

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

----□&□----



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỀ TÀI:
TÌM HIỂU VÀ ĐÁNH GIÁ MỘT SỐ THUẬT TOÁN
PHÂN CỤM CHO BÀI TOÁN PHÂN ĐOẠN ẢNH

Giảng viên: TS. Mai Tiến Dũng

Thành viên nhóm:

STT	MSSV	Họ Tên
1	19520148	Đinh Hoàng Lộc
2	19521388	Hoàng Tiến Dũng
3	19522485	Trương Minh Tuấn

TP. HỒ CHÍ MINH - 12/2021

LỜI CẢM ƠN

Sau quá trình học tập và rèn luyện tại Trường Đại học Công Nghệ Thông Tin, chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành đồ án môn học của mình.

Chúng em xin gửi lời cảm ơn chân thành đến thầy TS. Mai Tiến Dũng – Giảng viên phụ trách lớp CS231.M11 – Môn Nhập môn Thị giác máy tính đã tận tâm hướng dẫn, truyền đạt những kiến thức cũng như kinh nghiệm cho chúng em trong suốt thời gian học tập.

Trong quá trình làm đồ án môn học, mặc dù nhóm chúng em đã cố gắng nhưng chắc chắn sẽ không tránh được những sai sót không đáng có. Mong nhận được sự góp ý cũng như kinh nghiệm quý báu của các thầy và các bạn sinh viên để được hoàn thiện hơn và rút kinh nghiệm cho những môn học sau. Chúng em xin chân thành cảm ơn!

TP. Hồ Chí Minh, tháng 12 năm 2021.

MỤC LỤC

LÝ DO CHỌN ĐỀ TÀI	5
CHƯƠNG 1. GIỚI THIỆU CHUNG	6
1. Tổng quan về hệ thống xử lý ảnh	6
1.1. <i>Phân thu nhận ảnh (Image Acquisition)</i>	<i>6</i>
1.2. <i>Tiền xử lý (Image Processing)</i>	<i>6</i>
1.3. <i>Phân đoạn (Segmentation) hay phân vùng ảnh</i>	<i>6</i>
1.4. <i>Biểu diễn ảnh (Image Representation)</i>	<i>7</i>
1.5. <i>Nhận dạng và nội suy ảnh (Image Recognition and Interpretation)</i>	<i>7</i>
1.6. <i>Cơ sở tri thức (Knowledge Base)</i>	<i>7</i>
1.7. <i>Mô tả (Biểu diễn ảnh)</i>	<i>7</i>
2. Tổng quan về phân đoạn ảnh (Segmentation)	8
2.1. <i>Phân loại phân đoạn ảnh</i>	<i>9</i>
2.1.1. <i>Phân đoạn cá thể (Instance Segmentation)</i>	<i>10</i>
2.1.2. <i>Phân đoạn ngữ nghĩa (Semantic Segmentation)</i>	<i>10</i>
2.1.3. <i>Phân đoạn toàn cảnh (Panoptic Segmentation)</i>	<i>10</i>
2.2. <i>Ứng dụng của phân đoạn ảnh</i>	<i>11</i>
CHƯƠNG 2. TRÌNH BÀY LÝ THUYẾT CÁC THUẬT TOÁN PHÂN CỤM	12
1. Thuật toán K-Means Clustering	12
1.1. <i>Tổng quan về K-Means Clustering</i>	<i>12</i>
1.2. <i>Các bước thực hiện</i>	<i>12</i>
1.3. <i>Tóm tắt thuật toán</i>	<i>15</i>
1.4. <i>Ưu điểm và hạn chế của K-Means Clustering</i>	<i>16</i>
2. Thuật toán Meanshift	16
2.1. <i>Tổng quan về Meanshift</i>	<i>16</i>
2.2. <i>Các bước thực hiện</i>	<i>16</i>
2.3. <i>Ưu điểm và hạn chế của Meanshift</i>	<i>19</i>
3. Thuật toán Fuzzy C-Means (FCM)	20

3.1. Tổng quan về FCM	20
3.2. Ví dụ về các bước thực hiện trong FCM	20
3.3. Ưu điểm và hạn chế của FCM	23
4. Thuật toán Expectation Maximization (EM)	23
4.1. Tổng quan về EM	23
4.2. Các bước thực hiện	23
4.3. Ưu điểm và hạn chế của EM	24
4.4. Ví dụ về EM:	24
CHƯƠNG 3. CÀI ĐẶT THỰC NGHIỆM	27
1. Tổng quan về hệ thống xử lý ảnh	27
1.1. Môi trường thực nghiệm	27
1.2. Dữ liệu thực nghiệm	27
1.3. Các thư viện cần thiết	28
1.4. Các loại đặc trưng	28
2. Hệ số Silhouette	29
2.1. Tổng quan về hệ số Silhouette	29
2.2. Hệ số Silhouette	30
2.3. Silhouette Score	31
3. Cài đặt các thuật toán và chạy thử nghiệm	32
3.1. Thuật toán K-Means Clustering	32
3.2. Thuật toán Meanshift	32
3.3. Thuật toán Fuzzy C-Means	33
3.4. Thuật toán Expectation Maximization	33
4. Kết quả thực nghiệm	33
5. Nhận xét	35
CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	37
1. Kết luận	37
2. Hướng phát triển	37
TÀI LIỆU THAM KHẢO	39

LÝ DO CHỌN ĐỀ TÀI

Trong những năm gần đây, cùng với sự phát triển vượt bậc của công nghệ thông tin đã thúc đẩy sự phát triển của nhiều lĩnh vực xã hội khác như: y học, giáo dục, giải trí, kinh tế,... Lĩnh vực xử lý ảnh cũng như công nghệ thực tại ảo đã ra đời và thâm nhập mạnh mẽ vào đời sống con người.

Các hệ xử lý ảnh trong quá trình phân tích ảnh, tăng cường ảnh để nâng cao chất lượng ảnh. Do những nguyên nhân khác nhau: có thể do chất lượng thiết bị thu nhận ảnh, ảnh thu được sau quá trình thu nhận ảnh hoặc các phép biến đổi không tránh khỏi nhiễu hoặc khuyết thiếu do nguồn sáng, ảnh có thể bị suy biến. Sự sai sót này một phần bởi các thiết bị quang học và điện tử, phần khác bởi bản thân các phép biến đổi không phải là toàn ánh, nên có sự ánh xạ thiếu hụt đến những điểm trên ảnh kết quả. Để phân tích các đối tượng trong ảnh, chúng ta cần phải phân biệt được các đối tượng cần quan tâm với phần còn lại của ảnh. Những đối tượng này có thể tìm ra được nhờ các kỹ thuật phân đoạn ảnh, theo nghĩa tách phần tiền cảnh ra khỏi hậu cảnh trong ảnh.

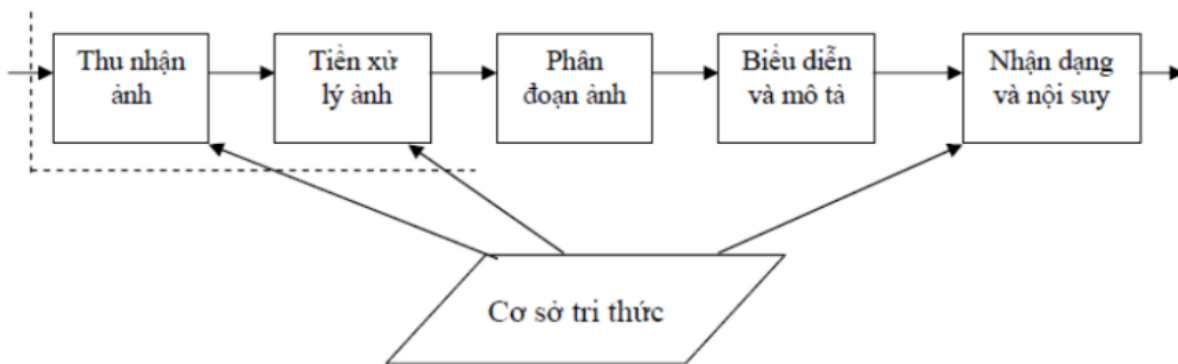
Vì vậy, các nhà khoa học đã phân đoạn ảnh ra để làm rõ nét hơn cho bức ảnh cần xử lý và tăng cường nâng cao chất lượng ảnh. Xuất phát trong hoàn cảnh đó “Tìm hiểu và đánh giá một số thuật toán phân cụm cho bài toán phân đoạn ảnh” đã được nhóm em chọn làm đề tài để nghiên cứu báo cáo.

CHƯƠNG 1. GIỚI THIỆU CHUNG

1. Tổng quan về hệ thống xử lý ảnh

Xử lý và nhận dạng hình ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển rất nhanh, kích thích cho nhiều trung tâm nghiên cứu, ứng dụng. Là một lĩnh vực nhiều thành phần bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh từ dữ liệu đa chiều trong thế giới thực để cho ra các thông tin số hoặc biểu tượng.

Mô tả các bước cơ bản xử lý ảnh bằng hình sau:



Hình 1: Các bước cơ bản trong xử lý ảnh

Sơ đồ này bao gồm các thành phần sau:

1.1. Phần thu nhận ảnh (Image Acquisition)

Ảnh có thể nhận qua camera màu hoặc đen trắng. Chất lượng một ảnh thu nhận được phụ thuộc vào thiết bị thu, môi trường (ánh sáng, phong cảnh).

1.2. Tiền xử lý (Image Processing)

Sau bộ thu nhận, ảnh có thể nhiễu độ tương phản thấp nên cần đưa vào bộ tiền xử lý. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để nâng cao chất lượng ảnh.

1.3. Phân đoạn (Segmentation) hay phân vùng ảnh

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

1.4. Biểu diễn ảnh (*Image Representation*)

Các vật thể sau khi được phân đoạn có thể được mô tả dưới dạng chuỗi các điểm và biểu diễn ảnh thường được sử dụng khi ta quan tâm đến đặc tính bên trong của vùng ảnh. Quá trình biểu diễn ảnh là việc biến đổi các số liệu của ảnh thành dạng thích hợp và cần thiết cho quá trình xử lý bằng máy tính.

1.5. Nhận dạng và nội suy ảnh (*Image Recognition and Interpretation*)

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản:

- Nhận dạng theo tham số.
- Nhận dạng theo cấu trúc.

Một số đối tượng nhận dạng khá phổ biến hiện nay đang được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản (text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người...

1.6. Cơ sở tri thức (*Knowledge Base*)

Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người.

1.7. Mô tả (*Biểu diễn ảnh*)

Từ Hình 1.1, ảnh sau khi số hoá sẽ được lưu vào bộ nhớ, hoặc chuyển sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòi hỏi dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh

thô đó được đặc tả (biểu diễn) lại (hay đơn giản là mã hoá) theo các đặc điểm của ảnh được gọi là các đặc trưng ảnh (Image Features) như biên ảnh (Boundary), vùng ảnh (Region). Một số phương pháp biểu diễn thường dùng:

- Biểu diễn bằng mã chạy (Run-Length Code).
- Biểu diễn bằng mã xích (Chaine-Code).
- Biểu diễn bằng mã tứ phân (Quad-Tree Code).

2. Tổng quan về phân đoạn ảnh (Segmentation)

Trong lĩnh vực Thị giác Máy tính (Computer Vision), phân đoạn ảnh là một quá trình chia một bức ảnh số thành nhiều phần khác nhau (tập hợp các điểm ảnh, hay có thể gọi là superpixels) để biểu diễn phân tích, nhận dạng ảnh. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

Mục tiêu của phân đoạn ảnh là để đơn giản hóa và hoặc thay đổi biểu diễn của một tấm ảnh vào điều gì đó có ý nghĩa hơn và dễ dàng để phân tích. Phân đoạn ảnh thường được sử dụng để xác định vị trí các đối tượng, đường biên (đường thẳng, cong,...). Hay nói cách khác phân đoạn ảnh là một quá trình gán nhãn (assigning a lable) cho mỗi điểm ảnh trong một bức ảnh, các điểm ảnh trong cùng một nhãn sẽ có những đặc tính giống nhau về màu sắc, cường độ hoặc kết cấu của ảnh.

Kết quả của việc phân đoạn ảnh là tập hợp các phân đoạn (segments) bao gồm có thể là toàn bộ bức ảnh hoặc tập hợp các đường biên chiết xuất từ hình ảnh. Các điểm ảnh trong cùng một đoạn có đặc tính tương tự nhau về màu sắc, cường độ hoặc kết cấu. Các đoạn lân cận thì khác nhau đáng kể về các đặc trưng trên.

Trong hầu hết các ứng dụng của lĩnh vực xử lý ảnh (image processing), thị giác máy tính, phân đoạn ảnh luôn đóng một vai trò cơ bản và thường là bước tiền xử lý đầu tiên trong toàn bộ quá trình trước khi thực hiện các thao tác khác ở mức cao hơn như nhận dạng đối tượng, biểu diễn đối tượng, nén ảnh dựa trên đối tượng, hay truy vấn ảnh dựa vào nội dung,... Vào những thời gian đầu, các phương pháp phân đoạn ảnh được đưa ra chủ yếu làm việc trên các ảnh mức xám do các hạn chế về phương tiện thu thập và lưu trữ. Ngày

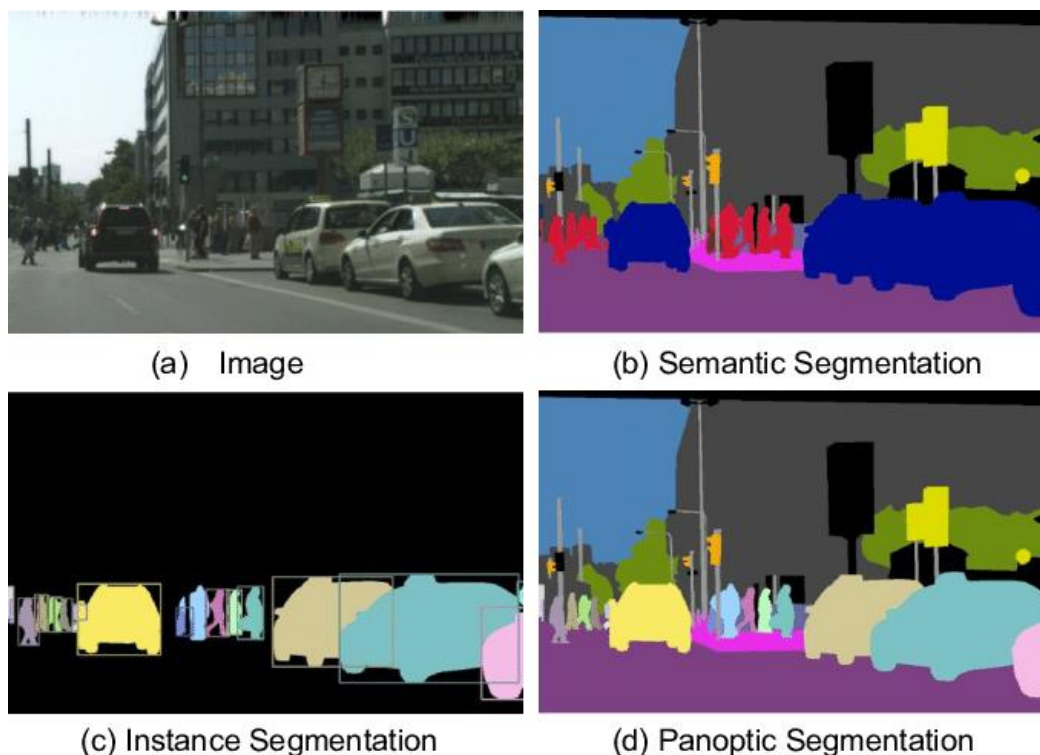
nay, cùng với sự phát triển về các phương tiện thu nhận và biểu diễn ảnh, các ảnh màu đã hầu như thay thế hoàn toàn các ảnh mức xám trong việc biểu diễn và lưu trữ thông tin do các ưu thế vượt trội hơn hẳn so với ảnh mức xám. Do đó, các kỹ thuật, thuật giải mới thực hiện việc phân đoạn ảnh trên các loại ảnh màu liên tục được phát triển để đáp ứng các nhu cầu mới. Các thuật giải, kỹ thuật này thường được phát triển dựa trên nền tảng các thuật giải phân đoạn ảnh mức xám đã có sẵn.

2.1. Phân loại phân đoạn ảnh

Có hai loại phân đoạn chính:

- *Phân đoạn theo cá thể (Instance Segmentation).*
- *Phân đoạn theo ngữ nghĩa (Semantic Segmentation).*

Gần đây có loại phân đoạn là *Phân đoạn khái quát (Panoptic Segmentation)* hay *Phân đoạn toàn cảnh* đang được phát triển trong lĩnh vực phân đoạn ảnh.



Hình 2: Các loại phân đoạn ảnh

Ta có thể thấy trong hình trên là một hình ảnh chứa các đối tượng có hình dạng được xác định rõ và có thể đếm được. Ví dụ: Người, xe hơi, cây cối, ...

Các vùng không có đối tượng mà ta có thể thấy là vùng nền vô định hình. Ví dụ: bầu trời, nước, tòa nhà.

Nhìn vào kết quả phân đoạn, ta có thể hiểu khái quát các loại phân đoạn ảnh này hoạt động như sau:

2.1.1. Phân đoạn cá thể (Instance Segmentation)

Nó phân đoạn các trường hợp khác nhau của mỗi loại ngữ nghĩa và do đó nó xuất hiện như một phần mở rộng của phân đoạn ngữ nghĩa. Mỗi cá thể của một đối tượng (ví dụ: người) được gán nhãn riêng biệt và các phân đoạn không có cá thể nào bị bỏ qua.

Ví dụ: Nếu ta nhìn vào hình (c), mỗi người/ô tô được dán nhãn với các màu khác nhau (tức là) mỗi người trong một hình ảnh được dán nhãn riêng biệt như person1, person2, car1, car2 thay vì gán nhãn chung như person giống Semantic Segmentation.

2.1.2. Phân đoạn ngữ nghĩa (Semantic Segmentation)

Phân loại rộng rãi các đối tượng thành các danh mục ngữ nghĩa như người, sách, hoa, xe hơi, ... Và đề cập đến nhiệm vụ xác định các lớp đối tượng khác nhau trong một hình ảnh. Ở đây tất cả các vùng của hình ảnh được xem xét và các đối tượng cùng loại được coi là một nhãn duy nhất.

Ví dụ: Nếu nhìn vào hình (b), tất cả các trường hợp của đối tượng là ô tô hoặc người có cùng màu, cũng có thể hiểu là chúng được gán một nhãn chung.

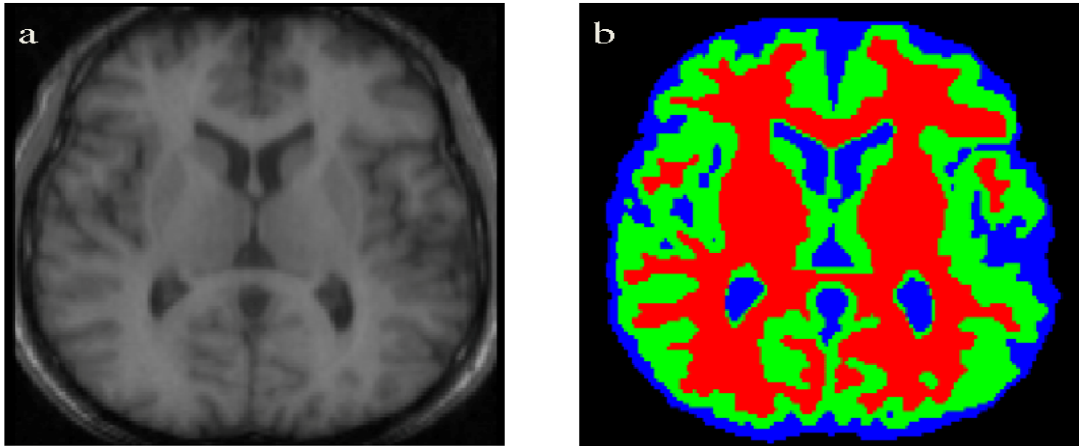
2.1.3. Phân đoạn toàn cảnh (Panoptic Segmentation)

Nó là sự kết hợp giữa Phân đoạn ngữ nghĩa và Phân đoạn cá thể sao cho tất cả các pixel được gán nhãn lớp và tất cả các thể hiện đối tượng được phân đoạn duy nhất. Ở đây, mỗi pixel được gán một nhãn cho từng lớp (ví dụ: người) và nếu một phân đoạn thuộc danh mục, thì mỗi thể hiện của nó được gán nhãn riêng lẻ. Chú thích mẫu, cú pháp cho từng chú thích: [`<class label>`, `<instance id>`].

Ví dụ: [“person”, “1”], [“building”, “None”].

2.2. Ứng dụng của phân đoạn ảnh

Phân đoạn ảnh giải quyết các vấn đề về như: hiểu ảnh, đối sánh và nhận dạng ảnh, dò tìm các đối tượng trong ảnh được ứng dụng trong nhiều lĩnh vực như y tế, công nghệ, giao thông, trí tuệ nhân tạo, ... Hiện nay, các nhà khoa học vẫn đang khai thác tiềm năng của phân đoạn ảnh trên nhiều lĩnh vực khác nhau.



Hình 3.1: Ảnh minh họa ứng dụng cho lĩnh vực y tế



Hình 3.2: Ảnh minh họa ứng dụng cho lĩnh vực giao thông



Hình 3.3: Ảnh minh họa ứng dụng công nghệ nhận dạng vân tay

CHƯƠNG 2. TRÌNH BÀY LÝ THUYẾT CÁC THUẬT TOÁN PHÂN CỤM

1. Thuật toán K-Means Clustering

1.1. Tổng quan về K-Means Clustering

K-Means Clustering là một thuật toán cơ bản của học máy không giám sát. Nó có thể được sử dụng để phân tách dữ liệu không nhãn thành các nhóm dựa trên các tính chất tương tự nhau.

1.2. Các bước thực hiện

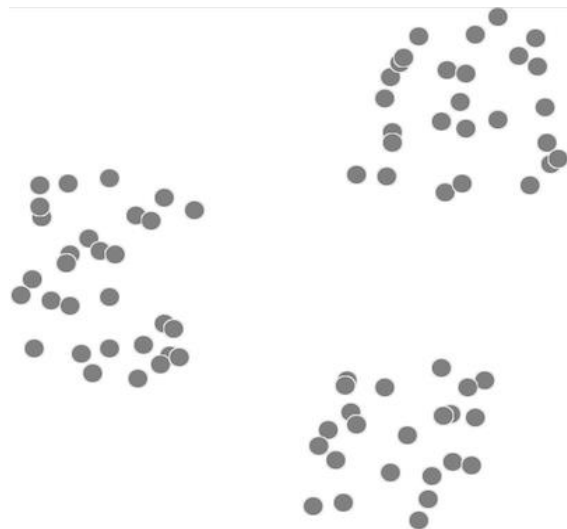
K-Means clustering sẽ nhóm các dữ liệu có tính chất giống nhau lại tạo thành các cụm.

Mỗi trung tâm cụm là đại diện của dữ liệu trong cụm đó. Các trung tâm cụm này sẽ thay thế tất cả các dữ liệu trong các cụm tương ứng của chúng.

Vì vậy, chúng ta chỉ cần lưu trữ nhãn cho từng pixel cho biết cụm mà pixel này thuộc về. Do đó chúng ta giữ bản ghi các dữ liệu của từng trung tâm cụm.

1.2.1. Bước 1

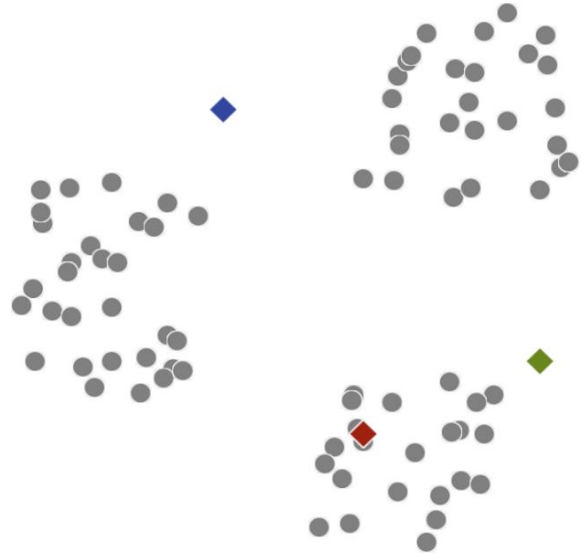
Đầu tiên là chuẩn bị dữ liệu cần phân cụm và quyết định số lượng cụm cần phân chia. Giả sử số cụm cần phân chia là 3.



Hình 4.1: Bước 1

1.2.2. Bước 2

Chọn ngẫu nhiên 3 điểm làm điểm trung tâm của cụm.

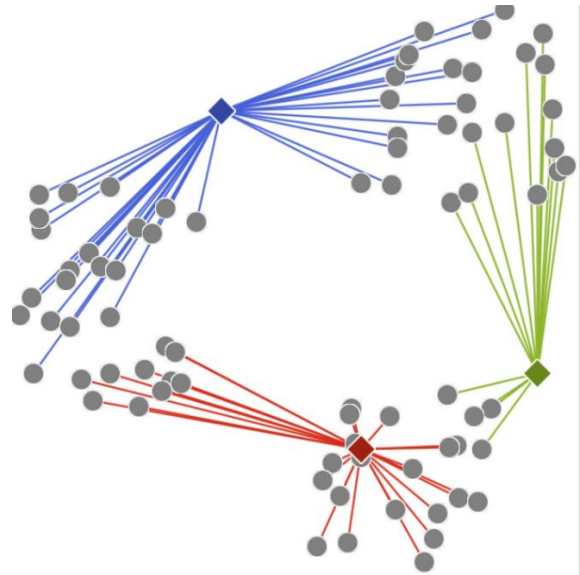


Hình 4.2: Bước 2

1.2.3. Bước 3

Tính toán khoảng cách từ 3 điểm được chọn đó đến các điểm không được chọn làm trung tâm. Phân mỗi điểm dữ liệu vào cụm có điểm trung tâm gần nó nhất.

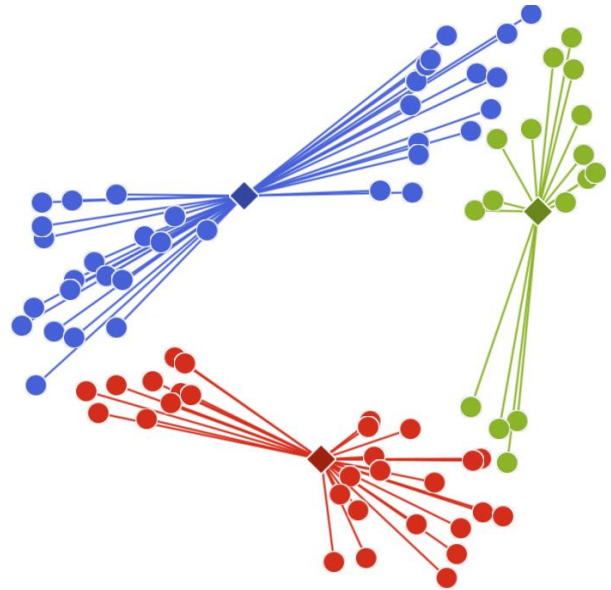
Khoảng cách ở đây biểu diễn bằng các đường màu xanh da trời, xanh lá và màu đỏ như thể hiện ở bên.



Hình 4.3: Bước 3

1.2.4. Bước 4

Qua bước 3, ta đã phân được thành 3 cụm. tiếp theo ta sẽ tính trung bình cộng giữa các điểm trong mỗi cụm, điểm vừa tính được chính là điểm trung tâm mới của cụm đó.

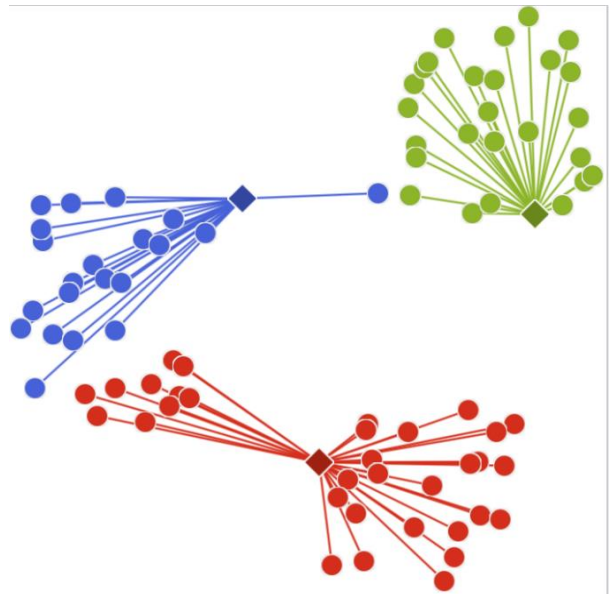


Hình 4.4: Bước 4

1.2.5. Bước 5

Một lần nữa tiến hành bước 3, tính toán lại khoảng cách các điểm đến các điểm trung tâm.

Sau đó phân loại lại các điểm dữ liệu về các cụm.

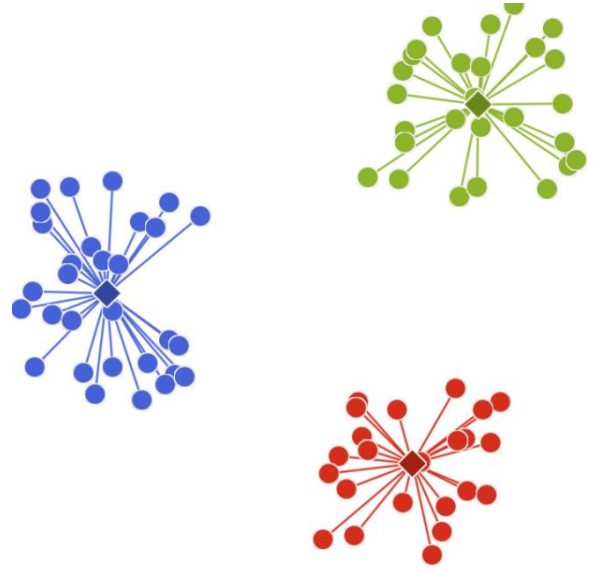


Hình 4.5: Bước 5

1.2.6. Bước 6

Sau đó lặp lại quá trình di chuyển trung tâm cụm và phân loại lại các điểm về các cụm gần nhất.

Quá trình này sẽ dừng khi sau khi dữ liệu sau khi phân cụm lại không thay đổi gì so với lần trước.

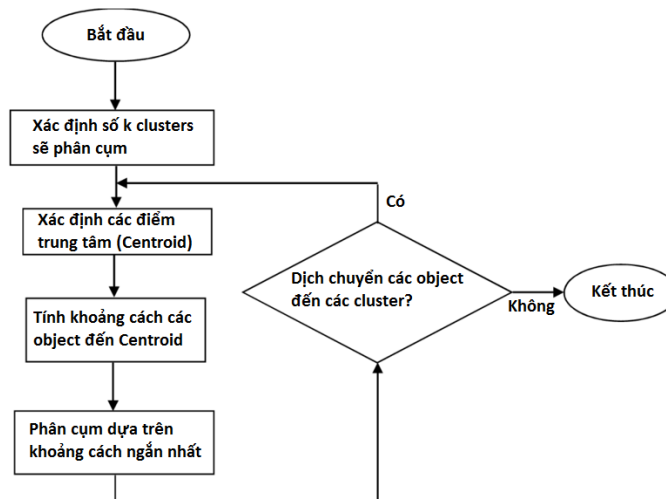


Hình 4.6: Bước 6

1.3. Tóm tắt thuật toán

- Đầu vào: Dữ liệu X và số lượng cluster cần tìm K.
- Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y.

* Lưu đồ thuật toán:



Hình 5: Lưu đồ thuật toán K-Means Clustering

Giải thích lưu đồ:

Bước 1: Chọn K điểm bất kì làm các center ban đầu.

Bước 2: Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.

Bước 3: Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.

Bước 4: Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cluster đó sau bước 2.

Bước 5: Quay lại bước 2.

1.4. Ưu điểm và hạn chế của K-Means Clustering

1.4.1. Ưu điểm

- Dễ dàng cài đặt và sử dụng.
- Hoạt động tốt đối với hình ảnh có các mảng màu tách biệt.

1.4.2. Hạn chế

- Center khởi tạo ban đầu mà quá lệch so với center cụm thì kết quả phân cụm không cao (có nghĩa là cụm dữ liệu được khám phá rất lệch so với thực tế).
- Nghiệm cuối cùng phụ thuộc vào các centers được khởi tạo ban đầu. Khi hệ số k lớn và số vòng lặp lớn thì thuật toán chạy chậm rãi.

2. Thuật toán Meanshift

2.1. Tổng quan về Meanshift

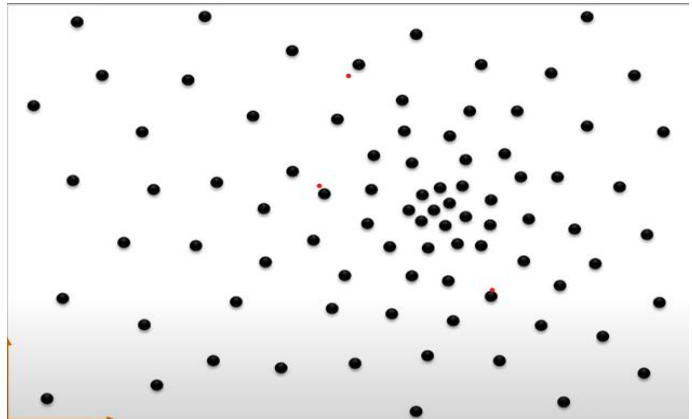
Là thuật toán phân cụm phổ biến và mạnh mẽ được sử dụng trong học không giám sát. Bằng cách lặp đi lặp lại việc gán các điểm dữ liệu đến các cụm về phía mode (mode là khu vực có mật độ điểm dữ liệu cao nhất, trong các khái niệm của Meanshift), nên vì thế nó còn có tên gọi là thuật toán Mode-Seeking. Thuật toán Mean-shift được áp dụng nhiều trong các lĩnh vực xử lý ảnh và thị giác máy tính

2.2. Các bước thực hiện

Nhìn chung cách hoạt động của thuật toán này khá là đơn giản, gồm các bước sau:

2.2.1. Bước 1

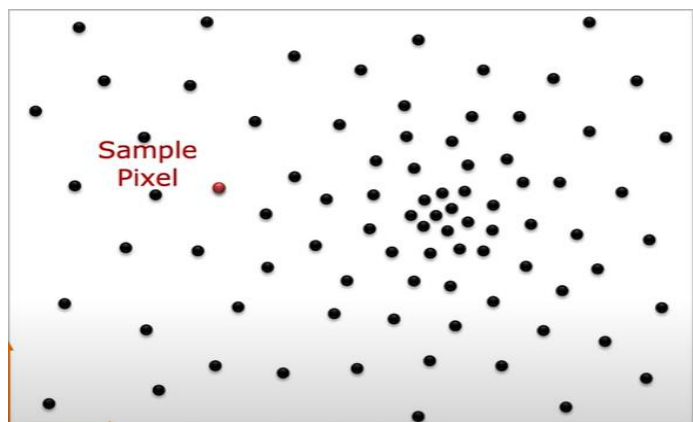
Giả sử, ta có một không gian đặc trưng hai chiều gồm các pixel phân bố không đều nhau.



Hình 6.1: Bước 1

2.2.2. Bước 2

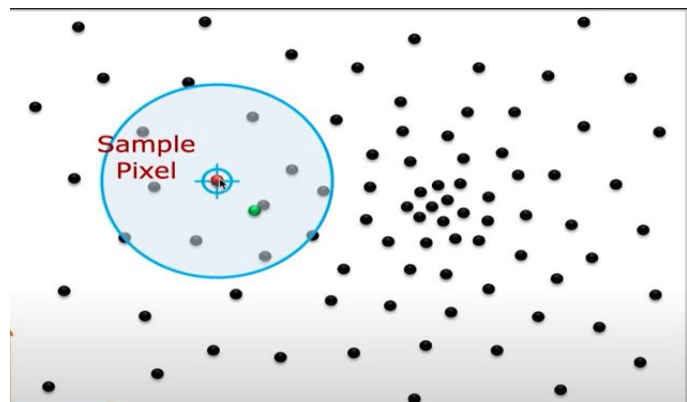
Ở hình 4.2.2, ta chọn một điểm dữ liệu mẫu.



Hình 6.2: Bước 2

2.2.3. Bước 3

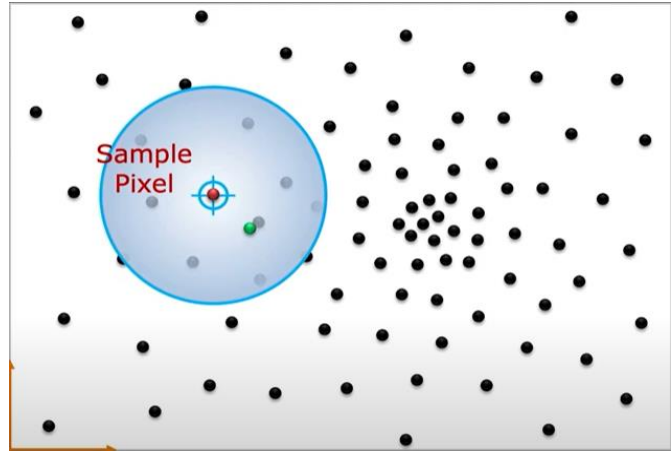
Tạo Window cho điểm dữ liệu mẫu.



Hình 6.3: Bước 3

2.2.4. Bước 4

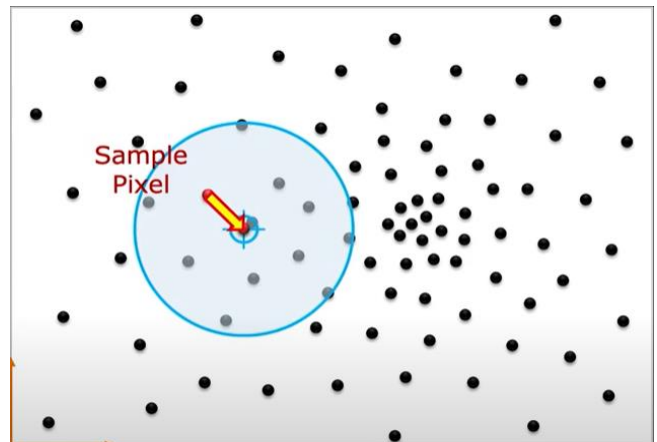
Tính toán lại giá trị trung bình của tất cả các điểm trong Window này (giả sử kết quả ta tính ra được nằm ở điểm màu xanh lá cây)



Hình 6.4: Bước 4

2.2.5. Bước 5

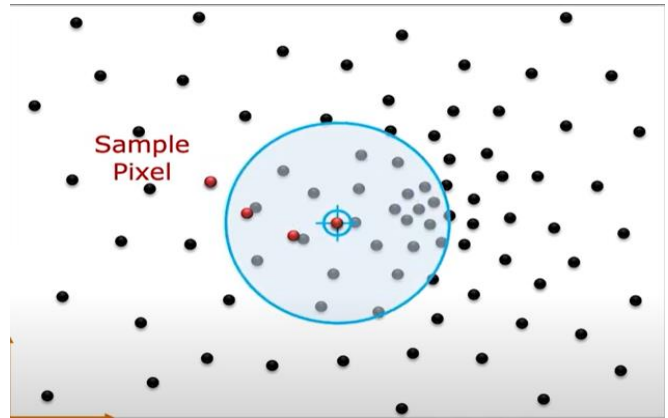
Từ pixel được chọn ban đầu, di chuyển đến điểm có giá trị trung bình mới vừa được tính.



Hình 6.5: Bước 5

2.2.6. Bước 6

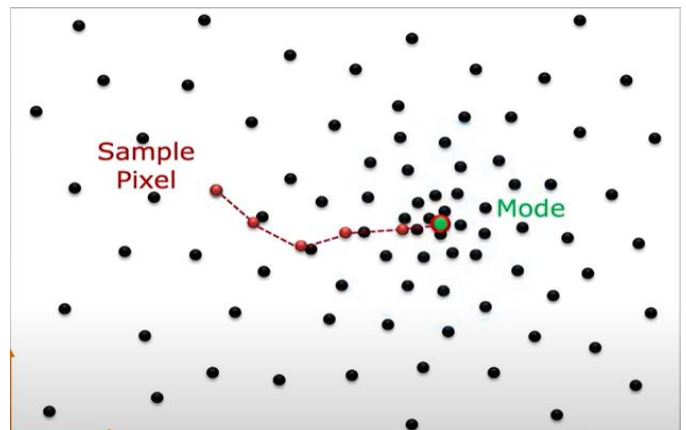
Tiếp tục tính giá trị trung bình trong Window dựa vào các điểm trung bình tìm được trước đó trong bán kính W .



Hình 6.6: Bước 6

2.2.7. Bước 7

Đến khi hội tụ (khi không có sự dịch chuyển trong lần tính kế tiếp) thì ta dừng thuật toán



Hình 6.7: Bước 7

Thực hiện tương tự cho các điểm dữ liệu. Sau khi kết thúc, với kết quả có nhiều Window chồng chéo lên nhau, Window chứa nhiều điểm nhất sẽ được giữ lại và các Window khác sẽ bị xóa đi. Gán các điểm dữ liệu cho Window mà chúng đang đứng

2.3. Ưu điểm và hạn chế của MeanShift

2.3.1. Ưu điểm

- Tìm được số lượng các cụm hợp lý nhất cho mỗi mức ảnh, xử lý tốt được các trường hợp ngoại lệ.

- Việc phân cụm chỉ phụ thuộc vào một tham số duy nhất là kích cỡ W của window.

2.3.2. Hạn chế

- Việc tính toán sẽ tương đối tốn kém do mỗi lần cập nhật lại giá trị trung bình mới để dịch chuyển ta phải tính toán lại giá trị đó và với mỗi pixel ta đều phải làm công việc tính toán giá trị trung bình đó.

3. Thuật toán Fuzzy C-Means (FCM)

3.1. Tổng quan về FCM

Là phương pháp phân cụm mà bộ dữ liệu được gom thành nhiều cụm mà với mỗi điểm dữ liệu trong bộ dữ liệu đều thuộc một phần trong mỗi cụm đó. Nói cách khác, một điểm có thể thuộc nhiều hơn một cụm. Thuật toán này được phát triển bởi Dunn năm 1973 và được cải tiến bởi Bezdek năm 1981.

FCM là một dạng soft-clustering (phân cụm mềm) khác với hard-clustering như k-means do một điểm dữ liệu không chỉ được phân vào một cụm mà có thể thuộc nhiều cụm tùy thuộc vào xác suất của nó đối với các cụm đó.

3.2. Ví dụ về các bước thực hiện trong FCM

Giả sử cho 4 điểm trên mặt phẳng Oxy có tọa độ:

$$x_1 = \{1; 3\}$$

$$x_2 = \{1,5; 3,2\}$$

$$x_3 = \{1,3; 2,8\}$$

$$x_4 = \{3; 1\}$$

3.2.1. Bước 1:

Khởi tạo giá trị c là số cụm là 2 và một ma trận phân vùng có kích cỡ 2x4

$$U^{(0)} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Và đưa ra ngưỡng dừng lặp lại cho thuật toán là:

$$\varepsilon \leq 0,01$$

Có nghĩa là:

$$\max_{i,k} \left| \mu_{ik}^{(\gamma+1)} - \mu_{ik}^{(\gamma)} \right| \leq 0,01$$

3.2.2. Bước 2:

Tính tọa độ của các tâm cụm:

$$V_{ij} = \frac{\sum_{k=1}^n (\mu_{ik})^2 * x_{kj}}{\sum_{k=1}^n (\mu_{ik})^2}$$

$$V_{ij} = \frac{\mu_1^2 * x_{1j} + \mu_2^2 * x_{2j} + \mu_3^2 * x_{3j} + \mu_4^2 * x_{4j}}{\mu_1^2 + \mu_2^2 + \mu_3^2 + \mu_4^2}$$

$$V_{11} = \frac{1 * 1 + 1 * 1.5 + 1 * 1.3 + 0 * 3}{1 + 1 + 1 + 0} \approx 1.26$$

$$V_{12} = \frac{1 * 3 + 1 * 3.2 + 1 * 2.8 + 0 * 1}{1 + 1 + 1 + 0} = 3$$

$$\text{Ta được tọa độ: } V_1 = \{1.26; 3\} \quad V_{21} = \frac{3*1}{1} = 3$$

$$V_{22} = \frac{1 * 1}{1} = 1$$

$$\text{Ta được tọa độ: } V_2 = \{3; 1\}$$

Tiếp theo, tính khoảng cách từ điểm đến các tâm cụm:

$$d_{11} = \sqrt{(1 - 1.26)^2 + (3 - 3)^2} = 0.26$$

$$d_{12} = 0.31$$

$$d_{13} = 0.2$$

$$d_{14} = 2.65$$

$$d_{21} = 2.82$$

$$d_{22} = 2.66$$

$$d_{23} = 2.47$$

$$d_{24} = 0$$

3.2.3. Bước 3:

Cập nhật lại ma trận phân vùng:

$$\mu_{ik}^{(\gamma+1)} = \left[\sum_{j=1}^c \left(\frac{d_{ik}^2}{d_{jk}^2} \right)^{\gamma} \right]^{-1}$$

$$\mu_{11} = \left[\left(\frac{d_{11}}{d_{11}} \right)^2 + \left(\frac{d_{11}}{d_{21}} \right)^2 \right]^{-1} = \left[\left(\frac{0.26}{0.26} \right)^2 + \left(\frac{0.26}{2.82} \right)^2 \right]^{-1} = 0.991$$

$$\mu_{12} = \left[1 + \left(\frac{0.31}{2.66} \right)^2 \right]^{-1} = 0.986$$

$$\mu_{13} = \left[1 + \left(\frac{0.2}{2.47} \right)^2 \right]^{-1} = 0.993$$

$$\mu_{14} = 0$$

Ta được ma trận mới:

$$U^{(1)} = \begin{bmatrix} 0.991 & 0.986 & 0.993 & 0 \\ 0.009 & 0.014 & 0.007 & 1 \end{bmatrix}$$

3.2.4. Bước 4:

Kiểm tra điều kiện hội tụ:

$$\max_{i,k} \left| \mu_{ik}^{(1)} - \mu_{ik}^{(0)} \right| = 0.0139 > 0.1$$

Nếu vẫn chưa thỏa mãn quay lại thực hiện các bước 2 và 3 cho tới khi nào đủ điều kiện

3.3. Ưu điểm và hạn chế của FCM

3.3.1. Ưu điểm

- Cho ra được kết quả tốt nhất với những bộ dữ liệu bị chong chéo.
- Điểm dữ liệu sẽ được gán thành viên với mỗi tâm cụm để biết được điểm này thuộc nhiều cụm trong bộ dữ liệu.

3.3.2. Hạn chế

- Phải khởi tạo trước số lượng cụm.
- Giá trị ngưỡng epsilon càng nhỏ sẽ đưa ra kết quả càng tốt nhưng sẽ làm số lần lặp lại tăng lên làm cho việc tính toán trở nên tốn kém hơn. Vì thế, ta phải đưa ra giá trị ngưỡng epsilon một cách hợp lý.

4. Thuật toán Expectation Maximization (EM)

4.1. Tổng quan về EM

Là một thuật toán gom nhóm dữ liệu được dùng trong các tác vụ khám phá tri thức.

Trong thống kê, thuật toán này sẽ lặp và tối ưu hóa khả năng nhìn thấy dữ liệu quan sát được thông qua việc ước lượng tham số cho mô hình thống kê cho các biến không quan sát được.

4.2. Các bước thực hiện

Thuật toán sẽ bắt đầu bằng cách tham số hóa cho mô hình dự đoán bằng cách đưa ra một giá trị trung bình và độ lệch chuẩn (nếu chúng ta sử dụng mô hình phân phối bình thường) rồi thực hiện lặp đi lặp lại các bước sau:

E-step: Dựa vào giá trị tham số ban đầu, tính toán xác suất gán nhãn các điểm dữ liệu vào một nhóm.

M-step: Tính toán lại các tham số đó dựa vào xác suất mới vừa tính được ở E-step

Lặp đi lặp lại cho đến khi các nhóm gom được ổn định hay hội tụ.

4.3. Ưu điểm và hạn chế của EM

4.3.1. Ưu điểm

- Dễ dàng cài đặt và sử dụng.
- Dự đoán được các dữ liệu bị thiếu xuyên suốt quá trình lặp.
- Hữu ích cho tác vụ gom nhóm và hình thành mô hình qua các tham số.

4.3.2. Hạn chế

- Chỉ chạy nhanh ở các vòng lặp đầu nhưng chậm hơn ở các vòng lặp sau.
- Không phải lúc nào cũng tìm được tham số tối ưu mà sẽ có trường hợp bị mắc kẹt ở điểm tối ưu cục bộ thay vì toàn cục.

4.4. Ví dụ về EM:

Cho biết C1 và C2 là 2 đồng xu

θ_1 và θ_2 lần lượt là xác suất tung được mặt ngửa của C1 và C2

Ta hãy giá trị của θ_1 và θ_2 bằng cách tung 2 đồng xu đó nhiều lần.

Giả sử ta chọn 5 lần vào một trong hai đồng xu đó một cách ngẫu nhiên. Mỗi lần chọn đồng xu ta tung 10 lần.

Nếu như ta không biết đâu là C1 và C2, hãy giả sử ước lượng cho 2 giá trị θ_1 và θ_2 :

$$\theta_1 = 0.6$$

$$\theta_2 = 0.5$$

Và kết quả chúng ta thu được như sau:

STT	Chú thích: H(Head): mặt ngửa, T(Tail): mặt sấp									
1	H	T	T	T	H	H	T	H	T	H
2	H	H	H	H	T	H	H	H	H	H
3	H	T	H	H	H	H	H	T	H	H
4	H	T	H	T	T	T	H	H	T	T
5	T	H	H	H	T	H	H	H	T	H

Bước E:

Ta sẽ sử dụng công thức phân phối nhị thức để đi tìm khả năng (likelihood) của 2 đồng xu bằng công thức

$$L(C) = \Theta^k(1 - \Theta)^{n-k}$$

Theo công thức này ta sẽ tìm likelihood cho 10 lần tung đồng xu đầu tiên (gồm 5 lần ngửa và 5 lần sấp như trên)

$$L(A) = 0.6^5(1 - 0.6)^5 = 0.0007963$$

$$L(B) = 0.5^5(1 - 0.5)^5 = 0.0009766$$

Chuẩn hóa dữ liệu đưa về xác suất:

$$P(A) = \frac{L(A)}{L(A)+L(B)} = 0.45$$

$$P(B) = 1 - P(A) = 0.55$$

Bước M:

Dựa vào xác suất này ta tìm được likelihood của mặt sấp và mặt ngửa

Đồng xu thứ nhất		Đồng xu thứ hai	
L(H)	L(T)	L(H)	L(T)
2.2	2.2	2.8	2.8
7.2	0.8	1.8	0.2
5.9	1.5	2.1	0.5
1.4	2.1	2.6	3.9
4.5	1.9	2.5	1.1

Với đồng xu thứ nhất:

$$\sum L(H) = 21.2 \quad \sum L(T) = 8.5 \quad \Theta_1 = \frac{21.2}{21.2+8.5} = 0.71$$

Với đồng xu thứ hai:

$$\sum L(H) = 11.8 \quad \sum L(T) = 8.5 \quad \Theta_1 = \frac{11.8}{11.8+8.5} = 0.58$$

Và hai giá trị Θ_1 và Θ_2 sẽ được dùng để tính toán trong lần lặp tiếp theo. Chúng ta sẽ dùng thuật toán khi giá trị này ổn định.

CHƯƠNG 3. CÀI ĐẶT THỰC NGHIỆM

1. Tổng quan về hệ thống xử lý ảnh

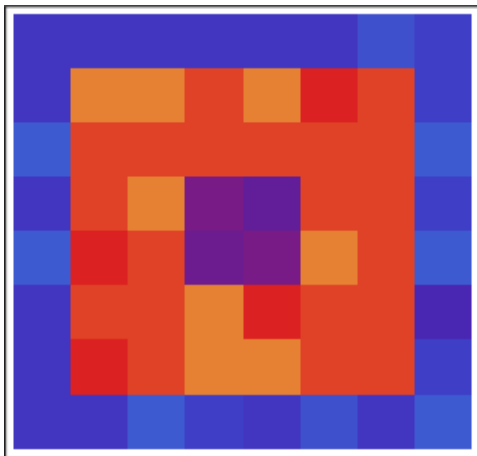
1.1. Môi trường thực nghiệm

Nhóm tiến hành thử nghiệm các thuật toán phân cụm ở Google Colaboratory (gọi tắt là “Colab”) là một sản phẩm của Google Research. Chọn Colab là nơi thử nghiệm là vì:

- Không yêu cầu cấu hình
- Dễ dàng quan sát, theo dõi, chỉnh sửa với các thành viên trong một nhóm.
- Tích hợp nhiều thư viện hỗ trợ cho môn học.
- Ngoài ra, Colab cung cấp GPU miễn phí (GPU không dùng trong đồ án).

1.2. Dữ liệu thực nghiệm

Tiến hành thử nghiệm các thuật toán trên bốn ảnh sau:





Hình 7: Các ảnh dùng cho thực nghiệm

1.3. Các thư viện cần thiết

Với thuật toán Kmeans, Mean shift, Expectation Maximization Clustering dùng thư viện: Sklearn

- Kmeans: `from sklearn.cluster import KMeans`
- Mean shift: `from sklearn.cluster import MeanShift, estimate_bandwidth`
- Expectation Maximization Clustering:


```
from sklearn.mixture import GaussianMixture
import scipy.stats
```
- Với thuật toán Fuzzy Clustering dùng thư viện: fuzzy-c-means


```
!pip install fuzzy-c-means
from fcmeans import FCM
```
- Một số thư viện khác: os, numpy, matplotlib, opencv.

1.4. Các loại đặc trưng

1.4.1. R, G, B

- Các đặc trưng được dùng để phân cụm là đặc trưng màu của ảnh: Red, Green, Blue.
- Cài đặt:

```
nrow, ncol, nchl = img.shape
g = []
g = img.reshape(nrow*ncol, nchl)
```

1.4.2. R, G, B, X, Y

- Gồm các đặc trưng màu và vị trí của điểm ảnh: Red, Green, Blue, X, Y.

```
nrow, ncol, nchl = img.shape
g = []
for i in range(nrow):
    for j in range(ncol):
        temp = [img[i][j][0], img[i][j][1], img[i][j][2],
i, j]
        g.append(temp)
```

1.4.3. R, G, B, X / num_of_rows * 255, Y / num_of_columns * 255 (RGBX'Y')

- Trong quá trình thực nghiệm, nhóm nhận thấy với đặc trưng R, B, G, X, Y sẽ tạo ra những kết quả không tốt do ảnh hưởng của vị trí. Nhóm đã đề xuất scale vị trí của điểm ảnh về đoạn [0, 255], cùng khoảng với đặc trưng màu RGB.

```
nrow, ncol, nchl = img.shape
g = []
for i in range(nrow):
    for j in range(ncol):
        temp = [img[i][j][0], img[i][j][1], img[i][j][2],
int(i/nrow*255), int(j/ncol*255)]
        g.append(temp)
```

2. Hệ số Silhouette

2.1. Tổng quan về hệ số Silhouette

- Phương pháp Silhouette là phương pháp đánh giá các kết quả phân cụm phổ biến và được sử dụng nhiều nhất.

- Hệ số Silhouette mục đích để đo lường mức độ tối ưu khi một quan sát, một điểm dữ liệu được phân vào các cluster bất kỳ. Phương pháp Silhouette sẽ cho chúng ta biết những điểm dữ liệu hay những quan sát nào nằm gọn bên trong cụm (tốt) hay nằm gần ngoài rìa cụm (không tốt) để đánh giá hiệu quả phân cụm.
- Thông thường, phân cụm được coi là một phương pháp Không giám sát (Unsupervised), do đó rất khó để thiết lập một số liệu hiệu suất tốt.
- Lý do nhóm chúng em chọn Silhouette vì:
 - + Chỉ số Silhouette Index là chỉ số đánh giá các kết quả phân cụm phổ biến và được sử dụng nhiều nhất.
 - + Silhouette có sẵn hàm trong sklearn -> rất thuận tiện cho việc sử dụng
 - + `from sklearn import metrics.silhouette_score`

2.2. Hệ số Silhouette

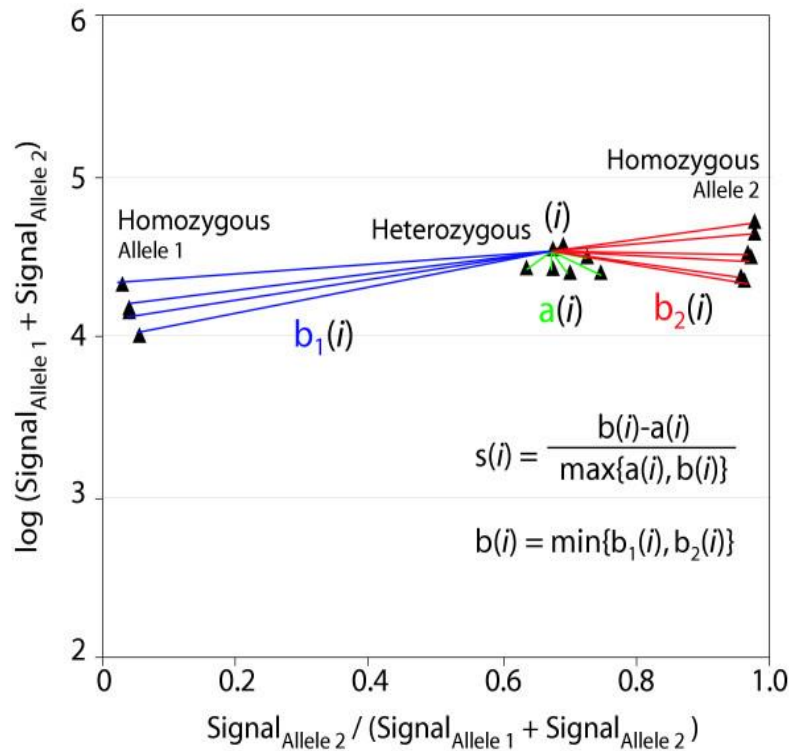
Silhouette đo lường khoảng cách của một điểm dữ liệu trong cụm đến Centroid, điểm trung tâm của cụm, và khoảng cách của chính điểm đó đến điểm trung tâm của cụm gần nhất (hoặc đến các điểm trung tâm cụm còn lại và lấy khoảng cách ngắn nhất).

Phạm vi [-1, 1]. Công thức:

$$s_i = (b_i - a_i) / \max(a_i, b_i)$$

- b_i là khoảng cách trung bình từ điểm i trong cluster A đến tất cả các điểm trong cluster B với cluster B là cluster láng giềng gần nhất. (tính giá trị trung bình tới từng cụm sau đó chọn giá trị trung bình nhỏ nhất làm $b(i)$)
- a_i là khoảng cách trung bình từ điểm i trong cluster A đến tất cả các điểm còn lại trong A

- $\max(b_i, a_i)$ tức lấy chọn giá trị lớn nhất giữa a_i và b_i



Hình 4.2.1: Ảnh minh họa về hệ số Silhouette

Sử dụng khoảng cách Euclidean để tính khoảng cách giữa các điểm:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Hệ số Silhouette cho biết:

- + Hệ số Silhouette gần 1 cho biết rằng mẫu ở xa các cụm lân cận.
- + Giá trị 0 cho biết mẫu nằm trên hoặc rất gần ranh giới quyết định giữa 2 cụm lân cận và
- + Các giá trị âm cho biết những mẫu đó **có thể** đã được gán cho cụm sai.

2.3. Silhouette Score

Silhouette Score được tính bằng giá trị trung bình của tất cả các hệ số Silhouette được tính trên tất cả các điểm dữ liệu.

Silhouette Score là thước đo mức độ của dữ liệu đã được phân nhóm.

Theo kinh nghiệm của tác giả trong tài liệu “Data mining and Predictive analytics” của nhà xuất bản Wiley:

- + Điểm trung bình Silhouette từ 0.5 trở lên, bằng chứng cho thấy có thể kết quả phân cụm sát với thực tế.
- + Điểm trung bình từ 0.25 đến 0.5 thì cần thêm kiến thức chuyên môn, kinh nghiệm để đánh giá thêm khả năng cluster có trong thực tế.
- + Điểm trung bình dưới 0.25, thì không nên tin tưởng cluster, và cần đi tìm nhiều bằng chứng khác.

Bản chất của Silhouette Score là dùng để đánh giá thuật toán phân cụm. Liệu ta có thể so sánh và đánh giá chất lượng phân đoạn ảnh thông qua Silhouette Score hay không? Câu trả lời là “Có”, tuy nhiên có một số điều kiện nhất định.

3. Cài đặt các thuật toán và chạy thử nghiệm

3.1. Thuật toán *K-Means Clustering*

```
k_means = KMeans(n_clusters = k, random_state = 0).fit(g)
tt = k_means.cluster_centers_[k_means.labels_]
t = tt[:,0:3]
img_res = t.reshape(nrow, ncol, nchl)
```

3.2. Thuật toán *Meanshift*

```
bandwidth_=estimate_bandwidth(g, quantile=0.2, n_samples=500)
ms = MeanShift(bandwidth=bandwidth_, bin_seeding=True)
ms.fit(g)
tt = ms.cluster_centers_[ms.labels_]
t = tt[:,0:3]
img_res = t.reshape(nrow, ncol, nchl)
```


3.3. Thuật toán Fuzzy C-Means

```
fcm = FCM(n_clusters=k)
fcm.fit(np.array(g))

fcm_labels = fcm.predict(np.array(g))
tt = fcm.centers[fcm_labels]
t = tt[:,0:3]

img_res = t.reshape(nrow, ncol, nchl)
```

3.4. Thuật toán Expectation Maximization

```
em = GaussianMixture(n_components=k, covariance_type='full',
                      max_iter=20,
                      random_state=0).fit(g)

centers = np.empty(shape=(em.n_components, g.shape[1]))

for i in range(em.n_components):

    density =
        scipy.stats.multivariate_normal(cov=em.covariances_
        [i],

                                         mean=em.means_[i]).logpdf(g)

    centers[i, :] = g[np.argmax(density)]

tt = centers[em.predict(g)]

t = tt[:,0:3]

img_res = t.reshape(nrow, ncol, nchl)
```

4. Kết quả thực nghiệm

Nhóm đã thực nghiệm một số trường hợp và có kết quả tại [Google Colab](#). Kết quả của các trường hợp nhóm đã thực nghiệm:

4.1. *Cố định số lượng cụm, cố định đặc trưng RGB và thay đổi thuật toán*

Thuật toán	Silhouette Score	Thời gian tính toán (s)
K-Means	0.889	63
FCM	0.841	62
Meanshift	0.837	68
EM	0.804	65

Bảng 1: Thực nghiệm với số cụm là 10, đặc trưng RGB và thay đổi thuật toán trên Hình 7.1

4.2. *Cố định số lượng cụm, cố định đặc trưng RGBXY và thay đổi thuật toán*

Thuật toán	Silhouette Score	Thời gian tính toán (s)
K-Means	0.36	62
FCM	0.3	62
Meanshift	0.36	62
EM	0.06	62

Bảng 2: Thực nghiệm với số cụm là 7, đặc trưng RGBXY và thay đổi thuật toán trên Hình 7.1

4.3. *Cố định số lượng cụm, cố định đặc trưng RGBX'Y' và thay đổi thuật toán*

Thuật toán	Silhouette Score	Thời gian tính toán (s)
K-Means	0.59	71
FCM	0.43	68
Meanshift	0.53	77
EM	-0.03	66

Bảng 3: Thực nghiệm với số cụm là 8, đặc trưng RGBX'Y' và thay đổi thuật toán trên Hình 7.1

4.4. *Cố định cụm, cố định thuật toán và thay đổi đặc trưng*

4.4.1. *K-Means*

Đặc trưng	Silhouette Score	Thời gian tính toán (s)
RGB	0.51	41
RGBXY	0.38	41
RGBX'Y'	0.45	41

Bảng 4: Thực nghiệm với số cụm là 6, thuật toán K-Means và thay đổi đặc trưng trên Hình 7.4

4.4.2. *FCM*

Đặc trưng	Silhouette Score	Thời gian tính toán (s)
RGB	0.37	108
RGBXY	0.3	111
RGBX'Y'	0.37	110

Bảng 5: Thực nghiệm với số cụm là 8, thuật toán FCM và thay đổi đặc trưng trên Hình 7.2

4.4.3. Meanshift

Đặc trưng	Silhouette Score	Thời gian tính toán (s)
RGB	0.69	101
RGBXY	0.37	101
RGBX'Y'	0.47	101

Bảng 6: Thực nghiệm với số cụm là 3, thuật toán Meanshift và thay đổi đặc trưng trên Hình 7.5

4.4.4. EM

Đặc trưng	Silhouette Score	Thời gian tính toán (s)
RGB	0.47	10
RGBXY	0.25	11
RGBX'Y'	0.26	10

Bảng 7: Thực nghiệm với số cụm là 8, thuật toán EM và thay đổi đặc trưng trên Hình 7.6

4.5. Cố định thuật toán, cố định đặc trưng và thay đổi số cụm

4.5.1. FCM

Số lượng cụm	Silhouette Score	Thời gian tính toán (s)
2	0.45	118
8	0.36	114
15	0.32	113
20	0.31	113

Bảng 8: Thực nghiệm đặc trưng RGBX'Y', thuật toán FCM và thay đổi số cụm trên Hình 7.2

4.5.2. EM

Số lượng cụm	Silhouette Score	Thời gian tính toán (s)
2	0.43	122
8	0.23	126
15	0.17	114
20	0.18	114

Bảng 9: Thực nghiệm đặc trưng RGBX'Y', thuật toán EM và thay đổi số cụm trên Hình 7.2

5. Nhận xét

Sau khi thực hiện các thử nghiệm:

- Cố định số lượng cụm, cố định loại đặc trưng và thay đổi thuật toán

- Cố định số lượng cụm, cố định thuật toán và thay đổi loại đặc trưng
- Cố định thuật toán, cố định đặc trưng và thay đổi số lượng cụm

Nhóm đã thu được kết quả và rút ra được một số nhận xét sau:

- Thời gian tính toán Silhouette Score khá lâu (60s - 120s)
- Với sự gia tăng giá trị của số lượng cụm, hình ảnh trở nên rõ ràng hơn vì thuật toán có thể phân loại nhiều lớp/cụm hơn. Tuy nhiên không phải số lượng cụm càng cao thì Silhouette Score càng cao.
- Không thể dùng Silhouette Score để đánh giá kết quả phân đoạn ảnh này tốt hơn kết quả phân đoạn ảnh kia khi số cụm khác nhau.
- Với việc cố định số cụm, ta có thể đánh giá chất lượng phân cụm trên một ảnh khi thay đổi thuật toán, đặc trưng đầu vào.

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Qua quá trình nghiên cứu các thuật toán phân cụm cho bài toán phân đoạn ảnh, chúng tôi rút ra được những kết luận sau:

- Không thể chọn một thuật toán phân cụm, bộ siêu tham số hay loại đặc trưng cụ thể nào để có thể áp dụng cho tất cả các trường hợp trong bài toán phân đoạn ảnh. Vì vậy, với mỗi trường hợp ảnh khác nhau, sẽ có những thuật toán phân cụm, bộ siêu tham số và loại đặc trưng phù hợp để giải quyết được bài toán đó.
- Cũng như kết luận trên thì với mỗi yêu cầu, kỹ thuật được người dùng sử dụng và bài toán được đưa ra sẽ cho độ hiệu quả khác nhau.
- Người dùng cần phải có kiến thức nhất định về các thuật toán và tính chất của ảnh đầu vào để có thể hình dung chính xác nhất về yêu cầu của bài toán phân đoạn nhằm lựa chọn kỹ thuật phân đoạn cho phù hợp.

2. Hướng phát triển

Qua bài nghiên cứu này, chúng tôi sẽ cố gắng thực hiện những việc sau để có thể cải tiến các phương pháp phân đoạn ảnh:

- Tìm hiểu thêm các phương pháp lựa chọn tham số phù hợp, tối ưu hơn cho từng thuật toán trong phần đã nghiên cứu.
- Nghiên cứu kết hợp các kỹ thuật hiện đại trong việc cải tiến thêm kỹ thuật đã nghiên cứu. Ví dụ như kết hợp mạng học sâu CNN để tăng độ hiệu quả cho bài toán phân đoạn .
- Nghiên cứu thêm các thuật toán có thể kết hợp cùng với các kỹ thuật trong phần nghiên cứu này để tìm ra thêm điểm mạnh cũng như mở rộng, khai thác thêm ứng dụng của chúng.
- Nghiên cứu, tìm hiểu thêm các phương pháp đánh giá kết quả bài toán phân đoạn ảnh được chính xác hơn.
- Ứng dụng kết quả của phân đoạn ảnh đã nghiên cứu được vào một bài toán thực tế cụ thể như ứng dụng vào bài toán hình ảnh y tế, nhận dạng đối tượng,... để

có thể đánh giá một cách khách quan hơn về hiệu quả của các phương pháp phân đoạn ảnh đã nghiên cứu.

TÀI LIỆU THAM KHẢO

- [1] <https://tailieu.vn/doc/tieu-luan-nghien-cuu-ung-dung-phan-vung-anh-trong-xu-ly-anh-1614079.html>
- [2] <https://text.123docz.net/document/2584448-tim-hieu-cac-phuong-phap-phan-doan-anh.htm>
- [3] <https://tieuluan.info/phn-on-nh.html>
- [4] <https://ichi.pro/vi/moi-thu-ban-can-biet-ve-phan-doan-hinh-anh-218272983555575>
- [5] <https://machinelearningcoban.com/2017/01/01/kmeans/#-thao-luan>
- [6] <https://gastack.vn/programming/4831813/image-segmentation-using-mean-shift-explained>
- [7] <https://vncoder.vn/bai-hoc/unsupervised-learning-clustering-phan-1-404>
- [8] <https://phamdinhhkhanh.github.io/2020/06/10/ImageSegmentation.html>
- [9] <https://www.youtube.com/watch?v=VhYt7nxOKKs>
- [10] https://www.youtube.com/watch?v=zr50h_91gOw
- [11] https://www.youtube.com/watch?v=Zz1cmz_DOMU
- [12] [https://datajobs.com/data-science-repo/Expectation-Maximization-Primer-\[Do-and-Batzoglou\].pdf](https://datajobs.com/data-science-repo/Expectation-Maximization-Primer-[Do-and-Batzoglou].pdf)
- [13] <https://cupdf.com/document/image-segmentation-56983813997c7.html>
- [14] <https://ongxuanhong.wordpress.com/2015/09/21/expectation-maximization-em-hoi-gi-dap-nay/>
- [15] https://www.youtube.com/watch?v=5TPldC_dC0s
- [16] [https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering))
- [17] <https://towardsdatascience.com/a-comparison-between-k-means-clustering-and-expectation-maximization-estimation-for-clustering-8c75a1193eb7>
- [18] <https://huytranvan2010.github.io/Meanshift-Object-Tracking/>
- [19] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

[20] <https://bigdatauni.com/tin-tuc/cac-phuong-phap-danh-gia-trong-thuat-toan-clustering.html>

[21] <https://ichi.pro/vi/danh-gia-muc-do-tot-cua-viec-phan-cum-doi-voi-truong-hop-hoc-tap-khong-giam-sat-225313176016113>

[22] <https://bigdatauni.com/tin-tuc/cac-phuong-phap-danh-gia-trong-thuat-toan-clustering.html>

[23] <https://bigdatauni.com/tin-tuc/phuong-phap-clustering.html>