

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

————— * —————

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

**ỨNG DỤNG HỌC MÁY ĐỂ PHÂN VÙNG ẢNH VÀ PHÁT
HIỆN BẤT THƯỜNG TRONG GIÁM SÁT RỪNG**

Sinh viên thực hiện: **Trần Văn Thành**

Lớp CNTT2.01 – K58

Giáo viên hướng dẫn 1: **PGS. Phạm Văn Hải**

Giáo viên hướng dẫn 2: **PGS. Đỗ Phan Thuận**

HÀ NỘI 1 - 2019

Phiếu giao nhiệm vụ đồ án tốt nghiệp

1. Thông tin về sinh viên

Họ và tên sinh viên: Trần Văn Thành

Điện thoại liên lạc: 0978779701

Email: thanhtranvan.hust@gmail.com

Lớp: CNTT2.01 - K58

Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại: Viện CNTT&TT - Đại học Bách Khoa Hà Nội

Từ ngày 31/ 08/ 2018 đến ngày 28/ 12/ 2018

2. Mục đích nội dung của ĐATN

Phần 1: Thuật toán K-means clustering và ứng dụng cho bài toán phân vùng ảnh.

Phần 2: Phát hiện đám cháy trong ảnh và video thời gian thực.

3. Các nhiệm vụ cụ thể của ĐATN

1. Cải tiến thuật toán phân cụm K-means, áp dụng phương pháp cải tiến BK-means cho bài toán phân vùng ảnh màu. Đánh giá và so sánh kết quả của phương pháp cải tiến BK-means với thuật toán K-means cơ bản và kết quả trên các công trình nghiên cứu khác. Ứng dụng trong quản lý tài nguyên và giám sát rừng.
2. Xử lý ảnh phát hiện đám cháy trong ảnh và video thời gian thực dựa trên không gian màu YCbCr.

4. Lời cam đoan của sinh viên

Tôi – Trần Văn Thành - cam kết các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 28 tháng 12 năm 2018

Trần Văn Thành

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành và cho phép bảo vệ

.....
.....

Hà Nội, ngày 28 tháng 12 năm 2018

PSG. Phạm Văn Hải

Lời cảm ơn

Đi qua những năm tháng Bách Khoa, tôi mới biết tuổi trẻ đáng trân trọng như thế nào. Trân trọng, không hẳn là vì có những khó khăn, thách thức tưởng chừng như gục ngã, không hẳn là vì ta biết mình trưởng thành đến đâu mà đơn giản là vì ta đã làm tất cả điều đó cùng ai.

Cảm ơn Bách Khoa! 5 năm, có lẽ chẳng đáng gì so với cuộc đời nhưng có thể đó là tất cả của tuổi thanh xuân. Không muốn biết Bách Khoa cho mình bao nhiêu, lấy đi những gì, chỉ biết rằng tuổi trẻ có Bách Khoa và chắc chắn sẽ không bao giờ quên điều đó.

Em xin chân thành cảm ơn thầy Phạm Văn Hải giảng viên bộ môn HTTT cùng thầy Đỗ Phan Thuận giảng viên bộ môn KHMT đã hướng dẫn em hoàn thành đồ án này. Thầy đã cung cấp kiến thức, hướng dẫn tận tình, chỉ dẫn cho em từ những bước đầu tiếp cận và trong suốt quá trình thực hiện đề tài này. Qua sự hướng dẫn của hai thầy, em không chỉ học được nhiều kiến thức bổ ích mà còn học được những phương pháp nghiên cứu khoa học, thái độ và tinh thần nghiêm túc hết mình trong công việc.

Mặc dù em đã có những cố gắng hoàn thành đề tài, xong do thời gian có hạn vì vậy em cũng không thể tìm hiểu kỹ và đầy đủ hết các khía cạnh mà đề tài đề cập đến. Trong thời gian sắp tới em rất mong nhận được sự chỉ dẫn tận tình của thầy hơn nữa để giúp em có thể hoàn thiện được đề tài hơn và áp dụng đề tài vào thực tế để đem lại lợi ích thiết thực theo đúng định hướng và mục tiêu mà đề tài đã đưa ra.

Bách Khoa! Tháng 12 năm 2018

Xin chân thành cảm ơn!

Mục Lục

| | |
|---|----|
| Phiếu giao nhiệm vụ đồ án tốt nghiệp..... | 1 |
| Lời cảm ơn | 2 |
| Mục Lục | 3 |
| Danh mục bảng | 4 |
| Danh mục hình vẽ | 4 |
| Danh mục công thức | 5 |
| Danh mục các từ viết tắt và thuật ngữ..... | 6 |
| Chương 1. Giới thiệu đề tài..... | 7 |
| 1.1. Tổng quan | 7 |
| 1.2. Mục đích đồ án..... | 7 |
| 1.3. Phát biểu bài toán..... | 8 |
| 1.4. Dữ liệu đầu vào | 8 |
| 1.5. Dữ liệu đầu ra..... | 8 |
| 1.6. Công cụ lựa chọn | 8 |
| 1.7. Tóm tắt nội dung đồ án | 9 |
| Chương 2. Giới thiệu về học không giám sát và thuật toán phân cụm K-means | 10 |
| 2.1. Tổng quan | 10 |
| 2.2. Học không giám sát (Unsupervised Learning)..... | 11 |
| 2.3. Thuật toán phân cụm K-means | 13 |
| 2.3.1. Giới thiệu | 13 |
| 2.3.2. Phân tích toán học | 13 |
| 2.3.3. Thuật toán K-means cơ bản | 17 |
| Chương 3. Thị giác máy tính trong phân vùng ảnh..... | 18 |
| 3.1. Bài toán phân vùng ảnh màu..... | 18 |
| 3.1.1. Bài toán phân vùng ảnh màu (color image segmentation)..... | 18 |
| 3.1.2. Biểu diễn dữ liệu..... | 20 |
| 3.1.3. Tiền xử lý ảnh | 21 |
| 3.2. Xây dựng mô hình..... | 24 |
| 3.2.1. Phương pháp cải tiến BK-means..... | 24 |
| 3.2.2. Lựa chọn tham số..... | 26 |
| 3.3. Phương pháp đánh giá..... | 28 |
| 3.4. Kết quả thử nghiệm..... | 31 |
| Chương 4. Phát hiện đám cháy trong ảnh và video sử dụng không gian màu YCbCr..... | 40 |

| | |
|---|----|
| 4.1. Biểu diễn dữ liệu | 40 |
| 4.2. Xây dựng mô hình..... | 42 |
| 4.3. Kết quả thử nghiệm và đánh giá | 48 |
| 4.3.1. Kết quả | 48 |
| 4.3.2. Đánh giá | 49 |
| Chương 5. Kết luận và hướng phát triển..... | 53 |
| 5.1. Kết luận..... | 53 |
| 5.2. Hướng phát triển | 54 |
| Phụ Lục | 55 |
| Tài liệu tham khảo..... | 57 |

Danh mục bảng

| | |
|---|----|
| Bảng 3.1: Các công việc cải tiến K-means trước đây [12] | 19 |
| Bảng 3.2: Kết quả thử nghiệm phương pháp BK-means dựa trên ảnh Ground Truth..... | 31 |
| Bảng 3.3: Kết quả phân cụm 481×321 điểm ảnh với K=2 | 31 |
| Bảng 3.4: Kết quả phân cụm same cluster và different cluster với K=2 | 31 |
| Bảng 3.5: Tỷ lệ TPR, FPR, FNR, TNR cho bài toán phân cụm K=2..... | 31 |
| Bảng 3.6: Tỷ lệ RI Precision , Recall, F1 cho bài toán phân cụm K=2 | 31 |
| Bảng 3.7: Hình ảnh phân vùng bằng BK-means và ảnh Ground Truth..... | 32 |
| Bảng 3.8: So sánh kết quả phân vùng ảnh BK-means và K-means trên bộ dữ liệu ảnh vệ tinh..... | 33 |
| Bảng 3.9: Kết quả lựa chọn tham số K clusters..... | 37 |
| Bảng 3.10: So sánh kết quả phương pháp đề xuất BK-means và thuật toán K-means..... | 37 |
| Bảng 3.11: Kết quả phát hiện thay đổi từ ảnh vệ tinh | 38 |
| Bảng 4.1: Giá trị trung bình các kênh màu R, G, B của vùng lửa | 43 |
| Bảng 4.2: Giá trị trung bình các kênh Y, Cb, Cr của vùng lửa..... | 45 |
| Bảng 4.3: Quy tắc xác định điểm ảnh cháy [19][25]..... | 47 |
| Bảng 4.4: Kết quả phát hiện đám cháy trong ảnh..... | 48 |
| Bảng 4.5: Unnormalized confusion matrix | 49 |
| Bảng 4.6: Normalized confuion matrix | 49 |
| Bảng 4.7: Kết quả phân loại ảnh chứa đám cháy | 49 |
| Bảng 4.8: Tỷ lệ phân loại các lớp dựa trên normalized confusion matrix [1] | 50 |
| Bảng 4.9: Kết quả tỷ lệ TPR, FPR, TNR, FNR..... | 51 |
| Bảng 4.10: Tỷ lệ Precision, Recall và F1 cho bài toán phân loại ảnh đám cháy..... | 52 |

Danh mục hình vẽ

| | |
|--|----|
| Hình 2.1: Môi quan hệ giữa AI, Machine Learning và Deep Learning AI. | 10 |
| Hình 2.2: Bài toán phân cụm với 3 cluster [1]. | 13 |
| Hình 3.1: Ảnh màu trong không gian màu RGB..... | 18 |
| Hình 3.2: Biểu diễn ảnh màu RGB bằng ma trận 3 chiều..... | 20 |

| | |
|---|----|
| Hình 3.3: Đơn vị lượng tử trung bình – MQU (Mean Quantization Unit) [3]. | 22 |
| Hình 3.4: Cấu trúc cây trong SMQT [3]. | 22 |
| Hình 3.5: Kết quả tiền xử lý ảnh. | 23 |
| Hình 3.6: Sơ đồ phương pháp phân vùng ảnh. | 24 |
| Hình 3.7: Ảnh vệ tinh khu vực hồ Chiquita Mar 1998-2011. | 35 |
| Hình 3.8: Lựa chọn tham số K cho ảnh 2011.jpg. | 36 |
| Hình 3.9: Lựa chọn tham số K cho ảnh 1998.jpg. | 36 |
| Hình 3.10: Biểu đồ so sánh hiệu năng phương pháp BK-means và K-means. | 39 |
| Hình 4.1: Không gian màu RGB. | 41 |
| Hình 4.2: Không gian màu YCbCr [9]. | 41 |
| Hình 4.3: Mô hình phát hiện đám cháy trong ảnh [19]. | 42 |
| Hình 4.4: Ảnh gốc – (a), tách các kênh màu R – (b), G – (c), B – (d). | 42 |
| Hình 4.5: Ảnh gốc – (a), tách các kênh màu Y – (b), Cb – (c), Cr – (d). | 45 |
| Hình 4.6: Giá trị ngưỡng (điểm e) cho tỷ lệ phát hiện cháy > 70% và tỷ lệ báo cháy giả < 20% [19][25]. | 46 |
| Hình 4.7: Biểu đồ vùng cháy cho kênh Cb và Cr. | 47 |
| Hình 4.8: Phương pháp tính Precision và Recall cho bài toán phân loại nhị phân [1]. | 51 |
| Hình 0.1: Giao diện chương trình. | 56 |

Danh mục công thức

| | |
|--|----|
| Công thức 2.1: Label vector của điểm dữ liệu. | 14 |
| Công thức 2.2: Hàm khoảng cách [1]. | 14 |
| Công thức 2.3: Sai số cho toàn bộ dữ liệu [1]. | 14 |
| Công thức 2.4: Tối ưu hàm $\mathcal{L}(Y, M)$ [1]. | 14 |
| Công thức 2.5: Tính label vector cho từng điểm dữ liệu [1]. | 15 |
| Công thức 2.6: Công thức thu gọn của Công thức 2.5 . | 15 |
| Công thức 2.7: Tính label vector cho từng điểm dữ liệu. | 16 |
| Công thức 2.8: Đạo hàm của hàm khoảng cách [1]. | 16 |
| Công thức 2.9: Nghiệm của bài toán. | 16 |
| Công thức 3.1: Chuyển từ hệ màu RGB sang Lab. | 21 |
| Công thức 3.2: Tổng bình phương lỗi trong cụm [17]. | 26 |
| Công thức 3.3: Công thức tính Silhouette [17]. | 27 |
| Công thức 3.4: Sai số bình phương trung bình [6]. | 28 |
| Công thức 3.5: Tỷ lệ nhiễu tín hiệu [6]. | 28 |
| Công thức 3.6: Tỷ lệ nhiễu tín hiệu cực đại [6]. | 28 |
| Công thức 3.7: Chỉ số tương đồng về cấu trúc [6]. | 28 |
| Công thức 3.8: Sai số trung bình tuyệt đối [6]. | 29 |
| Công thức 3.9: Chỉ số cấu trúc về nội dung [6]. | 29 |
| Công thức 4.1: Trung bình trên kênh màu Red [19]. | 43 |
| Công thức 4.2: Trung bình trên kênh màu Green [19]. | 43 |
| Công thức 4.3: Trung bình trên kênh màu Blue [19]. | 43 |
| Công thức 4.4: Quy tắc 1 [19]. | 43 |
| Công thức 4.5: Quy tắc 2 [19]. | 43 |
| Công thức 4.6: Chuyển đổi không gian màu RGB sang YCbCr [19]. | 44 |
| Công thức 4.7: Trung bình kênh độ sáng Y [19]. | 44 |

| | |
|---|----|
| Công thức 4.8: Trung bình kênh màu Cb [19]..... | 44 |
| Công thức 4.9: Trung bình kênh màu Cr [19]. | 44 |
| Công thức 4.10: Quy tắc 3 [19][25]..... | 45 |
| Công thức 4.11: Quy tắc 4 [19][25]..... | 45 |
| Công thức 4.12: Quy tắc 5 [19][25]..... | 46 |
| Công thức 4.13: Quy tắc 6 [19][25]..... | 46 |
| Công thức 4.14: Quy tắc 7 [19][25]..... | 47 |
| Công thức 4.15: Tính độ chính xác Accuracy trên toàn bộ các lớp [17]. | 50 |
| Công thức 4.16: Tính Precision [1]. | 51 |
| Công thức 4.17: Tính Recal [1]. | 51 |
| Công thức 4.18: Tính $F1 - score$ [1]. | 52 |
| Công thức 4.19: Tính $F\beta$ [1]. | 52 |
| Công thức 5.1: Đề xuất phát hiện đám khói trong ảnh [24]. | 54 |

Danh mục các từ viết tắt và thuật ngữ

| | |
|------------------------------------|---|
| AI- Artificial Intelligence | Trí tuệ nhân tạo |
| ML- Machine Learning | Máy học, Học máy |
| DL- Deep Learning | Học sâu |
| BK-means | Một phương pháp cải tiến của K-means |
| SMQT- Successive Mean Quantization | Một phương pháp nâng cao chất lượng ảnh |
| Transform | |
| MQU -Mean Quantization Unit | Đơn vị lượng tử trung bình trong SMQT |

Chương 1. Giới thiệu đề tài

1.1. Tổng quan

Rừng được xem là “lá phổi” của trái đất, rừng có vai trò quan trọng trong việc duy trì cân bằng sinh thái và sự đa dạng sinh học trên trái đất. Bởi vậy bảo vệ rừng và nguồn tài nguyên rừng luôn trở thành một yêu cầu, nhiệm vụ không thể trì hoãn đối với các quốc gia trên thế giới, trong đó có Việt Nam. Đây là một thách thức vô cùng lớn đòi hỏi mỗi cá nhân, tổ chức thuộc các cấp trong một quốc gia và trên thế giới nhận thức được vai trò và nhiệm vụ của mình trong công tác phục hồi và bảo vệ rừng.

Xử lý ảnh viễn thám nói chung và phân vùng ảnh (hay phân cụm) viễn thám nói riêng là vấn đề được nghiên cứu từ rất lâu và hiện tại vẫn đang được quan tâm. Phân cụm là một quy trình dùng để trích chọn những nét chính của các đối tượng nền bởi việc định nghĩa các vùng tương ứng. Nhiệm vụ của chức năng phân vùng ảnh là từ ảnh ban đầu, tiến hành xử lý và phân chia thành các vùng, các cụm khác nhau.

Xử lý ảnh vệ tinh để tính toán được sự thay đổi của đô thị hóa, hay sự sói mòn đất, nạn chặt phá rừng, mức độ thiệt hại của thiên tai. Theo phương pháp truyền thống từ xưa dùng các bản vẽ, dựa vào bản đồ và các chuyến đi thực nghiệm, nhưng không phải nơi đâu trên trái đất con người cũng có thể đặt chân đến để kiểm tra đo đạc. Vì vậy ngày nay với các công cụ khoa học hiện đại, từ vệ tinh con người có thể thu được các hình ảnh từ bất cứ nơi đâu trên trái đất. Qua các ảnh vệ tinh chụp được một khu vực trên mặt đất, sau khi xử lý bài toán phân vùng ảnh thành các vùng như cây, đất, nước... Ta tính được tỷ lệ phần trăm diện tích của các khu vực trong tấm ảnh, tính được diện tích các vùng thay đổi qua chu kỳ thời gian, và so sánh kết quả từ đó kết luận được độ che phủ rừng, nạn chặt phá rừng, hay sự đô thị hóa... và trợ giúp đưa ra quyết định.

Phát hiện đám cháy trong ảnh và video thời gian thực. Hiện nay, các hệ thống báo cháy tự động trong các tòa nhà cao tầng, văn phòng,... là các cảm biến khói – thiết bị dùng để phát hiện khói – một thành phần khởi đầu của sự cháy. Nhược điểm của các cảm biến báo cháy là không thể sử dụng ở ngoài trời, các khu vực rộng lớn và có gió. Các cảm biến này thường gây ra các hiện tượng báo cháy giả và có thể làm tốn kém tiền của, dẫn đến hậu quả nghiêm trọng.

1.2. Mục đích đề án

Mục đích của đề tài là áp dụng hiệu quả các kỹ thuật học máy và thị giác máy tính cho bài toán phân vùng ảnh để hỗ trợ ra quyết định trong giám sát rừng, phát hiện những bất thường trong quản lý tài nguyên thiên nhiên. Đồng thời điều chỉnh các tham số thích hợp cho mô hình để đạt được kết quả có độ chính xác tốt nhất.

1.3. Phát biểu bài toán

Bài toán đặt ra: xây dựng mô hình học máy cho bài toán phân vùng ảnh (Image Segmentation) đề xuất phương pháp cải tiến BK-means dựa trên thuật toán phân cụm K-means cơ bản. Tách vùng cháy đồng thời phát hiện đám cháy trong ảnh và video thời gian thực lấy từ các camera báo cháy, kết hợp với các cảm biến khói để giảm thiểu hiện tượng báo cháy giả.

1.4. Dữ liệu đầu vào

Tập dữ liệu gồm 600 ảnh Natural Images và 1000 ảnh Ground Truth (ảnh phân đoạn bằng tay), tất cả các ảnh đều có kích thước 481x321 pixel:

https://people.eecs.berkeley.edu/~yang/software/lossy_segmentation/

Tập dữ liệu ảnh vệ tinh thu thập từ: <https://earthengine.google.com/timelapse/>

Tập dữ liệu kiểm thử gồm 5331 ảnh, trong đó gồm 2400 ảnh có đám cháy được gán nhãn *fire*, và 2931 ảnh không có đám cháy được gán nhãn *not_fire*. Tập dữ liệu này được thu tập từ google image.

1.5. Dữ liệu đầu ra

Đối với bài toán phân vùng ảnh vệ tinh: ảnh đầu ra sau khi được phân vùng sẽ được gán nhãn và tỷ lệ % diện tích các vùng ảnh, đánh giá và so sánh kết quả mô hình.

Với bài toán phát hiện đám cháy trong ảnh và video: đầu ra sẽ là 1 ảnh hoặc tập các ảnh là các frame được cắt từ video đầu vào sau đó được gán nhãn là cháy hoặc không cháy.

1.6. Công cụ lựa chọn

Phương pháp xử lý ảnh được sử dụng để giải quyết bài toán là phương pháp BK-means là một cải tiến dựa trên thuật toán phân cụm K-means cơ bản, được cài đặt và triển khai trên môi trường hệ điều hành Windows. Ngôn ngữ để triển khai bài toán là python v2.7 và matlab cùng thư viện sử dụng là OpenCV, scikit-learn và các thư viện toán học khác, các thư viện này sẽ giúp việc triển khai các hàm toán học và xử lý ảnh trong python một cách dễ dàng hơn. Công cụ sử dụng để lập trình là IDE Pycharm v2018.2 và Matlab v2015a.

1.7. Tóm tắt nội dung đề án

Đề án bao gồm phụ lục và 5 chương:

Chương 1: Giới thiệu đề tài: Chương này trình bày về nhiệm vụ cần thực hiện trong khuôn khổ đề án tốt nghiệp, trình bày định hướng giải quyết các vấn đề đặt ra. Giới thiệu tóm tắt về cơ sở lý thuyết và công cụ được lựa chọn để giải quyết các vấn đề đặt ra.

Chương 2: Giới thiệu về học không giám sát và thuật toán phân cụm K-means: Chương này trình bày những lý thuyết cơ bản về Unsupervised Learning, cung cấp một cái nhìn tổng quát về học máy, và những vấn đề về học không giám sát, thuật toán phân cụm K-means.

Chương 3: Thị giác máy tính trong phân vùng ảnh: Chương này chủ yếu trình bày việc áp dụng phương pháp cải tiến BK-means dựa trên thuật toán K-means cơ bản cho bài toán phân vùng ảnh.

Chương 4: Phát hiện đám cháy trong ảnh và video sử dụng không gian màu YCbCr: Chương này trình bày các quy tắc để phát hiện và phân loại ảnh có chứa đám cháy, áp dụng cho bài toán thực tế phát hiện đám cháy trong ảnh và video thời gian thực.

Chương 5: Kết luận và hướng phát triển: Ở chương này, chúng ta đưa ra kết luận tổng kết các kết quả ban đầu, khó khăn và thuận lợi của đề án. Các định hướng phát triển tiếp theo.

Phụ lục: Ở phần này sẽ nói về cách cài đặt, chạy chương trình giải quyết bài toán.

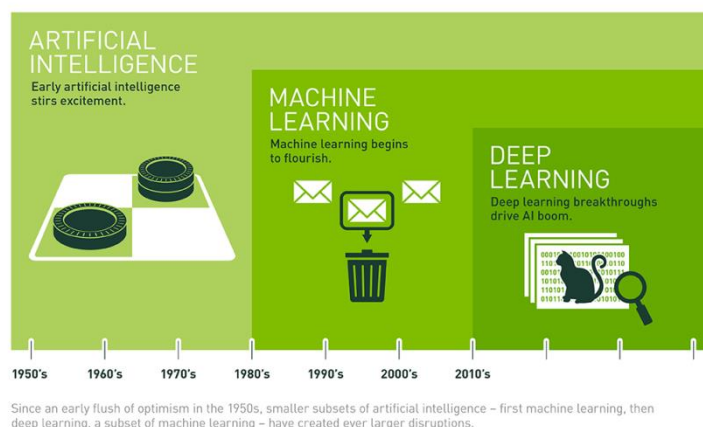
Chương 2. Giới thiệu về học không giám sát và thuật toán phân cụm K-means

2.1. Tổng quan

Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư 4.0 (1 - động cơ hơi nước, 2 - năng lượng điện và động cơ đốt trong, 3 – khoa học máy tính và internet). Trí Tuệ Nhân Tạo đang xuất hiện trong mọi lĩnh vực của đời sống mà có thể chúng ta không nhận ra. Xe tự lái của Google và Tesla, hệ thống tự găng thẻ (tag) khuôn mặt người dùng của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, ..., đó chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning [1].

Học máy (machine learning - ML) là một tập con của AI. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

Sự phát triển của các hệ thống tính toán cùng với đó, lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn như google hay facebook đã giúp Machine Learning tiến thêm một bước dài. Một lĩnh vực mới được ra đời gọi là học sâu (deep learning - DL). Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc [1].



Hình 2.1: Mối quan hệ giữa AI, Machine Learning và Deep Learning [AI](#).

Lĩnh vực machine learning và deep learning là cực kỳ rộng lớn và có nhiều nhánh nhỏ. Chúng ta luôn nhớ rằng điều đơn giản nhất là điều quan trọng nhất. Khi bắt tay vào giải quyết một bài toán machine learning hay bất cứ một bài toán nào, chúng ta nên bắt đầu từ những thuật toán đơn giản nhất. Ta không nên nghĩ rằng chỉ có những thuật toán phức tạp mới có thể giải quyết được vấn đề. Những thuật toán phức tạp thường yêu cầu độ tính toán cao và nhạy cảm với cách chọn tham số đầu vào. Thay vào đó, những thuật toán đơn giản giúp chúng ta sớm có một mô hình tổng quát cho bài toán. Các thuật toán đơn giản thường được gọi là *baseline* [1], sẽ giúp chúng ta có cái nhìn ban đầu về sự phức tạp của mỗi bài toán. Việc cải thiện kết quả sẽ được dần cải thiện ở những bước sau.

Từ những lý do trên, em quyết định sử dụng một trong những thuật toán cơ bản nhất trong Unsupervised learning - thuật toán K-means clustering (phân cụm K-means) và dựa trên thuật toán này, em đề xuất phương pháp cải tiến có tên BK-means để giải quyết bài toán phân vùng ảnh.

2.2. Học không giám sát (Unsupervised Learning)

a. Supervised learning (học có giám sát)

Chúng ta cần nhắc lại về học có giám sát Supervised learning. Như chúng ta đã biết Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning. Về mặt toán học, Supervised learning là khi chúng ta có một tập hợp biến đầu vào $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ và một tập hợp nhãn tương ứng $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$, trong đó $\mathbf{x}_i, \mathbf{y}_i$ là các vector. Các cặp dữ liệu biết trước $\{\mathbf{x}_i, \mathbf{y}_i\}$ được gọi là tập *training data* (dữ liệu huấn luyện). Từ tập training data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập \mathbf{X} sang một phần tử (xấp xỉ) tương ứng của tập \mathbf{Y} :

$$\mathbf{y}_i \approx \mathbf{f}(\mathbf{x}_i), \forall i = 1, 2, \dots, N$$

Mục đích là xấp xỉ hàm số \mathbf{f} thật tốt để khi có một dữ liệu \mathbf{x} mới, chúng ta có thể tính được nhãn tương ứng của nó $\mathbf{y} = \mathbf{f}(\mathbf{x})$.

Thuật toán supervised learning còn được tiếp tục chia nhỏ ra thành hai loại chính:

- Classification (Phân loại)

Một bài toán được gọi là *classification* nếu các *label* của *input data* được chia thành một số hữu hạn nhóm. Với y chỉ nhận giá trị từ một tập rời rạc, chẳng hạn {cá, cây, quả, mèo}. Ví dụ: Gmail xác định xem một email có phải là *spam* hay *non_spam*.

- Regression (Hồi quy)

Nếu *label* không được chia thành các nhóm mà là một giá trị thực cụ thể. Ví dụ: một căn nhà rộng $x \text{ m}^2$, có y phòng ngủ và cách trung tâm thành phố $z \text{ km}$ sẽ có giá là bao nhiêu? Với y nhận giá trị là số thực.

b. Unsupervised Learning (Học không giám sát)

Trong thuật toán này, chúng ta không biết được *outcome* hay nhãn *label* mà chỉ có dữ liệu đầu vào. Thuật toán *unsupervised learning* sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (*clustering*) hoặc giảm số chiều của dữ liệu (*dimension reduction*) để thuận tiện trong việc lưu trữ và tính toán [1].

Về mặt toán học, *unsupervised learning* có nghĩa là khi chúng ta chỉ có dữ liệu vào \mathbf{X} mà không biết nhãn \mathbf{Y} tương ứng. Máy tính cần học một hàm $\mathbf{y} = \mathbf{f}(\mathbf{x})$ từ tập học cho trước $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$. \mathbf{Y} có thể là các cụm dữ liệu hoặc các cấu trúc ẩn [1].

Những thuật toán loại này được gọi là *unsupervised learning* vì không giống như *supervised learning*, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào.

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành ba loại:

- Clustering (phân nhóm)

Một bài toán phân nhóm toàn bộ dữ liệu \mathbf{X} thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng [1].

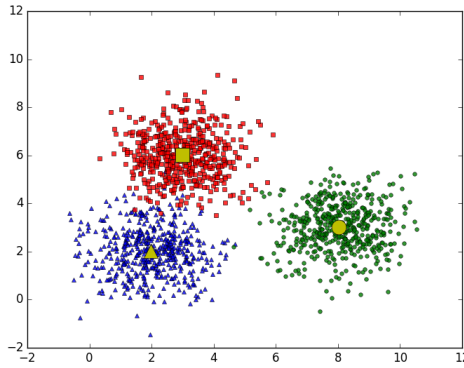
- Community detection: phát hiện các cộng đồng trong mạng xã hội.
- Trends detection: phát hiện xu hướng thị yếu.

2.3. Thuật toán phân cụm K-means

2.3.1. Giới thiệu

Trong thuật toán K-means clustering, chúng ta không biết nhãn *label* của từng điểm dữ liệu. Mục đích là làm thế nào để phân dữ liệu thành các cụm *cluster* khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.

Ý tưởng đơn giản nhất về *cluster* là tập hợp các điểm ở gần nhau trong một không gian nào đó (không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn). Hình bên dưới là một ví dụ về phân cụm với 3 cụm dữ liệu trong không gian 2 chiều.



Hình 2.2: Bài toán phân cụm với 3 cluster [1].

2.3.2. Phân tích toán học

Mục đích cuối cùng của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra *center* của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng. Giả sử rằng mỗi điểm dữ liệu chỉ thuộc vào đúng một nhóm.

Giả sử 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 $K < N$ là số cluster chúng ta muốn phân chia. Chúng ta cần tìm các center $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K \in \mathbb{R}^{d \times 1}$ và label của mỗi điểm dữ liệu.

Với mỗi điểm dữ liệu \mathbf{x}_i đặt $\mathbf{y}_i = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, \dots, \mathbf{y}_{iK}]$ là label vector của nó, trong đó nếu \mathbf{x}_i được phân vào *cluster* \mathbf{k} thì $\mathbf{y}_{ik} = 1$ và $\mathbf{y}_{ij} = 0, \forall j \neq k$. Điều này có nghĩa là có đúng một phần tử của vector \mathbf{y}_i là bằng 1 (tương ứng với cluster của \mathbf{x}_i), các phần tử còn lại bằng 0. Ví dụ: nếu một điểm dữ liệu có *label* vector là $[1, 0, 0, \dots, 0]$ thì nó thuộc vào *cluster* 1, nếu là $[0, 1, 0, \dots, 0]$ thì nó thuộc vào *cluster* 2, [1].. Ràng buộc của \mathbf{y}_i có thể viết dưới dạng toán học như sau:

$$y_{ik} \in \{0, 1\}, \sum_{k=1}^K y_{ik} = 1$$

Công thức 2.1: Label vector của điểm dữ liệu.

Hàm mất mát và bài toán tối ưu

Nếu ta coi center \mathbf{m}_k là center (hoặc representative) của mỗi *cluster* và ước lượng tất cả các điểm được phân vào *cluster* này bởi \mathbf{m}_k , thì một điểm dữ liệu \mathbf{x}_i được phân vào cluster k sẽ bị sai số là $(\mathbf{x}_i - \mathbf{m}_k)$. Ta mong muốn sai số này có trị tuyệt đối nhỏ nhất nên sẽ tìm cách để đại lượng sau đây đạt giá trị nhỏ nhất:

$$\|\mathbf{x}_i - \mathbf{m}_k\|_2^2$$

Hơn nữa, vì \mathbf{x}_i được phân vào *cluster* k nên $y_{ik} = 1, y_{ij} = 0, \forall j \neq k$. Khi đó, biểu thức bên trên sẽ được viết lại là:

$$y_{ik} \|\mathbf{x}_i - \mathbf{m}_k\|_2^2 = \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Công thức 2.2: Hàm khoảng cách [1].

Sai số cho toàn bộ dữ liệu sẽ là:

$$\mathcal{L}(\mathbf{Y}, \mathbf{M}) = \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Công thức 2.3: Sai số cho toàn bộ dữ liệu [1].

Trong đó $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \dots; \mathbf{y}_N]$, $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K]$ lần lượt là các ma trận được tạo bởi *label* vector của mỗi điểm dữ liệu và *center* của mỗi *cluster*. Hàm số mất mát trong bài toán K-means clustering là hàm $\mathcal{L}(\mathbf{Y}, \mathbf{M})$ với ràng buộc như được nêu trong phương trình [1].

Ta cần tối ưu bài toán sau:

$$\mathbf{Y}, \mathbf{M} = \arg \min_{\mathbf{Y}, \mathbf{M}} \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Công thức 2.4: Tối ưu hàm $\mathcal{L}(\mathbf{Y}, \mathbf{M})$ [1].

$$\text{Thỏa mãn: } y_{ij} \in \{0, 1\} \forall i, j; \sum_{j=1}^K y_{ij} = 1 \forall i$$

Ta cần tối ưu 2 biến số Y và M để hàm mất mát $\mathcal{L}(Y, M)$ đạt giá trị nhỏ nhất.

Thuật toán tối ưu hàm mất mát

Bài toán ở **Công thức 2.4** là một bài toán khó tìm điểm tối ưu vì nó có thêm các điều kiện ràng buộc. Bài toán này thuộc loại *mix-integer programming* (điều kiện biến là số nguyên) - là loại rất khó tìm nghiệm tối ưu toàn cục (*global optimal point*, tức nghiệm làm cho hàm mất mát đạt giá trị nhỏ nhất có thể). Tuy nhiên, trong một số trường hợp ta vẫn có thể tìm được phương pháp để tìm được nghiệm gần đúng hoặc điểm cực tiểu. Điểm cực tiểu ở đây chưa chắc đã phải là điểm làm cho hàm số đạt giá trị nhỏ nhất (cực tiểu toàn cục), mà nó có thể là điểm cực tiểu địa phương (điểm cực tiểu cục bộ).

Một cách đơn giản để giải bài toán **Công thức 2.4** là xen kẽ giải Y và M khi biến còn lại được cố định làm tham số.

Cố định M và tìm Y

Giả sử đã tìm được các centers, hãy tìm các label vector để hàm mất mát đạt giá trị nhỏ nhất. Điều này tương đương với việc tìm cluster cho mỗi điểm dữ liệu.

Khi các centers là cố định, bài toán tìm label vector cho toàn bộ dữ liệu có thể được chia nhỏ thành bài toán tìm label vector cho từng điểm dữ liệu x_i như sau:

$$y_i = \arg \min_{y_i} \sum_{j=1}^K y_{ij} \|x_i - m_j\|_2^2$$

Công thức 2.5: Tính label vector cho từng điểm dữ liệu [1].

$$\text{Thỏa mãn: } y_{ij} \in \{0,1\} \forall i, j; \sum_{j=1}^K y_{ij} = 1 \forall i$$

Vì chỉ có một phần tử của *label vector* y_i bằng 1 nên bài toán có thể tiếp tục được viết dưới dạng đơn giản hơn:

$$j = \arg \min_j \|x_i - m_j\|_2^2$$

Công thức 2.6: Công thức thu gọn của **Công thức 2.5**.

Vì $\|x_i - m_j\|_2^2$ chính là bình phương khoảng cách tính từ điểm x_i đến *center* m_i , ta có thể kết luận rằng mỗi điểm x_i thuộc vào *cluster* có *center* gần nó nhất! Từ đó ta có thể dễ dàng suy ra *label vector* của từng điểm dữ liệu.

Cố định Y và tìm M

Giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất. Một khi chúng ta đã xác định được label vector cho từng điểm dữ liệu, bài toán tìm center cho mỗi cluster được rút gọn thành:

$$\mathbf{m}_j = \arg \min_{\mathbf{m}_j} \sum_{i=1}^N y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Công thức 2.7: Tính label vector cho từng điểm dữ liệu.

Lúc này, ta có thể tìm nghiệm bằng phương pháp giải phương trình đạo hàm bằng 0, vì hàm cần tối ưu là một hàm liên tục và có đạo hàm xác định tại mọi điểm. Và quan trọng hơn, hàm này là hàm lồi (convex) theo m_j nên chúng ta sẽ tìm được giá trị nhỏ nhất và điểm tối ưu tương ứng.

Đặt $f(m_j) = \sum_{i=1}^N y_{ij} \|\mathbf{x}_i - m_j\|_2^2$, ta có đạo hàm

$$\frac{\partial f(m_j)}{\partial m_j} = 2 \sum_{i=1}^N y_{ij} (m_j - x_i)$$

Công thức 2.8: Đạo hàm của hàm khoảng cách [1].

Giải phương trình đạo hàm bằng 0 ta có:

$$\begin{aligned} m_j \sum_{i=1}^N y_{ij} &= \sum_{i=1}^N y_{ij} x_i \\ \Rightarrow m_j &= \frac{\sum_{i=1}^N y_{ij} x_i}{\sum_{i=1}^N y_{ij}} \end{aligned}$$

Công thức 2.9: Nghiệm của bài toán.

Ở đây mẫu số chính là phép đếm số lượng các điểm dữ liệu trong cluster j . Còn tử số chính là tổng các điểm dữ liệu trong cluster j . Nghĩa là m_j chính là trung bình cộng của các điểm trong cluster j .

2.3.3. Thuật toán K-means cơ bản

Đầu vào: Dữ liệu X và số cụm cluster cần tìm K .

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y .

1. Chọn K điểm bất kỳ làm các center ban đầu.
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
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.
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.
5. Quay lại bước 2.

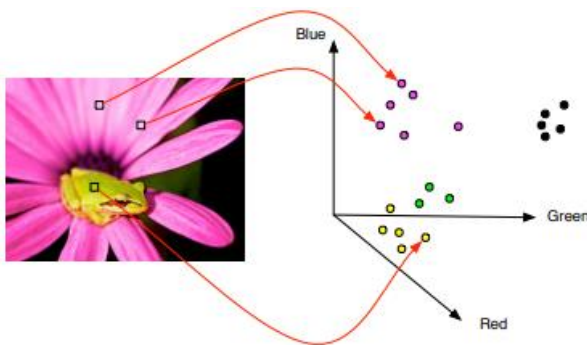
Thuật toán sẽ dừng lại sau một số hữu hạn vòng lặp (luôn hội tụ sau một số hữu hạn lần lặp). Vì hàm mất mát là một số dương và sau mỗi bước 2 hoặc 3, giá trị của hàm mất mát bị giảm đi. Số lượng cách phân nhóm cho toàn bộ dữ liệu là hữu hạn nên đến một lúc nào đó, hàm mất mát sẽ không thể thay đổi, và khi đó thuật toán sẽ dừng.

Chương 3. Thị giác máy tính trong phân vùng ảnh

3.1. Bài toán phân vùng ảnh màu

3.1.1. Bài toán phân vùng ảnh màu (color image segmentation)

Hình ảnh xám (grayscale) là một loại hình ảnh đơn giản có chứa một miền và mỗi pixel trong hình ảnh có thể là được đại diện bởi một số nguyên $[0,255]$ tức là nó chỉ mang thông tin cường độ [11]. Đối với ảnh màu, nó là một hình ảnh có chứa 3 thông tin đó là 2 thuộc tính về không gian tọa độ và thuộc tính còn lại là màu sắc, màu sắc của nó có thể là các không gian màu như RGB, HSV,... Nó mang đúng nghĩa của 1 ma trận 3 chiều. Dưới đây là mô tả một ảnh màu trong không gian màu RGB.



Hình 3.1: Ảnh màu trong không gian màu RGB.

(Nguồn: [Image segmentation using K-means](#))

Thuật toán K-means đã được sử dụng rất nhiều trong nghiên cứu và được cài đặt trong các phần mềm xử lý ảnh viễn thám. Tuy nhiên, khi phân vùng ảnh màu kích thước lớn, tốc độ hội tụ của thuật toán vẫn rất chậm. Trong đồ án này, tôi đề xuất một phương pháp cải tiến thuật toán phân cụm K-Means, cải tiến việc dự đoán tối ưu số cụm K và khởi tạo tâm cụm ban đầu hiệu quả, kết hợp với bước tiền xử lý ảnh đầu vào nhằm tăng tốc độ phân vùng ảnh màu có kích thước lớn, và cho kết quả đầu ra tốt đối với ảnh bị nhiễu.

Bảng 3.1: Các công việc cải tiến K-means trước đây [12]

| Tác giả, năm, Tài liệu tham khảo | Cải tiến K-means |
|--|--|
| J.B.McQueen, 1967, [26]. | Đề xuất một chiến lược học tập để xác định một nhóm các cụm gốc cho phân cụm. |
| J.Tou and R.Gonzales, 1974, [27]. | Đề xuất phương pháp tìm kiếm cụm đơn giản. |
| Y. Linde, A. Buzo, R. M. Gray, 1980, [28]. | Đề xuất phương pháp chia nhị phân cho phân cụm. |
| G. P. Babu, M. N. Murty, 1993, [29]. | Đề xuất phương pháp lập trình di truyền dựa trên tâm cụm tối ưu gần nhất. |
| C.Huang, R. Harris , 1993, [30]. | Dựa trên nguyên tắc phân tích cấu trúc, đề xuất tìm kiếm trực tiếp bằng phương pháp chia nhị phân. |
| I. Katsavounidis, C. C. J. Kuo, Z. Zhen, 1994, [31]. | Đề xuất một phương pháp khởi tạo tâm cụm bằng lựa chọn một điểm làm tâm cụm đầu tiên trên các cạnh (biên) của dữ liệu. Điểm mà ở xa nhất từ hạt giống đầu tiên sẽ được xem xét và được chọn làm tâm cụm thứ hai. |
| M. B. A. Daoud, S. A. Roberts, 1996, [32]. | Giới thiệu một phương pháp để phân chia toàn bộ dữ liệu thành hai nhóm khác nhau và các điểm được phân phối ngẫu nhiên trong nhóm. |
| P. S. Bradley, U. M. Fayyad, 1998, [33]. | Đề xuất một phương pháp để chọn ví dụ trung tâm nhất là tâm cụm đầu tiên. |
| A. Likas, N. Vlassis, J. J. Verbeek, 2003, [34]. | Đề xuất một thuật toán K-means, trong đó tăng dần số cụm cho đến khi tìm thấy K cho đến khi hội tụ. |
| S. S. Khan, A. Ahmad, 2004, [35]. | Giới thiệu một khởi tạo centroid phương pháp dựa trên mật độ ngưng tụ dữ liệu đa quy mô. Trong phương pháp này, mật độ của dữ liệu tại điểm được ước tính, và sau đó sắp xếp điểm dựa trên mật độ của chúng. |
| Bo Zhao, Zhongxiang Zhu, Enrong Mao and Zhenghe Song , 2007, [36]. | Đề xuất một phương pháp phân vùng ảnh dựa trên tối ưu hóa đàn kiến và phân cụm K-means. |
| Nor Ashidi Mat Isa, Samy A. Salamah, Umi Kalthum Ngah, 2009, [37]. | Đề xuất này thuật toán liên tục kiểm tra sự tương thích của từng centroid trong giai đoạn phân cụm. Nếu trung tâm không thể đáp ứng các tiêu chí, trọng tâm di chuyển đến nhóm dữ liệu mà trung tâm cụm đã đóng hoặc còn hoạt động nhiều nhất. |

3.1.2. Biểu diễn dữ liệu

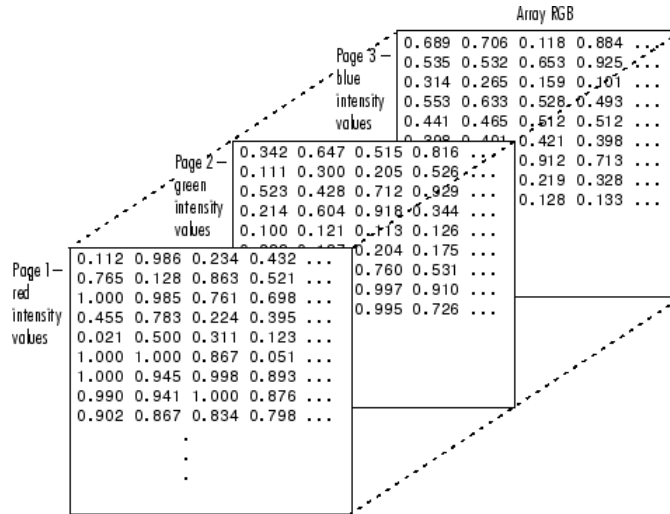
Ý tưởng cơ bản để phân đoạn hình ảnh bằng thuật toán phân cụm K-means là tạo các cụm dựa trên giá trị màu của mỗi pixel, vì thế độ chói (hay độ sáng) của mỗi pixel sẽ bị bỏ qua, ta chỉ quan tâm đến các thành phần về màu của pixel [4][10]. Vì vậy trong phần này, tôi áp dụng phân cụm K-means cho ảnh trên không gian màu $L^*a^*b^*$. Trong không gian màu $L^*a^*b^*$, loại bỏ thành phần về độ sáng L , và chỉ giữ lại 2 thành phần về màu là kênh màu a^* và b^* [4][10]. Lúc này kích thước dữ liệu đầu vào giảm, giúp cho thuật toán chạy nhanh hơn và phân vùng ảnh chính xác hơn. Sau đó ta cần *reshape* lại kích thước của ma trận ảnh từ ma trận 3 chiều sang ma trận 2 chiều.

Ảnh số thực chất là một ma trận các điểm ảnh. Đối với ảnh màu, mỗi điểm ảnh sẽ có 3 thuộc tính tương ứng với 3 chiều dữ liệu. Hai chiều về không gian chính là tọa độ hay vị trí của pixel đó trong ma trận điểm ảnh, chiều thứ ba là giá trị về màu sắc (màu ở đây có thể là không gian màu RGB). Dưới đây là phương pháp reshape một ảnh 3D về một mảng 2D:

$$I(h, w, c) \rightarrow A(h \times w, c)$$

Trong đó: h , w là kích thước và c là số kênh màu của ảnh I .

Mỗi điểm ảnh lúc này sẽ là 1 hàng của ma trận A , ma trận A sẽ có $h \times w$ hàng (ứng với tổng số pixel trong ảnh), và có c cột ($c = 3$ nếu là không gian màu RGB).



Hình 3.2: Biểu diễn ảnh màu RGB bằng ma trận 3 chiều.

(Nguồn: [Image 3D](#))

Phương pháp chuyển đổi ảnh từ hệ màu RGB sang hệ màu L*a*b*:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.357580 & 0.119193 & 0.950227 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

(Nguồn: [brucelindbloom](#))

Công thức 3.1: Chuyển từ hệ màu RGB sang Lab.

$$X = X/X_n, \quad X_n = 0.950456$$

$$Z = Z/Z_n, \quad Z_n = 1.088754$$

$$L = \begin{cases} 116 \times Y^{1/3} - 16, & Y > 0.008856 \\ 903.3 \times Y, & Y \leq 0.008856 \end{cases}$$

$$a = 500 \times (f(X) - f(Y)) + \delta$$

$$b = 200 \times (f(Y) - f(Z)) + \delta$$

$$\text{Trong đó: } f(t) = \begin{cases} t^{1/3}, & t > 0.008856 \\ 7.787t + 16/116, & t \leq 0.008856 \end{cases}$$

$$\text{và } \delta = \begin{cases} 128, & \text{cho ảnh 8 - bits} \\ 0, & \text{cho ảnh dấu chấm động} \end{cases}$$

$$0 \leq L \leq 100, -127 \leq a \leq 127, -127 \leq b \leq 127$$

3.1.3. Tiền xử lý ảnh

Là quá trình sử dụng các kỹ thuật xử lý ảnh để nâng cao chất lượng ảnh. Phương pháp sử dụng SMQT (Successive Mean Quantization Transform). Đơn vị cơ bản của SMQT là MQU (Mean Quantization Unit) trong đó bao gồm việc tính giá trị trung bình của tất cả các pixel trong ảnh, sau đó giá trị trung bình được sử dụng để định lượng giá trị của dữ liệu thành 0 hoặc 1, tùy thuộc vào giá trị của pixel thấp hơn hoặc cao hơn giá trị trung bình [3].

MQU:

$$\text{Pixel (V)} \rightarrow \text{Mean} \rightarrow M\{0,1\}$$

| | | | | | | |
|----|----|----|----|-----|-----|-----|
| 93 | 91 | 71 | 82 | 93 | 110 | 119 |
| 90 | 81 | 68 | 78 | 93 | 110 | 118 |
| 65 | 66 | 70 | 82 | 93 | 111 | 118 |
| 64 | 65 | 73 | 79 | 96 | 111 | 121 |
| 65 | 67 | 68 | 80 | 97 | 123 | 127 |
| 65 | 67 | 69 | 82 | 98 | 121 | 122 |
| 66 | 65 | 71 | 83 | 96 | 127 | 117 |
| 63 | 66 | 71 | 82 | 101 | 130 | 113 |
| 63 | 68 | 76 | 86 | 97 | 120 | 110 |
| 67 | 67 | 73 | 85 | 98 | 95 | 85 |
| 70 | 74 | 74 | 85 | 97 | 87 | 92 |
| 73 | 71 | 73 | 84 | 98 | 89 | 85 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Hình 3.3: Đơn vị lượng tử trung bình – MQU (Mean Quantization Unit) [3].

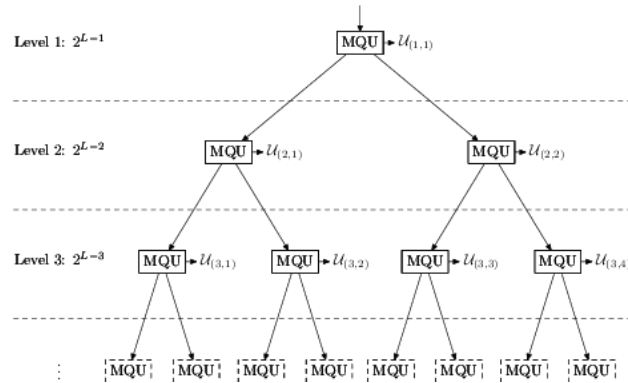
$$\text{Mean} = 87.5595$$

$$M(x) = \begin{cases} 1, & \text{if } V(x) > \text{Mean} \\ 0, & \text{else} \end{cases}$$

SMQT có thể được coi là một cây của các phép tính MQU, trong đó trọng số được đưa ra tùy thuộc vào mức độ hiện tại của cây [3].

$$\text{Weight} = 2^{L-i}$$

Trong đó L là mức tối đa của cây, i là mức hiện tại.



Hình 3.4: Cấu trúc cây trong SMQT [3].

Trong ảnh RGB, SMQT có thể được áp dụng theo hai cách khác nhau. Đầu tiên, áp dụng SMQT cho ba kênh RGB. Cho $D_{RGB}(X)$ là tất cả các giá trị dữ liệu bất kể kênh nào, thì:

$$SMQT_L: D_{RGB}(X) \rightarrow M_{RGB}(X) [3]$$

Điều này sẽ dẫn đến sự tăng cường độ tương phản phi tuyến bảo tồn thứ tự của các giá trị RGB cho mỗi pixel, nhưng với khoảng cách thay đổi giữa các giá trị R, G và B trong mỗi pixel. Cách thứ hai là áp dụng nó một cách riêng biệt trong mỗi ba kênh. Đặt

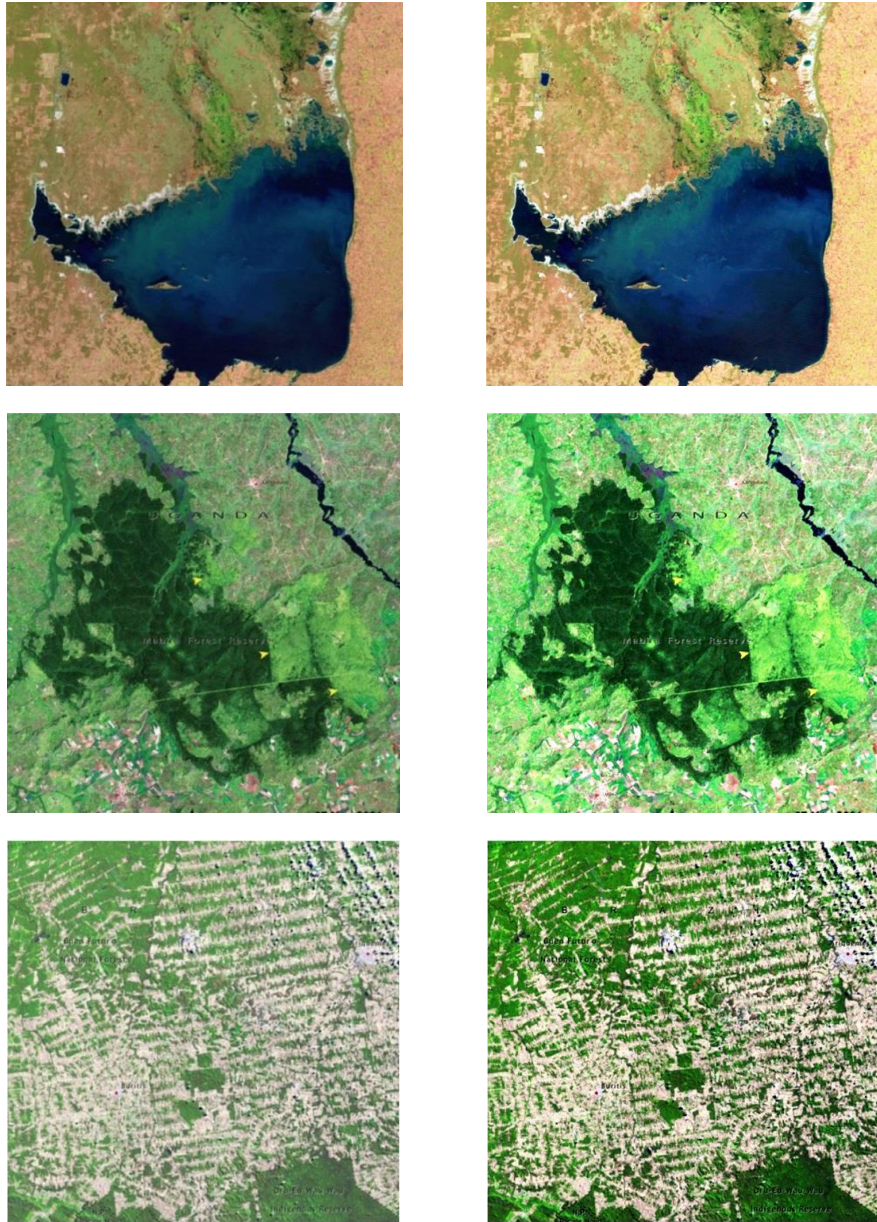
$D_R(X)$, $D_G(X)$ và $D_B(X)$ là tất cả các giá trị dữ liệu của kênh màu R, G và B tương ứng [3], thì ta có:

$$SMQT_L: D_R(X) \rightarrow M_R(X)$$

$$SMQT_L: D_G(X) \rightarrow M_G(X)$$

$$SMQT_L: D_B(X) \rightarrow M_B(X)$$

Cuối cùng, các pixel được tăng cường $M_{RGB}(X)$ thu được bằng cách ghép các kênh màu $M_R(X)$, $M_G(X)$ và $M_B(X)$. Dưới đây là hình ảnh sau khi thực hiện quá trình tiền xử lý bằng phương pháp SMQT:



Hình 3.5: Kết quả tiền xử lý ảnh.

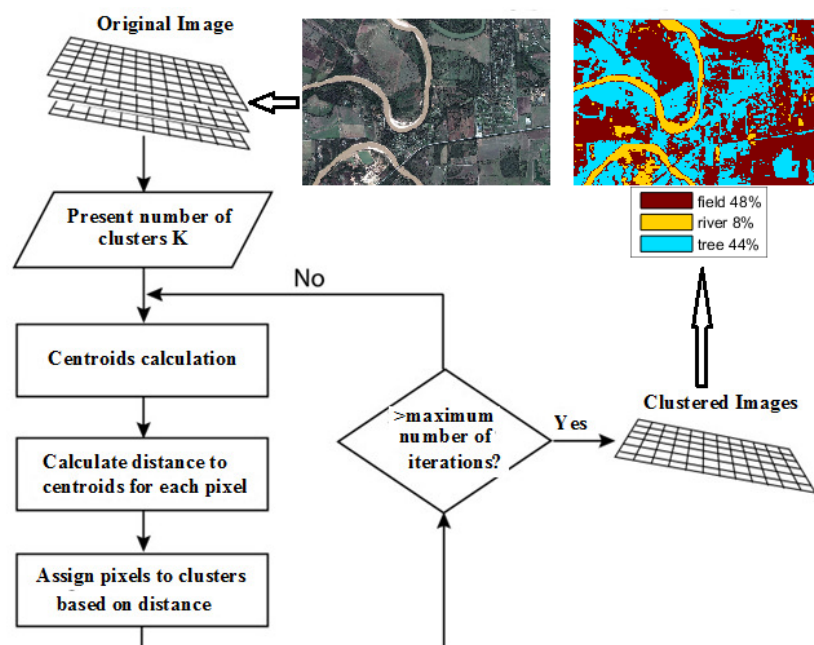
3.2. Xây dựng mô hình

3.2.1. Phương pháp cải tiến BK-means

Hiệu năng của thuật toán K-means bị ảnh hưởng rất nhiều ở bước khởi tạo trung tâm K cụm đầu tiên. Tùy vào các center ban đầu mà thuật toán có thể có tốc độ hội tụ rất chậm, hoặc thậm chí cho chúng ta nghiệm không chính xác (chỉ là local minimum - điểm cực tiểu địa phương - mà không phải giá trị nhỏ nhất). Đó là một nhược điểm lớn của K-means cho bài toán phân vùng ảnh nói riêng và phân cụm nói chung. Vì thế ta sẽ đi khắc phục nhược điểm này để có được bước khởi tạo trung tâm K ban đầu tốt nhất [1].

Một nhược điểm nữa ta sẽ cần khắc phục đó là K-means cho kết quả không tốt đối với các hình ảnh bị nhiễu, hay có chất lượng thấp. Vì thế trước khi thực hiện phân cụm ảnh với K-means, ta sẽ nâng cao chất lượng ảnh đầu vào, tăng cường đặc trưng cho các điểm ảnh.

Thuật toán K-means được chứng minh là hội tụ và có độ phức tạp tính toán là $O(tkn)$ với t là số lần lặp, k là số cụm, n là số đối tượng của tập dữ liệu vào. Thông thường $k \ll n$ và $t \ll n$ thuật toán thường kết thúc tại một điểm tối ưu cục bộ.



Hình 3.6: Sơ đồ phương pháp phân vùng ảnh.

Đề xuất phương pháp BK-means cho bài toán phân vùng ảnh màu dựa trên thuật toán phân cụm K-means cơ bản:

Đầu vào: Ảnh X và số cụm cluster cần tìm K.

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y.

1. Đọc ảnh màu X.
2. Tiền xử lý ảnh X sử dụng phương pháp SMQT [3.1.3].
3. Chuyển đổi ảnh X từ không gian màu RGB sang Lab. (Trong không gian L^*a^*b loại bỏ kênh độ sáng L, chỉ lấy 2 kênh màu a^* và b^*) [3.1.2].
4. Biểu diễn dữ liệu: Định dạng ảnh X từ không gian 3D chuyển thành không gian 2D [3.1.2] thu được ma trận điểm ảnh \hat{X} .
5. Lựa chọn tham số K từ ma trận \hat{X} : sử dụng phương pháp Elbow và Silhouette [3.2.2].
6. Khởi tạo K trung tâm cụm đầu tiên: Sắp xếp ma trận \hat{X} theo thứ tự không giảm, chia toàn bộ dữ liệu hình ảnh được sắp xếp thành K phần đều nhau (K cụm đã dự đoán ở bước 4). Tính giá trị trung bình của từng phần ở bước 5 và giá trị trung bình đó được gán thành trung tâm cho cụm tương ứng và lưu lại thành trung tâm cụm ban đầu [4].
7. Áp dụng thuật toán phân cụm K-means [2.3.3] cho ma trận ảnh với số cụm K và trung tâm cụm ban đầu đã tối ưu ở bước 5 và bước 6.
 - a. Lưu lại các nhãn của cụm và khoảng cách của tất cả các đối tượng đến cụm gần nhất trong mỗi lần lặp, vì có thể được sử dụng trong lần lặp lại tiếp theo [5].
 - b. Tính khoảng cách giữa đối tượng hiện tại và trung tâm cụm mới, nếu khoảng cách này nhỏ hơn hoặc bằng khoảng cách đến trung tâm cụm hiện tại thì đối tượng vẫn ở trong cụm được gán cho lần lặp trước đó [5].

3.2.2. Lựa chọn tham số

Trước khi áp dụng thuật toán K-means clustering cho bộ dữ liệu ta cần tối ưu tham số K để có được kết quả tốt nhất.

a) Phương pháp khuỷu tay (Elbow)

Ý tưởng là đối với một phạm vi các giá trị thích hợp của K, tính sai số của tổng lỗi trong cụm (wcss - within-cluster sum of errors) để tìm ra giá trị K tối ưu nhất. Ta có thể tìm thấy K ở vị trí khuỷu tay, tức là trước khi gặp khuỷu tay lỗi giảm nhanh dần và sau khi qua khuỷu tay lỗi giảm không đáng kể [17].

$$wcss(C) = \sum_{i=1}^K \sum_{o \in C_i} d(o, cen_i)^2$$

Công thức 3.2: Tổng bình phương lỗi trong cụm [17].

(Nguồn: [Clustering Algorithms and Evaluations](#))

Trong đó: o_i là đối tượng và cen_i là trung tâm của cụm thứ i, d là hàm khoảng cách Euclidean, K là số cụm.

b) Phương pháp Average Silhouette

Phương pháp này thể hiện tính nhất quán trong các *cluster*, giá trị *silhouette* là thước đo mức độ tương tự của một đối tượng với cụm (sự gắn kết) của chính nó so với các cụm khác (sự tách biệt) hay mức độ phân tách giữa các cụm [17]. Silhouette nằm trong phạm vi -1 đến $+1$, trong đó giá trị cao biểu thị rằng đối tượng được kết hợp tốt với cụm của chính nó và tách rời với các cụm lân cận. Nếu hầu hết các đối tượng có giá trị cao, thì cấu hình phân cụm là phù hợp. Nếu nhiều điểm có giá trị thấp hoặc âm, thì cấu hình phân cụm có thể có quá nhiều hoặc quá ít cụm. Silhouette được tính như sau:

Tính khoảng cách trung bình từ tất cả các điểm dữ liệu trong cùng một cụm $a(o_i)$. Tính khoảng cách trung bình từ tất cả các điểm dữ liệu trong cụm lân cận (gần nhất) $b(o_i)$ [17]. Để có được giá trị silhouette cho một đối tượng o_i trong một cụm Cluster A, chúng ta so sánh trung bình khoảng cách a giữa o_i và tất cả các đối tượng khác trong Cluster A với khoảng cách trung bình b giữa o_i và tất cả các đối tượng trong cụm Cluster B lân cận. Đối với từng đối tượng o_i thỏa mãn $-1 \leq sil(o_i) \leq 1$. Nếu $sil(o_i)$ lớn, trung bình khoảng cách đối tượng trong cụm nhỏ hơn hơn khoảng cách trung bình đến các đối tượng trong cụm lân cận, vì vậy o_i được phân loại tốt [17]. Nếu $sil(o_i)$ là nhỏ, khoảng cách đối tượng trung bình trong cụm lớn hơn khoảng cách trung bình đến các đối tượng trong cụm lân cận, vì vậy o_i đã bị phân loại sai.

$$a(o_i) = \frac{1}{|C_A| - 1} \sum_{o_j \in C_A, o_j \neq o_i} d(o_i, o_j)$$

$$b(o_i) = \min_{C_B \neq C_A} \frac{1}{|C_B|} \sum_{o_j \in C_B} d(o_i, o_j)$$

$$sil(o_i) = \frac{b(o_i) - a(o_i)}{\max\{b(o_i), a(o_i)\}}$$

Công thức 3.3: Công thức tính Silhouette [17].

(Nguồn: [Clustering Algorithms and Evaluations](#))

$$\text{Hoặc là } S_i = \begin{cases} 1 - a_i/b_i, & \text{if } a_i < b_i \\ 0, & \text{if } a_i = b_i \\ b_i/a_i - 1, & \text{if } a_i > b_i \end{cases}$$

$$-1 \leq S_i \leq 1$$

Nếu $S_i = 0$ đối tượng i rất gần với các cụm lân cận.

Nếu $S_i = 1$ đối tượng nằm cách xa các cụm lân cận.

Nếu $S_i = -1$ đối tượng được gán cho các sai cụm.

3.3. Phương pháp đánh giá

- Sai số bình phương trung bình: Mean Squared Error (MSE)

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (I_{ij} - \hat{I}_{ij})^2$$

Công thức 3.4: Sai số bình phương trung bình [6].

Trong đó: \hat{I} – origin image, I – segmented image,

M, N là kích thước của ảnh I .

$$RMSE = \sqrt{MSE}$$

Giá trị RMSE càng nhỏ thì ảnh đầu ra sẽ có chất lượng càng tốt.

- Tỷ lệ nhiễu tín hiệu: Signal-to-noise ratio (SNR)

$$SNR = \frac{\sum_{i=1}^M \sum_{j=1}^N I_{ij}^2}{\sum_{i=1}^M \sum_{j=1}^N (I_{ij} - \hat{I}_{ij})^2}$$

Công thức 3.5: Tỷ lệ nhiễu tín hiệu [6].

Trong đó: \hat{I} – origin image, I – segmented image,

M, N là kích thước của ảnh I .

- Tỷ lệ nhiễu tín hiệu cực đại: Peak Signal-to-Noise Ratio (PSNR)

$$PSNR = 10 \times \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Công thức 3.6: Tỷ lệ nhiễu tín hiệu cực đại [6].

Giá trị PSNR cao hơn tương ứng với một hình ảnh chất lượng tốt hơn.

- Chỉ số tương đồng về cấu trúc: Structural Similarity Index Metric (SSIM)

$$SSIM(x, y) = \frac{(2\mu_x + \mu_y)(2\sigma_{xy})}{(\mu_x^2 + \mu_y^2)(\sigma_x^2 + \sigma_y^2)}$$

Công thức 3.7: Chỉ số tương đồng về cấu trúc [6].

Trong đó:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i, \quad \mu_y = \frac{1}{N} \sum_{i=1}^N y_i$$

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2}$$

$$\sigma_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2}$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$$

N là tổng số pixel, x – segmented image, y – origin image.

- Sai số trung bình tuyệt đối: Mean Absolute Error (MAE)

Phương pháp này được sử dụng để phát hiện hiệu ứng làm mờ trong bất kỳ hình ảnh thời gian thực nào, phổ biến nhất là hình ảnh vệ tinh. Thông thường, ảnh vệ tinh bị mờ do nhiễu loạn khí quyển.

$$MAE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |I_{ij} - \hat{I}_{ij}|$$

Công thức 3.8: Sai số trung bình tuyệt đối [6].

Trong đó: \hat{I} – origin image, I – segmented image,

M, N là kích thước của ảnh I .

- Chỉ số cấu trúc về nội dung: Structural Content (SC)

Các nội dung cấu trúc liên quan đến sắp xếp không gian của pixel trong một hình ảnh, đưa ra sự tương đồng giữa hai hình ảnh. Khi hai hình ảnh càng giống nhau được so sánh thì SC của nó có giá trị tiến gần đến 1 (tối đa).

$$SC = \frac{\sum_{i=1}^M \sum_{j=1}^N I_{ij}^2}{\sum_{i=1}^M \sum_{j=1}^N \hat{I}_{ij}^2}$$

Công thức 3.9: Chỉ số cấu trúc về nội dung [6].

Trong đó: \hat{I} – origin image, I – segmented image,

M, N là kích thước của ảnh I .

- Đánh giá dựa trên kết quả phân cụm

Trong mỗi $\frac{N(N-1)}{2}$ cặp điểm ảnh trong từng lớp dữ liệu, ta muốn gán từng cặp pixel này cho cùng một cụm nếu và chỉ khi chúng giống nhau. Một quyết định dương tính thực sự (TP) khi gán hai điểm ảnh tương tự cho cùng một cụm, một quyết định phủ định (TN) thực sự là khi gán hai điểm ảnh khác nhau cho các cụm khác nhau. Có hai loại lỗi chúng ta có thể phạm phải. Một là (FP) quyết định gán hai điểm ảnh khác nhau cho cùng một cụm. Hai là (FN) quyết định gán hai điểm ảnh tương tự nhau cho các cụm khác nhau. Chỉ số RI cho biết tỷ lệ phần trăm của các quyết định chính xác.

$$RI = \frac{TP + TN}{TP + FP + TN + FN}$$

Xem **Công thức 4.15**

$$Precision = \frac{TP}{TP + FP}$$

Xem **Công thức 4.16**

$$Recall = \frac{TP}{TP + FN}$$

Xem **Công thức 4.17**

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Xem **Công thức 4.18**

3.4. Kết quả thử nghiệm

Bảng 3.2: Kết quả thử nghiệm phương pháp BK-means dựa trên ảnh Ground Truth

| Bảng 3.7 | Cluster K | Time(s) | RMSE | MAE | SNR | PSNR | SC | SSIM |
|-----------------|------------------|----------------|-------------|------------|------------|-------------|-----------|-------------|
| Ảnh 1 | 2 | 8.76 | 0.0478 | 0.0143 | 96.16 | 74.53 | 0.98 | 0.9422 |
| Ảnh 2 | 2 | 3.79 | 0.0347 | 0.0146 | 154.5 | 77.31 | 0.99 | 0.9469 |

Kết quả khi phân tích hình ảnh 1 trong **Bảng 3.7** với số cụm K=2, kích thước ảnh 481×321 pixel:

Bảng 3.3: Kết quả phân cụm 481×321 điểm ảnh với K=2

| Total=154401 | Predicted as 0 | Predicted as 1 | |
|---------------------|-----------------------|-----------------------|--------|
| Actual: 0 | 140693 | 609 | 141302 |
| Actual: 1 | 4145 | 8954 | 13099 |

Bảng 3.4: Kết quả phân cụm *same cluster* và *different cluster* với K=2

| | Same cluster | Different cluster |
|--------------------------|---------------------|--------------------------|
| Same classes | TP = 9946045935 | FN = 1262289427 |
| Different classes | FP = 588625471 | TN = 122796367 |

Trong đó: Same Classes là Actual Positive, Different Classes là Actual Negative,

Same cluster là Predicted as Positive, Same Classes là Predicted Negative.

Bảng 3.5: Tỷ lệ TPR, FPR, FNR, TNR cho bài toán phân cụm K=2


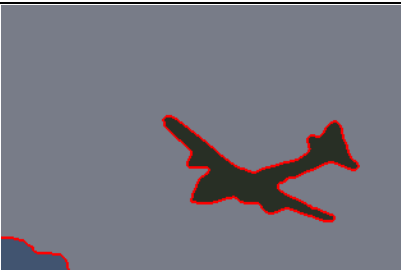




| | Predicted as Positive | Predicted as Negative |
|-------------------------|------------------------------|------------------------------|
| Actual: Positive | TPR = 0.987804 | FNR = 0.012196 |
| Actual: Negative | FPR = 0.318019 | TNR = 0.681981 |

Bảng 3.6: Tỷ lệ RI Precision, Recall, F_1 cho bài toán phân cụm K=2

| | |
|-------------------------|----------|
| RI | 0.940316 |
| Precision | 0.944125 |
| Recall | 0.987804 |
| F_1 | 0.965471 |

Công thức tính (TP, FP, TN, FN) và (TPR, FPR, FNR, TNR) và (RI, Precision, Recall, F_1) tham khảo mục [4.3.2]

Bảng 3.7: Hình ảnh phân vùng bằng BK-means và ảnh Ground Truth

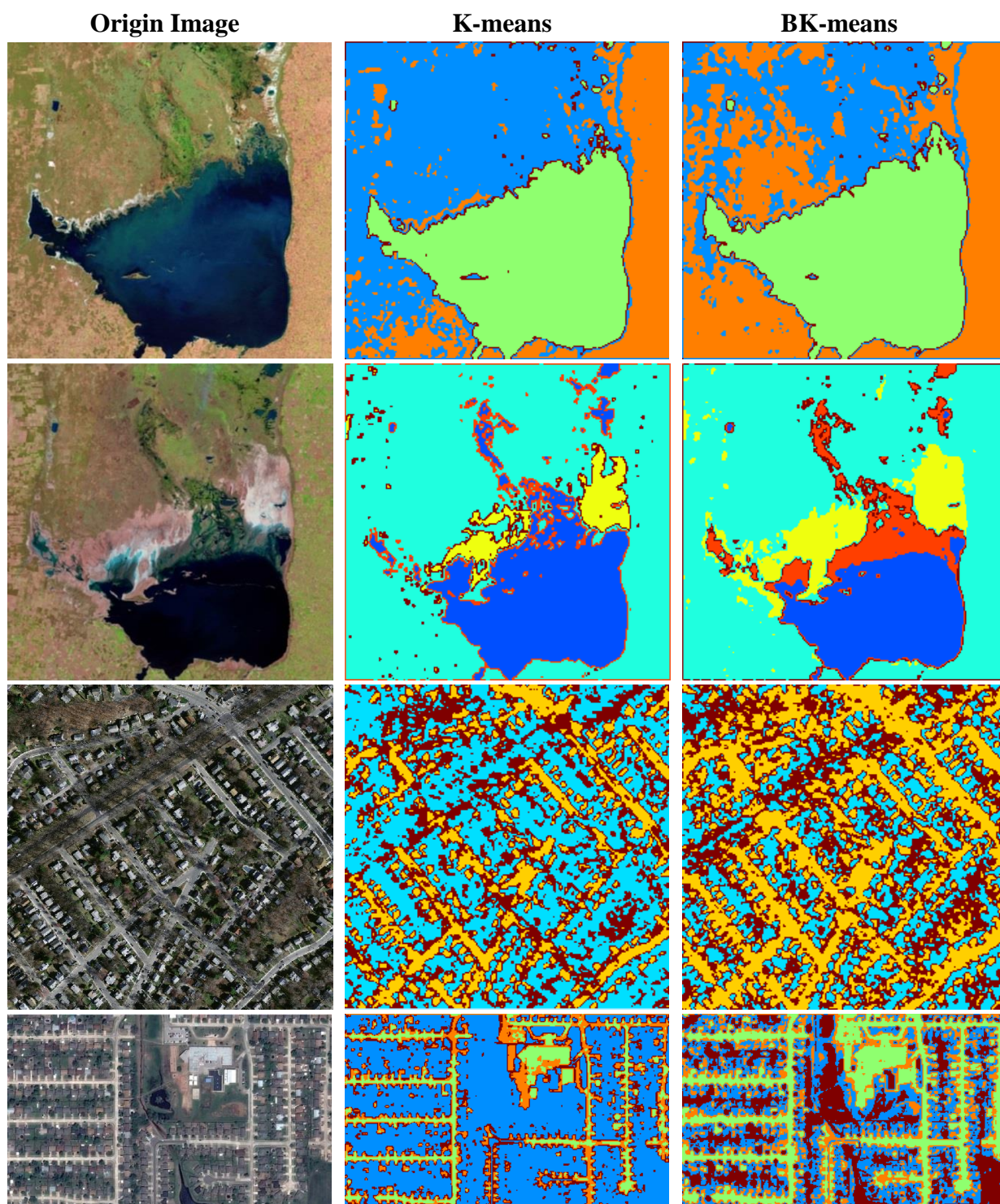
| Ảnh gốc ¹ | Ground Truth ² | BK-means |
|---|--|---|
|  |  |  |
|  |  |  |

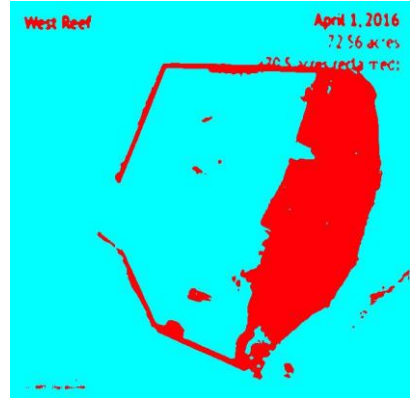
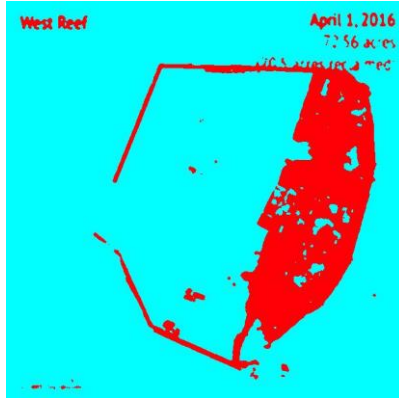
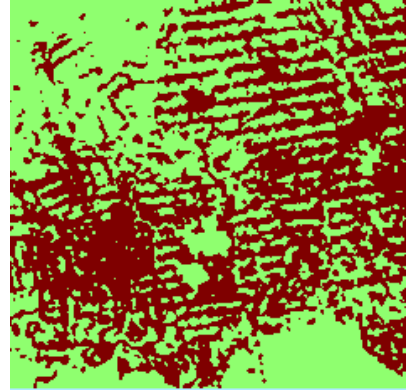
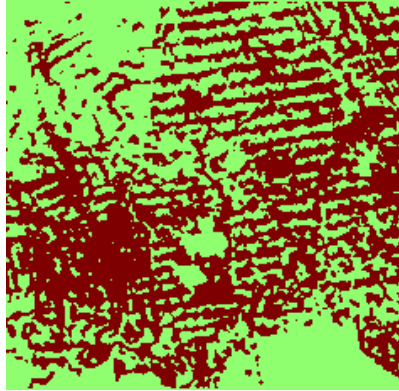
(Nguồn ảnh gốc và Ground Truth: [lossy segmentation](https://people.eecs.berkeley.edu/~yang/software/lossy_segmentation/))

¹ Nguồn https://people.eecs.berkeley.edu/~yang/software/lossy_segmentation/

²

Bảng 3.8: So sánh kết quả phân vùng ảnh BK-means và K-means trên bộ dữ liệu ảnh vệ tinh

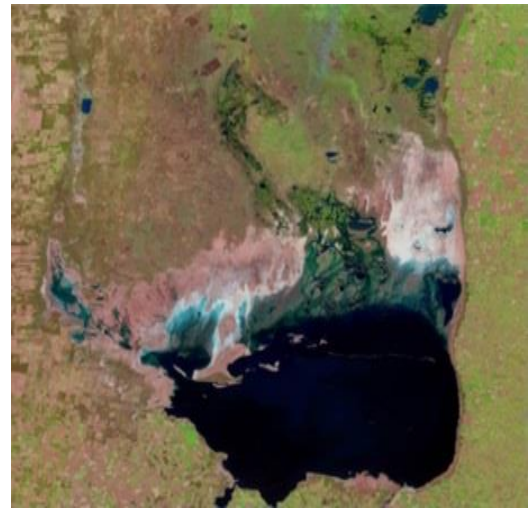




Để chi tiết hơn, tôi sẽ sử dụng 2 hình ảnh dưới đây để thấy rõ nhất sự khác biệt thay đổi của phương pháp đề xuất BK-means so với phương pháp K-means cơ bản, và các lợi ích mà mô hình mang lại cho ứng dụng trong thực tiễn. Thực hiện áp dụng phương pháp đề xuất và thuật toán K-means để tự động nhận diện ra vùng hồ nước, vùng rừng cây và thảm thực vật, vùng đất, vùng cát ven hồ nước. Bằng trực quan chúng ta cũng có thể dự đoán được bức ảnh có 5 màu chủ đạo: xanh lục ở hồ nước, xanh lam vùng rừng cây và thảm thực vật, màu nâu ở đất, màu trắng của cát, màu xám của khu vực đầm lầy, điều đó tương ứng với 5 cluster.



1998.jpg

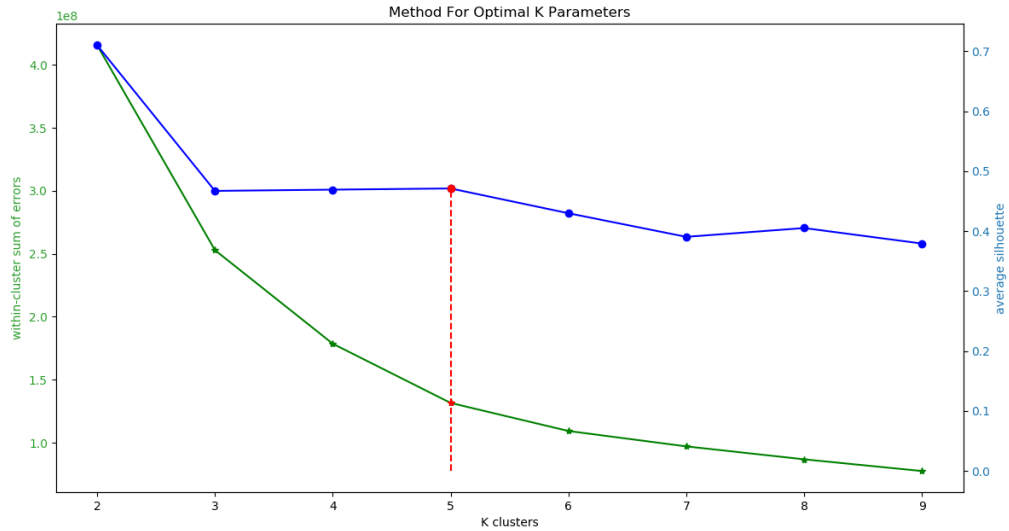


2011.jpg

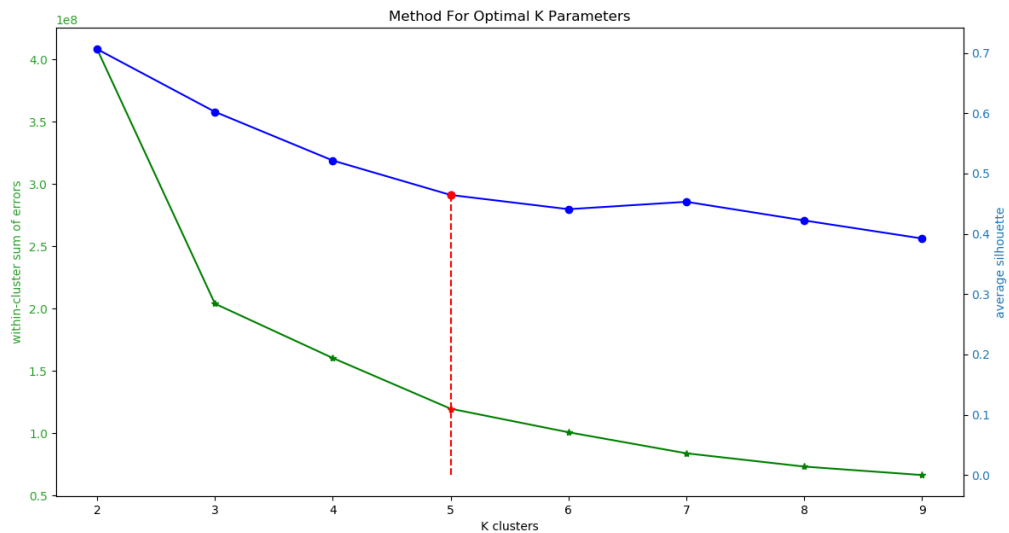
Hình 3.7: Ảnh vệ tinh khu vực hồ Chiquita Mar 1998-2011.

(Nguồn: [nasa](#))

Để rõ ràng hơn, tôi thực hiện tối ưu tham số K cho hình ảnh 2011.jpg sử dụng 2 phương pháp Elbow và Average Silhouette [3.2.2], kết quả được biểu diễn bằng đồ thị dưới đây:



Hình 3.8: Lựa chọn tham số K cho ảnh 2011.jpg.



Hình 3.9: Lựa chọn tham số K cho ảnh 1998.jpg.

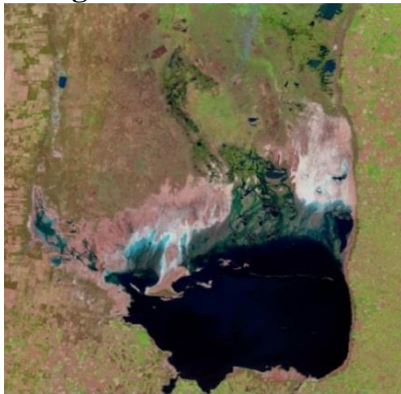
Khi số cụm K nằm trong khoảng [2,9], rõ ràng từ đồ thị cho thấy tại vị trí khuỷu tay (K=5) giá trị tổng lỗi trong cụm (wcss) sẽ giảm chậm dần theo phương pháp Elbow, và cũng tại vị trí đó giá trị Average Silhouette là lớn nhất. Khi đó K=5 sẽ là số cụm tối ưu nhất cần tìm.

Bảng 3.9: Kết quả lựa chọn tham số K clusters

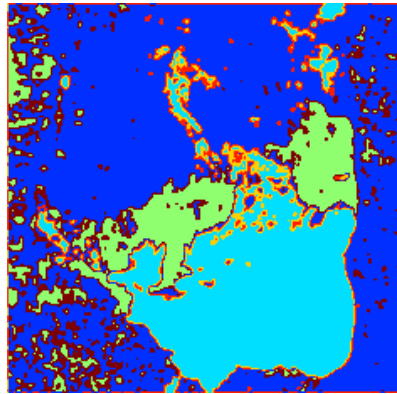
| Ảnh | K | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|------|-----------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1998 | wcss× 10 ⁸ | 4.0831 | 2.0403 | 1.6044 | 1.1976 | 1.0088 | 0.8401 | 0.7334 | 0.6655 |
| | silhouette | 0.7063 | 0.6023 | 0.5216 | 0.4644 | 0.4408 | 0.4531 | 0.4222 | 0.3923 |
| 2011 | wcss× 10 ⁸ | 4.1562 | 2.5291 | 1.7860 | 1.3163 | 1.0935 | 0.9713 | 0.8687 | 0.7756 |
| | silhouette | 0.7101 | 0.4671 | 0.4691 | 0.4711 | 0.4298 | 0.3904 | 0.4051 | 0.3793 |

Bảng 3.10: So sánh kết quả phương pháp đề xuất BK-means và thuật toán K-means

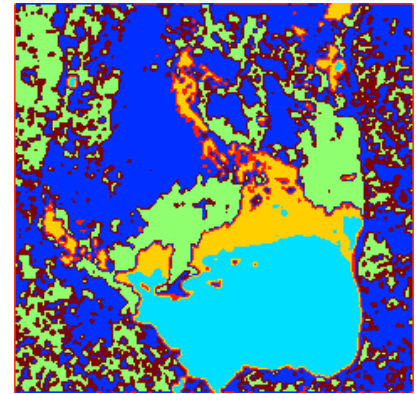
Ảnh gốc



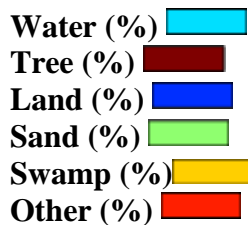
K-means



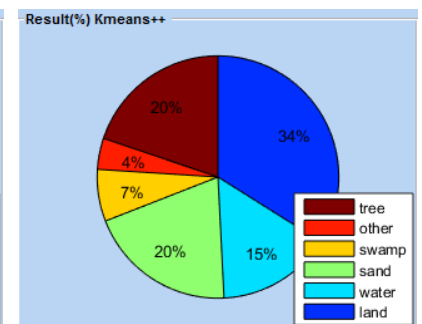
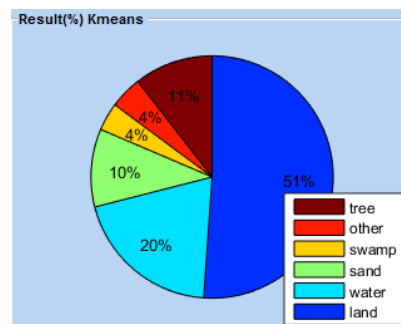
BK-means



Biểu đồ vùng

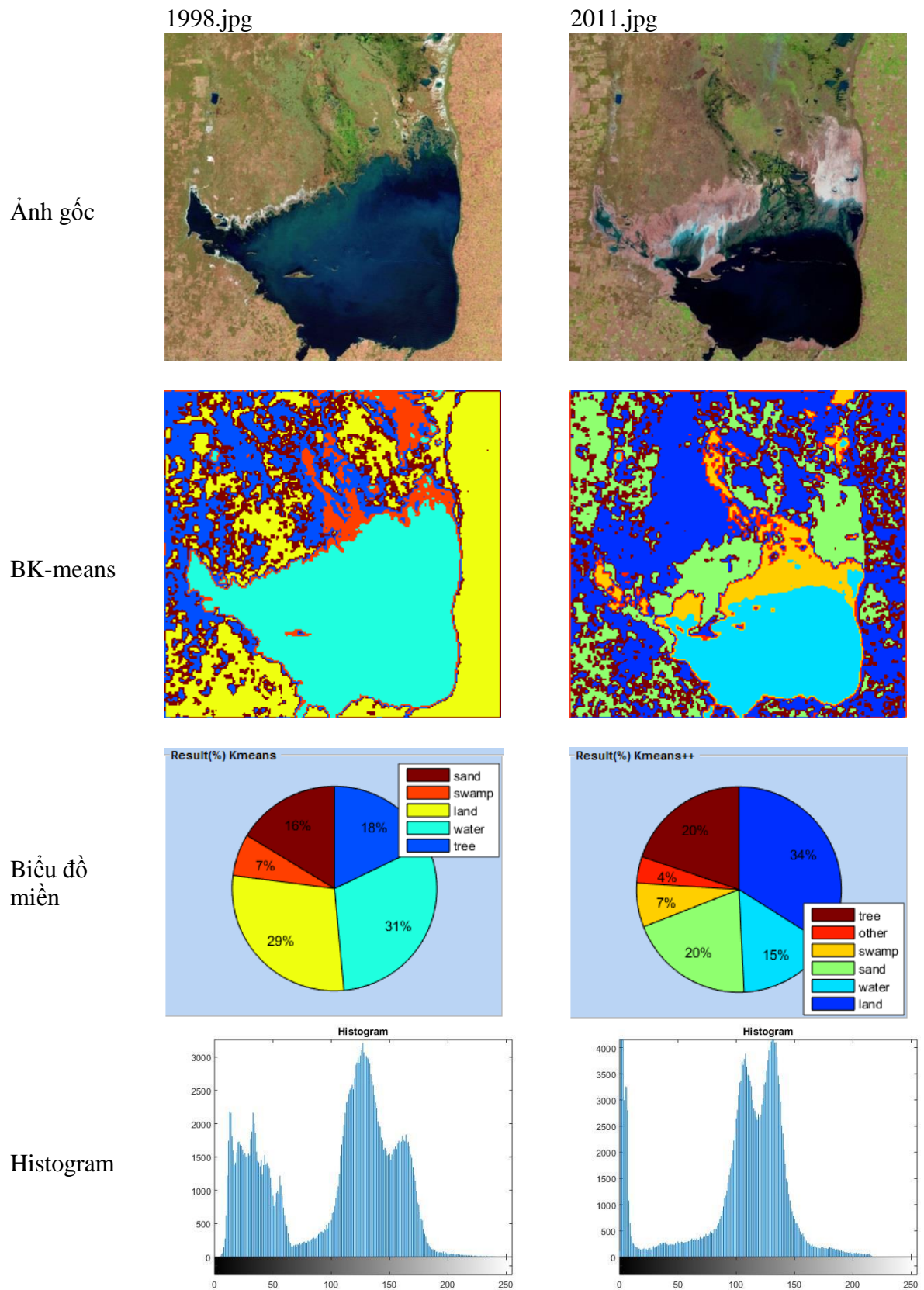


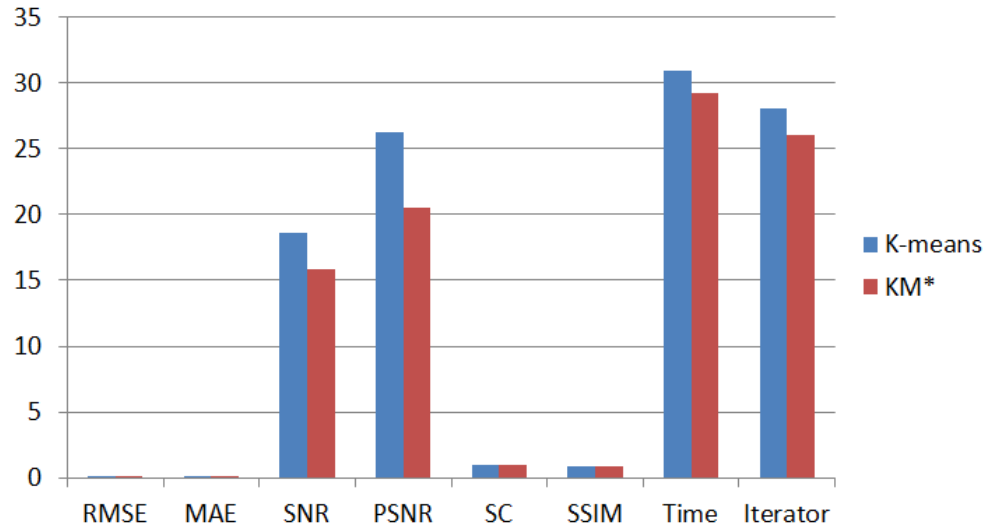
Time (s)
Iterations
RMSE
MAE
SNR
PSNR
SC
SSIM



| | | |
|------------|---------|---------|
| Water (%) | 20 | 15 |
| Tree (%) | 11 | 20 |
| Land (%) | 51 | 34 |
| Sand (%) | 10 | 20 |
| Swamp (%) | 4 | 7 |
| Other (%) | 4 | 4 |
| Time (s) | 30.8658 | 29.2071 |
| Iterations | 28 | 26 |
| RMSE | 0.0486 | 0.0944 |
| MAE | 0.0362 | 0.0735 |
| SNR | 18.5860 | 15.8679 |
| PSNR | 26.2638 | 20.5045 |
| SC | 0.9863 | 0.9768 |
| SSIM | 0.8550 | 0.8369 |

Bảng 3.11: Kết quả phát hiện thay đổi từ ảnh vệ tinh





Hình 3.10: Biểu đồ so sánh hiệu năng phương pháp BK-means và K-means.

Từ các số liệu đánh giá và kết quả trên, nhận thấy phương pháp được đề xuất cho kết quả khá tốt. Các hình ảnh được phân vùng bằng phương pháp đề xuất có độ chi tiết và chính xác cao hơn phương pháp K-means cơ bản thể hiện rõ nhất khi ảnh đầu vào là các hình ảnh bị nhiễu và có chất lượng thấp. Hơn nữa phương pháp đề xuất BK-means cho thời gian chạy thuật toán nhanh hơn và số vòng lặp cũng giảm đáng kể so với phương pháp K-means cơ bản.

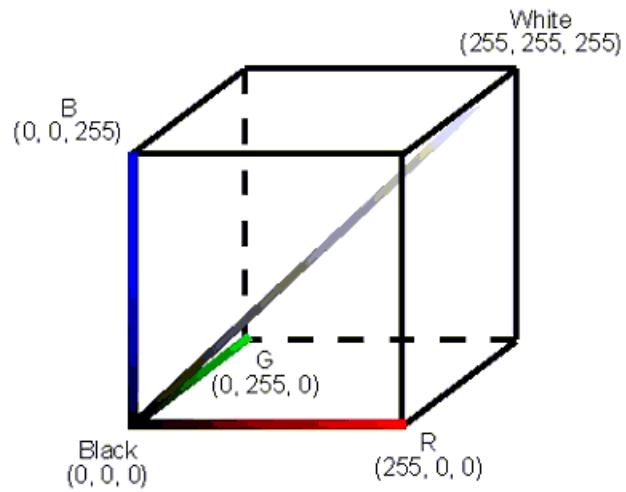
Chương 4. Phát hiện đám cháy trong ảnh và video sử dụng không gian màu YCbCr.

Hỏa hoạn gây thiệt hại lớn cho cuộc sống, tính mạng và tài sản của con người, do đó phát hiện cháy sớm là việc rất quan trọng. Đầu báo cháy, đầu báo khói và đầu báo nhiệt độ đã được sử dụng rộng rãi để bảo vệ tài sản và đưa ra cảnh báo về hỏa hoạn. Phương pháp truyền thống này dựa trên cảm biến và có nhiều nhược điểm: nó có độ truyền trễ, và chúng được áp dụng chủ yếu cho các khu vực trong nhà và không thể được sử dụng cho các khu vực ngoài trời để theo dõi và đưa ra cảnh báo ở những khu vực rộng lớn. Trong khi phát hiện cháy dựa trên thị giác có nhiều lợi thế: một khu vực rộng lớn có thể theo dõi, xác định vị trí chính xác của vụ cháy và cảnh báo dọc theo với camera giám sát. Dữ liệu hình ảnh được trích xuất từ video bởi camera giám sát. Video sẽ được liên tục cắt thành các frames-khung hình, và các frames này sẽ liên tục được đưa vào hệ thống xử lý ảnh để phát hiện vùng lửa. Khi hệ thống nhận diện được vùng lửa trong ảnh tại thời điểm đó video sẽ ngay lập tức được báo động có hỏa hoạn tại nơi khu vực đặt camera giám sát. Phương pháp này giúp tăng tỷ lệ báo cháy thật và giảm thiểu sự báo động giả từ cảm biến khói.

4.1. Biểu diễn dữ liệu

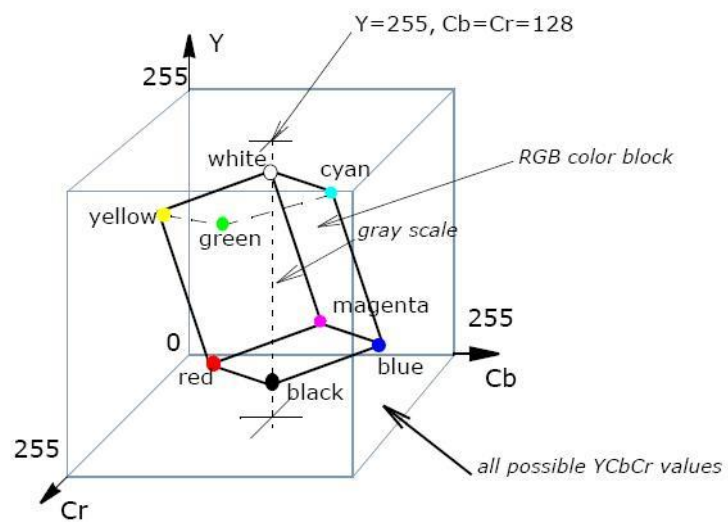
Sử dụng không gian màu RGB và YCbCr, ưu điểm khi sử dụng không gian màu YCbCr là nó có thể tách độ sáng hiệu quả hơn không gian màu RGB [19].

Một ảnh số trong không gian màu RGB có 3 mặt phẳng R, G, B tương ứng với 3 kênh màu. Các sự kết hợp giữa các mặt phẳng màu RGB cho phép các thiết bị hiển thị biểu diễn màu trong môi trường kỹ thuật số. Mỗi mặt phẳng màu được lượng hóa thành các mức rời rạc. Nói chung có 256 (8 bit cho mỗi kênh màu) lượng tử hóa được sử dụng cho mỗi mặt phẳng, ví dụ màu trắng được biểu thị bằng $(R, G, B) = (255, 255, 255)$ và màu đen được biểu thị bằng $(R, G, B) = (0, 0, 0)$. Một ảnh màu bao gồm nhiều pixel, trong đó mỗi pixel được biểu thị bằng vị trí không gian trong lưới hình chữ nhật (x, y) và vector màu $(R(x, y), G(x, y), B(x, y))$ tương ứng với vị trí không gian (x, y) là vị trí tọa độ của pixel [19].



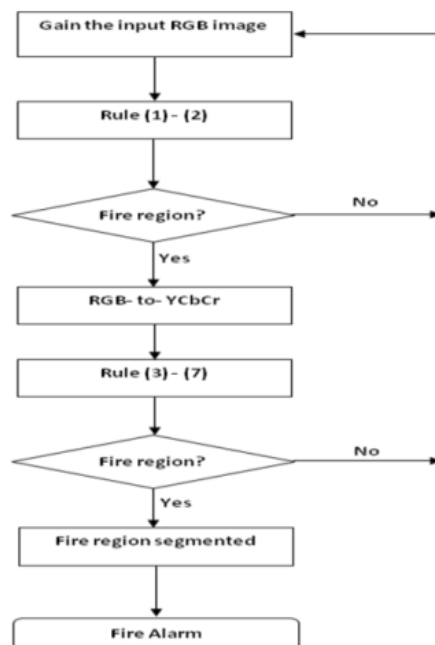
Hình 4.1: Không gian màu RGB.

(Nguồn: [rgb](#))



Hình 4.2: Không gian màu YCbCr [9].

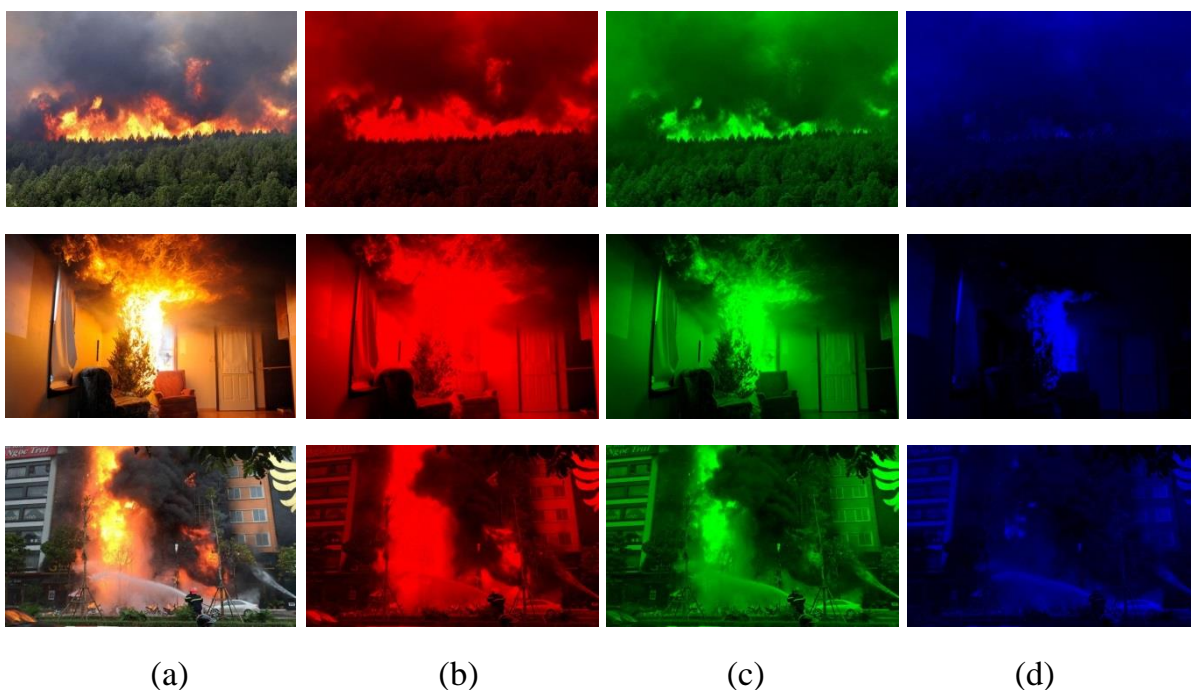
4.2. Xây dựng mô hình



Hình 4.3: Mô hình phát hiện đám cháy trong ảnh [19].

Quy tắc 1

Có thể nhận thấy tại các vùng cháy trong ảnh, kênh R có giá trị cường độ cao hơn kênh G, và kênh G có giá trị cường độ cao hơn giá trị cường độ kênh B.



Hình 4.4: Ảnh gốc – (a), tách các kênh màu R – (b), G – (c), B – (d).

$$R_{mean}(x, y) = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N R(x, y)$$

Công thức 4.1: Trung bình trên kênh màu Red [19].

$$G_{mean}(x, y) = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N G(x, y)$$

Công thức 4.2: Trung bình trên kênh màu Green [19].

$$B_{mean}(x, y) = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N B(x, y)$$

Công thức 4.3: Trung bình trên kênh màu Blue [19].

Bảng 4.1: Giá trị trung bình các kênh màu R, G, B của vùng lửa

| Hàng | Mean R | Mean G | Mean B |
|------|----------|---------|---------|
| 1 | 100.8567 | 87.3496 | 83.1776 |
| 2 | 152.0827 | 87.6628 | 35.8172 |
| 3 | 105.8824 | 81.5731 | 63.5662 |

Hình ảnh được phân vùng các điểm chớp cháy của nó như hình trên với kênh màu G. Sau đó chúng ta tính giá trị trung bình của các kênh màu R, G và B tại vị trí các vùng lửa được phân nhận dạng từ ảnh gốc. Vì vậy, đối với pixel tại vị trí tọa độ (x, y) nếu là pixel lửa thì quy tắc bên sau đây phải được thỏa mãn:

$$R_1(x, y) = \begin{cases} 1, & \text{if } R(x, y) > G(x, y) > B(x, y) \\ 0, & \text{otherwise} \end{cases}$$

Công thức 4.4: Quy tắc 1 [19].

Quy tắc 2

Xác định một số giá trị ngưỡng cho điểm ảnh lửa:

$$R_2(x, y) = \begin{cases} 1, & \text{if } (R(x, y) > 190) \cap (G(x, y) > 100) \cap (B(x, y) < 140) \\ 0, & \text{otherwise} \end{cases}$$

Công thức 4.5: Quy tắc 2 [19].

Khi hình ảnh được chuyển đổi từ không gian màu RGB sang YCbCr, cường độ và không gian màu sẽ dễ dàng phân biệt. Điều này giúp việc nhận dạng vùng lửa dễ dàng trong không gian màu YCbCr.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

Công thức 4.6: Chuyển đổi không gian màu RGB sang YCbCr [19].

Trong đó: Y là độ sáng, Cb và Cr là các thành phần màu Blue và màu Red .

Giá trị trung bình của ba thành phần Y, Cb và Cr trên toàn bộ bức ảnh được biểu thị bằng Ymean, Cbmean và Crmean tương ứng được tính như sau:

$$Y_{mean}(x, y) = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N Y(x, y)$$

Công thức 4.7: Trung bình kênh độ sáng Y [19].

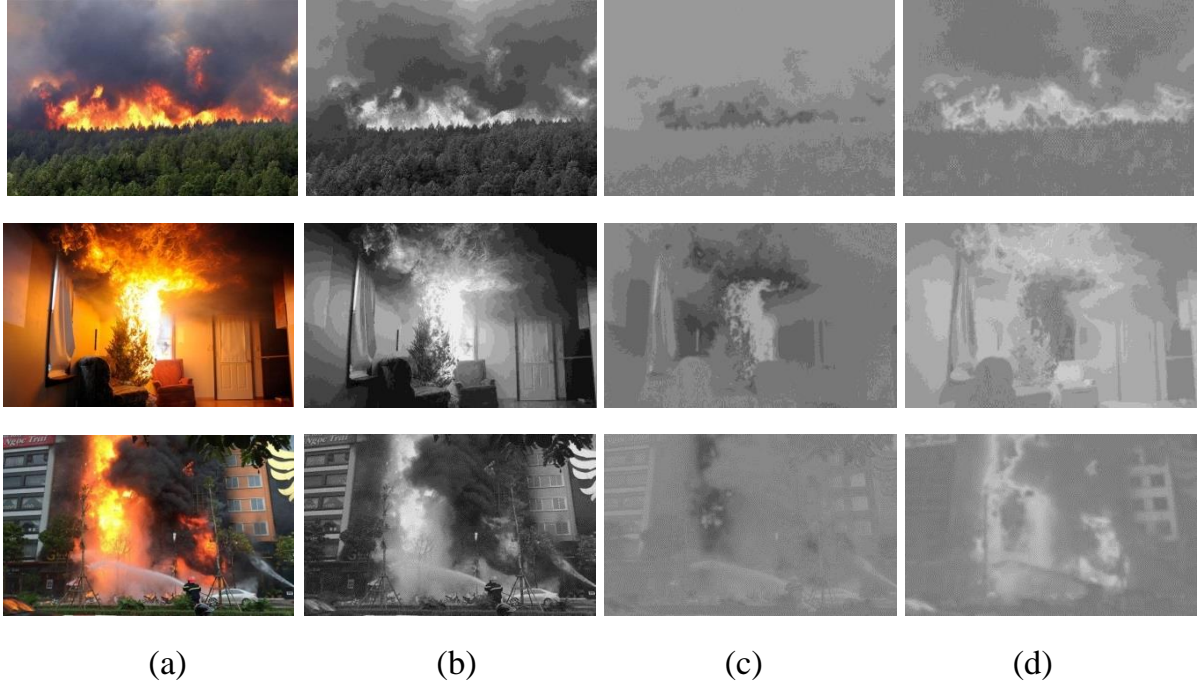
$$Cb_{mean}(x, y) = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N Cb(x, y)$$

Công thức 4.8: Trung bình kênh màu Cb [19].

$$Cr_{mean}(x, y) = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N Cr(x, y)$$

Công thức 4.9: Trung bình kênh màu Cr [19].

Trong đó: (x, y) là vị trí tọa độ của pixel, M và N kích thước của ảnh.



Hình 4.5: Ảnh gốc – (a), tách các kênh màu Y – (b), Cb – (c), Cr – (d).

Bảng 4.2: Giá trị trung bình các kênh Y, Cb, Cr của vùng lửa

| Hàng | Mean Y | Mean Cb | Mean Cr |
|------|----------|----------|----------|
| 1 | 94.0257 | 139.1689 | 134.2668 |
| 2 | 102.7483 | 118.2278 | 160.0604 |
| 3 | 90.5568 | 132.2079 | 139.9889 |

Quy tắc 3 và 4

$$R_3(x, y) = \begin{cases} 1, & \text{if } Y(x, y) > Cb(x, y) \\ 0, & \text{otherwise} \end{cases}$$

Công thức 4.10: Quy tắc 3 [19][25].

$$R_4(x, y) = \begin{cases} 1, & \text{if } Cr(x, y) > Cb(x, y) \\ 0, & \text{otherwise} \end{cases}$$

Công thức 4.11: Quy tắc 4 [19][25].

Quy tắc 5

Vùng ngọn lửa nói chung là vùng sáng nhất trong bối cảnh nhìn thấy của bức ảnh hay camera. Giá trị trung bình trên toàn bộ tấm ảnh của ba kênh Ymean, Cbmean và Crmean chứa thông tin có giá trị. Đối với vùng ngọn lửa, giá trị của thành phần Y là

lớn hơn thành phần Y trung bình trên toàn bộ bức ảnh. Trong khi giá trị của thành phần Cb tại vùng lửa nhỏ hơn giá trị Cb trung bình của trên toàn bộ bức ảnh. Hơn nữa, thành phần Cr của vùng lửa lớn hơn Cr trung bình trên toàn bộ bức ảnh [21][25]:

$$R_5(x, y) = \begin{cases} 1, & \text{if } (Y(x, y) \geq Y_{mean}(x, y)) \cap (Cb(x, y) \leq Cb_{mean}(x, y)) \cap (Cr(x, y) \leq Cr_{mean}(x, y)) \\ 0, & \text{otherwise} \end{cases}$$

Công thức 4.12: Quy tắc 5 [19][25].

Trong đó: $R_5(x, y)$ chỉ ra rằng bất kỳ điểm ảnh nào thỏa mãn điều kiện được đưa ra trong phương trình trên đều có thể được coi là điểm ảnh cháy.

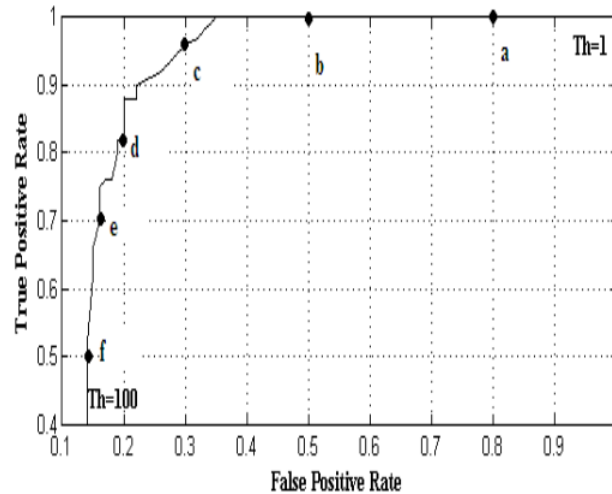
Quy tắc 6

Có sự khác biệt đáng kể giữa Cb và các thành phần Cr của các điểm ảnh cháy. Đối với pixel lửa, thành phần Cb chủ yếu là "đen" (cường độ thấp hơn) trong khi thành phần Cr chủ yếu là "trắng" (cường độ cao hơn) [19][25]. Thực tế này có thể được dịch sang một quy tắc khác như sau:

$$R_6(x, y) = \begin{cases} 1, & \text{if } |Cb(x, y) - Cr(x, y)| \geq Th \\ 0, & \text{otherwise} \end{cases}$$

Công thức 4.13: Quy tắc 6 [19][25].

Giá trị của Th được chọn sao cho tỷ lệ phát hiện đám cháy (tỷ lệ TPR hay Recall) vượt quá 70% và tỷ lệ báo động giả (tỷ lệ FPR) nhỏ hơn 30% (điểm e) tương ứng với Th = 75 [19].



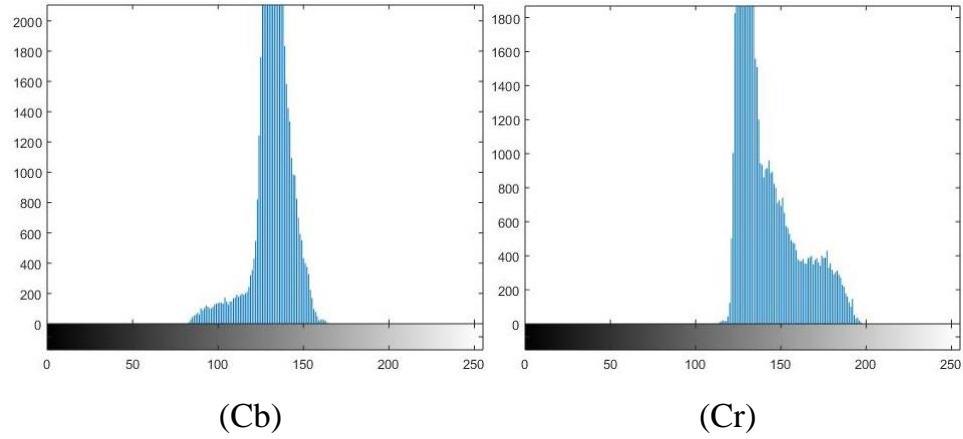
Hình 4.6: Giá trị ngưỡng (điểm e) cho tỷ lệ phát hiện cháy > 70% và tỷ lệ báo cháy giả < 20% [19][25].

Quy tắc 7

Chúng ta đã xác định một số giá trị ngưỡng cho điểm ảnh lửa. Ta chỉ xét giá trị ngưỡng cho 2 kênh màu Cb và Cr, không xét kênh độ sáng Y vì nó là thành phần độ sáng và nó phụ thuộc vào điều kiện chiếu sáng.

$$R_7(x, y) = \begin{cases} 1, & \text{if } (Cb(x, y) \leq 150) \cap (Cr(x, y) \geq 120) \\ 0, & \text{otherwise} \end{cases}$$

Công thức 4.14: Quy tắc 7 [19][25].



Hình 4.7: Biểu đồ vùng cháy cho kênh Cb và Cr.

Một điểm ảnh là pixel vùng lửa nếu thỏa mãn đồng thời 7 quy tắc trên.

Bảng 4.3: Quy tắc xác định điểm ảnh cháy [19][25]

| Color space | Rules |
|-------------|--|
| RGB | 1) $R > G > B$ |
| | 2) $R > R_{mean} \cap G > G_{mean} \cap B > B_{mean}$ |
| YCbCr | 3) $Y(x, y) \geq Cb(x, y)$ |
| | 4) $Cr(x, y) \geq Cb(x, y)$ |
| | 5) $Y(x, y) \geq Y_{mean} \cap Cb(x, y) \leq Cb_{mean} \cap Cr(x, y) \geq Cr_{mean}$ |
| | 6) $ Cb(x, y) - Cr(x, y) \geq Th, (Th = 75)$ |
| | 7) $(Cb(x, y) \leq 150) \cap (Cr(x, y) \geq 120)$ |

4.3. Kết quả thử nghiệm và đánh giá

4.3.1. Kết quả

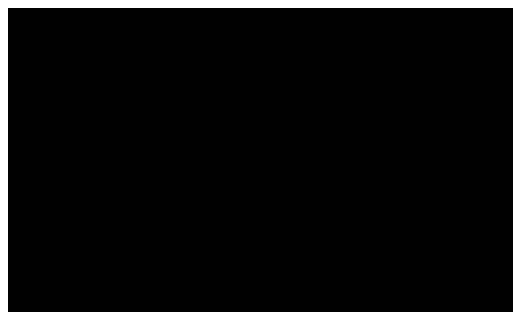
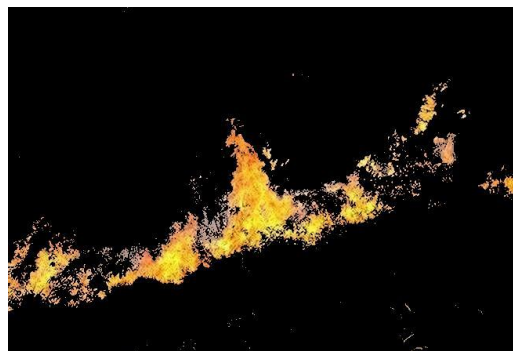
Tập dữ liệu kiểm thử gồm 2 tập con, tập *test_fire* chứa 2400 ảnh có đám cháy được gán nhãn 1 và tập *not_fire* chứa 2931 ảnh không có đám cháy được gán nhãn 1.

Bảng 4.4: Kết quả phát hiện đám cháy trong ảnh

Origin Image



Fire Detection



4.3.2. Đánh giá

Khi xây dựng một mô hình Machine Learning, chúng ta cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không và để so sánh khả năng của các mô hình.

Trong bài toán phát hiện đám cháy, ta có thể quy nó về bài toán phân loại ảnh (image classification) thành 2 lớp là *fire* và *not-fire*.

Accuracy độ chính xác là tỉ lệ giữa số ảnh được phân loại đúng và tổng số ảnh có trong tập dữ liệu kiểm thử. Ma trận nhầm lẫn chưa chuẩn hóa - *unnormalized confusion matrix*:

Bảng 4.5: Unnormalized confusion matrix

| Total=5331 | Predicted as fire | Predicted as not-fire | |
|------------------|-------------------|-----------------------|------|
| Actual: fire | 1689 | 711 | 2400 |
| Actual: not-fire | 566 | 2365 | 2931 |

Bảng 4.6: Normalized confusion matrix

| | Predicted as fire | Predicted as not-fire |
|------------------|-------------------|-----------------------|
| Actual: fire | 0.7038 | 0.2962 |
| Actual: not-fire | 0.1931 | 0.8069 |

Trong bài toán xác định có đám cháy hay không hay không thì việc không bị sót (Miss Detection Rate) quan trọng hơn là việc báo động nhầm (False Alarm Rate). Trong hai lớp dữ liệu *fire* và *not-fire* có một lớp nghiêm trọng hơn lớp kia và cần phải được dự đoán chính xác, đó là lớp *fire*. Ta định nghĩa lớp *fire* cần được xác định đúng là lớp Positive (P-dương tính), lớp còn lại *not-fire* gọi là Negative (N-âm tính). Ta định nghĩa True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) dựa trên *unnormalized confusion matrix* chưa chuẩn hoá như sau [1]:

Bảng 4.7: Kết quả phân loại ảnh chứa đám cháy

| | Predicted as Positive | Predicted as Negative |
|------------------|---------------------------|---------------------------|
| Actual: Positive | True Positive (TP) = 1689 | False Negative (FN) = 711 |
| Actual: Negative | False Positive (FP) = 566 | True Negative (TN) = 2365 |

True Positive (TP): số lượng điểm của lớp positive được phân loại đúng là positive.

True Negative (TN): số lượng điểm của lớp negative được phân loại đúng là negative.

False Positive (FP): số lượng điểm của lớp negative bị phân loại nhầm thành positive.

False Negative (FN): số lượng điểm của lớp positive bị phân loại nhầm thành negative.

$$RI = \frac{TP + TN}{TP + FN + FP + TN}$$

Công thức 4.15: Tính độ chính xác Accuracy trên toàn bộ các lớp [17].

$$RI = 76.33\%$$

$$\text{Error} = 1 - RI = 23.67\%$$

Ở đây ta chỉ quan tâm đến TPR, FNR, FPR, TNR (R - Rate) dựa trên normalized confusion matrix như sau:

Bảng 4.8: Tỷ lệ phân loại các lớp dựa trên normalized confusion matrix [1]

| | Predicted as Positive | Predicted as Negative |
|------------------|------------------------------|------------------------------|
| Actual: Positive | $TPR = \frac{TP}{(TP + FN)}$ | $FNR = \frac{FN}{(TP + FN)}$ |
| Actual: Negative | $FPR = \frac{FP}{(FP + TN)}$ | $TNR = \frac{TN}{(FP + TN)}$ |

True positive rate (TPR), false negative rate (FNR), false positive rate (FPR), true negative rate (TNR):

FPR—False Positive Rate còn được gọi là False Alarm Rate (tỉ lệ báo động nhầm), FNR—False Negative Rate còn được gọi là Miss Detection Rate (tỉ lệ bỏ sót). Trong bài toán phát hiện đám cháy, thà báo nhầm còn hơn bỏ sót, tức là ta có thể chấp nhận False Alarm Rate cao để đạt được Miss Detection Rate thấp [1].

Receiver Operating Characteristic curve - ROC

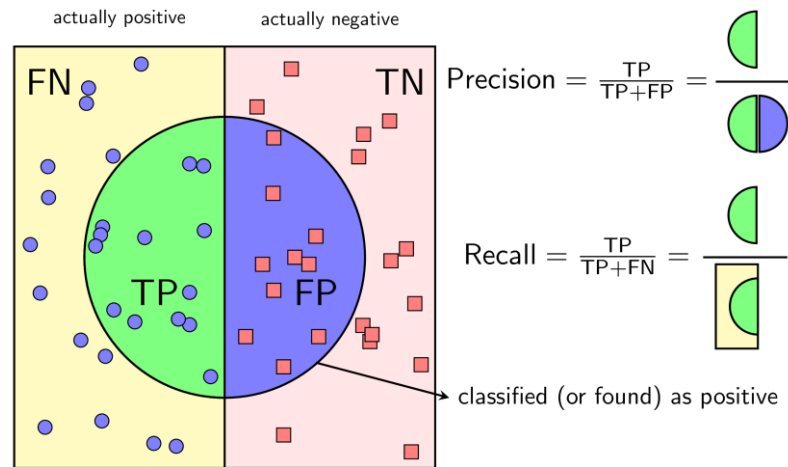
Trong bài toán phân loại ảnh thành 2 lớp là có đám cháy và không có đám cháy, việc tăng hay giảm FNR, FPR có thể được thực hiện bằng việc thay đổi một ngưỡng Th (threshold). Điều chỉnh ngưỡng Th để tăng mức độ báo nhầm (FPR) để giảm mức độ bỏ sót (FNR), tăng FNR đồng nghĩa với việc giảm TPR vì tổng của chúng luôn bằng 1. Tỉ lệ các điểm được phân loại là Positive sẽ tăng lên sẽ kéo theo cả False Positive Rate và True Positive Rate cùng tăng lên (cột thứ nhất trong ma trận tăng lên). Từ đây suy ra cả FNR và TNR đều giảm [1].

Ngược lại, nếu ta muốn bỏ sót còn hơn báo nhầm, tất nhiên là ở mức độ nào đó, ta cần tăng threshold lên một số lớn hơn 50. Khi đó, hầu hết các điểm dữ liệu sẽ được dự đoán thuộc lớp 0, tức Negative, và cả TNF và FNR đều tăng lên, kéo theo TPR và FPR giảm xuống [1].

Như vậy, ứng với mỗi giá trị của threshold, ta sẽ thu được một cặp (FPR, TPR). Biểu diễn các điểm (FPR, TPR) trên đồ thị khi thay đổi threshold từ 0 tới 100 ta sẽ thu được một đường được gọi là Receiver Operating Characteristic curve hay ROC curve [19].

Bảng 4.9: Kết quả tỷ lệ TPR, FPR, TNR, FNR

| | Predicted as Positive | Predicted as Negative |
|-------------------------|-----------------------|-----------------------|
| Actual: Positive | TPR = 0.7038 | FNR = 0.2962 |
| Actual: Negative | FPR = 0.1931 | TNR = 0.8069 |



Hình 4.8: Phương pháp tính Precision và Recall cho bài toán phân loại nhị phân [1].

Với một cách xác định một lớp là *positive*, Precision được định nghĩa là tỉ lệ số điểm *true positive* trong số những điểm được phân loại là *positive* (TP + FP).

$$\text{Precision} = \frac{TP}{TP + FP}$$

Công thức 4.16: Tính Precision [1].

Recall được định nghĩa là tỉ lệ số điểm *true positive* trong số những điểm thực sự là *positive* (TP + FN).

$$\text{Recall} = \frac{TP}{TP + FN}$$

Công thức 4.17: Tính Recall [1].

Precision cao đồng nghĩa với việc độ chính xác của việc phát hiện ảnh đám cháy là cao. Recall cao đồng nghĩa với việc True Positive Rate cao, tức tỉ lệ bỏ sót các ảnh thực sự *positive* là thấp [1].

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Công thức 4.18: Tính F_1 – score [1].

F_1 có giá trị nằm trong nửa khoảng $(0,1]$. F_1 càng cao, bộ phân lớp càng tốt. Khi cả recall và precision đều bằng 1 (tốt nhất có thể), $F_1 = 1$.

$$F_\beta = (1 + \beta^2) \frac{precision \times recall}{\beta^2 \times precision + recall}$$

Công thức 4.19: Tính F_β [1].

F_1 chính là một trường hợp đặc biệt của F_β khi $\beta = 1$. Khi $\beta > 1$, recall được coi trọng hơn precision, còn khi $\beta < 1$, precision được coi trọng hơn. Hai đại lượng β thường được sử dụng là $\beta = 2$ và $\beta = 0.5$ [1].

Bảng 4.10: Tỷ lệ Precision, Recall và F_1 cho bài toán phân loại ảnh đám cháy

| | |
|-------------------------|--------|
| Precision | 0.7490 |
| Recall | 0.7038 |
| F_1 | 0.7257 |

Chương 5. Kết luận và hướng phát triển

5.1. Kết luận

Dù phải trải qua một số khó khăn, vướng mắc nhưng nhờ sự hướng dẫn tận tình của các thầy, đồ án đã hoàn thành và đạt được những mục tiêu cơ bản đặt ra khi thực hiện:

- Tìm hiểu và nghiên cứu về Học không giám sát trong học máy.
- Cài đặt và cải tiến mô hình BK-means dựa trên thuật toán phân cụm K-means và áp dụng cho bài toán phân vùng ảnh và phát hiện đám cháy trong ảnh sử dụng không gian màu YCbCr.
- Trong đồ án này em đã đề xuất phương pháp phân vùng ảnh BK-means dựa trên thuật toán phân cụm K-means bằng cách sử dụng không gian màu $L^*a^*b^*$ phù hợp cho kết quả tốt hơn. Thực hiện việc khởi tạo trung tâm cụm ban đầu dựa trên mật độ, thay vì random như trong thuật toán K-means nhằm giảm thiểu số bước lặp, giúp giảm thời gian chạy của thuật toán.
- Hoàn thiện đồ án và báo cáo trong thời gian đã định.

Bên cạnh những thành công đã đạt được thì vẫn còn một số tồn tại sau:

- Tập dữ liệu còn đơn giản, chưa đủ lớn.
- Giao diện chương trình còn sơ sài, chưa đầy đủ chức năng. Tốc độ nhận diện đám cháy qua video realtime còn chậm phản hồi, độ chính xác chưa cao.
- Chưa thể phát hiện khói trong đám cháy khi chưa có lửa, những hình ảnh mờ và bị nhiễu.

5.2. Hướng phát triển

Mặc dù các bài toán đặt ra đã giải quyết được vấn đề, phương pháp đề xuất BK-means cải thiện hiệu quả thấy rõ so với phương pháp K-means cơ bản, nhưng vẫn còn khá đơn giản chưa có tính ứng dụng thực tiễn, chỉ ở quy mô nghiên cứu. Vì vậy, trong thời gian tới, hy vọng có thể áp dụng thuật toán với bộ dữ liệu phức tạp hơn, và sử dụng các phương pháp mới tiên tiến hơn kết hợp với các thuật toán trong Deep Learning để nâng cao khả năng nhận diện và phân vùng ảnh. Đặc biệt là có thể triển khai và tận dụng tối đa khả năng tính toán của GPU. Trong thời gian làm đồ án, do còn nhiều thiếu sót, rất mong các thầy cô cũng như các chuyên gia thông cảm và góp ý để đề tài có thể hoàn thiện một cách tốt nhất. Trong thời gian làm đồ án, do còn nhiều thiếu sót, rất mong các thầy cô cũng như các chuyên gia thông cảm và góp ý để đề tài có thể hoàn thiện một cách tốt nhất.

Dưới đây là một số hướng phát triển tiếp của đồ án.

Phát hiện vùng khói trong ảnh

Tương tự như phát hiện vùng lửa, ta có thể mô hình hóa các điểm ảnh khói. Nhưng điểm ảnh khói không hiển thị các đặc điểm màu sắc như pixel lửa. Lúc đầu, khi nhiệt độ khói thấp, dự kiến khói sẽ hiển thị từ màu trắng-xanh đến trắng. Về sau nhiệt độ của khói tăng lên và màu chuyển từ màu xám đen đến màu đen. Vì vậy, tôi có định hướng ý tưởng xây dựng các điểm ảnh khói như sau:

$$|R(x, y) - G(x, y)| \leq Th$$

$$|G(x, y) - B(x, y)| \leq Th$$

$$|B(x, y) - R(x, y)| \leq Th$$

Công thức 5.1: Đề xuất phát hiện đám khói trong ảnh [24].

Trong đó Th là ngưỡng toàn cầu dao động từ 15 đến 25, các điểm ảnh khói nên có tương tự cường độ trong các kênh màu RGB của chúng. Vì thông tin khói sẽ được sử dụng cho hệ thống phát hiện cháy sớm nhất, nên phát hiện các mẫu khói khi khói có nhiệt độ thấp. Đây là trường hợp, mà các mẫu khói có màu từ trắng-xanh đến trắng, có nghĩa là độ bão hòa của màu sắc nên càng thấp càng tốt. Sử dụng ý tưởng này, quy tắc sau được sử dụng khi không gian màu HSV được sử dụng [24]:

$$S(x, y) \leq 0.1$$

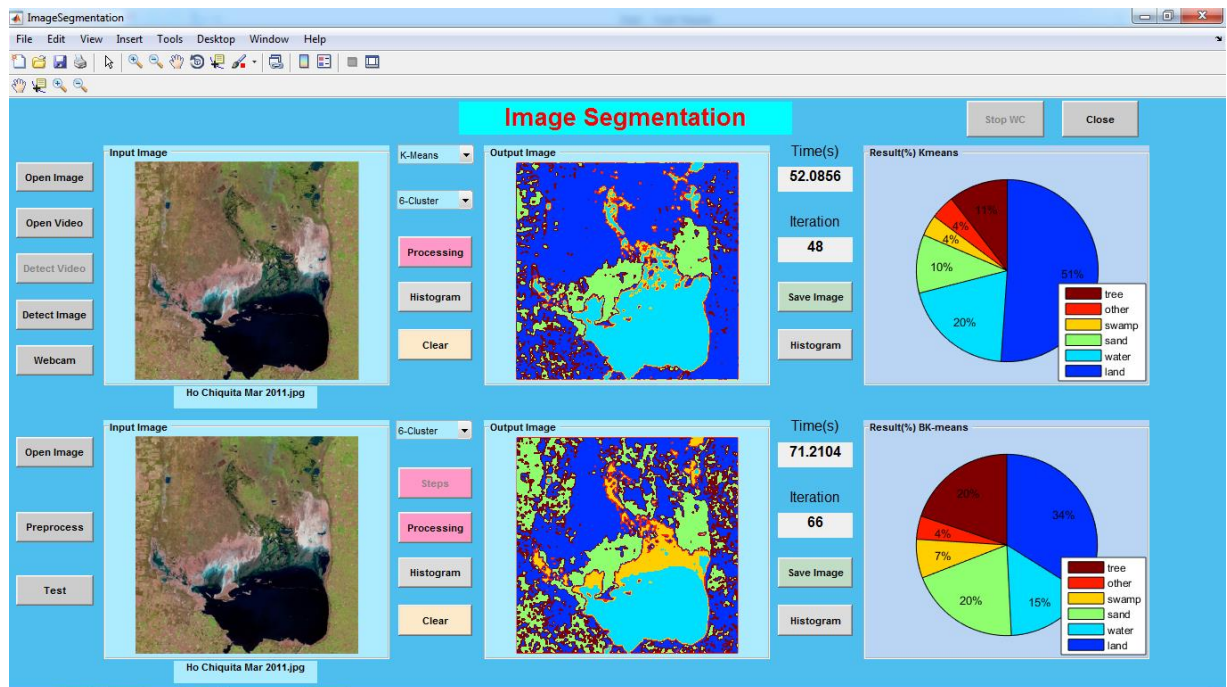
Phát hiện chuyển động của lửa và khói trong video

Để phân biệt được khói và các hình ảnh tương tự như sương mù hay mây trời, phân biệt lửa và các hình ảnh có màu sắc giống lửa như ánh đèn công suất lớn...ta sẽ cần phát hiện sự chuyển động của khói và lửa trong video đám cháy.

Phụ Lục

Hướng dẫn chạy chương trình trên Matlab

1. Giải nén project “ImageSegmentation.zip”.
2. Ta được thư mục ImageSegmentation chứa các thư mục và file sau:
 - a. Thư mục “input_images”: chứa các file ảnh đầu vào để xử lý ảnh cho bài toán phân vùng ảnh và phát hiện bất thường trong giám sát rừng.
 - b. Thư mục “dataset” là thư mục chứa dữ liệu kiểm thử cho bài toán phát hiện đám cháy trong ảnh.
 - c. Thư mục “python” chứa code phần chương trình chạy bằng python gồm 2 file “bkmeans.py” và “kmeans_sklearn.py” và các file ảnh đầu vào phải được đặt trong cùng 1 thư mục. Để chạy chương trình, yêu cầu máy tính phải được cài đặt sẵn python v2.7 hoặc mới hơn. Trong thư mục hiện tại mở cửa sổ cmd và chạy lệnh “python bkmeans.py” hoặc “python kmeans_sklearn.py” để chạy code chương trình sử dụng thư viện sklearn.
 - d. Thư mục “video” chứa video đầu vào, video đầu ra và các thư mục chứa ảnh kết quả từng bước xử lý ảnh cho bài toán phát hiện đám cháy trong ảnh, video có sẵn, hoặc video thời gian thực.
 - e. Thư mục “report” chứa báo cáo đồ án, hướng dẫn cài đặt chương trình.
 - f. File “ImageSegmentation.m” chứa code chương trình chính.
 - g. File “ImageSegmentation.fig” là 1 file Matlab Figure thiết kế giao diện chương trình trên matlab, có thể click vào file này để chạy luôn chương trình trong cửa sổ matlab.
 - h. Các files “realVideo.m”, “test_fire.m”, “colorspace.m”, “fire.m” chứa các hàm xử lý ảnh và video.
3. Trong thư mục “ImageSegmentation”, mở file “ImageSegmentation.m” bằng phần mềm Matlab, yêu cầu máy tính phải được cài sẵn phần mềm matlab phiên bản 2015a hoặc cao hơn.
4. Trên giao diện matlab chọn tab EDITOR, click nút RUN (f5) để chạy chương trình. Hoặc double click vào file “ImageSegmentation.fig” để khởi chạy chương trình.



Hình 0.1: Giao diện chương trình.

Trong đồ án, các kết quả đạt được khi thực hiện trên máy tính Lenovo Thinkpad X230, Intel® Core™ i5 CPU 3320M @ 2.6GHz × 4, memory 8 GiB, graphics Intel®HD Graphics, trên môi trường HĐH Windows 7-64bit.

Phần mềm sử dụng: Matlab v2015a, PyCharm 2018.2 với môi trường Python 2.7, thư viện scikit-learn.

Tài liệu tham khảo

- [1] Vũ Hữu Tiệp, Machine Learning cơ bản, Nhà xuất bản khoa học và kỹ thuật, 2018.
- [2] ZhongwuWang, John R.Jensen, JunghoIm., An automatic region-based image segmentation algorithm for remote sensing applications, Environmental Modelling & Software, Elsevier Science Publishers, Volume 25, Issue 10, October 2010, Pages 1149-1165.
- [3] Julián Calderón González & Òscar Daniel Carmona Salazar., Image Enhancement with Matlab Algorithms, Blekinge Institute of Technology Department of Applied Signal Processing SE-371 79, Karlskrona Swede.
- [4] Sona Kajla & Rajiv Bansal., Efficient Improved K means Clustering for Image Segmentation, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2016.
- [5] Ankit Bisla, 2012UCP1684 Peeyush Yadav, 2012UCP1687., Image Segmentation Using K-means Clustering Algorithm, Machine Learning Project.
- [6] S. Rajkumar & G. Malathi.,A Comparative Analysis on Image Quality Assessment for Real Time Satellite Images, School of Computing Science and Engineering, VIT University, Chennai - 600127, Tamil Nadu, India., Indian Journal of Science and Technology, Vol. 9 (34), September 2016.
- [7] Carl Sundelius., Deep Fusion of Imaging Modalities for Semantic Segmentation of Satellite Imagery, Master of Science Thesis in Electrical Engineering Department of Electrical Engineering, Linköping University, 2017.
- [8] Zhensong Chen, Zhiquan Qi, Fan Meng, Limeng Cui & Yong Shi., Image Segmentation via Improving Clustering Algorithms with Density and Distance, Information Technology and Quantitative Management (ITQM 2015), Elsevier Science Publishers, 55: 1015-1022, 2015.
- [9] Neethu John, Amitha Viswanath, Sowmya V and Soman K P., Analysis of Various Color Space Models onEffective Single Image Super Resolution, August 2016.

- [10] Gunjan Mathur, Hemant Purohit., Performance Analysis of Color Image Segmentation using K- Means Clustering Algorithm in Different Color Spaces, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), Volume 4, Issue 6, Ver. III (Nov - Dec. 2014).
- [11] Mohammad Naved Qureshi & Mohd Vasim Ahamad., An Improved Method for Image Segmentation Using K-Means Clustering with Neutrosophic Logic, International Conference on Computational Intelligence and Data Science (ICCIDS 2018), Elsevier Science Publishers, 132: 534-540, 2018.
- [12] Ganesan P & Dr V.Rajini., Application of Modified K-Means Clustering Algorithm for Satellite Image Segmentation based on Color Information, International Journal of Advanced Research in Computer Science, Vol. 4, No.5, May 2013.
- [13] Mohsen Ghanea, Payman Moallem & Mehdi Momeni., Automatic building extraction in dense urban areas through GeoEye multispectral imagery, International Journal of Remote Sensing, 35:13, 5094-5119, 2014.
- [14] Sarthak panda., Color Image Segmentation Using K-means Clustering and Thresholding Technique, International Journal of Engineering Science and Computing, Institute of technical education and research, Bhubaneswar, Odisha, 2015.
- [15] Ajay Kumar & Shishir Kumar., Color Image Segmentation via Improved K-Means Algorithm, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 3, 2016.
- [16] B.Basavaprasad, Ravindra S.Hegadi., Color Image Segmentation Using Adaptive GrowCut Method, International Conference on Advanced Computing Technologies and Applications (ICACTA 2015), Elsevier Science Publishers, 45: 328-335, 2015.
- [17] S. M. Aqil Burney, Humera Tariq., K-Means Cluster Analysis for Image Segmentation, International Journal of Computer Applications (0975 – 8887) Volume 96– No.4, June 2014.
- [18] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze., Introduction to Information Retrieval, Cambridge University Press, 2009.

- [19] Vipin V, Image Processing Based Forest Fire Detection, International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 2, February 2012.
- [20] A. Enis Çetin, Kosmas Dimitropoulos, Benedict Gouverneur, Nikos Grammalidis, Osman Günay, Y. Hakan Habiboglu, B. Ugur Töreyn, Steven Verstockt., Video fire detection – Review, Digital Signal Processing, Elsevier Science Publishers, 23 (2013) 1827–1843.
- [21] Viktor Tuba, Romana Capor-Hrosik, Eva Tuba., Forest Fires Detection in Digital Images Based on Color Features, International Journal of Environmental Science, John Naisbitt University, Volume 2, 2017
- [22] Rui Chen, Yuanyuan Luo, Mohanmad Reza Alsharif., Forest Fire Detection Algorithm Based on Digital Image, Department of Information Engineering, Faculty of Engineering, University of the Ryukyus, Okinawa, Japan, Journal Of Software, Vol. 8, No. 8, August 2013.
- [23] Ku. R.A.Agrawal, Prof. S.T.Khandare., Fire Detection Using Image Processing, International Journal of Advanced Engineering and Global Technology, Vol-03, Issue-12, December 2015.
- [24] Sarkawas Nagma Mehaboobsab, Azad Kulsum Alamgeer, Mokashi Muzna Noor Ahmed, Ansari Mohd Salim Badruddin., Fire Detection System Using Image Processing, Department of Electronics and Telecommunication Engineering Anjuman-I-Islam's Kalsekar Technical Campus, 2015-2016.
- [25] Norsyahirah Izzati binti Zaidi, Nor Anis Aneza binti Lokman, Mohd Razali bin Daud, Hendriyawan Achmad and Khor Ai Chia., Fire Recognition Using RGB and YCbCr Color Space, ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 21, November 2015.
- [26] J. B. MacQueen, "Some methods for classification and analysis of multivariate observation", In: Le Cam, L.M., Neyman, J. (Eds.), University of California, 1967.
- [27] J. Tou, R. Gonzales, "Pattern Recognition Principles", Addison-Wesley, Reading, MA., 1974.

- [28] Y. Linde, A. Buzo, R. M. Gray, “An algorithm for vector quantizer design”, IEEE Trans. Commun. 28, 1980, 84–95.
- [29] G. P. Babu, M. N. Murty, “A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm”, Pattern Recognition Lett. 14 (10), 1993, 763–769.
- [30] C. Huang, R. Harris “A comparison of several codebook generation approaches”, IEEE Trans. Image Process. 2 (1), 1993, 108–112.
- [31] I. Katsavounidis, C. C. J. Kuo, Z. Zhen, “A new initialization technique for generalized lloyd iteration”, Signal Process. Lett. IEEE 1(10), 1994, 144–146.
- [32] M. B. A. Daoud, S. A. Roberts, “New methods for the initialization of clusters”, Pattern Recognition Lett. 17 (5), 1996, 451–45.
- [33] P. S. Bradley, U. M. Fayyad, “Refining initial points for K-means clustering”, In: Proc. 15th Internat. Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA, 1998, pp. 91–99.
- [34] A. Likas, N. Vlassis, J. J. Verbeek, 2003. “The global Kmeans clustering algorithm”, Pattern Recognition 36, 451–461.
- [35] S. S. Khan, A. Ahmad, 2004 “Cluster center initialization algorithm for k means clustering”, Pattern Recognition Lett. 25 (11), 1293–1302.
- [36] Bo Zhao, Zhongxiang Zhu, Enrong Mao and Zhenghe Song “Image Segmentation Based on Ant Colony Optimization and K-Means Clustering” Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China.
- [37] Nor Ashidi Mat Isa, Samy A. Salamah, Umi Kalthum Ngah “Adaptive Fuzzy Moving K-means Clustering Algorithm for Image Segmentation ” , IEEE Transactions on Consumer Electronics, Vol. 55, No. 4, Nov 2009.
- [38] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, U. Stilla., Classification with an edge: Improving semantic image segmentation with boundary detection, ISPRS

Journal of Photogrammetry and Remote Sensing, Volume 135, January 2018, Pages 158-172.

- [39] Nicolas Audebert, Bertrand Le Saux, Sébastien Lefèvre., Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 140, June 2018, Pages 20-32.