

ỨNG DỤNG THUẬT TOÁN PHÂN CỤM K-MEANS TRONG NÉN ẢNH *

Phạm Anh Tuấn, Đỗ Tâm Linh An

*Posts and Telecommunications Institute of Technology,
Km10, Đường Nguyễn Trãi, Q.Hà Đông, Hà Nội*

2020
July

Tóm Tắt

Được cho là sự kết hợp của văn bản, âm thanh hay hình ảnh đặt trong môi trường số - Đa Phương Tiện được biết đến như một đại lượng thông tin mà dữ liệu được biểu diễn dưới dạng "Kép". Việc tối ưu hoá không gian lưu trữ dữ liệu mà chất lượng không bị giảm nhiều là điều cần thiết. Bên cạnh việc cải tiến các hệ thống giao thức truyền thông, thì việc nén dữ liệu cũng quang trọng không kém. Tối ưu hoá dữ liệu cũng có rất nhiều hiệu quả trong khả năng truyền tải, xử lý thông tin giữa các mạng lưới thiết bị trong thời đại số hiện nay. Trong khuôn khổ báo cáo này chúng ta chỉ xét đến việc nén hình ảnh trong phân ngành xử lý hình ảnh, cụ thể bằng việc sử dụng học máy trong việc nâng cao hiệu quả nén ảnh.

*Bài tiểu luận giới thiệu về phương pháp phân cụm cơ bản áp dụng trong nén hình ảnh và sơ qua các phương pháp cải tiến thuật toán này.

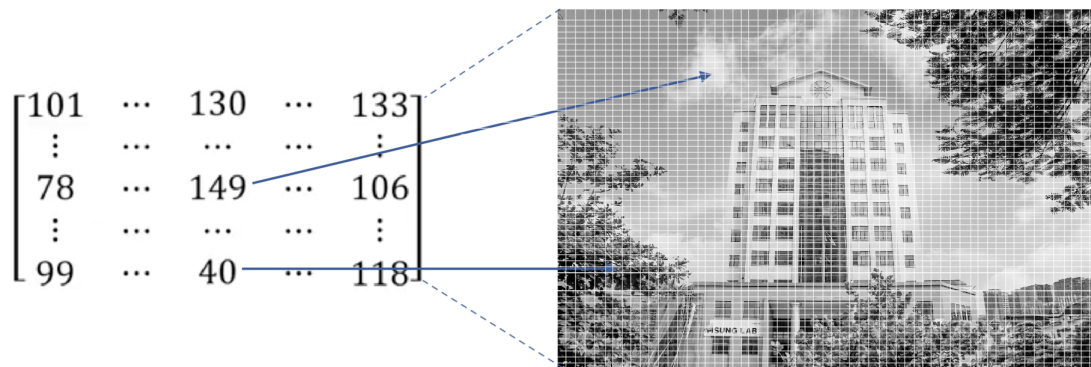
Mục lục

1	Giới thiệu	3
2	Đề xuất và phân tích thuật toán K-Means	6
2.1	Ý tưởng	6
2.2	Giải thuật	7
2.3	Hàm mất mát và bài toán tối ưu	7
2.4	Thuật toán tối ưu hàm mất mát	8
2.4.1	Cố định M, tâm của cụm	8
2.4.2	Cố định Y, Label Vector	9
3	Chương trình và tính hiệu quả (Chưa cập nhật)	9
4	Nhận định và cải tiến thuật toán	9
4.1	Nhận định	9
4.2	Cải tiến	10
4.2.1	Biến đổi wavelet	10
4.2.2	Phân cụm Kmeans ảnh cực tiểu	11
4.2.3	Phân cụm Kmeans theo ảnh gốc	11
4.2.4	Thử nghiệm và kết quả	11
5	Kết luận	12
6	Tài liệu tham khảo	13

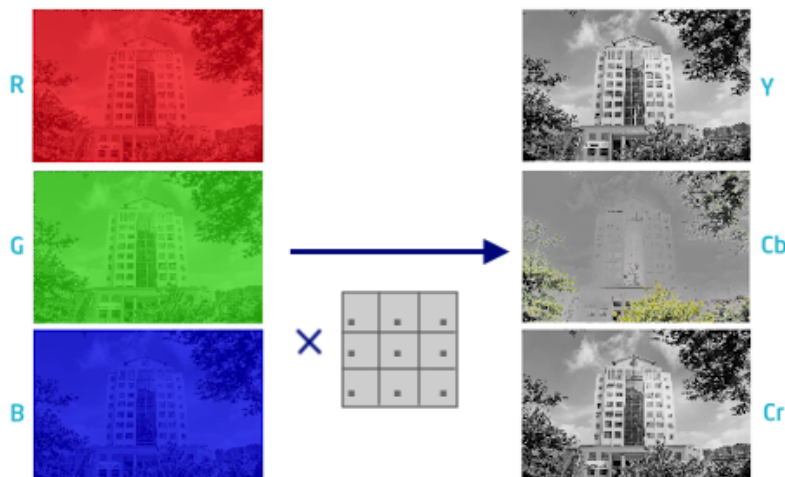
1 Giới thiệu

Việc xử lý ảnh nói chung hay nén ảnh nói riêng luôn được chú trọng quan tâm và cải thiện từ khi xuất hiện lần đầu cho đến nay. Trong khoảng thời gian đó đã có rất nhiều phương pháp nén ảnh tối ưu từ nén không tổn hao và nén tổn hao. Ở nén tổn hao thì các phương pháp đa dạng hơn có thể kể đến: Nén cơ bản (*Baseline Sequential Mode*), nén lũy tiến (*Progressive Mode*) và nén phân cấp (*Hierarchical Mode*). Các phương pháp này chỉ khác nhau trong phương thức ta mã hoá các hệ số **DCT**.

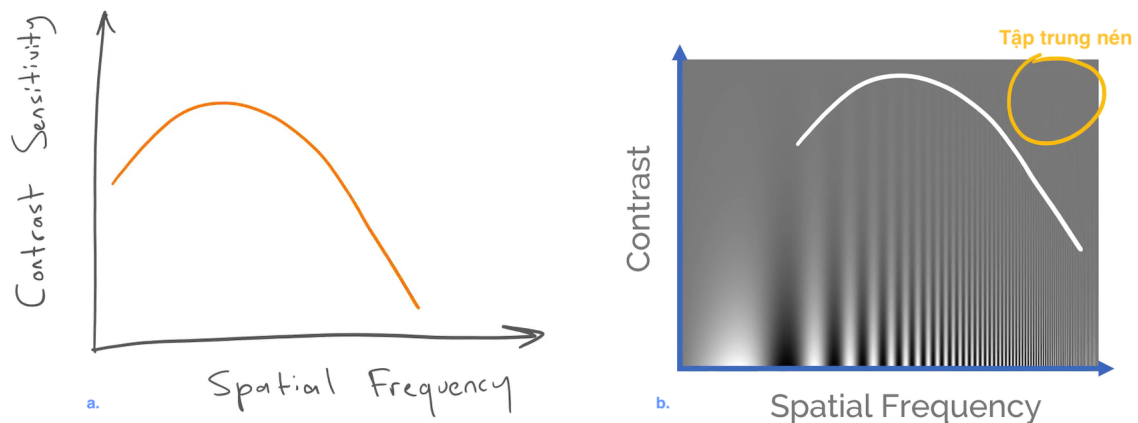
Sơ lược về cách hoạt động của thuật toán **DCT** (Biến đổi cosin rời rạc) là dựa vào các đặc tính của thị giác con người để loại bỏ bớt thông tin "Dư thừa" có trong một bức ảnh. Ý tưởng của phép biến đổi này bắt đầu bằng việc chia bức ảnh nhiều ô vuông nhỏ và biểu diễn nó dưới dạng ma trận. Trị số lớn ta biểu diễn cho cường độ Pixel lớn và ngược lại đối với trị số nhỏ trong ma trận.



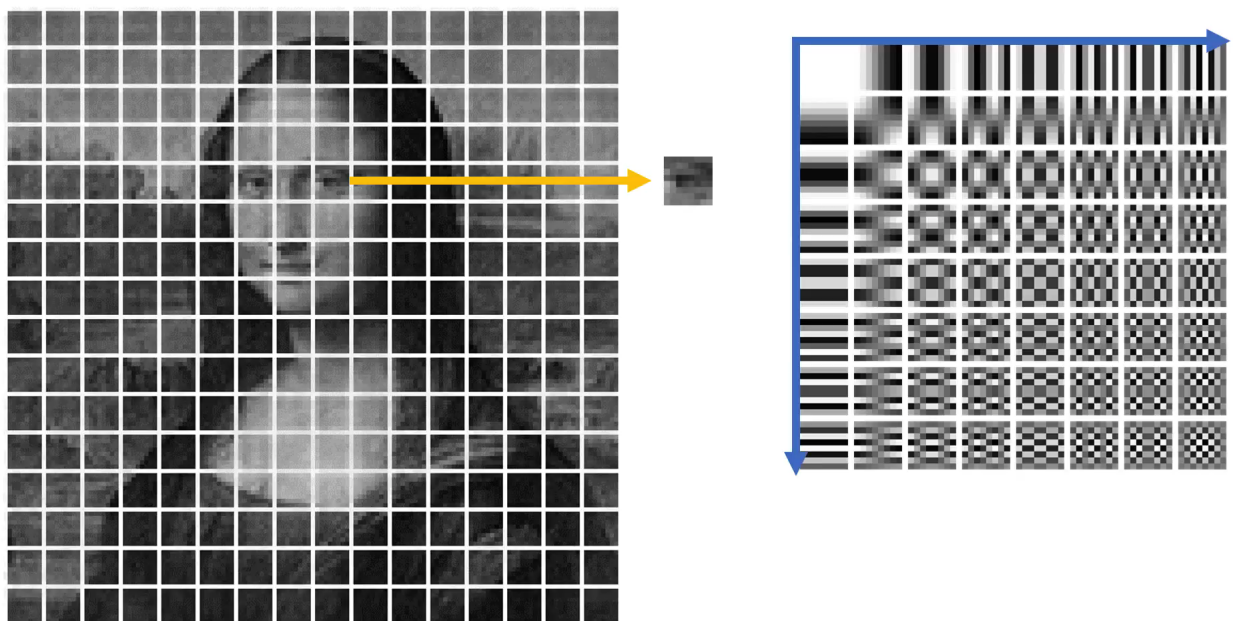
Hình ảnh kỹ thuật số ngày nay thường lưu dưới các kênh màu RGB. Tuy nhiên trong nén ảnh chúng ta lại sử dụng một kênh màu khác. Như đã đề cập ở trên, nén ảnh tổn hao chú trọng vào việc loại bỏ thông tin dư thừa mà ta không dễ nhận thấy trong một bức ảnh. Chính vì vậy hệ YCbCr được sinh ra để thay thế. Với kênh Y để biểu diễn cường độ ánh sáng và 2 kênh còn lại là Cb và Cr.



Tiếp sau khi chuyển đổi không gian màu của bức ảnh, ta sẽ dựa vào đặc điểm thị giác con người để điều chỉnh thích hợp các kênh màu để giảm lượng lưu trữ nhưng đảm bảo lượng thông tin trong bức ảnh. Cụ thể, các kênh màu như **Cb** và **Cr** sẽ được giảm xuống so với kênh **Y** do mắt chúng ta nhạy cảm với thay đổi ánh sáng so với màu sắc. Bên cạnh đó khả năng nhận biết thông tin có trong bức ảnh còn phụ thuộc vào tương quan độ tương phản và tần số, được biểu diễn qua biểu đồ dưới đây. Như hình **b)** bên dưới, phần thông tin hình ảnh với tần số cao và độ tương phản thấp hầu như không nhận thấy được phần không gian dưới đường kẻ trắng. Đồng nghĩa với việc những khu vực trên ngoài đường kẻ trắng sẽ được tập trung nén.



Vậy DCT đã làm gì? Đó là chia hình ảnh thành các khối 8×8 và lượng tử hoá chúng thành các đại diện cho miền tần số. Rõ ràng hơn, bằng việc so sánh từ khối vuông 8×8 với 64 mẫu tần số với sự tăng dần tần số từ trái sang phải và từ trên xuống dưới như hình.



Chúng ta sẽ sử dụng phương trình (1) cho việc biến đổi các lưới ô 8*8 sử dụng biểu đồ bước sóng như hình phía trên. Quá trình tương tự biến đổi Fourier hai chiều, sự khác biệt là sử dụng hàm Cosin và hàm Gamma:

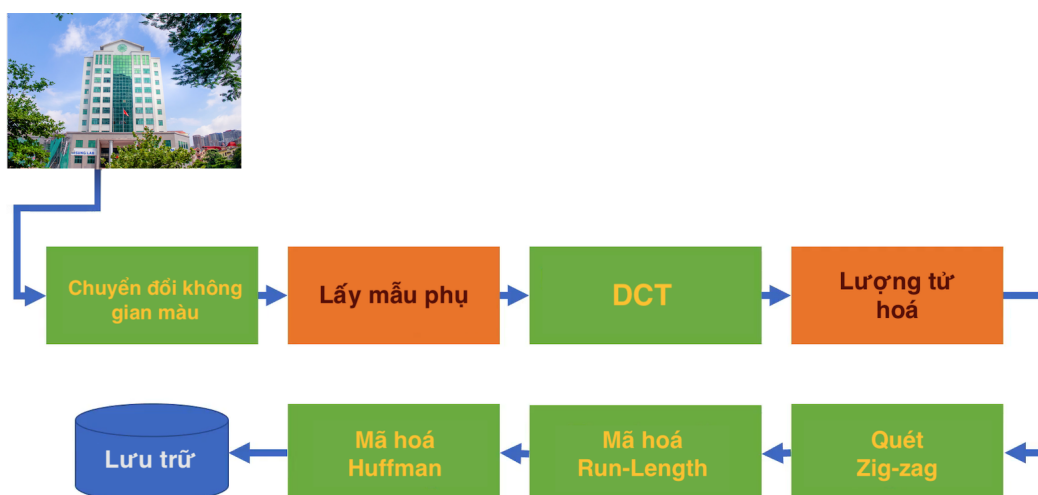
$$\mathbf{F}(\mathbf{u}, \mathbf{v}) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (\wedge(i) \wedge(j)) \left[\cos\left(\frac{u \cdot \pi}{2N}(2i+1)\right) \cos\left(\frac{v \cdot \pi}{2M}(2j+1)\right) \right]$$

Sau cùng, với ma trận biểu thị độ nhạy cường độ Pixel ta biểu diễn thành một khối tương tự mà đơn vị ma trận lại biểu diễn một thành phần tần số cụ thể. Phương pháp này là phương pháp biến đổi Cosin rời rạc

Discrete Cosine Transform



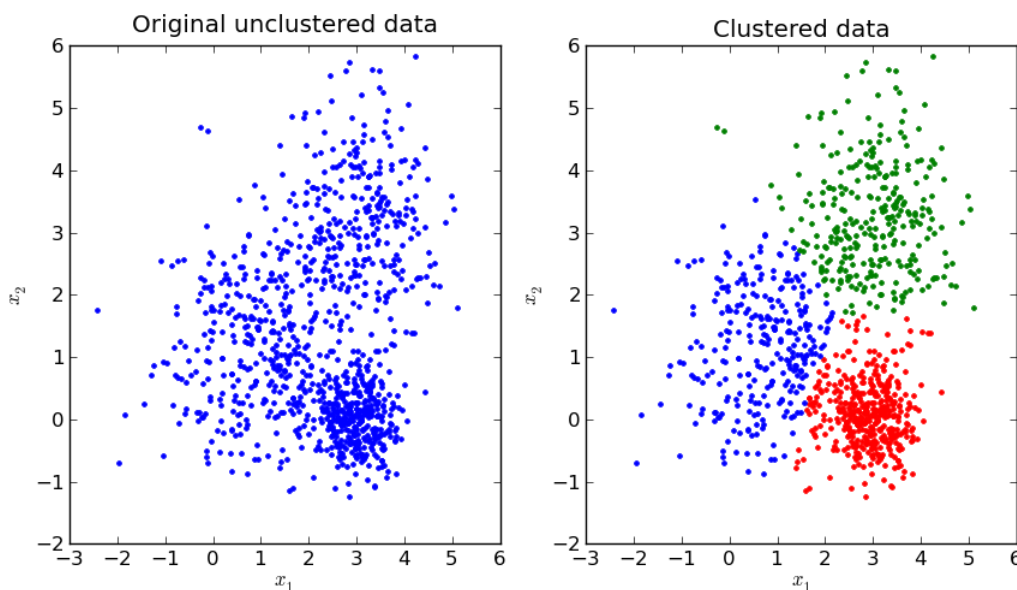
Bản thân DCT không nén hình ảnh mà cung cấp thông tin bản chất của quá trình nén - thứ thật sự diễn ra sau quá trình biến đổi Cosin rời rạc (DCT). Sơ qua toàn bộ quá trình để có một bức ảnh nén hoàn chỉnh sẽ được biểu diễn ở hình bên dưới:



2 Đề xuất và phân tích thuật toán K-Means

2.1 Ý tưởng

Quay trở lại với phương pháp nén ảnh bằng phân cụm K-Means. Phương pháp này có những ưu điểm vượt trội như khả năng hoạt động đơn giản của thuật toán (giữ nguyên kênh màu RGB mà không cần chuyển đổi kênh màu như kỹ thuật cũ, không cần công đoạn chuyển đổi tần số v.v ...), xử lý hiệu quả với bộ dữ liệu lớn,... Đây là một thuật toán không quá mới trong khoa học máy tính và đặc biệt là học máy. Ý tưởng của thuật toán này là việc phân chia một bộ dữ liệu thành các cụm khác nhau mà các thành viên trong cụm đó chung hay có những đặc điểm tương đồng. Việc phân cụm sẽ được diễn ra tự động với số lần lặp để xác định chính xác điểm tụ của mỗi cụm ¹. Trong nén ảnh, ta sẽ coi dữ liệu là các điểm trong không gian, các điểm dữ liệu này càng gần nhau thì càng giống .



Từ ý tưởng và nguyên lý làm việc của thuật toán, ta đưa ra một mô hình toán học bắt đầu bằng việc đặt một đầu vào dữ liệu $X \in R^{nd}$ trong đó N là số điểm dữ liệu và d là chiều cho từng mẫu dữ liệu đó. Thuật toán sẽ có nhiệm vụ chính là chia N dữ liệu đó thành K cụm (thỏa mãn $K \leq N$) mà các dữ liệu trong mỗi cụm là giống nhau. Đầu ra của thuật toán là xác định và xuất ra trung tâm (center) hoặc nhãn (label) của cụm dữ liệu đó.

¹Điểm tụ là điểm xác định trung tâm của cụm dữ liệu đó.

2.2 Giải thuật

Giả sử chúng ta có N điểm dữ liệu là $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ và thuật toán bắt đầu bằng việc khởi tạo k điểm bất kì làm điểm trung tâm của Cluster (cụm dữ liệu)

$$\mathbb{C}^{(0)} = \{m_1^{(0)}, m_2^{(0)}, \dots, m_k^{(0)}\} \quad (1)$$

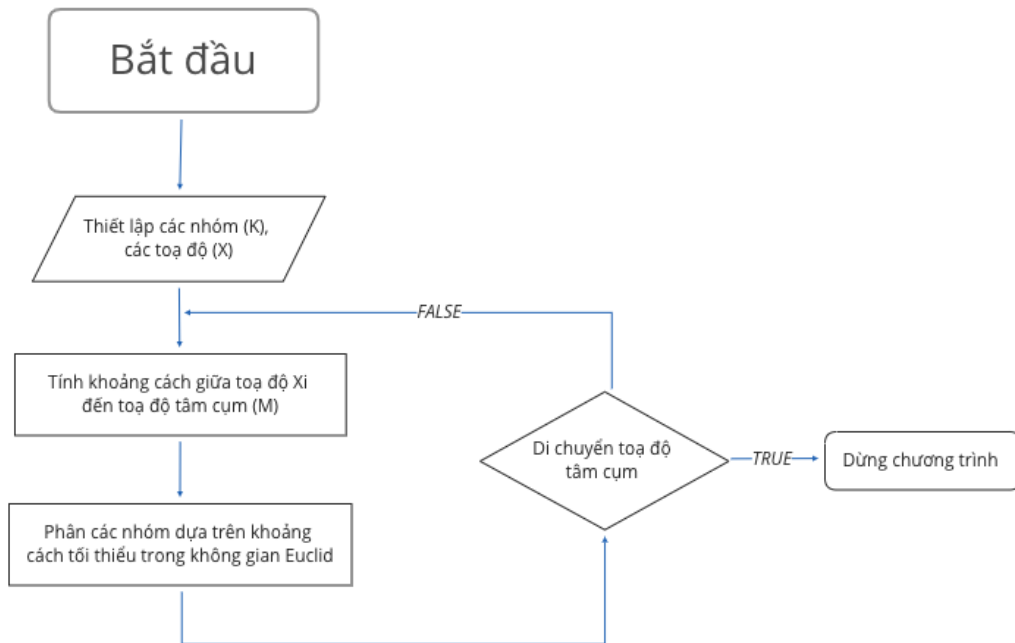
Giá trị sau khi khởi tạo sẽ tiến đến quá trình nhóm dữ liệu, ở bước này các điểm dữ liệu sẽ được nhóm đến các cụm dữ liệu có trung tâm gần nhất với nó :

$$\mathbb{S}_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2\} \quad , \forall j, 1 \leq j \leq k \quad (2)$$

Các tâm cụm sẽ được cập nhật lại bằng cách lấy trung bình cộng khoảng cách tới các điểm dữ liệu và sẽ lặp lại qua phương trình 2 cho đến khi các cụm của các nhóm không thay đổi thêm thì dừng giải thuật:

$$m_i^{(t+1)} = \frac{1}{|\mathbb{S}_i^{(t)}|} \sum_{x_j \in \mathbb{S}_i^{(t)}} x_j \quad (3)$$

Lưu đồ biểu diễn của thuật toán được biểu diễn như hình dưới



2.3 Hàm mất mát và bài toán tối ưu

Trước khi đi sâu vào bài toán, với mỗi điểm \mathbf{x}_i nếu được phân vào một cụm \mathbf{k} ta đặt $\mathbf{y}_i = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, \dots, \mathbf{y}_{iK}]$ là Label Vector của nó. Khi đó $\mathbf{y}_{jk} = 1$ và $\mathbf{y}_{ij} = 0, \forall j \neq k$ từ đó dẫn tới phương trình ràng buộc của \mathbf{y}_i như sau:

$$y_{ik} \in \{0, 1\}, \quad \sum_{k=1}^K y_{ik} = 1 \quad (4)$$

Khi đó, nếu ta coi m_k là tâm mỗi cụm và các điểm để xác định được phân vào một cụm (Cluster) được xác định bởi m_k . Khi đó sai số của một điểm dữ liệu x_i khi phân vào cụm sẽ là $(x_i - m_k)$. Như mục tiêu ban đầu của thuật toán là khiến trị tuyệt đối của sai số này nhỏ nhất:

$$\|\mathbf{x}_i - \mathbf{m}_k\|^2 \rightarrow Min \quad (5)$$

Theo phương trình (4) trước đó, biểu thức (5) được viết lại thành:

$$y_{ik} \|\mathbf{x}_i - \mathbf{m}_k\|^2 = \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2 \quad (6)$$

Với $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \dots; \mathbf{y}_N]$, $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K]$ lần lượt là các ma trận Label Vector và tâm cụm. Bài toán ta cần tối ưu quay trở về dạng sau, (*argmin* là điều kiện giá trị của biến số để hàm số đang xét đạt giá trị nhỏ nhất):

$$\mathbf{Y}, \mathbf{M} = \arg \min_{\mathbf{Y}, \mathbf{M}} \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2 \quad (7)$$

2.4 Thuật toán tối ưu hàm mất mát

Ở phương trình (7) rất khó để tìm điểm tối ưu hay chính xác là việc tìm nghiệm tối ưu toàn cục (global optimal point) do có thêm các điều kiện ràng buộc. Bài toán trước mắt ta sẽ tập trung vào phương pháp tìm điểm gần đúng hay điểm cực tiểu. \mathbf{Y} và \mathbf{M} là 2 biến có thể thay đổi được và được tận dụng để xử lý bài toán này

2.4.1 Cố định M, tâm của cụm

Giả sử các Centers đã được biết hoặc cho trước, khi đó cần tìm các Label Vector để hàm mất mát đạt giá trị cực tiểu. Khi các Centers là cố định, phương trình (7) được rút gọn lại thành:

$$\mathbf{y}_i = \arg \min_{\mathbf{y}_i} \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2 \quad (8)$$

Ta biết chỉ một phần tử của Label Vector y_i bằng 1 nên phương trình (8) có thể tiếp tục viết lại dưới dạng

$$j = \arg \min_j \|\mathbf{x}_i - \mathbf{m}_j\|_2^2 \quad (9)$$

Từ (9) ta nhận thấy $\|\mathbf{x}_i - \mathbf{m}_j\|^2$ chính là bình phương khoảng cách từ \mathbf{x}_i tới Center \mathbf{m}_j . Rút ra kết luận: **Mỗi điểm \mathbf{x}_i thuộc về Cụm có Tâm gần đó nhất.**

2.4.2 Cố định Y, Label Vector

Giả sử Label Vector được cố định, phương trình (7) được rút gọn thành

$$\mathbf{m}_j = \arg \min_{\mathbf{m}_j} \sum_{i=1}^N y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2 \quad (10)$$

Nhận thấy rằng phương trình (10) là một hàm lồi và khả dĩ trên đoạn $i \in [1, N]$ nghiệm của phương trình có được bằng phương pháp giải đạo hàm bằng 0

Đặt $l(\mathbf{m}_j)$ thay cho $\arg \min$, ta có đạo hàm cho phương trình (10) như sau:

$$\frac{\partial l(\mathbf{m}_j)}{\partial \mathbf{m}_j} = 2 \sum_{i=1}^N y_{ij} (\mathbf{m}_j - \mathbf{x}_i) \quad (11)$$

Giải phương trình với đạo hàm bằng (0) ta được:

$$\mathbf{m}_j \sum_{i=1}^N y_{ij} = \sum_{i=1}^N y_{ij} \mathbf{x}_i \Rightarrow \mathbf{m}_j = \frac{\sum_{i=1}^N y_{ij} \mathbf{x}_i}{\sum_{i=1}^N y_{ij}} \quad (12)$$

Điểm đặc biệt của phương trình (12) đó là mẫu số chính là phép đếm số lượng các điểm dữ liệu có trong **Cluster** j còn tử số là tổng các điểm dữ liệu có trong **Cluster** j . Rút ra kết luận một cách tổng quan rằng: \mathbf{m}_j chính là trung bình cộng các điểm trong **Cluster** j

3 Chương trình và tính hiệu quả (Chưa cập nhật)

Bằng việc đưa thuật toán về K-Means vào trong ứng dụng viết bởi C++, đã đưa ra dữ liệu về tính hiệu quả cũng như khả năng của thuật toán này. Do mỗi bức ảnh chứa 3 không gian màu, đã thu được kết quả sau

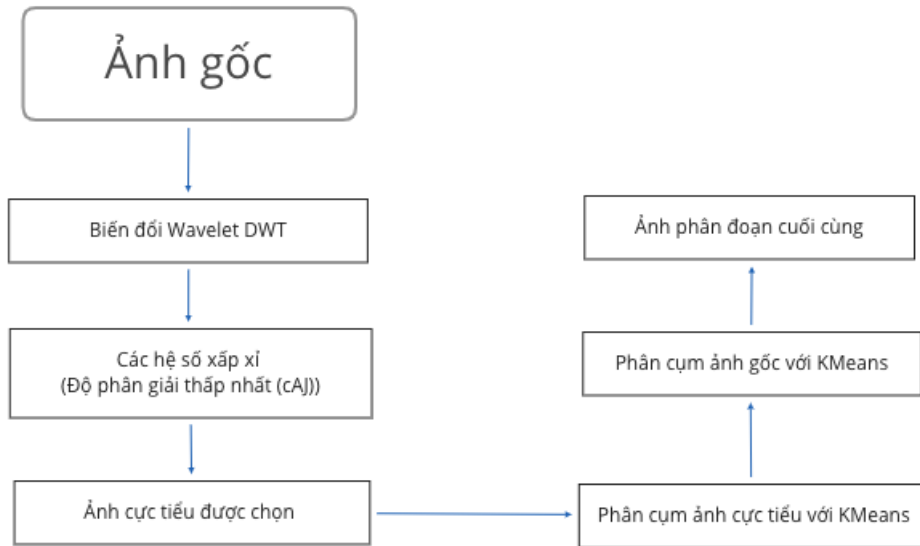
4 Nhận định và cải tiến thuật toán

4.1 Nhận định

Từ kết quả của chương trình, cũng như qua biểu diễn toán học của thuật toán, dễ dàng nhận thấy những đặc điểm nổi bật của thuật toán này cũng chính là yếu điểm của nó. Cụ thể các ảnh khi được xử lý rất đơn giản và nhanh chóng được thực hiện, tuy nhiên tốc độ xử lý của thuật toán sẽ trở nên chậm chạp và sẽ không phù hợp với một tập dữ liệu cũng như cụm phức tạp. Độ hội tụ của tâm cụm sẽ chậm đi khi xử lý một bức ảnh lớn (nhất là những bức ảnh thiên văn học). Ngoài ra, do ở bước đầu trong việc triển khai thuật toán, tâm cụm được khởi tạo ngẫu nhiên, chính vì vậy nếu các cụm được tạo "không tốt" như quá gần nhau ... thì việc xác định các tâm cụm hội tụ về đúng điểm là rất lâu là đôi khi có thể gây lỗi cho chương trình. Và các nhà khoa học đang không ngừng cải thiện làm sao cho tâm cụm được khởi tạo tốt nhất.

4.2 Cải tiến

Qua phần nhận định trên, chúng ta đã thấy việc xác định các tâm cụm (mặc dù là ngẫu nhiên) ảnh hưởng rất nhiều đến hiệu quả của thuật toán. Chính vì vậy thuật toán cải tiến dưới đây - wiKmeans áp dụng phép biến đổi Wavelet sẽ tập trung vào cải thiện phân vùng ảnh. Lưu đồ thuật toán của wiKmeans được biểu diễn như hình dưới:

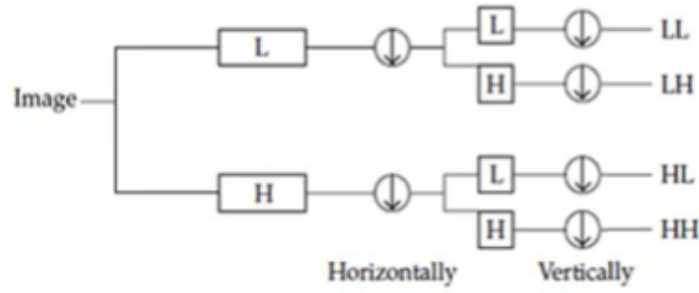


4.2.1 Biến đổi wavelet

Ở bước này, phép biến đổi wavelet được dùng giảm kích thước ảnh. Sau khi thực hiện phép biến ta thu được hệ số *wavelet* là hàm co giãn và vị trí của sóng nhỏ. Biến đổi sóng nhỏ có thể được biểu diễn tương tự như biến đổi *Fourier*:

$$\mathbf{F}(\mathbf{w}) = \int_{-\infty}^{\infty} f(t) e^{j\mathbf{w}t} dt$$
$$\mathbf{C}(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} s(t) \omega(\text{scale}, \text{position}, t) dt$$

Tập hệ số Wavelet có thể thu được nhờ phép biến đổi sóng nhỏ rời rạc (Discrete Wavelet Transform - DWT). Với phần lớn ảnh số thì nội dung tần số thấp là quan trọng nhất, giữ được hầu như các đặc tính của ảnh đầu vào của phép biến đầu với kích thước giảm bốn lần. Sau khi áp dụng bộ lọc thông thấp theo hai hướng (LL) ta thu được ảnh xấp xỉ (cA1) của ảnh gốc. Nếu áp dụng bộ lọc thông thấp cho chiều ngang và bộ lọc thông cao cho chiều dọc ảnh (LH) ta có tập hệ số ngang (cH1) của ảnh gốc. Tương tự ta có tập hệ số dọc (cV1) và tập hệ số chéo (cD1). Lặp tiến trình trên bằng con (LL) để sinh ra các hệ số ở mức 2 tiếp theo. Hình 2 mô tả DWT ảnh theo thuật toán hình kim tự tháp của Mallat [4].



Vậy ảnh gốc S được biểu diễn trên cơ sở các hệ số biến đổi sóng con của nó như sau:

$$\mathbf{S} = cA_1 + \{cH_1 + cV_1 + cD_1\}$$

Thực hiện lặp tiến trình cho đến khi mức độ chi tiết là mẫu hay pixel. Tại mức J , ảnh gốc được biểu diễn bởi:

$$\mathbf{S} = cA_J + \{cH_i + cV_i + cD_i\}_{i=1}^J$$

Các hệ số xấp xỉ được tính toán như sau:

$$cA_{j-1} = cA_J + \{cH_j + cV_j + cD_j\}$$

Mỗi lần thực hiện phân rã wavelet, kích thước của ảnh xấp xỉ cA_j giảm đi bốn lần so với lần thực hiện trước đó (mỗi chiều giảm xuống một nửa). Như vậy, giả sử chúng ta phân rã 3 mức cho ảnh đầu vào, ta thu được ảnh xấp xỉ có kích thước giảm xuống 64 lần.

4.2.2 Phân cụm Kmeans ảnh cực tiểu

Tiến hành phân cụm ảnh xấp xỉ cực tiểu lựa chọn với thuật toán KMeans. Sau khi đã phân cụm tất cả các ô, ta được tập tâm cụm như sau:

$$\mathbf{V}_{Init} = \{\mathbf{v}_k; 1 \leq k \leq c\}$$

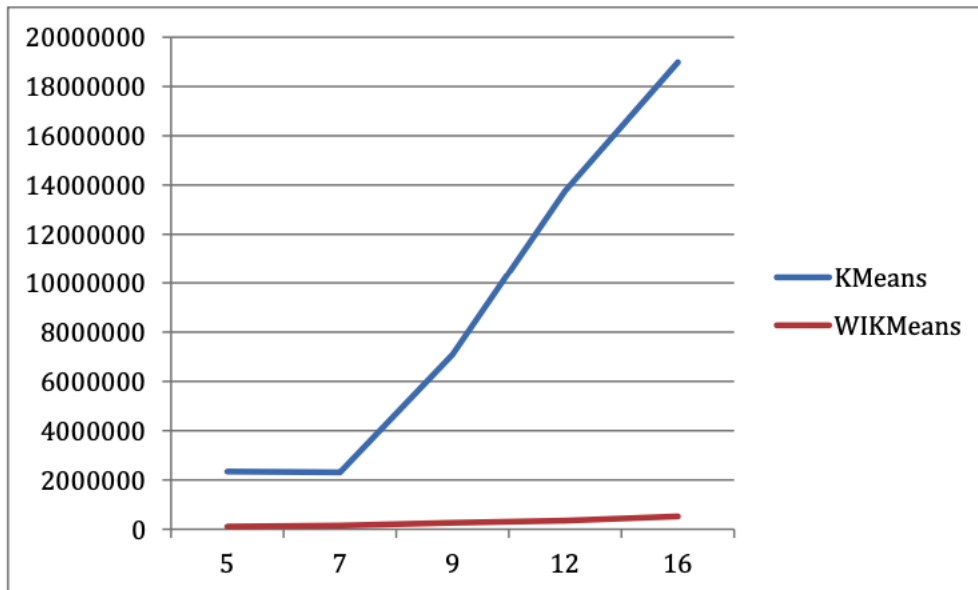
4.2.3 Phân cụm Kmeans theo ảnh gốc

Thực hiện phân việc phân cụm ảnh gốc với thuật toán KMeans với tập tâm cụm khởi tạo \mathbf{V}_{Init} thu được trong B2.

4.2.4 Thử nghiệm và kết quả

Bằng việc áp dụng thuật toán đề xuất wiKMeans và so sánh kết quả với thuật toán Kmeans trước đó. Bảng sau đây cho thấy thời gian phân cụm của một bức ảnh trong hai thuật toán trong trường hợp số cụm bằng 5

Thuật toán	5	7	9	12	16
KMeans	2339797	2309922	7092406	13765729	18985531
WIKMeans	39656	107622	134891	247828	306203



Bảng sau đây đưa ra số liệu về khoảng cách các tâm cụm của thuật toán wiKMeans và KMeans trong trường hợp 5 cụm

Khoảng cách các tâm sinh ra từ wiKmeans

	1	2	3	4	5
1	0	67.62	112.240	178.03	239.33
2		0	47.15	112.83	171.41
3			0	69.08	128.22
4				0	70.87
5					0

Khoảng cách các tâm sinh ra từ Kmeans

	1	2	3	4	5
1	0	67.05	111.557	179.38	240.52
2		0	45.04	113.39	172.52
3			0	68.37	127.47
4				0	68.18
5					0

5 Kết luận

Kmeans là một thuật toán đơn giản với nhiều ưu điểm cải tiến so với các thuật toán nén ảnh thuần túy trước đó. Mặc dù vậy các nhược điểm của thuật toán như khởi tạo tâm cụm hay xử lý điểm tụ là điều rõ ràng. Đã có rất nhiều thuật toán dựa trên Kmeans

nhưng được cải thiện về các nhược điểm có thể kể Đến Kmeans++ ,K-SVD.. Tuy nhiên trong khuôn khổ bài báo cáo ta chỉ xét qua về ý tưởng ban đầu về thuật toán Kmeans và qua đó điểm qua việc sử dụng biến đổi wavelet để cải thiện thuật toán.

6 Tài liệu tham khảo

- [1] Joaquin Prez Ortega, Ma. Del, Roco Boone Rojas, and Mara J. Somodevilla "Research issues on k-means algorithm: An experimental trial using matlab".
- [2] Arthur,D.,Vassilvitskii,S.,2016.k-Means++:TheAdvantagesofCareful Seeding.Technical Report, Stanford.
- [3] Kanungo, Tapas; Mount, David M.; Netanyahu, Nathan S.; Piatko, Christine D.; Silverman, Ruth; Wu, Angela Y. (2002). "An efficient k-means clustering algorithm: Analysis and implementation"