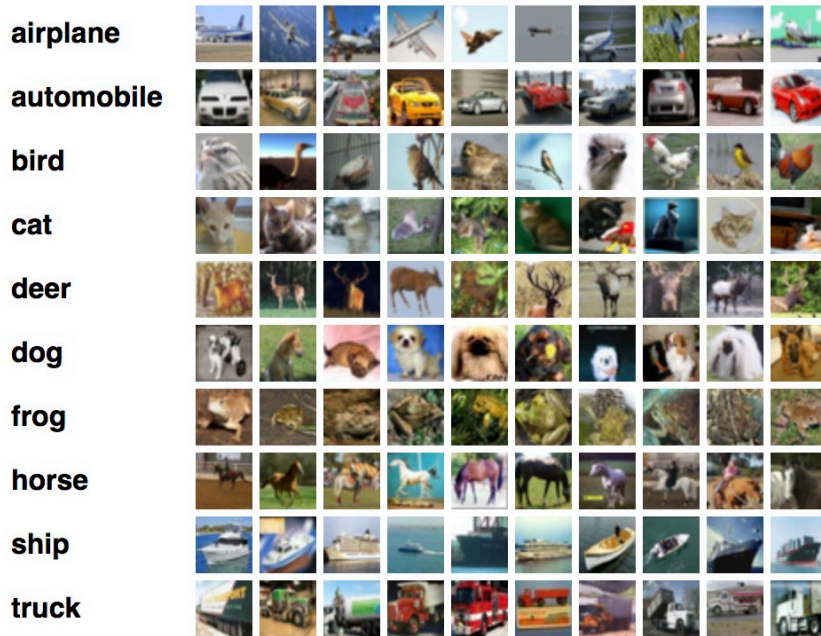
Làm thế nào để trích chọn đặc trưng?

- Có 2 cách tiếp cận
 - Cách tiếp cận dựa vào các phương pháp đặc thù cho từng bài toán cần giải quyết, dựa trên kinh nghiệm và thực nghiệm là chính (LBP, SIFT, Gabor wavelets ...) – nội dung của học phần.
 - Cách tiếp cận dựa vào các mạng nơ ron học sâu.



Phân lớp (classification) là gì?

- Phân lớp (classification): thao tác gán nhãn cho các dữ liệu input dựa trên đánh giá về độ tương đồng so sánh với dữ liệu chuẩn (có nhãn).



Phân lớp

Trong bài toán nhận dạng/phân lớp dữ liệu được chia thành 3 tập:

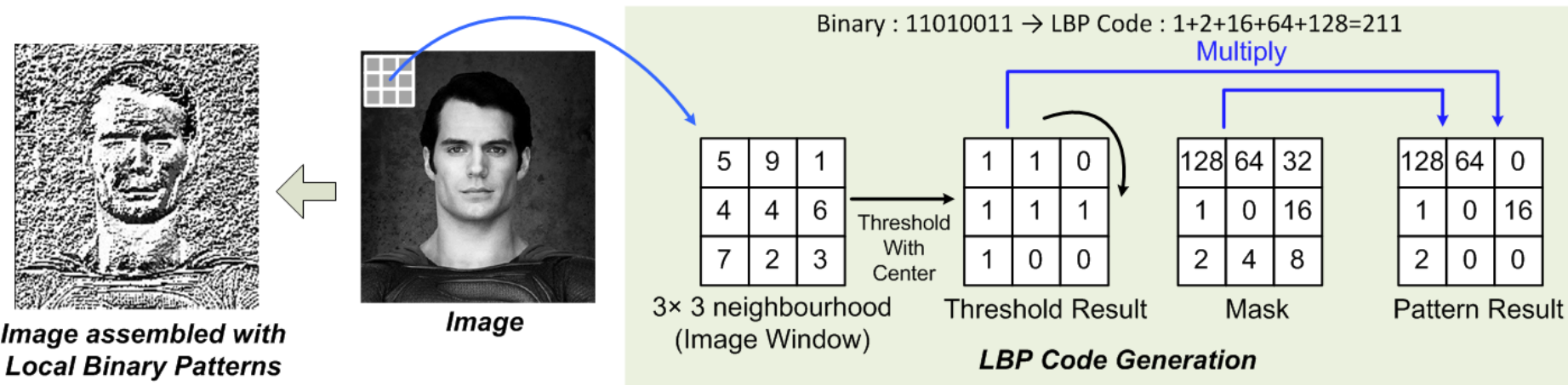
1. tập tham chiếu (reference set) chứa các dữ liệu có nhãn chính xác (with annotation information).
2. tập dữ liệu học (training set) chứa các dữ liệu (có thể có nhãn hoặc không) để hệ thống học và đưa ra mô hình dự đoán. Nếu thuật toán học đòi hỏi dữ liệu có nhãn thì gọi là học có giám sát (supervised learning) ngược lại là học không có giám sát (unsupervised learning).
3. tập dữ liệu kiểm thử (test/validation set): chứa các dữ liệu dùng để đánh giá độ chính xác của hệ thống. Có nhiều cách đánh giá khác nhau về độ chính xác của hệ thống, phụ thuộc vào bài toán và mục tiêu.

Phương pháp trích chọn đặc trưng nhị phân cục bộ LBP

- LBP = Local Binary Patterns
- LBP là phương pháp trích chọn đặc trưng hiệu quả được thiết kế ban đầu cho bài toán phân loại (classification/recognition) vân ảnh (texture), sau đó áp dụng và thành công nhất cho bài toán nhận dạng mặt người (danh tính, giới tính, ...)
- Ưu điểm của LBP là đơn giản, nhanh và hiệu quả.
- Các đặc trưng LBP có khả năng chống lại các thay đổi về ánh sáng của ảnh.

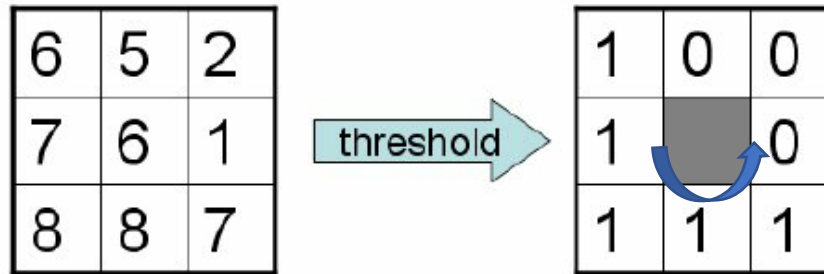
Phương pháp trích chọn đặc trưng nhị phân cục bộ LBP

- Phương pháp LBP dựa trên một thao tác cơ bản gọi là toán tử LBP (LBP operator). Toán tử này thực hiện công việc tại từng vị trí điểm ảnh input để sinh ra một ảnh output gọi là ảnh LBP, các giá trị output được gọi là các mẫu/đặc trưng LBP của ảnh.



Toán tử LBP

- Toán tử LBP khá đơn giản xét về thuật toán, được áp dụng tại mỗi điểm ảnh bằng cách so sánh giá trị mức xám của điểm ảnh đang xét với N ($N=4, 6, 8, 10, 12, 16$) điểm lân cận nằm cách đều trên 1 đường tròn bán kính r (1, 2, 3, 4, 5, 6, 7), nếu điểm đang xét nhỏ hơn ta được 1 bit 1, ngược lại là 1 bit 0 tại vị trí so sánh. Sau đó N bit được ghép với nhau (theo thứ tự như hình mũi tên bên dưới) để tạo thành 1 số ở cơ số 10 gọi là giá trị LBP tại điểm ảnh output tương ứng.



Binary code = **11110001**

LBP = $1 + 16 + 32 + 64 + 128 = 241$

Toán tử LBP – ví dụ minh họa

5	60	123
0	65	19
28	198	2



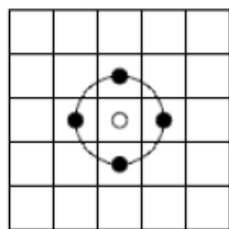
0	0	1
0		0
0	1	0

$$\begin{aligned} \text{LBP}_{\text{code}} &= (00100100)_2 = \\ &0*128+0*64+1*32+0*16+0*8+1*4+0*2+0*1 = 32+4 = \\ &36. \end{aligned}$$

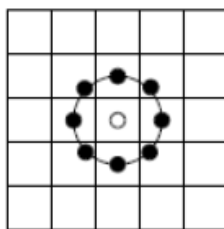
	36	

Toán tử LBP

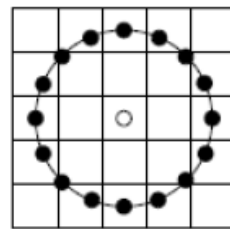
- Toán tử LBP cơ bản chỉ xét 8 điểm lân cận với bán kính $r = 1$. Sau đó các toán tử LBP được mở rộng để xét các điểm lân cận với bán kính r lớn hơn. Khi đó mức xám của các điểm lân cận được xét (cơ bản không nằm đúng trên ảnh) sẽ được nội dung song tuyến tính để tính và so sánh với điểm đang xét. Bên dưới là các mẫu LBP khác nhau.



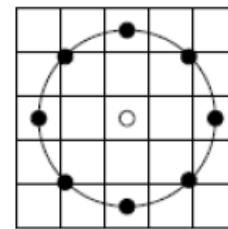
(4,1)



(8,1)



(16,2)

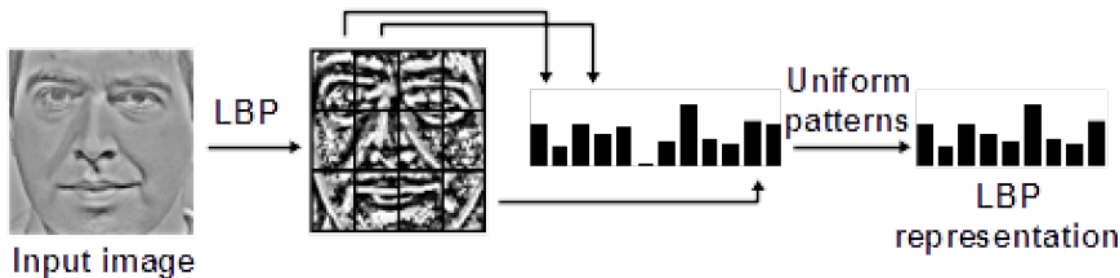


(8,2)

Hình 3.12: Các mẫu toán tử LBP

Phương pháp trích chọn đặc trưng LBP

- Từ ảnh LBP thu được sau khi áp dụng toán tử LBP, ảnh sẽ được chia thành các vùng con rời nhau và tính các chuỗi histogram, ghép lại với nhau tạo thành vector LBP cho bức ảnh.



Hình 3.14: Biểu diễn ảnh mặt trong nhận dạng với LBP

Mẫu đồng nhất (uniform patterns)

- Việc chia ảnh LBP thành các vùng con và tính histogram sau đó ghép lại là để đảm bảo các đặc trưng cục bộ, không gian của ảnh gốc được thể hiện trong vector kết quả.
- Theo thống kê từ cơ sở dữ liệu ảnh FERET thì chỉ có một số giá trị LBP xuất hiện nhiều (hơn 80%), các giá trị này có 1 đặc điểm chung là số lần thay đổi từ bit 0 sang 1 không vượt quá 2, các giá trị này gọi là Uniform patterns và được gán nhãn riêng (58 giá trị), các giá trị còn lại được gán chung 1 nhãn, cuối cùng 1 vector LBP được rút ngắn từ 256 giá trị xuống còn 59 giá trị.

Áp dụng LBP

- Các vector LBP sau đó sẽ được so sánh bằng các hàm khoảng cách (càng lớn thì độ tương đồng càng nhỏ). Hàm Chi-Square hay được dùng:

$$\chi^2(S, M) = \sum_i \frac{(S_i - M_i)^2}{|S_i + M_i|}$$

- LBP được dùng trong nhiều bài toán liên quan tới phân tích ảnh mặt (phát hiện, nhận dạng, xác định các thuộc tính liên quan tới ảnh mặt như cảm xúc, giới tính, độ tuổi ...).

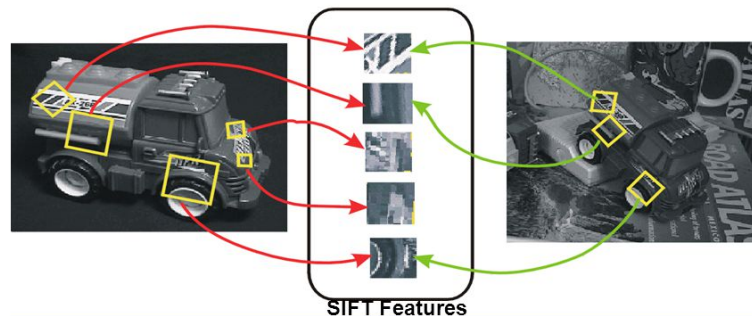
Trích chọn đặc trưng bất biến về tỉ lệ SIFT (Scale Invariant Feature Transform)

SIFT là phương pháp trích chọn đặc trưng cục bộ do David Lowe đề xuất ban đầu là để giải quyết bài toán đối sánh ảnh (image matching) với các yêu cầu:

- + Bất biến đối với các tỉ lệ khác nhau
- + Bất biến đối với các ảnh có hướng khác nhau.
- + Bất biến đối với thay đổi của ánh sáng.
- + Bất biến đối với các khung nhìn khác nhau

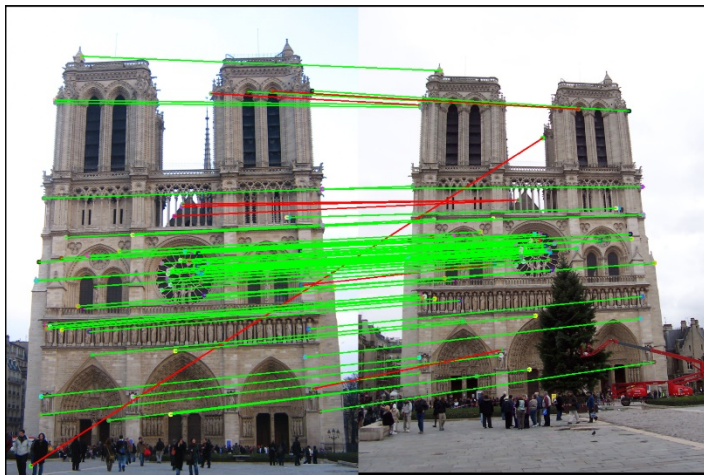
Idea of SIFT

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Trích chọn đặc trưng bất biến về tỉ lệ SIFT (Scale Invariant Feature Transform)

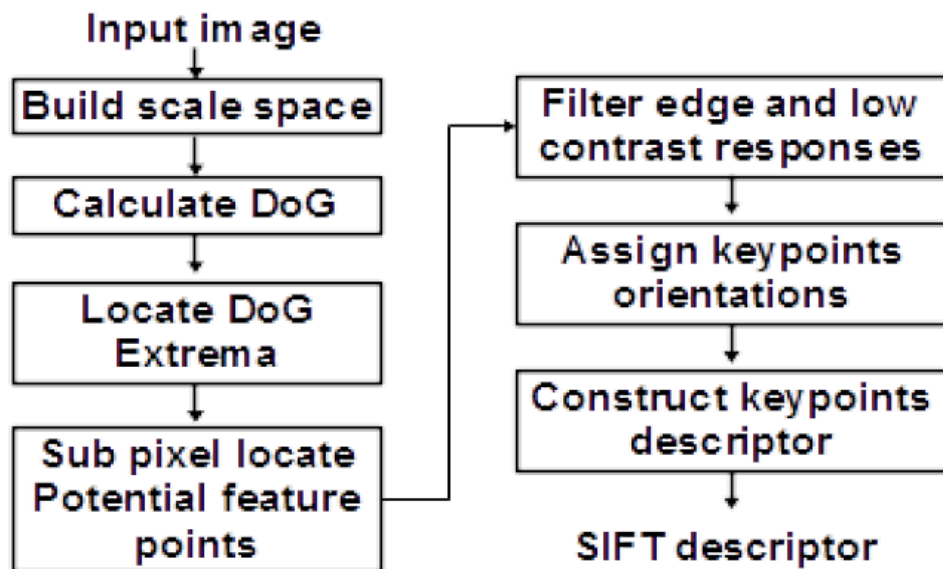
Hiện nay SIFT được dùng cho nhiều ứng dụng thực tế khác nhau: các hệ truy vấn ảnh, tạo ảnh panorama, nhận dạng hình ảnh, xử lý video ... SIFT cũng là cơ sở cho nhiều thuật toán trích chọn đặc trưng cục bộ khác như SURF, ORB ...



(b) the other spherical panoramic image

Trích chọn đặc trưng bất biến về tỉ lệ SIFT (Scale Invariant Feature Transform)

Thuật toán trích chọn đặc trưng SIFT gồm các bước:

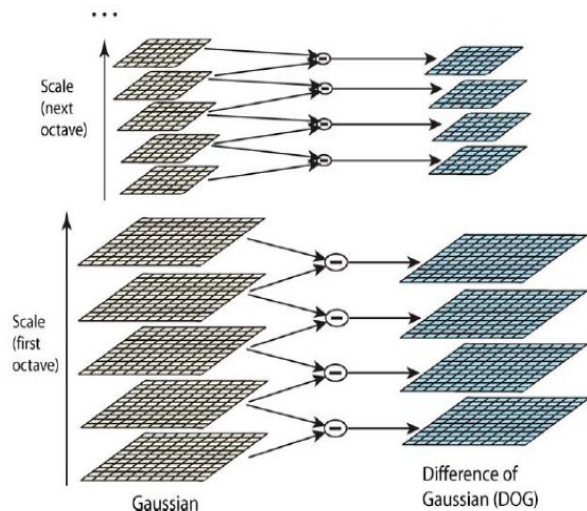


Hình 3.7: Các bước trích chọn đặc điểm SIFT

Trích chọn đặc trưng bất biến về tỉ lệ SIFT (Scale Invariant Feature Transform)

Bước 1: Xây dựng không gian ảnh từ 1 ảnh input (sinh ra các ảnh có kích thước khác nhau từ ảnh gốc: $1/2 * 1/2$, $1/4 * 1/4$, 2×2 ..)

Bước 2: Áp dụng các phép lọc Gauss trên các ảnh có cùng kích thước và tính ảnh DoG (Difference of Gaussian) là hiệu của 2 ảnh kết quả lọc Gauss.



Trích chọn đặc trưng bất biến về tỉ lệ SIFT (Scale Invariant Feature Transform)

Bước 3: Xác định các vị trí cực đại trên các ảnh DoG.

Bước 4: Áp dụng triển khai chuỗi Taylor với các điểm ảnh để xác định các điểm đặc trưng tiềm năng có dạng (x, y, σ) .

Bước 5: Loại bỏ các điểm có độ tương phản không tốt (theo giá trị thử nghiệm ..).

Bước 6: Tính đạo hàm của ảnh và tính hướng cho các điểm lân cận của điểm đặc trưng.

Bước 7: Tính histogram của 16 vùng con xung quanh mỗi điểm đặc trưng=> mỗi mô tả SIFT của 1 điểm đặc trưng sẽ là 1 vector có $16 \times 8 = 128$ giá trị.

Trích chọn đặc trưng bất biến về tỉ lệ SIFT (Scale Invariant Feature Transform)

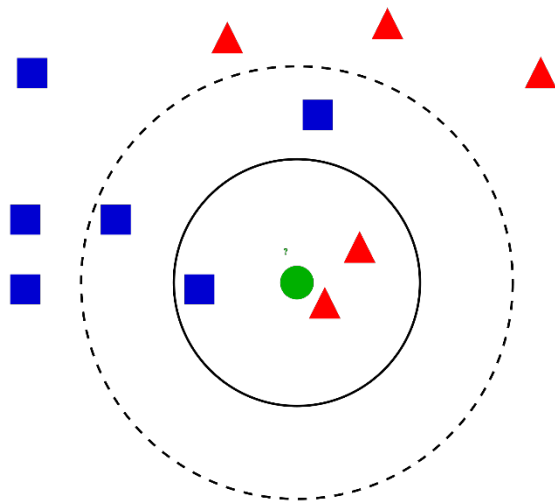
Phương pháp SIFT được cài đặt sẵn trong các công cụ, thư viện xử lý ảnh như OpenCV, Matlab, ... nhưng có một số vấn đề cần lưu ý:

- + SIFT có bản quyền
- + Không hiệu quả khi sự thay đổi trong ảnh lớn.
- + Không hiệu quả với các ảnh có biến dạng mềm.
- + Số điểm đặc trưng của mỗi bức ảnh sẽ khác nhau=> để đối sánh 2 ảnh thì phải chuẩn hóa số điểm đặc trưng giống nhau hoặc đối sánh các cặp và lưu lại các độ tương đồng cao nhất.

Phân lớp kNN (K nearest neighbors) – k láng giềng gần nhất

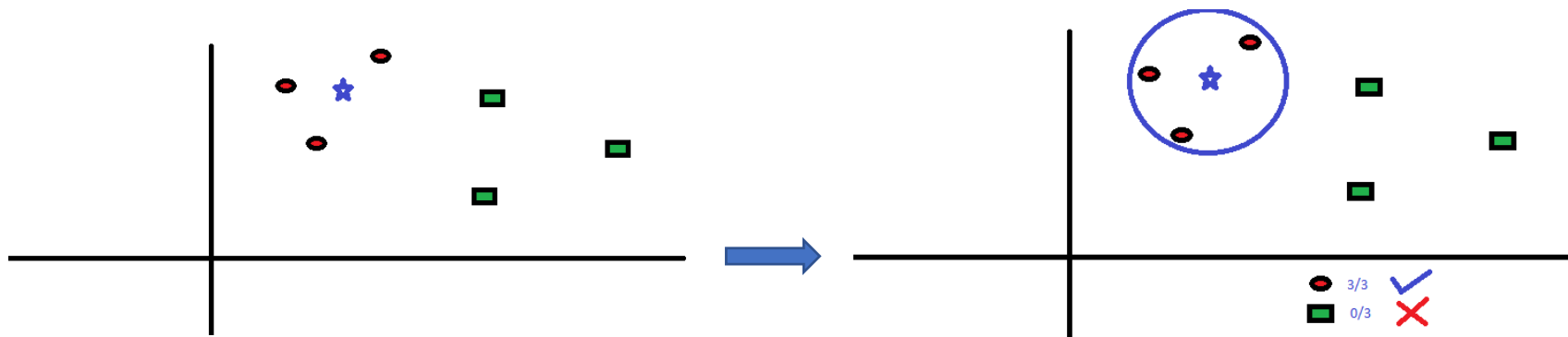
Giả sử có n lớp các đối tượng (người, các đồ vật ...), mỗi đối tượng có một tập các ảnh tương ứng. Chọn 1 số k lẻ (1, 3, 5 ...), một ảnh test sẽ được gán/nhận dạng. Nếu có $> k/2$ ảnh của một lớp nào đó có độ tương đồng (trong số k ảnh có độ tương đồng lớn nhất) thì kết luận ảnh test sẽ thuộc về lớp đó.

Hình tròn màu xanh
sẽ được phân lớp là
hình vuông hay hình
tam giác?



Phân lớp kNN (K nearest neighbors) – k láng giềng gần nhất

Ví dụ minh họa:



Phân lớp kNN (K nearest neighbors) – k láng giềng gần nhất

- Với $k = 1$: nhãn của ảnh test sẽ được gán bằng nhãn của ảnh có độ tương đồng lớn nhất với nó. Đây là trường hợp đơn giản nhất và hay sử dụng nhất.
- Với $k > 1$: tính độ tương đồng của ảnh test so với tất cả các ảnh trong tập training/reference sắp xếp theo chiều giảm dần. Nếu có ít nhất $k/2 + 1$ ảnh của một lớp nào đó nằm trong số k ảnh có độ tương đồng lớn nhất thì kết luận ảnh test thuộc về lớp đó.

Phân lớp kNN (K nearest neighbors) – k láng giềng gần nhất

- Độ tương đồng (similarity) giữa hai bức ảnh được tính bằng một hàm đo khoảng cách (distance) giữa 2 vector đặc trưng tương ứng của chúng. Khoảng cách càng lớn sẽ tương ứng với độ tương đồng càng thấp và ngược lại.
- Các hàm khoảng cách có thể sử dụng ví dụ như hàm Euclidean, Hamming, Cosine, Mahalanobis ...

Phân lớp kNN (K nearest neighbors) – k láng giềng gần nhất

Ví dụ với hàm Euclidean: $X = (x_1, x_2, \dots, x_n)$ và $Y = (y_1, y_2, \dots, y_n)$ thì khoảng cách giữa 2 vector sẽ là:

$$D(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Cho một hệ thống có các mẫu được mô tả bằng các vector 3 chiều như sau:

$$\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} \text{ và vector cần phân lớp } \begin{bmatrix} -1 \\ 1 \\ -2 \end{bmatrix}.$$

Nếu dùng hàm khoảng cách Euclidean để tính độ tương đồng thì khi phân lớp bằng k-NN kết quả là mẫu nào?

Phân lớp kNN (K nearest neighbors) – k láng giềng gần nhất

Gọi các vector lần lượt là v_1 , v_2 , v_3 , v_4 và vector test là v_t , ta có:

$$D(v_t, v_1) = \sqrt{22}$$

$$D(v_t, v_2) = \sqrt{5}$$

$$D(v_t, v_3) = \sqrt{24}$$

$$D(v_t, v_4) = \sqrt{17}$$

Suy ra ảnh test thuộc về lớp thứ 2 (v_2) vì chúng có độ tương đồng lớn nhất.

Trong phương pháp kNN nếu không nói gì thì $k = 1$.

Các giá trị k thường sử dụng là 1, 3, 5.

Phân lớp kNN (K nearest neighbors) – k láng giềng gần nhất

Gọi các vector lần lượt là v_1 , v_2 , v_3 , v_4 và vector test là v_t , ta có:

$$D(v_t, v_1) = \sqrt{22}$$

$$D(v_t, v_2) = \sqrt{5}$$

$$D(v_t, v_3) = \sqrt{24}$$

$$D(v_t, v_4) = \sqrt{17}$$

Suy ra ảnh test thuộc về lớp thứ 2 (v_2) vì chúng có độ tương đồng lớn nhất.

Trong phương pháp kNN nếu không nói gì thì $k = 1$.

Các giá trị k thường sử dụng là 1, 3, 5.

Phân lớp SVM (Support vector machine)

Nghiên cứu các tài liệu và viết 2 trang A4 để trả lời các câu hỏi sau:

1. SVM là gì, hoạt động như thế nào và phù hợp với bài toán nào?
2. Thế nào là SVM nhị phân?
3. Thế nào là SVM đa nhãn?
4. Ví dụ về chương trình (code) sử dụng phương pháp SVM?

Danh sách bài tập lớn

Mỗi sinh viên chọn 1 chủ đề sau:

1. Nhận dạng cử chỉ tay (gesture recognition).
2. Ghép ảnh panorama sử dụng SIFT/SURF trên di động.
3. Phát hiện và đếm phương tiện giao thông.
4. Phát hiện và đếm người sử dụng DNN.
5. Nhận dạng mặt người.
6. Nhận dạng giới tính dựa trên ảnh mặt người.
7. Nhận dạng cảm xúc dựa trên ảnh mặt người.
8. Nhận dạng biển báo giao thông.
9. Phát hiện và nhận dạng biển số xe.
10. Phát hiện và nhận dạng logo các nhãn hàng nổi tiếng.
11. Nhận dạng đánh đi.
12. Phát hiện đám cháy.

Bài tập

- Áp dụng toán tử LBP(8, 1) với các ảnh sau:

92	57	237	113
17	180	14	55
186	227	182	195
191	214	25	32
246	147	31	232

154	149	76	116
84	56	21	164
25	75	152	168
81	123	34	190
162	15	197	215

Bài tập

- Tính khoảng cách Chi-Square của các cặp vector sau:
- $V1 = (0, 1, 8, 2, 5, 3, 4)$, $V2 = (8, 4, 9, 2, 1, 0, 5)$
- $V1 = (2, 5, 9, 7, 1, 2, 6)$, $V2 = (6, 3, 7, 9, 10, 6, 1)$

Bài tập

- Viết chương trình áp dụng toán tử LBP để sinh ảnh LBP từ 1 ảnh input đọc từ file.
- Viết chương trình tính vector SIFT của 1 ảnh input.
- Viết chương trình tính độ tương đồng của 2 ảnh dựa trên SIFT vector của 2 ảnh.
- Viết chương trình đọc 1 thư mục ảnh, đọc 1 ảnh test từ thư mục khác và hiển thị các ảnh giống ảnh test nhất.
- Viết chương trình đọc n ảnh từ 1 thư mục sau đó dùng SIFT để khâu (stitch) thành 1 ảnh dạng panorama.

Questions?