

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN TỬ VIỄN THÔNG



BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH
KĨ THUẬT MÁY TÍNH
ĐỀ TÀI : PHÁT HIỆN CHUYỂN ĐỘNG
TRONG VIDEO VÀ ỨNG DỤNG

GVHD : TS. Trần Thị Minh Hạnh
TS. Hoàng Lê Uyên Thực

SVTH : Lý Huỳnh Hữu Trí
Nguyễn Ngọc Hiếu
Tô Đông Trung

19DTCLC2
19DTCLC2
19DTCLC2

Đà Nẵng, tháng 12 năm 2022

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT	1
DANH MỤC CÁC TỪ TIẾNG ANH	1
DANH MỤC HÌNH ẢNH	2
LỜI MỞ ĐẦU	3
CHƯƠNG I . TỔNG QUAN	4
1.1 Giới thiệu chương.....	4
1.2 Giới thiệu đề tài	4
1.3 Mục tiêu của đề tài	5
1.4 Phương pháp nghiên cứu	6
1.5 Kết luận chương	6
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT	7
2.1 . Giới thiệu chương.....	7
2.2 . Các khái niệm cơ bản về video	8
2.2.1 Khái niệm video :	8
2.2.2. Chuyển động (Motion)	8
2.2.3 Nền (BackGround)	8
2.3 . Giới thiệu về ngôn ngữ lập trình Python , thư viện OpenCV và các thư viện liên quan	9
2.3.1 Ngôn ngữ Python.....	9
2.3.2. OpenCV	9
2.3.3. Numpy	9
2.3.4 Một số Mô_đun liên quan.....	10
2.4. Tìm hiểu về giải thuật trừ nền ứng dụng phát hiện chuyển động.....	10
2.4.1 . Khái niệm:	10
2.4.2 Các thuật toán trừ nền	10
2.5. Tổng quan về Mixture of Gaussian	11
2.5.1. Phân phối chuẩn.....	11
2.5.2 Mô hình Gaussian Mixture (GMM)	13
2.6 Tìm hiểu Phương pháp Mixture of Gaussian (MOG2).....	15
2.6.1 Bài toán đặt ra	15
2.6.2 Hướng giải quyết bài toán bằng Gaussian Mixture Model (GMM)	15
2.7 Tổng quan thuật toán Gaussian Mixture cho BackgroundSubtracsion :	22
2.7.1 Quy trình lập mô hình (Modeling Process)	22

2.7.2. Thích ứng mô hình (Cập nhật mô hình)	25
2.7.3 Ước tính mô hình nền	29
2.7.4 Phân loại điểm ảnh (Pixel).....	30
2.9 Thuật toán phân cụm (Thuật toán K-Mean Clustering)	31
2.9.1 Định nghĩa K-Mean Clustering	31
2.9.2 Nguyên lý của K-mean Clustering	31
2.9.3 Kết luận :	33
2.10 Tìm hiểu về xử lý ảnh	33
2.10.1 Tổng quan xử lý ảnh.....	33
2.10.2 Một số khái niệm cơ bản trong xử lý ảnh.....	34
2.10.3 Giới thiệu phép hình thái học trong xử lý ảnh.....	34
2.10.4 Phép giãn (Nở) ảnh :.....	34
2.10.5 Phép co ảnh.....	37
2.11 Phương pháp đánh giá mô hình.....	39
2.12 Ma trận nhầm lẫn (Confusion Matrix)	40
2.13 Kết luận chương	41
CHƯƠNG 3 : THỬ NGHIỆM VÀ ĐÁNH GIÁ	42
3.1 Giới thiệu chương.....	42
3.2 Sơ đồ khối của quy trình trừ nền sử dụng GMM	42
3.3 Lưu đồ thuật toán trừ nền sử dụng GMM	43
3.3.1 Phân tích lưu đồ thuật toán từng khối.....	44
3.4 Sử dụng hỗn hợp Gaussian 2 (MoG2) trong OpenCV	46
3.5 Các bài toán thực hiện	46
3.5.1 Phát hiện xe chuyển động trên đường và thực hiện theo dõi, đếm số lượng xe	46
3.5.2 Phát hiện trộm di chuyển vào trong nhà và thực hiện báo động.....	51
3.6 Đánh giá mô hình	53
3.6.1 Nhận xét đánh giá	54
3.6.2 Kết Luận	54
Kết Luận Và Hướng Phát Triển.....	55
1. Kết quả đạt được:	55
2. Hướng phát triển :	56
Phụ Lục	57
Tài liệu tham khảo	64

DANH MỤC CÁC TỪ VIẾT TẮT

CV	Computer Vision
GMM	Gaussian Mixture Model
MoG	Mixture of Gaussians
OpenCV	Open Source Computer Vision Library
NumPy	Numerical Python
IOU	Intersection Over Union
TP	True Positive
FP	False Positive
TN	True Negative
TN	False Negative

DANH MỤC CÁC TỪ TIẾNG ANH

Pixel	Điểm ảnh
Frame	Khung hình
Train	Huấn luyện
Detect	Phát hiện
Tracking	Theo dõi (Theo vết)
Background	Nền
Background Subtracsion	Trừ nền
Foreground	Tiền cảnh

DANH MỤC HÌNH ẢNH

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT	7
Hình 2.1. Mô hình Gaussian hỗn hợp	12
Hình 2.2. Bài toán về mô hình Gaussian.....	13
Hình 2.3. Kết quả bài toán về mô hình Gaussian dưới dạng biểu đồ.....	13
Hình 2.4 phân phối đa chiều với số cụm $K=3$ đối với dữ liệu 1 chiều và 2 chiều	14
Hình 2.5. Xét một điểm ảnh trên một khung hình tại thời điểm xác định.....	15
Hình 2.6 .Biểu đồ biểu thị cường độ màu sắc theo thời gian tại một điểm ảnh	16
Hình 2.7. Mô hình hóa sự phân bố cường độ sáng tại một điểm ảnh.....	17
Hình 2.8. Sử dụng mô hình Gauss cho chuồng đầu tiên tại 1 điểm ảnh	17
Hình 2.9. Biểu diễn biểu đồ dưới dạng tập hợp của 3 gaussian	18
Hình 2.10. Biểu diễn biểu đồ dưới dạng hỗn hợp các Gaussian	19
Hình 2.11. Biểu diễn biểu đồ của điểm ảnh dưới dạng hỗn hợp các Gaussian.....	20
Hình 2.12. Kết quả sau khi trừ nền	21
Hình 2.13. Mô tả về mô hình Gaussian với số cụm $K = 3$	23
Hình 2.14. Mô phỏng hỗn hợp Gaussian trên một điểm ảnh (pixel).....	24
Hình 2.15. Khởi tạo mô hình Gaussian Mixture	24
Hình 2.16. Kết quả của quá trình cập nhật Gaussian	27
Hình 2.17. Kết quả của quá trình khởi tạo ra một Gaussian mới	28
Hình 2.18 .Ví dụ về quá trình lựa chọn mô hình nền.....	29
Hình 2.19.Mình hoạt tính khoảng cách giữa 2 điểm dựa trên Euclide.....	32
Hình 2.20 .Ảnh minh hoạt ý tưởng của thuật toán K-Means	33
CHƯƠNG 3 : THỬ NGHIỆM VÀ ĐÁNH GIÁ.....	42
Hình 3.1 Quy trình trừ nền sử dụng Gaussian Mixture Model	42
Hình 3.2 Lưu đồ thuật toán của quá trình trừ nền	43
Hình 3.3 Lưu đồ thuật toán của quá trình khởi tạo mô hình Gauss	44
Hình 3.4 Lưu đồ thuật toán của quá trình huấn luyện mô hình Gauss.....	45
Hình 3.5 Hàm khởi tạo MOG2 với các tham số mặc định trong OpenCV	46
Hình 3.6 Sơ đồ thuật toán hiện theo dõi (tracking) và đếm số lượng xe trên đường	47
Hình 3.7 Kết quả mô hình đối với bài toán phát hiện chuyển động , theo dõi và đếm xe	50
Hình 3.8 Lưu đồ thuật toán phát hiện cảnh báo khi có trộm vào nhà	51
Hình 3.9 Kết quả mô hình đối với bài toán phát hiện trộm vào nhà , phát chuông cảnh báo ...	52

LỜI MỞ ĐẦU

Trong thời đại ngày nay công nghệ thông tin hầu như đã thâm nhập vào toàn bộ các lĩnh vực đời sống xã hội. Xã hội càng phát triển thì nhu cầu về công nghệ thông tin ngày càng cao, do vậy dữ liệu số hầu như không còn xa lạ đối với mỗi người chúng ta. Trong mọi lĩnh vực các ứng dụng công nghệ thông tin đã trợ giúp con người rất nhiều. Hiện nay, thông tin hình ảnh đóng vai trò rất quan trọng trong trao đổi thông tin, bởi phần lớn các thông tin mà con người thu nhận được đều thông qua thị giác. Trong các lĩnh vực công nghệ thông tin thì lĩnh vực giám sát tự động đã và đang thu hút được nhiều sự quan tâm của các nhóm nghiên cứu trong và ngoài nước. Cùng với sự phát triển của sức mạnh máy tính, các hệ thống giám sát tự động ngày càng tinh vi và hiện đại đã trợ giúp con người rất nhiều trong lĩnh vực an ninh, giám sát giao thông v.v. Thời gian qua, hệ thống giám sát bằng camera đã trở thành lĩnh vực nghiên cứu rất phát triển. Trước các nhu cầu giám sát, đảm bảo an ninh mục tiêu, trụ sở, nhà riêng... của các tổ chức, cá nhân ngày càng cao đã thúc đẩy việc nghiên cứu, ứng dụng các sản phẩm giám sát thông qua camera theo dõi. Một số ứng dụng quan trọng của hệ thống giám sát camera trong các lĩnh vực như: Các thiết bị giám sát an ninh chuyên dụng sử dụng trong lực lượng vũ trang; các thiết bị giám sát an ninh dân dụng; các thiết bị giám sát, phân luồng giao thông; công nghệ nghiên cứu, chế tạo Robot thông qua việc phân tích các cử động của con người... Ở nước ta hiện nay, lĩnh vực giám sát Video cũng có những bước phát triển đáng kể. Tuy nhiên nó chỉ mới dựa trên nền tảng phần cứng và cũng chưa áp dụng nhiều trong thực tế. Việc giải quyết bài toán này theo hướng tiếp cận sử dụng phần mềm chưa được quan tâm phát triển. Một hệ thống giám sát camera bao gồm nhiều chức năng: Phát hiện chuyển động, phân loại, theo dấu, cảnh báo, phân tích hành động. Phát hiện chuyển động được xác định là một trong những chức năng đầu tiên và quan trọng nhất của một hệ thống giám sát camera. Nó không chỉ có chức năng trích xuất đối tượng chuyển động mà còn có ý nghĩa quan trọng đối với ứng dụng thị giác máy tính như mã hóa video dựa trên đối tượng, phân tích chuyển động của con người và tương tác người – máy. Xuất phát từ các nhu cầu thực tế, nhóm chúng em đã quyết định chọn đề tài là: “Phát hiện chuyển động trong video và ứng dụng”

Để thực hiện và hoàn thành tốt đề tài này, nhóm chúng em xin cảm ơn cô TS. Trần Thị Minh Hạnh và cô TS. Hoàng Lê Uyên Thực . Cảm ơn hai Cô đã trực tiếp giúp đỡ và hướng dẫn nhóm trong thời gian thực hiện đề tài này

Xin chân Thành cảm ơn !

CHƯƠNG I. TỔNG QUAN

1.1 Giới thiệu chương

- Trong chương này ta sẽ tìm hiểu tầm quan trọng , tính cấp thiết của đề tài , tìm hiểu về thị giác máy tính .
- Đặt mục tiêu cho đề tài , tìm hiểu về phương pháp nghiên cứu để giải quyết đề tài

1.2 Giới thiệu đề tài

- Trong kỷ nguyên công nghệ thông tin hiện nay, với sự phát triển nhanh chóng của các công nghệ chế tạo thiết bị phần cứng ngày càng hiện đại, tinh vi thì ngành trí tuệ nhân tạo cũng không ngừng mở rộng để phù hợp với các yêu cầu của thực tế. Trong đó phải kể đến sự phát triển của các thiết bị thu nhận hình ảnh từ thế giới thực, chẳng hạn như các hệ thống giám sát bằng camera, song hành với nó là các vấn đề liên quan đến việc giám sát. Thách thức chính trong lĩnh vực này chính là việc xử lý các hình ảnh thu nhận được từ các hệ thống giám sát đó. Giám sát là một vấn đề được rất nhiều nhà nghiên cứu quan tâm đặc biệt bởi những ứng dụng thiết thực của nó cho đời sống xã hội. Chẳng hạn như các hệ thống giám sát các hành vi khả nghi của tội phạm, khủng bố ở các địa điểm nhạy cảm của các chính phủ. Hệ thống giám sát trong các viện bảo tàng, lưu trữ để chống trộm cắp các di vật đang được trưng bày. Hệ thống giám sát các hiện tượng bất bình thường, vi phạm pháp luật, tai nạn ở các điểm giao thông. Hệ thống giám sát phòng chống hỏa hoạn. Các hệ thống giám sát trong các siêu thị, cửa hàng, công ty để chống trộm cắp,... Thách thức chính cho ngành công nghệ phần mềm là đưa ra các giải pháp nhằm xây dựng một hệ thống giám sát tối ưu nhất nhằm giúp con người phát hiện chính xác và kịp thời các hiện tượng bất thường để có biện pháp xử lý nhanh chóng nhằm tránh các thiệt hại đáng tiếc cho xã hội. Dữ liệu thu được từ hệ thống camera giám sát thường được lưu trữ dưới dạng video. Như vậy công việc hiện nay của chúng ta là nghiên cứu các vấn đề liên quan đến việc xử lý video.
- Thị giác máy tính (tiếng Anh: computer vision) là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh và, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng, ví dụ trong các dạng quyết định. Việc phát triển lĩnh vực này có bối cảnh từ việc sao chép các khả năng thị giác con người bởi sự nhận diện và hiểu biết một hình ảnh mang tính điện tử. Sự nhận diện hình ảnh có thể xem là việc giải quyết vấn đề của các biểu tượng thông tin từ dữ liệu hình ảnh qua cách dùng các mô hình được xây dựng với sự giúp đỡ của các ngành lý thuyết học, thống kê, vật lý và hình học. Thị giác máy tính cũng được mô tả là sự tổng thể của một dải rộng các quá trình tự động và tích hợp và các thể hiện cho các nhận thức thị giác. Thị giác máy tính là một môn

học khoa học liên quan đến lý thuyết đằng sau các hệ thống nhân tạo có trích xuất các thông tin từ các hình ảnh. Dữ liệu hình ảnh có thể nhiều dạng, chẳng hạn như chuỗi video, các cảnh từ đa camera, hay dữ liệu đa chiều từ máy quét y học. Thị giác máy tính còn là một môn học kỹ thuật, trong đó tìm kiếm việc áp dụng các mô hình và các lý thuyết cho việc xây dựng các hệ thống thị giác máy tính. Các lĩnh vực con của thị giác máy tính bao gồm tái cấu trúc cảnh, dò tìm sự kiện, theo dõi video, nhận diện bố cục đối tượng, học, chỉ mục, đánh giá chuyển động và phục hồi ảnh.

1.3 Mục tiêu của đề tài

- Khi xã hội phát triển càng mạnh, yêu cầu về các thiết bị công nghệ càng cao. Như vậy, xử lý video là một mảnh đất màu mỡ cho các trung tâm nghiên cứu, các công ty đầu tư vào. Nhất là trong giai đoạn hệ thống nhúng đang phát triển và mở ra một kỷ nguyên mới cho ngành công nghệ phần mềm như hiện nay. Giám sát tự động là một hướng mới và có nhiều triển vọng trong sự phát triển tiếp theo của lĩnh vực nhận dạng và xử lý ảnh 2 chiều. Đồng thời, đó cũng là một hướng đi cho thiết kế chuyên dụng cho các thiết bị giám sát tự động. Việc phát hiện ra các đối tượng chuyển động trong video nhờ các kỹ thuật xử lý ảnh, trên cơ sở đó đoán nhận một số hành vi của đối tượng là một việc làm có ý nghĩa khoa học và thực tiễn. Nhất là trong hoàn cảnh Việt Nam chưa có nhiều những nghiên cứu và ứng dụng theo hướng này. Xuất phát từ thực tế đó, chúng em thấy việc nghiên cứu và đưa ra các phương pháp để xử lý video là vô cùng thiết thực. Nên nhóm quyết định ứng dụng lĩnh vực này với mục tiêu là gồm 2 ứng dụng chính đó là :

+ Đối với video ban ngày : Phát hiện chuyển động , thực hiện theo dõi và đếm số lượng xe đang di chuyển trên đường mục đích để đảm bảo an toàn trong tham gia giao thông , tìm kiếm và truy vết xe , thu thập thông tin .

+ Đối với video ban đêm : Phát hiện chuyển động và báo động khi có trộm đột nhập vào nhà dân , doanh nghiệp , hoặc ở những nơi thường xuyên xảy ra trộm cắp tài sản nhằm đảm bảo được trật tự an ninh, an toàn xã hội , đảm bảo về tài sản, vật chất của người dân , doanh nghiệp ,....Không bị mất cắp.

1.4 Phương pháp nghiên cứu

- Nghiên cứu và sử dụng phương pháp trừ nền. Bản chất của phương pháp này là phát hiện các đối tượng chuyển động từ sự khác biệt giữa khung hiện tại và khung tham chiếu
- Để cho đạt kết quả tốt nhất thì nhóm sử dụng thuật toán Gaussian Mixture Model (GMM) làm phương pháp nghiên cứu cho đề tài phát hiện chuyển động trong video

1.5 Kết luận chương

- Qua chương này chúng ta nắm rõ hơn về tầm quan trọng của đề tài phát hiện chuyển động trong video và ứng dụng nó ngoài thực tế
- Hiểu được rõ hơn về khái niệm thị giác máy tính và ứng dụng
- Có hai mục tiêu mà đề tài muốn hướng đến để ứng dụng trong thực tế:
 - + Với ứng dụng hệ thống phát hiện chuyển động vào ban ngày: Có thể phát hiện và đếm được xe đang di chuyển trên đường
 - + Với ứng dụng hệ thống phát hiện chuyển động vào ban đêm: Giám sát phát hiện được trộm vào nhà và phát chuông cảnh báo
- Với hai ứng dụng cùng với các tính năng mà mục tiêu đề tài đã đặt ra ở trên thì chúng ta sẽ có một cái nhìn sâu sắc hơn về tính thực tiễn của đề tài
- Để giải quyết các bài toán trên thì chúng ta đi tìm hiểu về một phương pháp được sử dụng rộng rãi trong lĩnh vực xử lý video là phương pháp trừ nền cùng với một thuật toán được sử dụng trong đề tài này là Gaussian Mixture Model (GMM) để cải thiện độ chính xác của bài toán.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 . Giới thiệu chương

- Trong chương này chúng ta tìm hiểu về các khái niệm cơ bản về :

- **Video:**

- + Khái niệm video
- + Chuyển động (Motion)
- + Nền (Background))

- **Python và thư viện trong Python :**

- + Khái niệm Python
- + Thư viện OpenCV
- + Thư viện hỗ trợ các phép toán học (Numpy)
- + Và một số module Python hỗ trợ và được sử dụng trong đồ án như (Randint , Time , Mixer ,...)

- **Tìm hiểu về giải thuật trừ nền và một số thuật toán liên quan đến giải thuật trừ nền :**

- + Khái niệm về trừ nền
- + Tìm hiểu các thuật toán trừ nền (**Frame Difference , Mean Filter, GMM** . Và ưu nhược điểm của từng thuật toán trên)

- **Tìm hiểu về phương pháp Mixture of Gaussian (MoG2) , bài toán đặt ra và cách dùng phương pháp Mixture of Gaussian để giải quyết bài toán**

- **Tiếp tục sẽ đi sâu hơn về lý thuyết của phương pháp Mixture of Gaussian :**

- + Phân phối chuẩn là gì ?
- + Ví dụ bài toán về phân phối chuẩn và cách giải quyết thể hiện dưới mô hình Gauss .
- + Gaussian Mixture Model là gì , ứng dụng của nó trong lĩnh vực thị giác máy tính .

- **Tổng quan thuật toán Gaussian Mixture cho BackgroundSubtracsion :**

- + Chúng ta sẽ tìm hiểu về tổng quan thuật toán như : Cách thuật toán này sử dụng trong giải thuật trừ nền , cách thuật toán này xây dựng lên một mô hình ban đầu và cập nhật mô hình Gauss thông qua một số công thức , cách để phân loại pixel thuộc về nền (Background) hay là tiền cảnh (Foreground) . Ngoài ra , chúng ta sẽ tìm hiểu tại sao chọn thuật toán MoG2 và một số tham số tác động đến hiệu suất

- Để hiểu hơn về cách phân loại thì chúng ta sẽ đi tìm hiểu thêm về một thuật toán đó là Kmean :
 - + Về nguyên lý hoạt động
 - + Kết luận lại thuật toán Kmean
- Ta sẽ tìm hiểu thêm về xử lý ảnh và ứng dụng xử lý ảnh trong đề án này :
 - + Các khái niệm cơ bản bản và một số thuật ngữ của một bức ảnh (điểm ảnh, Pixel, mức màu ,...)
 - + Một số phép toán hình thái học phục vụ trong việc khử nhiễu ở giai đoạn hậu xử lý của bức ảnh
- Phần cuối chúng ta sẽ tìm hiểu về một phương pháp ứng dụng để đánh giá mô hình phát hiện đối tượng đó là phương pháp Intersection Over Union (IoU)

2.2 . Các khái niệm cơ bản về video

2.2.1 Khái niệm video :

Video là tập hợp các khung hình, mỗi khung hình là một ảnh. dòng video gồm các chuỗi các khung hình liên tiếp, không thể chia nhỏ hơn, ứng với một thao tác camera đơn. Scene (cảnh) là các đơn vị logic của dòng video, một cảnh gồm các lia liên quan về không gian và liên kết về thời gian, cùng mô tả một nội dung ngữ nghĩa hoặc một tình tiết. Khi được chiếu, các khung hình lần lượt được hiển thị ở tốc độ nhất định. Tốc độ thường thấy ở các định dạng video khác nhau là 30 và 25 hình/s. Như vậy một giờ video sẽ có số khung hình tương ứng là 108000 hoặc là 90000. Dù là video ở định dạng nào thì nó cũng có dung lượng rất lớn và nếu xử lý với tất cả các khung hình thì thật không hiệu quả. Phân đoạn là quá trình phân tích và chia nội dung hình ảnh video thành các frame.

2.2.2. Chuyển động (Motion)

- Motion là thuộc tính quan trọng của video. Thông tin về chuyển động có thể được sinh ra bằng các kỹ thuật ghép khối hoặc luồng ánh sáng. Các đặc trưng chuyển động như mô men của trường chuyển động, biểu đồ chuyển động hoặc là các tham số chuyển động toàn cục có thể được trích chọn từ vector chuyển động. Các đặc trưng mức cao phản ánh di chuyển camera như quét camera (pan), nghiêng (tilt), phóng to (zoom out), thu nhỏ (zoom in) cũng có thể được trích chọn.

2.2.3 Nền (BackGround)

- BackGround được hiểu đơn giản như là phong nền , màu nền chủ đạo được hiển thị nhiều nhất trong suốt video . Về tổng quan, công dụng trước hết chúng ta có một yếu tố được định vị, xác định rõ để làm bậc cho những yếu tố khác phát triển lên trên nền (Background) đó.

2.3 . Giới thiệu về ngôn ngữ lập trình Python , thư viện OpenCV và các thư viện liên quan

2.3.1 Ngôn ngữ Python

- Python một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo.
- Ngày nay, Python trở nên phổ biến nhanh vì tính đơn giản và khả năng đọc mã. So với C/C++, Python hoạt động chậm hơn do Python là ngôn ngữ thông dịch

2.3.2. OpenCV

- OpenCV (Open Source Computer Vision Library) à một thư viện mã nguồn mở Python hỗ trợ các chức năng lập trình được sử dụng cho thị giác máy tính trong Trí tuệ nhân tạo, Học máy, Nhận dạng khuôn mặt, v.v.
- Trong OpenCV, CV là dạng viết tắt của thị giác máy tính, được định nghĩa là lĩnh vực nghiên cứu giúp máy tính hiểu được nội dung của các hình ảnh kỹ thuật số như ảnh và video.
- Đặc biệt là về xử lý ảnh ứng dụng trong thị giác máy tính. OpenCV hỗ trợ nhiều ngôn ngữ khác nhau như: Python, C++, ... và có sẵn trên các nền tảng như Windows, Linux, Mac OS, ...

2.3.3. Numpy

- NumPy là viết tắt của Numerical Python. NumPy được tạo ra vào năm 2005 bởi Travis Oliphant. Nó là một dự án mã nguồn mở cấu trúc dữ liệu mạnh mẽ, triển khai các mảng và ma trận đa chiều. Các cấu trúc dữ liệu này đảm bảo tính toán hiệu quả với ma trận và mảng. Việc triển khai thậm chí còn nhắm đến các ma trận và mảng khổng lồ. Bên cạnh đó, mô-đun cung cấp một thư viện lớn các hàm toán học cấp cao để hoạt động trên các ma trận và mảng này.

2.3.4 Một số Mô_đun liên quan

- Randint : tạo ra các số ngẫu nhiên
- Time , Date: Cung cấp các lớp để thao tác ngày và giờ.
- Mixer: Mô-đun của PyGame để tải và phát âm thanh

2.4. Tìm hiểu về giải thuật trừ nền ứng dụng phát hiện chuyển động

2.4.1 . Khái niệm:

- Background subtraction hay còn gọi là trừ nền là một trong những giải thuật đơn giản và phổ biến trong lĩnh vực Thị giác máy tính (Computer Vision). Thuật toán được sử dụng nhằm xác định đối tượng chuyển động trong camera nền tĩnh.
- Phép trừ nền là quá trình tách các đối tượng tiền cảnh ra khỏi nền trong một chuỗi các khung hình video. Nó được sử dụng rộng rãi trong các ứng dụng phát hiện và theo dõi đối tượng để loại bỏ phần nền

2.4.2 Các thuật toán trừ nền

a) Khung khác biệt (Frame Difference)

- Phương pháp Frame Differences là một **thuật toán để xác định chuyển động của một đối tượng**. Sử dụng thuật toán này, chúng ta có thể phân biệt một đối tượng chuyển động trong môi trường. Phép trừ nền là một trong những phương pháp phù hợp để cải thiện hơn nữa sự khác biệt khung hình, do đó tăng hiệu quả và độ chính xác của nó.

- Ưu điểm :

- Đây là phương pháp đơn giản nhất của giải thuật trừ nền. Ta sẽ tính hiệu giá trị pixel của khung hình hiện tại và khung hình trước đó nếu hiệu này lớn hơn ngưỡng **T** thì pixel tại vị trí đó thuộc về Foreground.

- Ưu điểm của phương pháp này là tốc độ tính toán nhanh do việc khởi tạo background image chỉ đơn giản là việc lấy khung hình trước đó.

- Nhược điểm :

- Phương pháp này là nó chỉ tốt với những đối tượng di chuyển liên tục nhưng khi có một đối tượng đứng yên trong khung hình quá lâu thì đối tượng này sẽ bị cho là background cùng với đó phương pháp này phụ thuộc rất nhiều vào ngưỡng **T** do đó với mỗi video ta đều phải chọn 1 ngưỡng phù hợp.

b) Bộ lọc trung bình (Mean Filter)

- Phương pháp bộ lọc trung bình (Mean Filter) sử dụng N khung hình phía trước khung hình đang xét để khởi tạo background image. Giả sử khung hình đang xét tại thời điểm t thì background tương ứng với nó sẽ được tính như sau:

$$B(x, y, t) = \frac{1}{N} \sum_{i=1}^N I(x, y, t) \quad (2.1)$$

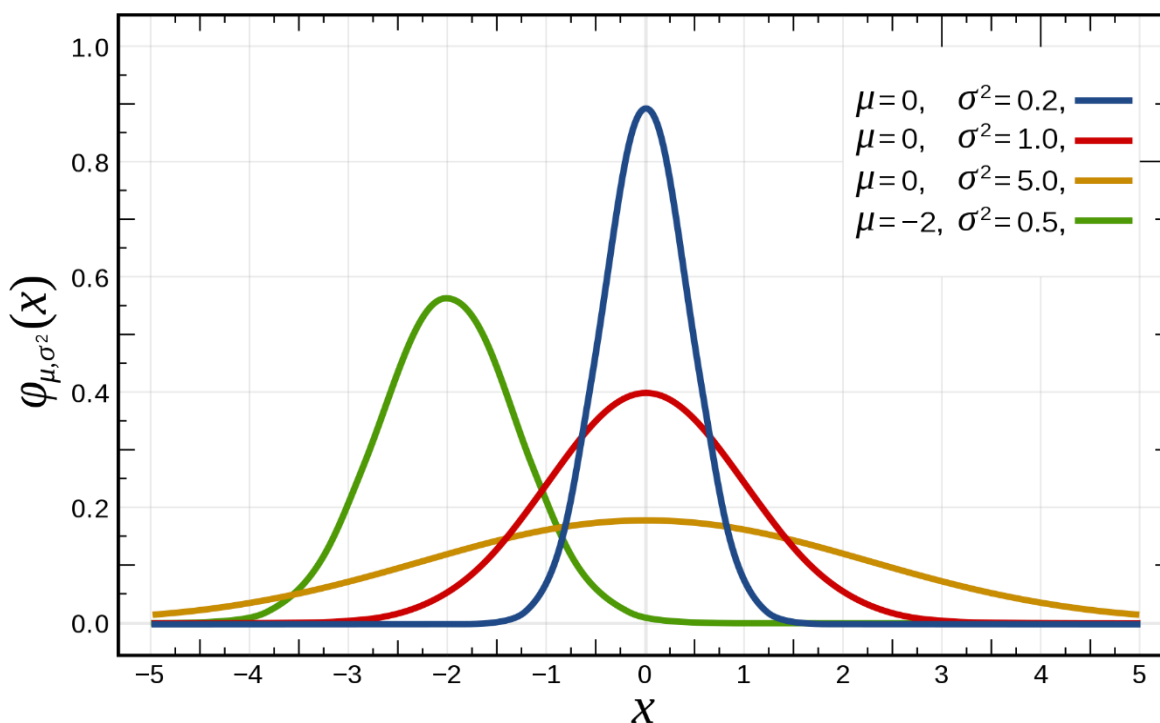
- Trong đó B là background image tại thời điểm t , N là số khung hình trước thời điểm t để dùng cho việc tính toán ra background image, $I(x, y, t)$ là khung hình tại thời điểm t .
- Sau khi khởi tạo background image thì phần còn lại của phương pháp này cũng giống với phương pháp frame difference.

2.5. Tổng quan về Mixture of Gaussian

2.5.1. Phân phối chuẩn

- **Phân bố Gaussian (Phân phối chuẩn)** : Là một loại phân bố xác suất liên tục đối xứng về giá trị trung bình của nó; hầu hết các cụm quan sát xung quanh giá trị trung bình, và quan sát càng xa giá trị trung bình, xác suất xảy ra của nó càng thấp. Giống như các phân bố xác suất khác, phân bố Gaussian mô tả cách phân phối kết quả của một biến ngẫu nhiên, Đồ thị của một hàm Gaussian tạo thành hình dạng chuông đặc trưng
- Phân bố Gaussian được đặt tên như vậy bởi vì nó được phát hiện lần đầu tiên bởi Carl Friedrich Gauss, được sử dụng rộng rãi trong xác suất và thống kê. Điều này phần lớn là do định lý giới hạn trung tâm, trong đó nói rằng một sự kiện là tổng của các sự kiện ngẫu nhiên nhưng giống hệt nhau có xu hướng hướng tới một phân phối chuẩn, bất kể sự phân bố của biến ngẫu nhiên. Nhiều hiện tượng tự nhiên, chẳng hạn như chiều cao, cân nặng, điểm kiểm tra và các hiện tượng khác, phù hợp với tiêu chí này, và do đó thể hiện sự phân phối bình thường.

- Đây là một phân phối xác suất cực kì quan trọng trong nhiều lĩnh vực .Nó là họ phân phối có dạng tổng quát giống nhau, chỉ khác tham số vị trí (giá trị trung bình μ) và tỉ lệ (phương sai σ).



Hình 2.1. Mô hình Gaussian hỗn hợp

Hàm mật độ xác suất của phân phối chuẩn với trung bình μ và phương sai σ^2 (hay độ lệch chuẩn σ) là một ví dụ của một hàm Gauss có công thức [1] :

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (2.2)$$

Trong đó :

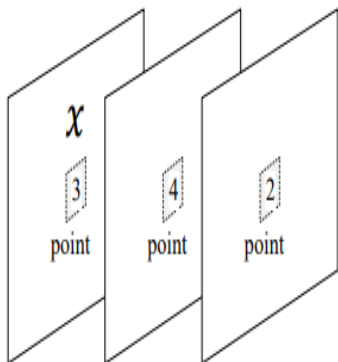
μ = Giá trị trung bình

σ = Độ lệch chuẩn

$\pi \approx 3.14159$

$e \approx 2.71828$

Bài toán đặt ra : Ví dụ \mathbf{x} là một pixel ngẫu nhiên biểu thị cường độ sáng của một điểm ảnh cụ thể, ta có giá trị cường độ sáng của \mathbf{x} tại 3 thời điểm liên tiếp của 3 frame tương ứng 3 giá trị là 2.0 ,3.0 ,4.0 khi áp dụng mô hình Gaussian được tính như sau :

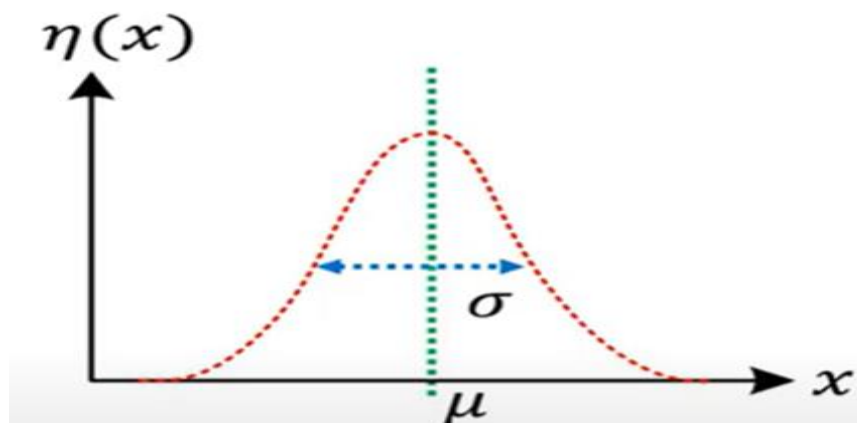


$$\mu = \frac{3.0 + 4.0 + 2.0}{3} = 3.0 ; \sigma = \sqrt{0.67} = 0.81$$

$$\eta(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi} \times 0.81} \exp \left(- \left(\frac{x - 3.0}{0.81} \right)^2 \right)$$

Hình 2.2. Bài toán về mô hình Gaussian

Và được thể hiện bằng mô hình Gauss như bên dưới



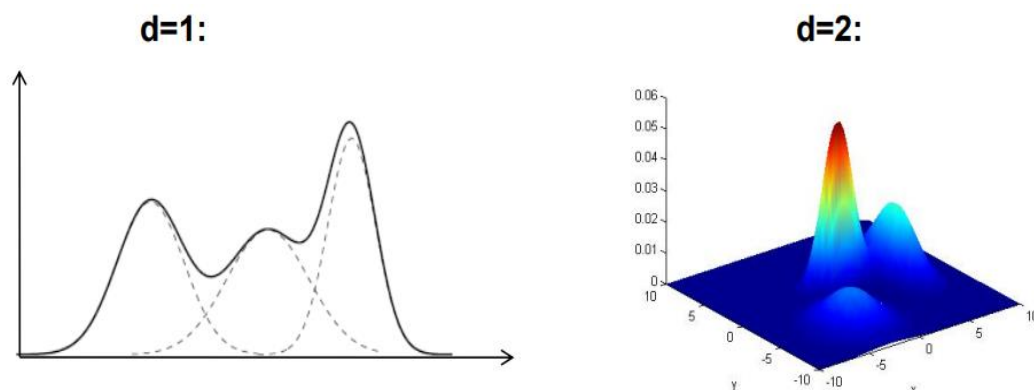
Hình 2.3. Kết quả bài toán về mô hình Gaussian dưới dạng biểu đồ

2.5.2 Mô hình Gaussian Mixture (GMM)

+ Gaussian Mixture Model (*viết tắt GMM*) là một mô hình phân cụm thuộc lớp bài toán học không giám sát mà phân phối xác suất của mỗi một cụm được giả định là phân phối Gaussian đa chiều. Sở dĩ mô hình được gọi là Mixture là vì xác suất của mỗi điểm dữ liệu không chỉ phụ thuộc vào một phân phối Gaussian duy nhất mà là kết hợp từ nhiều phân phối Gaussian khác nhau từ mỗi cụm.

+ Mô hình hỗn hợp Gaussian là một mô hình xác suất để đại diện cho các quần thể con phân bố bình thường trong một tổng thể tổng thể (Như ví về bài toán đặt ra ở trên xét trên 1 pixel thì 1 pixel có thể có nhiều quần thể con về cường độ màu sắc cụ thể như (đường , xe ô tô , và có khi là có tuyết hoặc giọt mưa đi qua một pixel))

Ví dụ về Mixture of Gaussian:



Hình 1.4 phân phối đa chiều với số cụm $K=3$ đối với dữ liệu 1 chiều và 2 chiều

- Mục tiêu của mô hình GMM là ước lượng tham số phù hợp nhất cho k cụm thông qua phương pháp ước lượng hợp lý tối đa
- Một số giả định của mô hình GMM:
 - + Có k cụm cần phân chia mà mỗi cụm tuân theo phân phối Gaussian đa chiều với tập tham số đặc trưng $\{(\mu_i, \Sigma_i)\}_{i=1}^k$
 - + z_k được giả định là một biến ngẫu nhiên nhận giá trị 1 nếu như quan sát \mathbf{x} rơi vào cụm thứ k , các trường hợp còn lại nhận giá trị 0.
 - + z_k được coi như là một biến ẩn (latent variable hoặc hidden variable) mà ta chưa biết giá trị của nó. Xác suất xảy ra của $p(z_k = 1|\mathbf{x})$ giúp chúng ta xác định tham số phân phối của Gaussian Mixture.
 - + Tập hợp các giá trị của z_k đối với các cụm sẽ tạo thành một phân phối xác suất sẽ tạo thành một phân phối xác suất $(\pi_1, \pi_2, \dots, \pi_k)$ trong đó $\pi_k = p(z_k = 1|\mathbf{x})$
 - + Một xác suất hỗn hợp tại một điểm dữ liệu \mathbf{X} sẽ được tính theo công thức Bayes như sau [2]:

$$\begin{aligned}
 p(\mathbf{x}) &= \sum_{c=1}^k p(z_c) p(\mathbf{x}|z_c) \\
 &= \sum_{c=1}^k p(z_c = 1) p(\mathbf{x}|\mu_c, \Sigma_c) \\
 &= \sum_{c=1}^k \pi_c p(\mathbf{x}|\mu_c, \Sigma_c) \\
 &= \sum_{c=1}^k \pi_c N(\mathbf{x}|\mu_c, \Sigma_c)
 \end{aligned} \tag{2.3}$$

+ Thành phần xác suất $p(\mathbf{x}|\mu_i, \Sigma_i)$ được tính từ phân phối Gaussian đa chiều và chúng đồng thời là mục tiêu mà chúng ta cần tham số hoá.

2.6 Tìm hiểu Phương pháp Mixture of Gaussian (MOG2)

2.6.1 Bài toán đặt ra

- Giả sử rằng trong một video đầu vào đang hoạt động ở một thời tiết bình thường có thể chuyển sang thời tiết xấu (trong video chứa cảnh đường phố có xe ô tô chạy qua và chúng ta muốn đếm những chiếc xe, theo dõi, hoặc tìm ra những chiếc xe vi phạm giao thông) nhưng những gì chúng ta có là những hạt mưa và tuyết trong video trên đó và hạt mưa (tuyết) có thể duy chuyển xung quanh (tăng thêm sự phức tạp của ảnh) thì với phương pháp trừ nền cơ bản nó thật sự không giải quyết tốt các vấn đề trên
- Để thật sự đưa ra một giải pháp hữu ích, thì chúng ta cần có khả năng lập mô hình các biến đổi cường độ tại mỗi điểm trong ảnh vì vậy chúng ta xem các biến thể cường độ như một phân bố và sau đó chúng ta điều chỉnh một mô hình cho phù hợp với phân phối, bằng cách xem xét các thành phần khác nhau này của sự phân bố, chúng ta đưa ra phán đoán rằng pixel sẽ thuộc về nền hay là tiền cảnh và mô hình này chúng ta nói đến là GMM (Gaussian Mixture Model).

2.6.2 Hướng giải quyết bài toán bằng Gaussian Mixture Model (GMM)

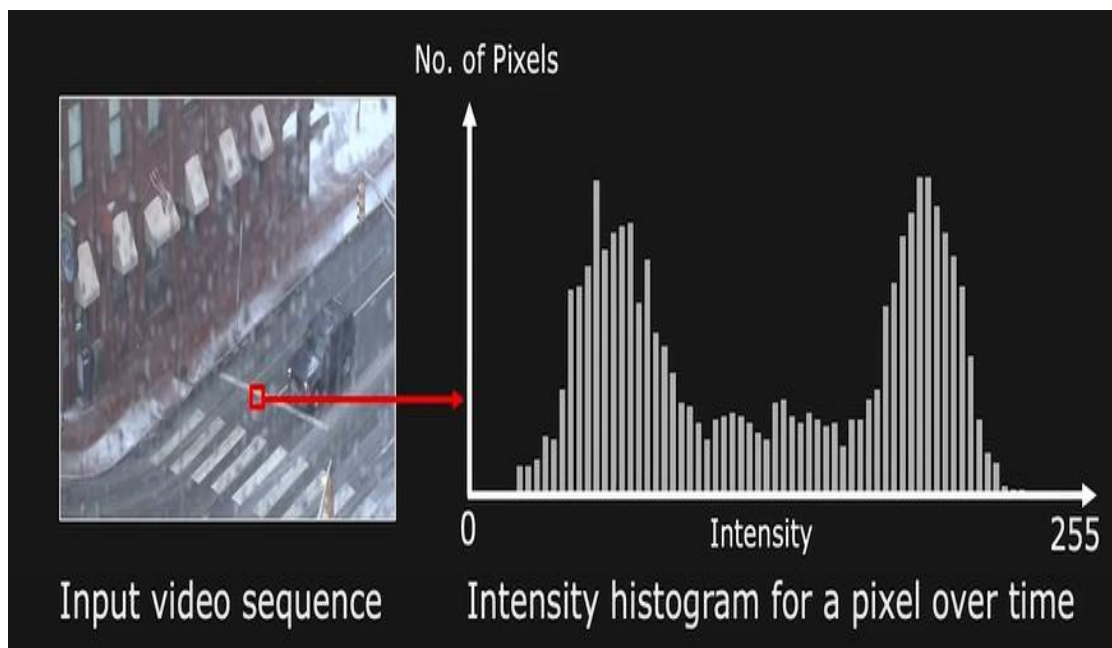
- Bài toán đặt ra ở trên được giải quyết bằng GMM như sau :

Hình 2.5 bên dưới: Đầu vào là một khung hình video. Chúng ta giả sử xét một pixel tại vòng màu đỏ và ta tiến hành xem xét mà tính toán biểu đồ của các mẫu độ sáng bằng Histogram tương ứng với ảnh đó



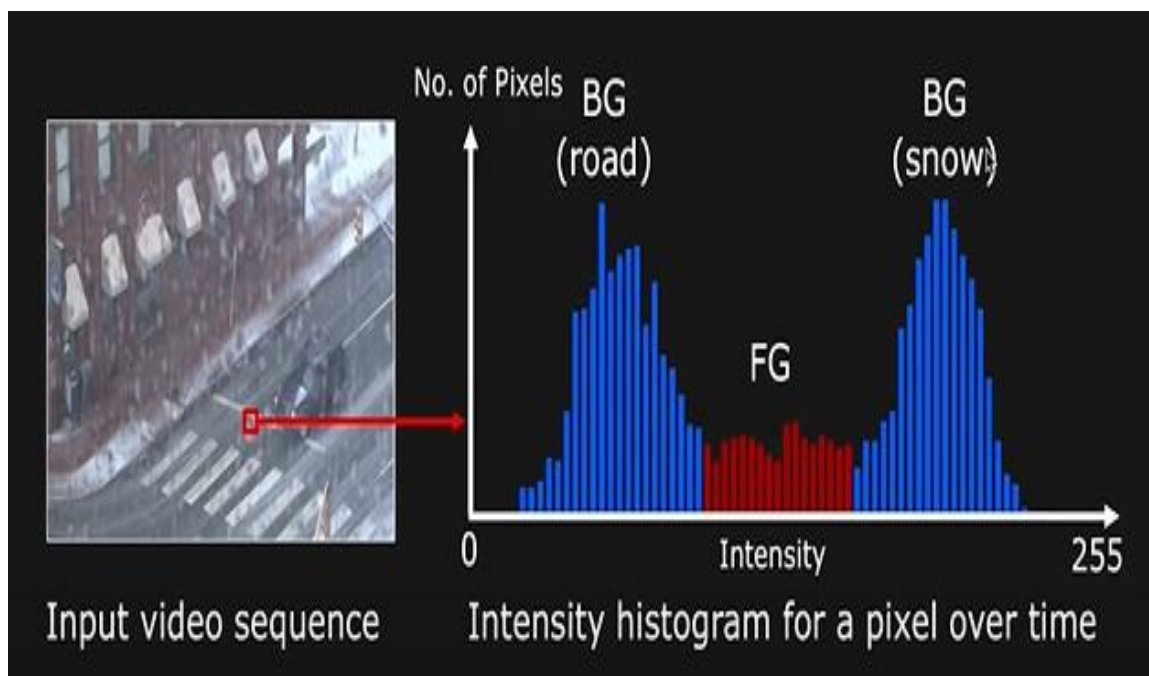
Hình 2.5. Xét một điểm ảnh trên một khung hình tại thời điểm xác định

- Ta thấy biểu đồ (hình 2.6) có giá trị cường độ sáng tại 1 pixel từ 0-255 (trục hoành), tần suất xuất hiện của cường độ ánh sáng (trục tung):



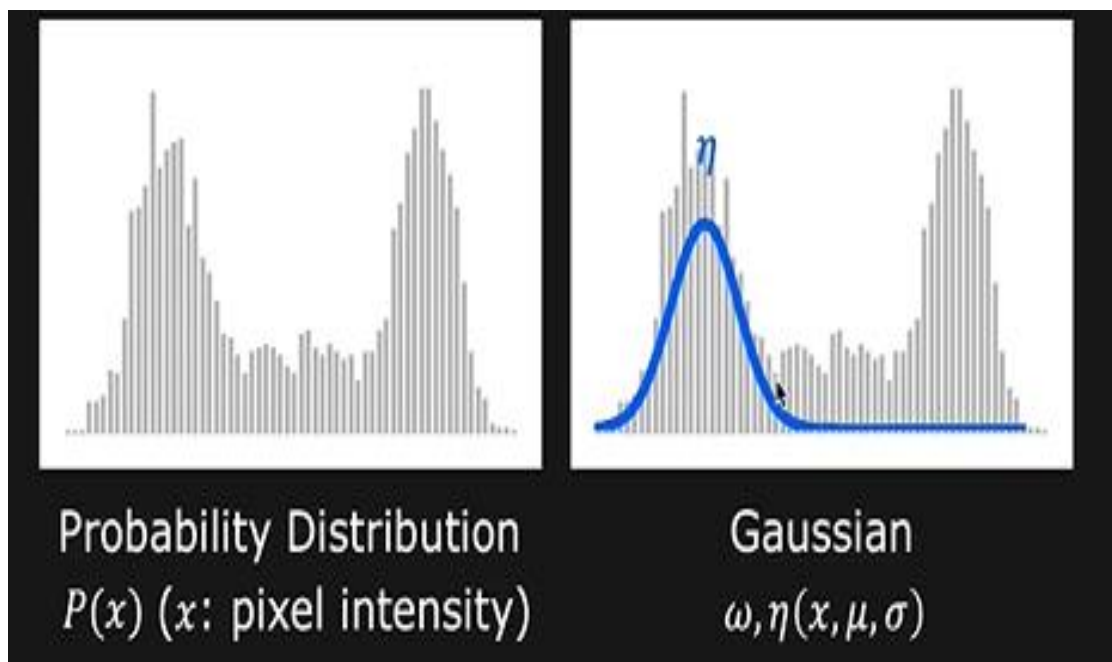
Hình 2.6 .Biểu đồ biểu thị cường độ màu sắc theo thời gian tại một điểm ảnh

- Chi tiết do cường độ sáng về nền như ta thấy ở hình 2.6 (nền (Background) sẽ là: đường , tuyết cũng được xem là thuộc về nền (Background)). Có thể thay đổi theo thời gian về độ chiếu sáng , trong trường hợp này có thể do ánh sáng hoặc giọt mưa , tuyết đi qua pixel và đôi khi đối tượng chuyển động đi ngang qua pixel ví dụ ở đây là xe ô tô nhưng hiếm khi xe đi qua pixel này mà thường thì do tuyết hay mưa đi qua
- Chúng ta có thể xem đây là một tập hợp các cường độ , mỗi hình chuông là một quần thể con (gồm: con đường , tuyết , và đôi khi có xe ô tô chạy qua dẫn đến cường độ thay đổi được thể hiện bằng xanh lam và sau đó đối tượng hoặc phương tiện di chuyển không thường xuyên (Cụ thể là ô tô) được hiển thị bằng màu đỏ (Hình 2.7) . Vì vậy trực giác ở đây là pixel nền vì nếu một chiếc ô tô đi vào và nằm trên pixel và nó vẫn ở đó và tại một số điểm chúng ta phải nói rằng chiếc xe đó là một phần của nền (Background) (Như hình 2.7)



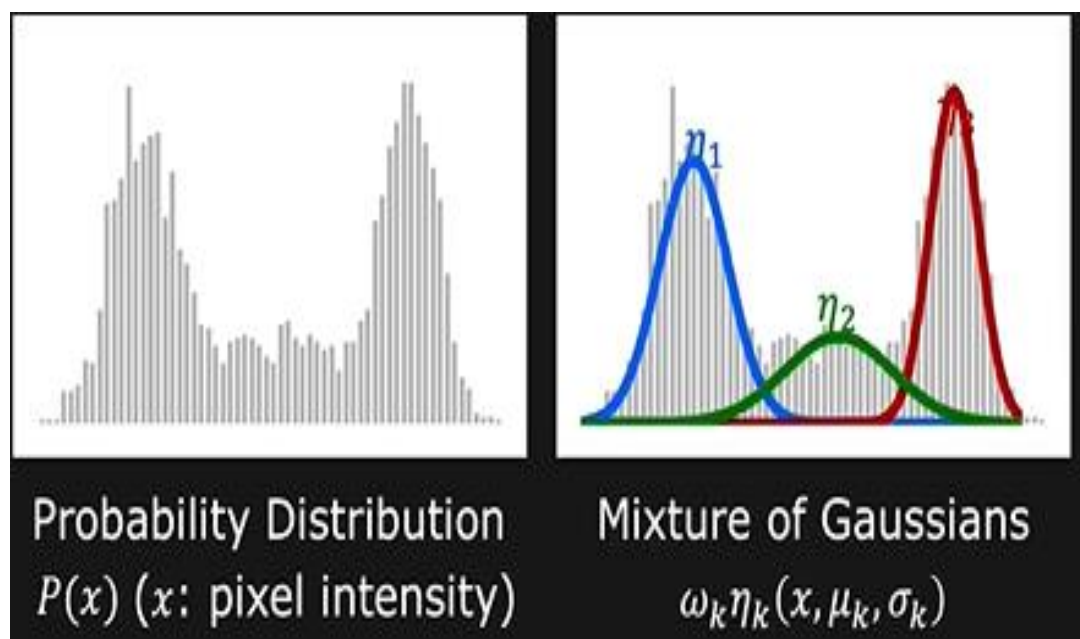
Hình 2.7. Mô hình hóa sự phân bố cường độ sáng tại một điểm ảnh

- Chúng ta sẽ thử và mô hình hóa sự phân bố này và sau đó phân loại pixel thuộc về Background (Nền) hay là Foreground (Tiền Cảnh) . Nghĩa là chúng ta sẽ sử dụng mô hình Gaussian và mô hình hóa thác đầu tiên sử dụng Gaussian nếu lắp một Gaussian và dữ liệu đó sẽ có dạng như (Hình 2.8)



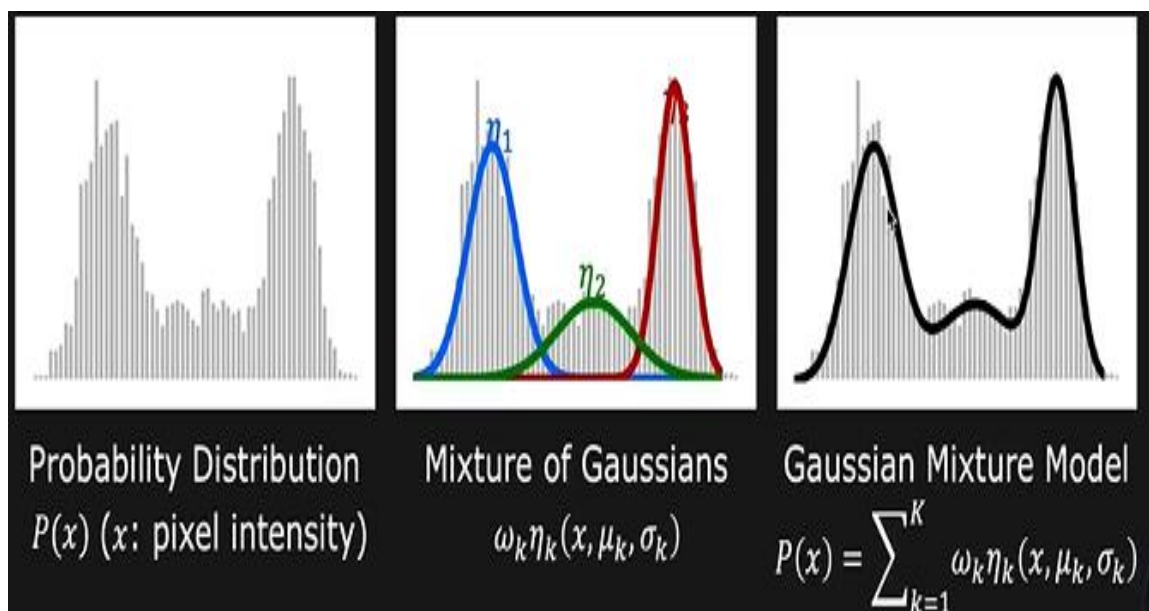
Hình 2.8. Sử dụng mô hình Gauss cho chuông đầu tiên tại 1 điểm ảnh

- Tóm tắt một chút về Gaussian: Như chúng ta đã biết chiều rộng (Độ lệch chuẩn) được cho bởi sigma (σ) và giá trị trung bình do Mean đưa ra kí hiệu (μ). Chúng ta sẽ có một tham số thứ ba đó là ω (Omega). ω (Omega) được gọi là trọng số liên quan đến Gaussian này điều này có thể phụ thuộc vào thực thể của từng vật thể chuyển động qua điểm ảnh (pixel) thì sẽ có những chiều cao của chuông khác nhau đó chính là Gaussian. Nên chúng ta có thể lập mô hình một trong những đỉnh với một Gaussian tương ứng , vì chúng ta đang xem xét các giá trị độ sáng . Với logic tương tự chúng ta có thể vẽ lên các Gaussian và bây giờ chúng ta xem toàn bộ biểu đồ biến thiên cường độ như tập hợp các Gaussian (Như hình 2.9)



Hình 2.9. Biểu diễn biểu đồ dưới dạng tập hợp của 3 gaussian

- Một biểu diễn khá tốt bằng cách sử dụng hỗn hợp Gaussian . Khi trộn chúng với nhau và chúng ta sẽ nhận được một biểu đồ thực tế (Hình 2.10)



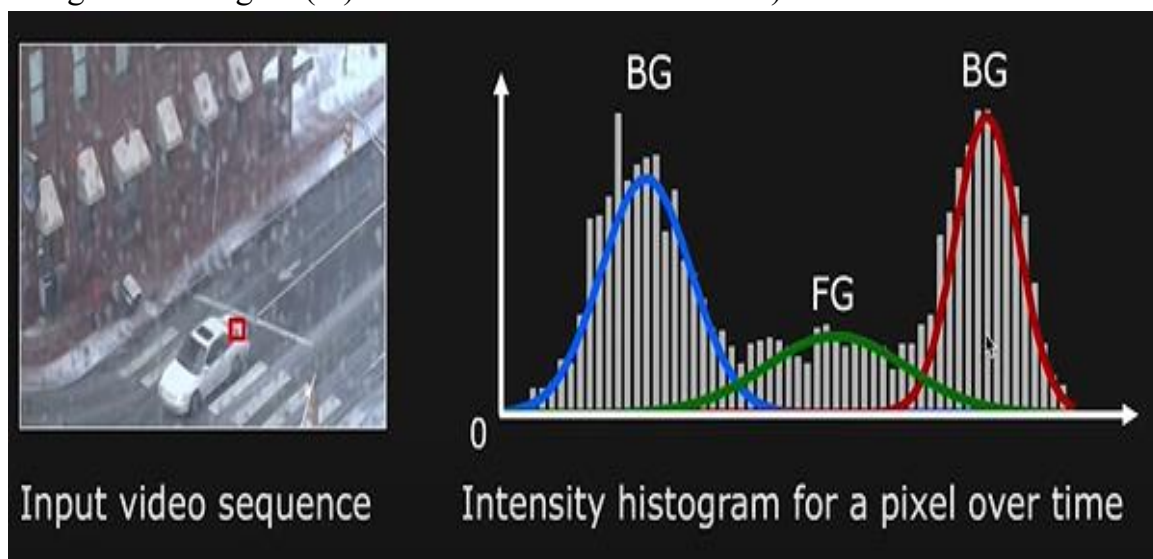
Hình 2.10. Biểu diễn biểu đồ dưới dạng hỗn hợp các Gaussian

- Lưu ý : Chúng ta chỉ xét một điểm ảnh (pixel) ở vòng tròn màu đỏ (một video có thể đọc vào nhiều frame nhưng chúng ta chỉ ví dụ là ngay tại pixel đó chỉ có 3 trường hợp xảy ra với pixel đang xét).
 ⇒ Vậy đó là ý tưởng mà chúng ta điều chỉnh một mô hình hỗn hợp Gaussian vào biểu đồ , và chúng ta sẽ có mô hình là tổng có trọng số của các Gaussian chính. Mỗi gaussian đều có giá trị trung bình riêng và chiều rộng của nó
- GMM Distribution : Weighted sum of K Gaussians

$$P(x) \cong \sum_{k=1}^K \omega_k \eta_k(x, \mu_k, \sigma_k) \text{ Such that } \sum_{k=1}^K \omega_k = 1 \quad (2.4)$$

- Để sử dụng mô hình GMM đạt độ chính xác cao nhất thì chúng ta phải chọn K (Số lượng mô hình Gaussian) tại một điểm ảnh (pixel) . nếu K quá lớn có thể tưởng tượng dữ liệu sẽ không ổn định , nếu K quá nhỏ sự phù hợp sẽ không chính xác và không đủ linh hoạt trong một mô hình vì thế để đưa ra một mô hình hỗn hợp Gaussian cho cường độ biến đổi màu sắc của pixel theo thời gian chúng ta sẽ có các vấn đề giải quyết đó là chúng ta phân loại đối tượng trong số Gaussian hình là tiền cảnh (Foreground) hay nền (Background) . Trước tiên muốn tìm ra thì chúng ta phải có một mô hình và tìm ra Gaussian đại diện cho tiền cảnh (Foreground) hoặc nền (Background) để chúng ta nhìn thấy được các điểm ảnh (pixel) trong tương lai (nghĩa là khung hình (Frame) tiếp theo), và phân loại chúng thành các thành phần khác nhau

- Hình 2.11 chúng ta thấy ba Gaussian được liên kết với phân phối cụ thể này (khi ω lớn và sigma (σ) nhỏ \Rightarrow sẽ là Gaussian nền)



Hình 2.11. Biểu diễn biểu đồ của điểm ảnh dưới dạng hỗn hợp các Gaussian

Trong hình 2.11. BG: kí hiệu nền (Background)
FG: kí hiệu tiền cảnh (Foreground)

- Nếu chia Omega và Sigma có độ lệch chuẩn lớn thì chúng ta gọi đó là nền . và bất cứ khi nào có Omega chia chi Sigma nhỏ sẽ được gọi là tiền cảnh

Công thức : ω / σ lớn \Rightarrow background , ω / σ nhỏ \Rightarrow foreground

Một điều gặp phải là cập nhật lại biểu đồ do quá trình xử lý video là đọc vào từng frame

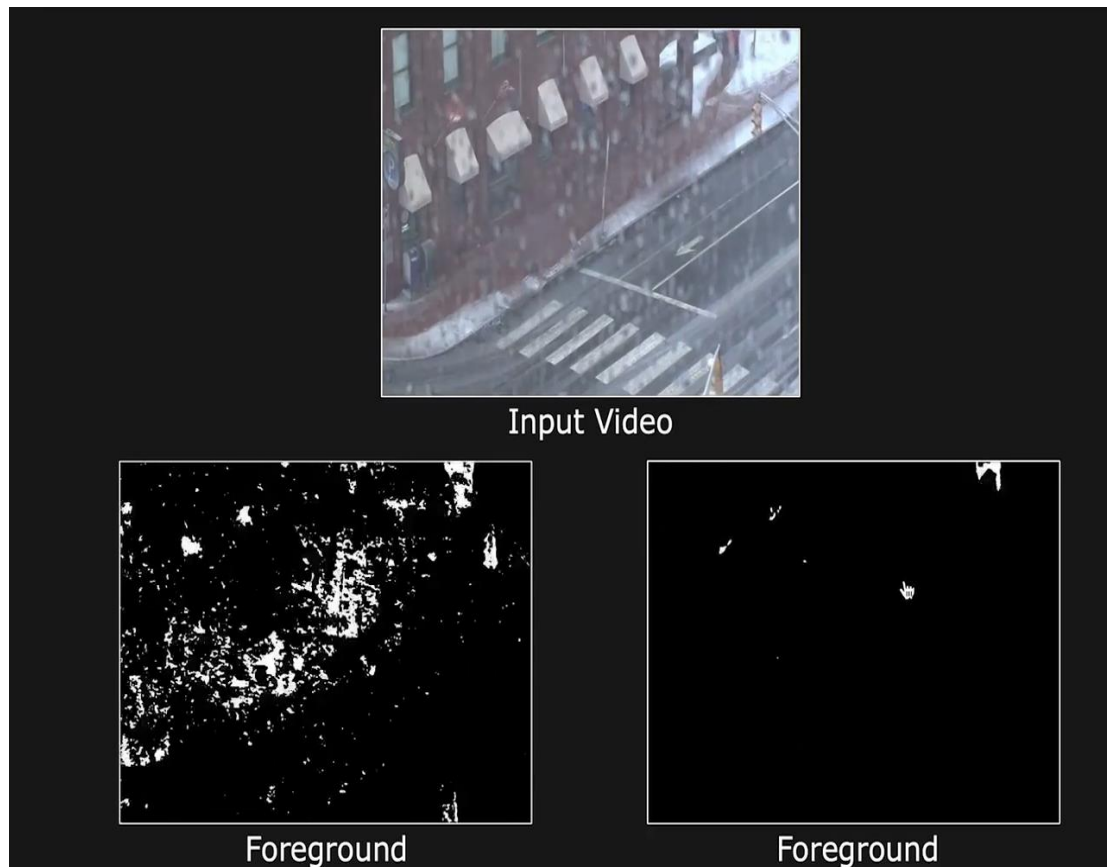
- Chúng ta không muốn là điều này cho mỗi khung hình mà chúng ta sẽ tính toán lại các mô hình hỗn hợp Gaussian và có thể là hàng triệu điểm ảnh (pixel) cho mỗi khung hình ta thấy quá rắc rối và tính toán phức tạp nên ta làm như sau :
 - + Bước 1: Khung hình đầu tiên của video chúng ta tiến hành chuẩn hóa biểu đồ để đảm bảo nó được phân phối
 - + Bước 2: Bây giờ chúng ta lắp một GMM và chúng ta chọn K bằng 3-5 để phân loại pixel và đối với mỗi khung hình tiếp theo
 - + Bước 3: Chúng ta sẽ lấy giá trị pixel X . Giá trị X và ta tìm Gaussian nào trong mô hình của chúng ta gần nhất về giá trị trung bình μ_k gần nhất với X. Đó là Gaussian mà giá trị pixel thuộc về muốn biết Gaussian đó là gì chúng ta nhìn vào Omega chia cho Sigma để phân biệt

Lưu ý: Để biết rằng điểm ảnh (Pixel) có thuộc về Gaussian hay không thì ta có công thức như sau :

$$\|X - \mu_k\| \text{ is minimum and } \|X - \mu_k\| < 2.5\sigma_k \quad (2.5)$$

+ B4: Chúng ta chỉ cần cập nhật những biểu đồ cho pixel , nếu biểu đồ chuẩn hóa khác với biểu đồ hiện tại mà nếu sự khác biệt đó lớn thì chúng ta sẽ thay biểu đồ mới và sửa lại mô hình GMM (Nhưng điều này hiếm khi xảy ra vì một pixel duy nhất sẽ không thay đổi đáng kể biểu đồ trên hàng triệu biểu đồ pixel nhưng theo thời gian thì có thể thay đổi biểu đồ đó (có thể là độ sáng thay đổi , hoặc thời tiết thay đổi đột ngột có thể xảy ra))

Kết quả cuối cùng :



Hình 2.12. Kết quả sau khi trừ nền

2.7 Tổng quan thuật toán Gaussian Mixture cho BackgroundSubtracsion :

- Các giá trị của một pixel cụ thể được mô hình hóa dưới dạng Mixture (hỗn hợp) các Gaussian Adaptive (Thích ứng) :

+ Tại sao lại Mixture (Hỗn hợp) ?

⇒ Vì 1 điểm ảnh (pixel) đang xét thì có thể có nhiều giá trị thay đổi theo thời gian (Hoặc chính xác hơn là nhiều bề mặt xuất hiện trong 1 điểm ảnh (pixel))

+ Tại sao lại Adaptive (Thích ứng) ?

⇒ Vì Điều kiện ánh sáng thay đổi

- Ở mỗi lần lặp lại, Gaussian được đánh giá bằng cách sử dụng phương pháp đơn giản phỏng đoán để xác định điểm ảnh (pixel) nào có khả năng tương thích với nền nhất

- Các điểm ảnh (pixel) không khớp với “Gaussian nền” được phân loại là tiền cảnh ,và các điểm ảnh (pixel) tiền cảnh được nhóm lại

2.7.1 Quy trình lập mô hình (Modeling Process)

- Mô hình hỗn hợp

- Tại bất kỳ thời điểm t nào, thông tin được biết về một pixel cụ thể, tọa độ (x_0, y_0) [4] :

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \quad (2.6)$$

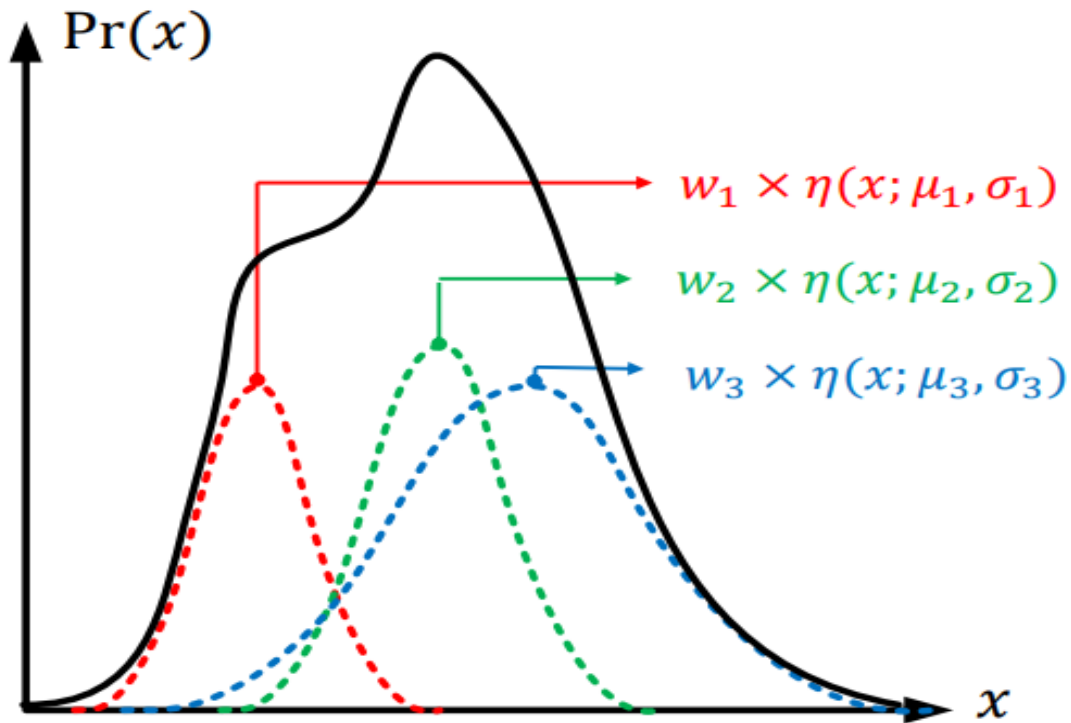
- Lịch sử này được mô hình hóa bằng hỗn hợp các bản phân phối K Gaussian [4]:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \mathcal{N}(\mathbf{X}_t | \mu_{i,t}, \Sigma_{i,t}) \quad (2.7)$$

Trong đó :

$$\mathcal{N}(\mathbf{X}_t | \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma_{i,t}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (\mathbf{x}_t - \mu_{i,t})} \quad (2.8)$$

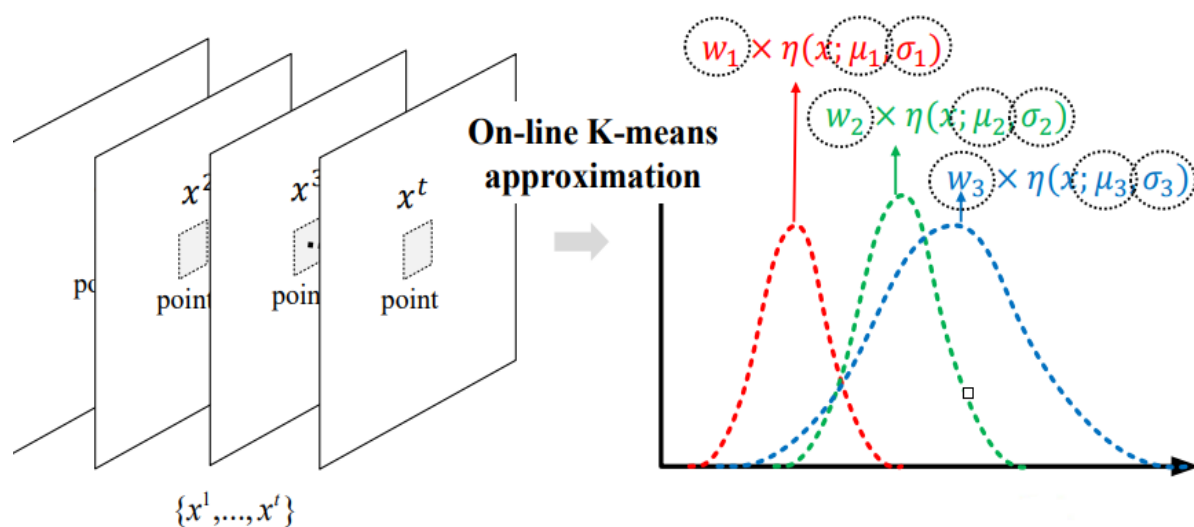
Ví dụ: Gaussian có số cụm là 3 ta có :



Hình 2.13. Mô tả về mô hình Gaussian với số cụm $K = 3$

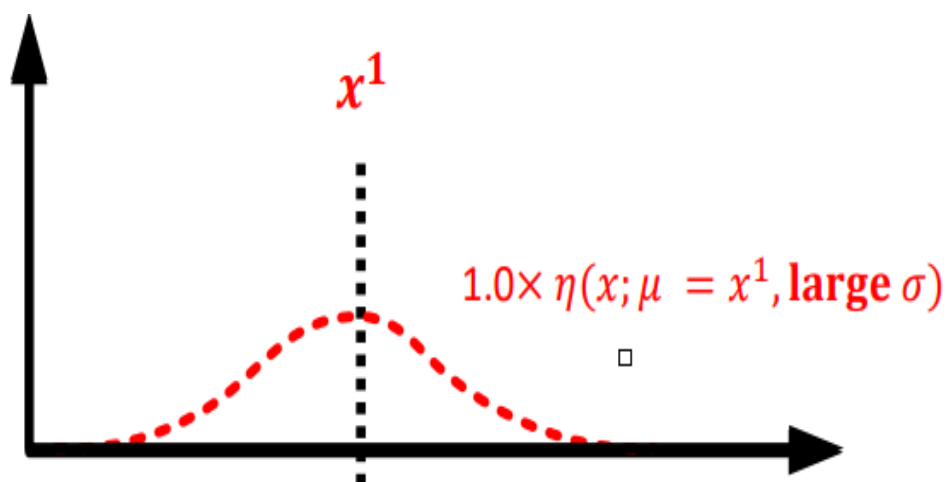
Mô Tả Mô hình :

- + Tất cả các điểm (Pixel) được coi là độc lập lẫn nhau
- + Sự phân bố cường độ của mọi điểm tại bất kì thời điểm nào được mô hình hóa dưới dạng GMM
- + Đối với mục đích tính toán sử dụng phép xấp xỉ K-Mean để mô hình hóa



Hình 2.14. Mô phỏng hỗn hợp Gaussian trên một điểm ảnh (pixel)

+ Khởi Tạo Gaussian để lập mô hình



Hình 2.15. Khởi tạo mô hình Gaussian Mixture

Trong đó : $W = 1$; $\mu_c = x^1$; Độ lệch chuẩn lớn

2.7.2. Thích ứng mô hình (Cập nhật mô hình)

- Tiếp theo dùng thuật toán xấp xỉ K-Mean kiểm tra xem pixel của khung (frame) tiếp theo $[X^t]$ (Với X là pixel đang xét, t là thời tại vị trí X đang xét)]

+ Sẽ có 2 trường hợp xảy ra :

Trường hợp 1 (Case 1): Có ít nhất 1 kết quả phù hợp với Gaussian của pixel đầu tiên và sau đó sẽ cập nhật lại mô hình nghĩa là: Nếu một giá trị pixel mới, X_{t+1} , có thể khớp với một trong các Gaussian hiện có (trong phạm vi $2,5\sigma$). Thì $\mu_{i,t+1}$ và $\sigma_{i,t+1}^2$ của Gaussian đó được cập nhật như sau [4] :

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho X_{t+1} \quad (2.9)$$

Và

$$\sigma_{i,t+1}^2 = (1 - \rho)\sigma_{i,t}^2 + \rho(X_{t+1} - \mu_{i,t+1})^2 \quad (2.10)$$

Trong đó :

$$\rho = \alpha \mathcal{N}(X_{t+1} | \mu_{i,t}, \sigma_{i,t}^2)$$

α : Tỷ lệ học tập

Trọng số trước của các Gaussian được cập nhật như sau [4]:

$$\rho = \alpha \mathcal{N}(X_{t+1} | \mu_{i,t}, \sigma_{i,t}^2) \quad (2.11)$$

Trong đó : $M_{i,t+1} = 1$ Cho Gaussian phù hợp ,và $M_{i,t+1} = 0$ cho tất cả những cái khác

Ví dụ : ($W_{i,t}$, $\mu_{i,t}$, $\sigma_{i,t}$) tại thời điểm X_{t+1} là ($W_{i,t+1}$, $\mu_{i,t+1}$, $\sigma_{i,t+1}$) sẽ được cập nhật như sau :

⇒ Giả sử : ($W_{i,t}$, $\mu_{i,t}$, $\sigma_{i,t}$) = (0.3 , 100 , 16) => tại thời điểm $X_{t+1} = 110$, cho tỷ lệ học tập là : 0.01

⇒ **Áp dụng công thức**

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha(M_{i,t+1})$$

⇒ $W_{i,t+1} = (1-0.01) * 0.3 + 0.01$

$$\Rightarrow W_{i,t+1} = 0.307$$

\Rightarrow **Phương trình trở thành :**

$$(W_{i,t+1}, \mu_{i,t+1}, \sigma_{i,t+1}) = (0.307, \mu_{i,t+1}, \sigma_{i,t+1})$$

Giá trị trung bình trước của các Gaussian được cập nhật như sau [4] :

$$\text{Mean Value : } \mu^{t+1} = (1 - \rho_i)\mu_i^t + \rho_i x^{t+1}$$

$$\rho_i = \alpha \times \eta(x^{t+1}, \mu_i^t, \sigma_i^t) \quad (2.12)$$

Tương tự ví dụ trên ta cập nhật lại giá trị Mean cho Gaussian như sau :

Với : $(W_{i,t}, \mu_{i,t}, \sigma_{i,t}) = (0.3, 100, 16) \Rightarrow$ tại thời điểm $X_{t+1} = 110$, cho tỷ lệ học tập là : 0.01

Áp dụng công thức :

$$\text{Mean Value : } \mu^{t+1} = (1 - \rho_i)\mu_i^t + \rho_i x^{t+1}$$

$$\rho_i = \alpha \times \eta(x^{t+1}, \mu_i^t, \sigma_i^t)$$

Trước hết ta tính :

$$\rho_i = 0.01 * n(X^{t+1} = (110, 100, 16))$$

$$= 0.01 * \frac{1}{\sqrt{2\pi} * 16} \exp\left(-\left(\frac{110-100}{16}\right)^2\right) = 0.00017$$

$$\Rightarrow \mu_{i+1}^t = (1 - 0.00017) * 100 + 0.00017 * 110 = 100.0017$$

\Rightarrow **Phương trình trở thành :**

$$(0.307, \mu_{i,t+1}, \sigma_{i,t+1}) = (0.307, 100.0017, \sigma_{i,t+1})$$

Độ lệch chuẩn trước của các Gaussian được cập nhật như sau [4] :

$$(\sigma_i^{(t+1)})^2 = (1 - \rho_i)(\sigma_i^t)^2 + \rho_i(x^{t+1} - \mu_i^{t+1})^2 \quad (2.13)$$

Tương tự ví dụ trên ta cập nhật lại giá trị độ lệch chuẩn cho Gaussian như sau:

Với : $(w_i, t, \mu_i, t, \sigma_i, t) = (0.3, 100, 16) \Rightarrow$ tại thời điểm $X_{t+1} = 110$, cho tỷ lệ học tập là : 0.01

Áp dụng công thức :

$$\begin{aligned} (\sigma_i^{(t+1)})^2 &= (1 - \rho_i)(\sigma_i^t)^2 + \rho_i(x^{t+1} - \mu_i^{t+1})^2 \\ &= (1 - 0.00017) * 16 * 16 + 0.00017 * (110 - 100.0017)^2 = 255.5523 \end{aligned}$$

\Rightarrow Độ lệch chuẩn là : $\sigma = \sqrt{255.5523} = 15.986$

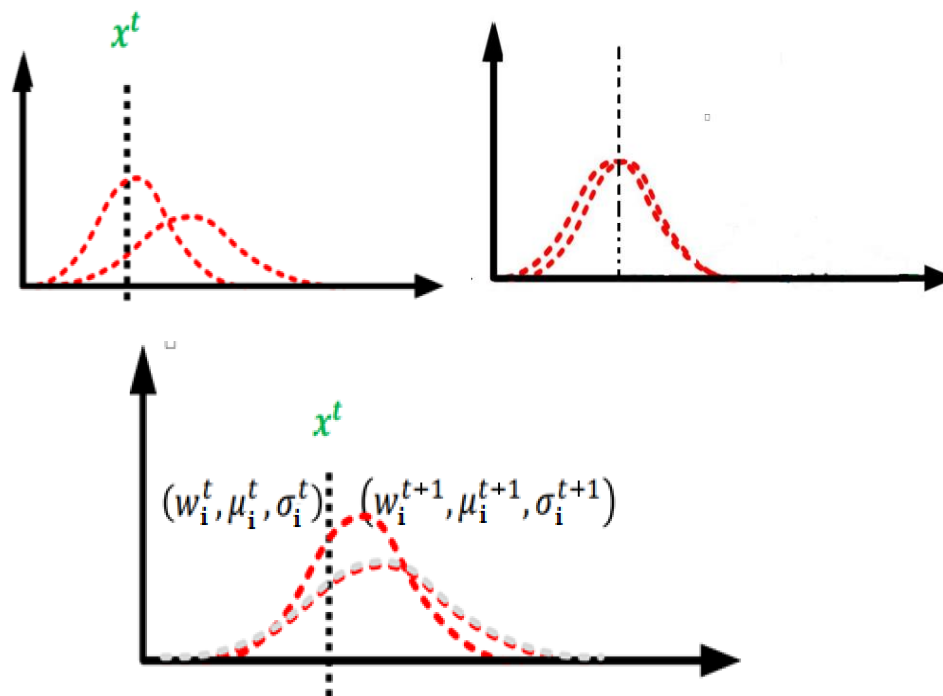
\Rightarrow Phương trình trở thành :

$$(w_i, t+1, \mu_i, t+1, \sigma_i, t+1) = (0.307, 100.0017, 15.986)$$

Vậy tại thời điểm tiếp theo (t+1) thì Gaussian sẽ cập nhật là :

$$(w_i, t+1, \mu_i, t+1, \sigma_i, t+1) = (0.307, 100.0017, 15.986)$$

Kết quả mô phỏng :



Hình 2.16. Kết quả của quá trình cập nhật Gaussian

Trường hợp 2 (Case 2) : Không có kết quả nào phù hợp

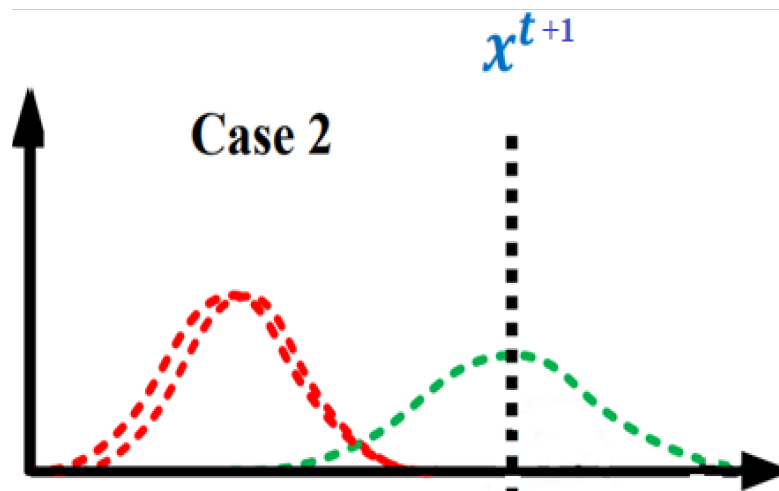
- Nếu X_{t+1} không khớp với bất kỳ Gaussian i nào hiện có, thì phân phối “ít có khả năng nhất” sẽ được thay thế bằng phân phối mới. chúng ta tạo một Gaussian có trọng số thấp nghĩa là bằng $X_t + 1$ và độ lệch chuẩn lớn cho mô hình

+ “ít có khả năng nhất” được xác định bằng ω/σ

Chú ý : ω/σ lớn \Rightarrow background , ω/σ nhỏ \Rightarrow foreground

+ Phân phối mới có $\mu_{t+1} = X_{t+1}$, phương sai cao và trọng số trước đó thấp.

Kết quả mô phỏng :



Hình 2.17. Kết quả của quá trình khởi tạo ra một Gaussian mới

2.7.3 Ước tính mô hình nền

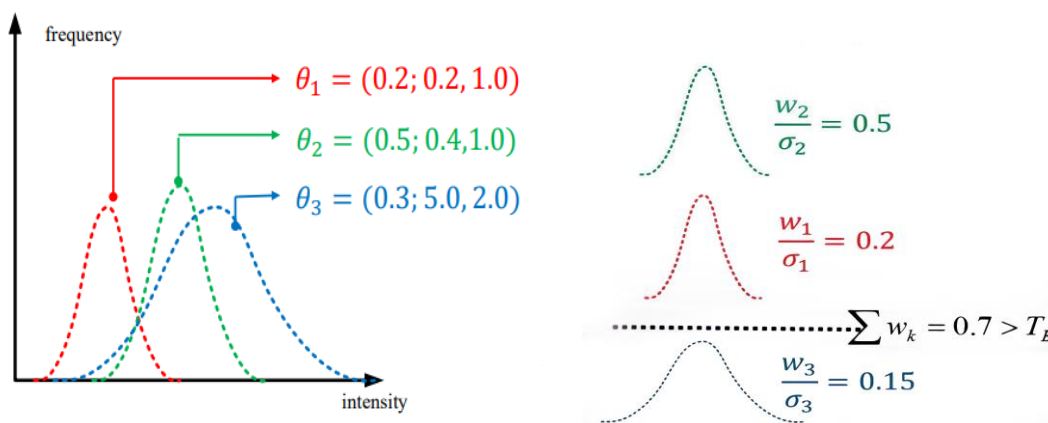
- Bước đầu tiên là chọn Gaussian tương ứng với nền từ I Gaussian đã được xây dựng , Vì cường độ nền là giá trị xuất hiện thường xuyên nên Gaussian nền phải có trọng số cao và phương sai thấp , nên Gaussian được sắp xếp theo giá trị của ω/σ từ cao đến thấp
- Sau đó, đơn giản là các bản phân phối B đầu tiên được chọn làm mô hình nền có tổng trọng số lớn hơn ngưỡng T chọn trước [5] :

$$B = \operatorname{argmin}_b \left(\sum_{i=1}^b \omega_i > T \right) \quad (2.14)$$

+ Trong đó T là phần tối thiểu của hình ảnh dự kiến sẽ làm nền(hay còn gọi là Slection Threshold)

+ Sau khi ước tính mô hình nền , các bản phân phối sẽ biết được phân phối nào sẽ trở thành mô hình nền và phân phối nào được coi là tiền cảnh.

Ví dụ : Có 3 Gaussian và chọn giả sử $T = 0.6$ (Slection Threshold = 0.6)

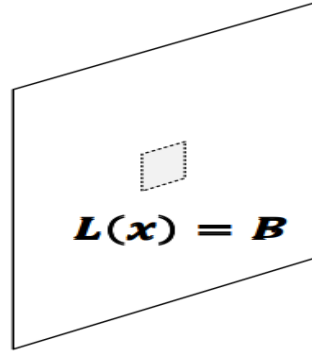


Hình 2.18 .Ví dụ về quá trình lựa chọn mô hình nền

⇒ $\sum w_k = 0.7 > T$ (Slection Threshold = 0.6) ⇒ **B = 0.7** thỏa điều kiện nên sẽ không lấy Gaussian màu xanh , và 2 Gaussian đầu tiên (Đỏ và xanh lá) sẽ thuộc về BackGround

2.7.4 Phân loại điểm ảnh (Pixel)

Ghi nhãn cho pixel $L(x)$



+ **TH 1:** $L(x) = B$, có ít nhất 1 kết quả phù hợp với Background Gaussians

+ **TH2 :** $L(x) = F$, không thuộc về Background hay còn gọi là ForeGround

2.8 Thuật toán MoG2 và một số tham số tác động đến hiệu suất

- Trong báo cáo này chúng ta sẽ dùng phương pháp MoG2 do tỷ lệ phức tạp thấp, tiêu thụ bộ nhớ ít và phù hợp với môi trường ngoài trời cùng với tính mạnh mẽ của nó và nó cũng có thể xử lý các phân phối đa phương thức. Trong MoG2, nền được gọi là khung tham số của các giá trị trong đó mỗi vị trí pixel được biểu thị bằng một số hàm Gaussian dưới dạng hàm phân phối xác suất như được đưa ra ở trên

- Trong MoG2, nền được gọi là khung tham số của các giá trị trong đó mỗi vị trí pixel được biểu thị bằng một số hàm Gaussian dưới dạng hàm phân phối xác suất như được đưa ra :

$$F(i_{t,0} = \mu) = \sum_{i=1}^k \omega_{i,t} \eta(\mu, \sigma) \quad (2.15)$$

Trong đó : η : Thành phần Gauss thứ i

μ : cường độ trung bình

σ : độ lệch chuẩn

$\omega_{i,t}$: dữ liệu được tính bởi thành phần thứ i

- Ngoài ra, đối với phương pháp MoG2, chỉ pixel nằm trong hệ số tỷ lệ của độ lệch chuẩn nền mới được coi là một phần của nền (Background). Điều này có thể được xác định bằng cách so sánh giá trị pixel với theo dõi thành phần Gaussian

Trong phương pháp MoG2, có 5 tham số sẽ tác động đáng kể đến kết quả phát hiện đối tượng [6]:

1. **Background component weight threshold (Ts)**: Ngưỡng trọng lượng thành phần nền
2. **Standard deviation scaling factor (D)**: Hệ số tỷ lệ độ lệch chuẩn
3. **Learning rate (ρ)**: tỷ lệ học tập
4. **Total number of Gaussian components (K)**: Tổng số thành phần Gaussian
5. **Maximum number of components M in the background model (M)**: Số thành phần M tối đa trong mô hình nền

2.9 Thuật toán phân cụm (Thuật toán K-Mean Clustering)

2.9.1 Định nghĩa K-Mean Clustering

- **K-Mean clustering** là một thuật toán phân cụm đơn giản thuộc loại học không giám sát (tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm

2.9.2 Nguyên lý của K-mean Clustering

Ý tưởng của thuật toán phân cụm K-mean:

- Trong thuật toán k-Means mỗi cụm dữ liệu được đặc trưng bởi một tâm (centroid). tâm là điểm đại diện nhất cho một cụm và có giá trị bằng trung bình của toàn bộ các quan sát nằm trong cụm. Chúng ta sẽ dựa vào khoảng cách từ mỗi quan sát tới các tâm để xác định nhãn cho chúng trùng thuộc về tâm gần nhất. Ban đầu thuật toán sẽ khởi tạo ngẫu nhiên một số lượng xác định trước tâm cụm. Sau đó tiến hành xác định nhãn cho từng điểm dữ liệu và tiếp tục cập nhật lại tâm cụm. Thuật toán sẽ dừng cho tới khi toàn bộ các điểm dữ liệu được phân về đúng cụm hoặc số lượt cập nhật tâm chạm ngưỡng.

Cụ thể các bước của thuật toán k-Means được tóm tắt như sau:

- 1.- Khởi tạo ngẫu nhiên K tâm cụm $\mu_1, \mu_2, \dots, \mu_k$

2.- Lặp lại quá trình cập nhật tâm cụm cho tới khi dừng:

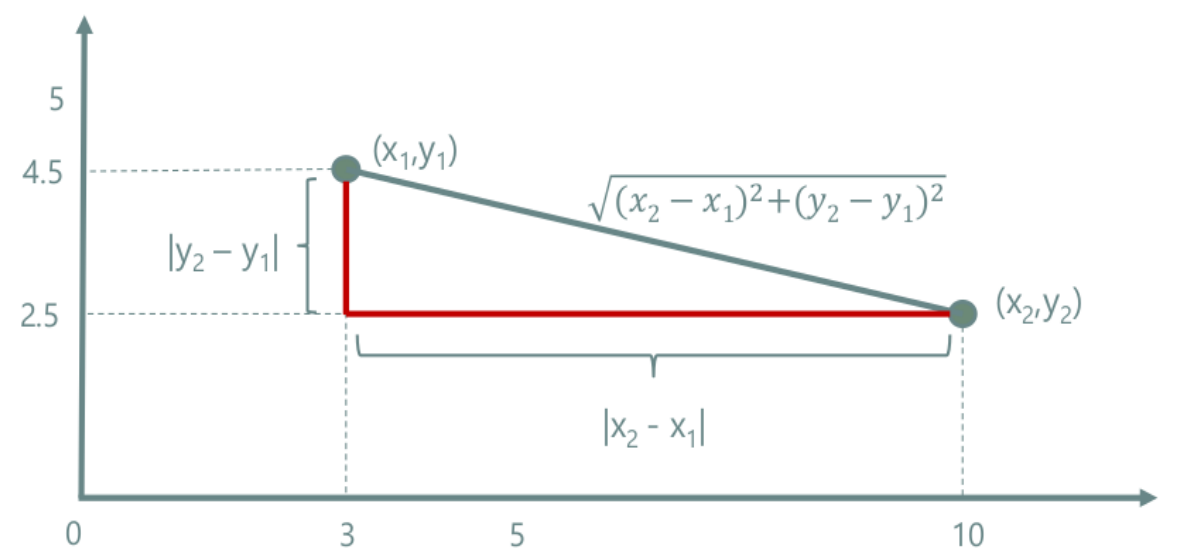
a. Xác định nhãn cho từng điểm dữ liệu c_i dựa vào khoảng cách tới từng tâm cụm [7] :

$$c_i = \arg \min_j \| \mathbf{x}_i - \mu_j \|_2^2 \quad (2.16)$$

b. Tính toán lại tâm cho từng cụm theo trung bình của toàn bộ các điểm dữ liệu trong một cụm [7] :

$$\mu_j = \frac{\sum_{i=1}^n \mathbf{1}(c_i = j) \mathbf{x}_i}{\sum_{i=1}^n \mathbf{1}(c_i = j)} \quad (2.17)$$

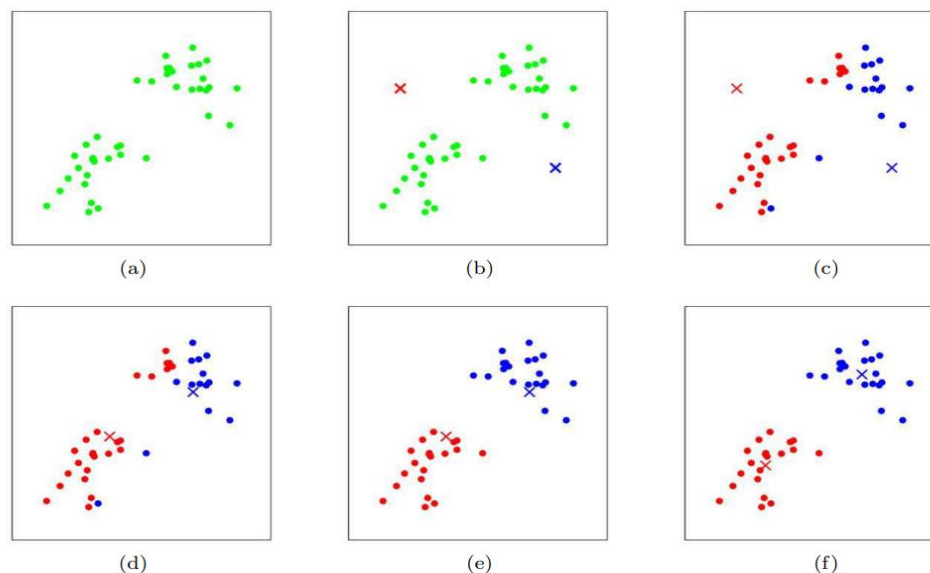
Trong công thức 2.a thì $\| \mathbf{x} \|_2^2$ là bình phương của norm chuẩn bậc 2, kí hiệu là L_2 , norm chuẩn bậc 2 là một độ đo khoảng cách thường được sử dụng trong machine learning.



Hình 2.19. Minh hoạt tính khoảng cách giữa 2 điểm dựa trên Euclidean

Bản chất của 2 bước này là :

- Gán nhãn cho mỗi điểm dữ liệu bằng với nhãn của tâm cụm gần nhất.
- Dịch chuyển dần dần tâm cụm μ_j tới trung bình của những điểm dữ liệu mà được phân về j.



Hình 2.20 .Ảnh minh hoạt ý tưởng của thuật toán K-Means

2.9.3 Kết luận :

-Thuật toán k-Means là phương pháp đơn giản và thường được áp dụng trong các bài toán phân cụm. Thuật toán này dựa trên khoảng cách để cập nhật lại nhãn cho các quan sát về tâm gần nhất và tâm cụm sau đó được tính theo trung bình của toàn bộ các quan sát bên trong cụm. thuật toán sẽ hội tụ sau hữu hạn bước [8].

-Tuy nhiên thuật K-Means vẫn là thuật toán tồn tại những hạn chế đó là cần phải xác định trước tâm cụm, vị trí tâm của cụm chịu sự phụ thuộc vào vị trí khởi tạo ban đầu của chúng, Trong trường hợp các bộ dữ liệu có phân phối phức tạp và mất cân bằng thì thuật toán sẽ không phân cụm chính xác.

2.10 Tìm hiểu về xử lý ảnh

2.10.1 Tổng quan xử lý ảnh

- Xử lý ảnh là đối tượng nghiên cứu của lĩnh vực thị giác máy tính, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới và tuân theo ý muốn của người sử dụng. . Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây

- Xử lý ảnh gồm quá trình phân tích, phân lớp đối tượng, làm tăng chất lượng phân đoạn, tìm biên, gán nhãn cho vùng. Xử lý ảnh được áp dụng rộng rãi trong đời sống và các lĩnh vực khác: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, Photoshop, Nhận dạng khuôn mặt, nhận dạng chữ viết, nhận dạng biển số xe , xử lý ảnh thiên văn, ảnh y tế, ...Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người.

2.10.2 Một số khái niệm cơ bản trong xử lý ảnh

* **Ảnh và điểm ảnh** : Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại 1 toạ độ trong không gian của đối tượng và ảnh được xem như là 1 tập hợp các điểm ảnh

* **Pixel** : Ảnh gồm nhiều điểm ảnh được gọi là pixel . mỗi pixel gồm một cặp toạ độ x, y và màu . cặp toạ độ x,y tạo nên độ phân giải . khi được số hóa thì thường được biểu diễn dưới dạng ma trận 2 chiều có kích thước $N \times M$ (với N là hàng , M là cột) , và ảnh gồm $N \times M = \text{Số Pixel}$

* **Mức xám, màu** : Là số các giá trị có thể có của các điểm ảnh của ảnh , mức xám là cường độ sáng của điểm ảnh , màu là màu của điểm ảnh (R,G,B).

2.10.3 Giới thiệu phép hình thái học trong xử lý ảnh

- **Phép toán hình thái học** là tập hợp các phép toán dùng để phân tích hình ảnh và xử lý tín hiệu số đa chiều theo mẫu , theo cấu trúc được chọn

- **Yếu tố cấu trúc** : Là yếu tố do chúng ta quy định , có thể là bất kì hình dạng nào hoặc kích thước nào

- **Fit** : Tất cả các pixel trong yếu tố cấu trúc sẽ phủ hết các pixel trên ảnh

- **Hit** : Bất kì yếu tố cấu trúc nào phủ lên trên 1 pixel ảnh

- **Miss** : trường hợp còn lại (các pixel trong cấu trúc sẽ không phủ lên trên 1 pixel ảnh nào)

- Điều kiện :

+ Đối tượng là X : ảnh xám hoặc ảnh nhị phân

+ Phần tử cấu trúc S : là mặt nạ bất kì mà các phần tử của nó tạo nên mô típ . Tiến hành rê mặt nạ đi khắp ảnh và tính giá trị điểm ảnh có các điểm lân cận với mô típ của mặt nạ theo cách lấy hội (Phép AND) và tuyển (Phép OR)

+ B_x là phép dịch chuyển vị trí B đến vị trí x

- Mục đích :

+ Nghiên cứu cấu trúc hình học của đối tượng ảnh

+ Đơn giản hóa bằng cách loại bỏ các thông tin không cần thiết

2.10.4 Phép giãn (Nở) ảnh :

Khái niệm :

Phép toán này có tác dụng làm cho đối tượng ban đầu trong ảnh tăng lên về kích thước (giãn nở ra).

Công thức :

$$X \oplus B = \bigcup_{x \in X} B_x \quad (2.18)$$

Trong đó : + X : Ma trận điểm ảnh của ảnh nhị phân.

+ B: Là phần tử cấu trúc. Phép giãn nở (Dilation) ảnh sẽ cho ra hợp các B_x với x thuộc X , dễ dàng thấy rằng đây là một phép tổng giữa X và B.

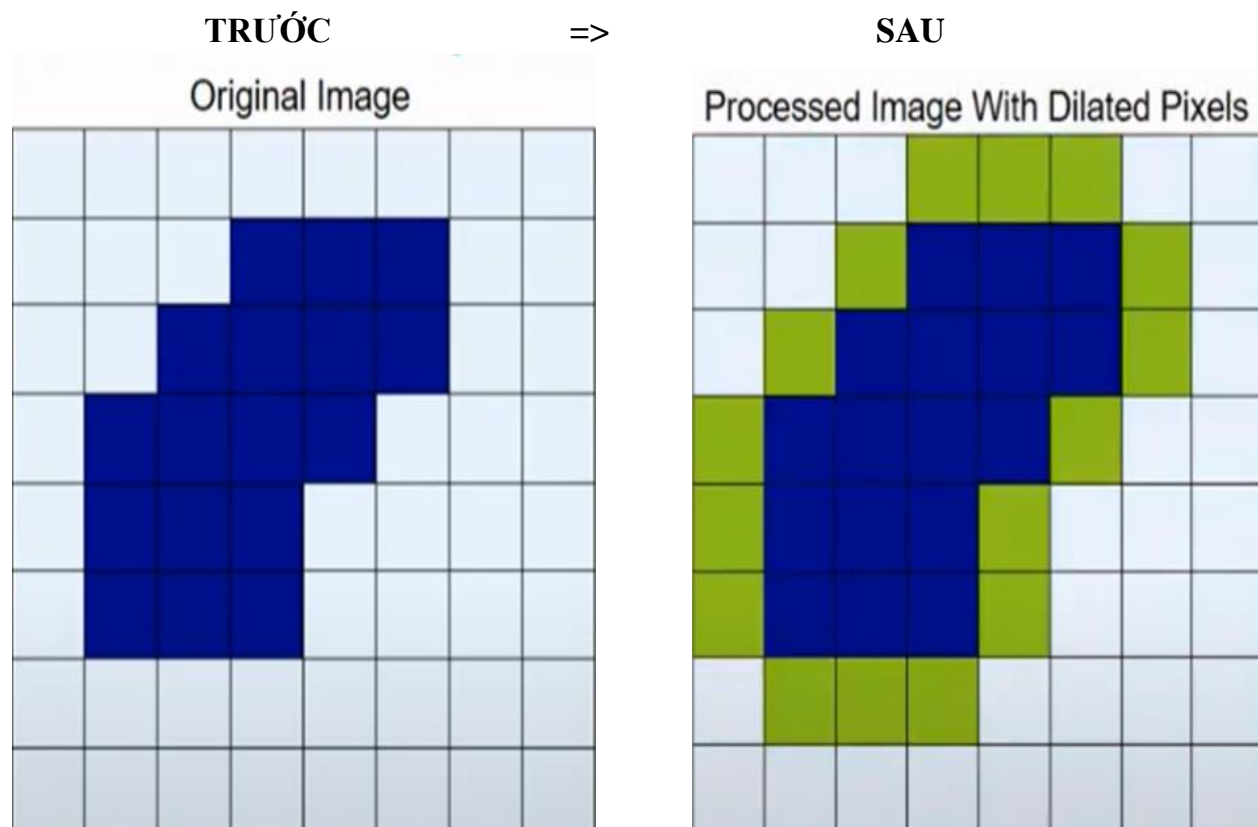
- Ví dụ với phép giãn (Nở) ảnh:

+ Phần tử cấu trúc S:



+ Cách hoạt động :di chuyển đến từng pixel của ảnh nếu trùng 1 phần , hoặc trùng hoàn toàn thì Tâm phần tử cấu trúc chuyển thành pixel thuộc ảnh

+ Kết quả mô phỏng quá trình trước và sau khi giãn ảnh



- Ví dụ với mô phỏng tính toán ảnh dưới dạng ma trận :

Giãn ảnh với ma trận

$$\begin{vmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{vmatrix}$$

Phần tử cấu trúc

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{vmatrix}$$

Kết quả sau khi giãn ảnh dưới dạng ma trận

$$\begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix}$$

2.10.5 Phép co ảnh

Khái niệm :

- Phép toán co (Erosion) là một trong hai hoạt động cơ bản (khác phép giãn nở) trong hình thái học có ứng dụng trong việc giảm kích thước của đối tượng, co hẹp đối tượng, tách rời các đối tượng gần nhau

Công thức :

$$X \ominus B = \left\{ \frac{x}{B_x} \subseteq X \right\} \quad (2.19)$$

Trong đó :

+ X : Ma trận điểm ảnh của ảnh nhị phân.

+ B: Là phần tử cấu trúc. Phép giãn nở (Dilation) ảnh sẽ cho ra hợp các điểm x sao cho B_x thuộc X , dễ dàng thấy rằng đây là một phép tổng giữa X và B.

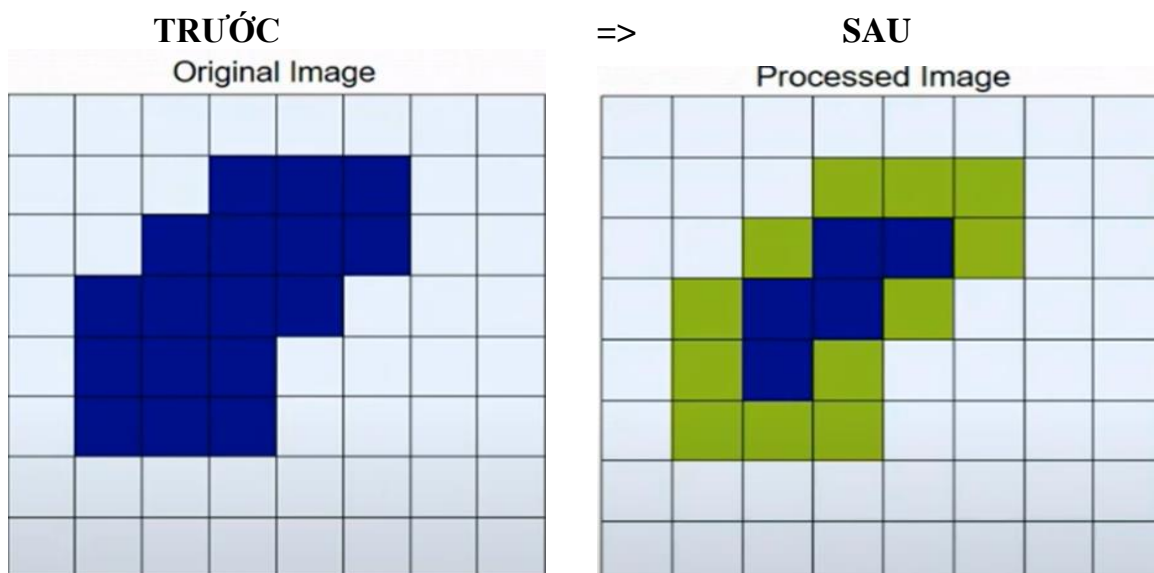
• Ví dụ phép co ảnh:

+ Phần tử cấu trúc S :



+ Cách thực hiện : Bỏ các vị trí trùng một phần

+ Kết quả mô phỏng quá trình trước và sau khi co ảnh



- Ví dụ với mô phỏng tính toán ảnh dưới dạng ma trận

Co ảnh với ma trận

0	0	1	1	1	0	1	0
1	0	0	1	1	1	1	0
0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	0
1	0	1	1	1	0	0	1
1	1	0	1	1	0	0	1
0	1	1	0	0	1	1	0
0	1	0	1	1	1	0	0

Phần tử cấu trúc

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{vmatrix}$$

Kết quả sau khi co ảnh dưới dạng ma trận

$$\begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

2.11 Phương pháp đánh giá mô hình

- Để đánh giá mô hình phát hiện đối tượng, ta sử dụng các thông số như confusion matrix, precision, recall, Average Precision, mean Average Precision.
- IOU (Intersection Over Union) là chỉ số đánh giá được sử dụng để đo độ chính xác của mô hình phát hiện đối tượng trên tập dữ liệu cụ thể. Dựa vào tỉ lệ IOU để đo lường mức độ giao nhau giữa đường bao thực và đường bao dự đoán.

$$IOU = \frac{\text{diện tích phần giao nhau}}{\text{diện tích phần giao nhau và không giao nhau}} \quad (2.20)$$

- + Nếu chỉ số $IOU > 0.5$ thì đối tượng được nhận dạng đúng gọi là True positive (TP).
- + Nếu chỉ số $IOU < 0.5$ thì đối tượng được nhận dạng sai gọi là False positive (FP).
- + Nếu mô hình không dự đoán được đường bao đối tượng đó gọi là False negative (FN).
- Precision là đơn vị dùng để đo lường mức độ chính xác về dự đoán của mô hình hay nói cách khác là tỉ lệ phần trăm dự đoán của mô hình là chính xác.

$$Precision = \frac{TP}{TP + FP} \quad (2.21)$$

- Recall dùng để đo lường độ nhạy của mô hình.

$$Recall = \frac{TP}{TP + FN} \quad (2.22)$$

- Average precision (AP) là chỉ số dùng để đánh giá mô hình dựa trên sự thay đổi của ngưỡng IOU, tức là sẽ chọn N ngưỡng, với mỗi ngưỡng sẽ có một cặp giá trị precision, recall tương ứng là P_n, R_n với $n = 1, 2, \dots, N$ và AP được tính theo công thức sau:

$$AP = \sum_n (R_n - R_{n-1})P_1 \quad (2.23)$$

- Mean Average Precision (mAP) là trung bình của AP đối với tất cả các lớp.

2.12 Ma trận nhầm lẫn (Confusion Matrix)

- Ma trận nhầm lẫn (Confusion Matrix) Là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại:

	Actual Values	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

- Để đơn giản hóa, ta sẽ sử dụng lại bài toán về phát hiện đối tượng chuyển động để giải thích 4 chỉ số này. Trong bài toán phát hiện đối tượng chuyển động có 2 lớp: lớp có đối tượng chuyển động được chuẩn đoán Positive và lớp không có đối tượng chuyển động là Negative
- True Positive (TP): Những đối tượng chuyển động mà ta đoán là có là đang chuyển động .
- True Negative (TN): Những đối tượng ta đoán là không chuyển động đúng là đang không chuyển động
- False Positive (FP): Những đối tượng ta đoán là đang chuyển động ra không chuyển động
- False Negative (FN): Những đối tượng ta đoán là không chuyển động thật ra đang chuyển động
- Từ 4 chỉ số này, ta có 2 con số để đánh giá mức độ tin cậy của một mô hình: Precision: Đây là tỷ lệ giữa những đối tượng thật sự chuyển động so với tất cả các đối tượng được dự đoán là có chuyển động . Chỉ số này được tính theo công thức :

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

- Recall: Trong những đối tượng có chuyển động , bao nhiêu trong số đối tượng được dự đoán đúng bởi mô hình của chúng ta ?. Chỉ số này được tính theo công thức:

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

2.13 Kết luận chương

- **Qua chương này chúng ta biết được những vấn đề sau:**
 - + Các khái niệm cơ bản nhất về Video , công cụ hỗ trợ trong việc lập trình (Python và các thư viện liên quan)
 - + Hiểu được bản chất của một số giải thuật trừ nền , ưu nhược điểm của từng giải thuật
- **Về phương pháp sử dụng để giải quyết bài toán:**
 - + Nắm rõ được phương pháp Gaussian Mixture Model (GMM) và ứng dụng GMM cho trừ nền (BackgroundSubtracsion)
 - + Hiểu được cách mà phương pháp Gaussian Mixture Model xây dựng lên một mô hình và cập nhật mô hình Gauss thông qua một số công thức , ngoài ra còn biết được thuật toán MOG2 và các tham số tác động đến hiệu suất
 - + Nguyên lý hoạt động thuật toán phân loại K-Mean Clustering và kết luận lại thuật toán
- **Về xử lý ảnh :**
 - + Hiểu được các khái niệm cơ bản nhất của một bức ảnh kỹ thuật số
 - + Hiểu được bản chất của các phép hình thái học trong việc xử lý nhiễu sau khi trừ nền bằng Gaussian Mixture Model (Đây là giai đoạn hậu xử lý của bài toán trừ nền)
- Phần cuối chúng ta sẽ tìm hiểu về một phương pháp ứng dụng để đánh giá mô hình phát hiện đối tượng đó là phương pháp Intersection Over Union (IoU) .Cụ thể phương pháp này là xác định thương của phần giao (phép giao giữa phát hiện đối tượng bằng thuật toán với phần phát hiện đúng) và phần hợp (phép hợp nghĩa là phần phát hiện đối tượng bằng thuật toán + phần không bắt được đối tượng của phát hiện đúng). Thường thì đối với phát hiện chuyển động trong video người ta thường đánh giá dựa vào định tính hơn là định lượng và một khái niệm đó là ma trận nhầm lẫn (**Confusion Matrix**)

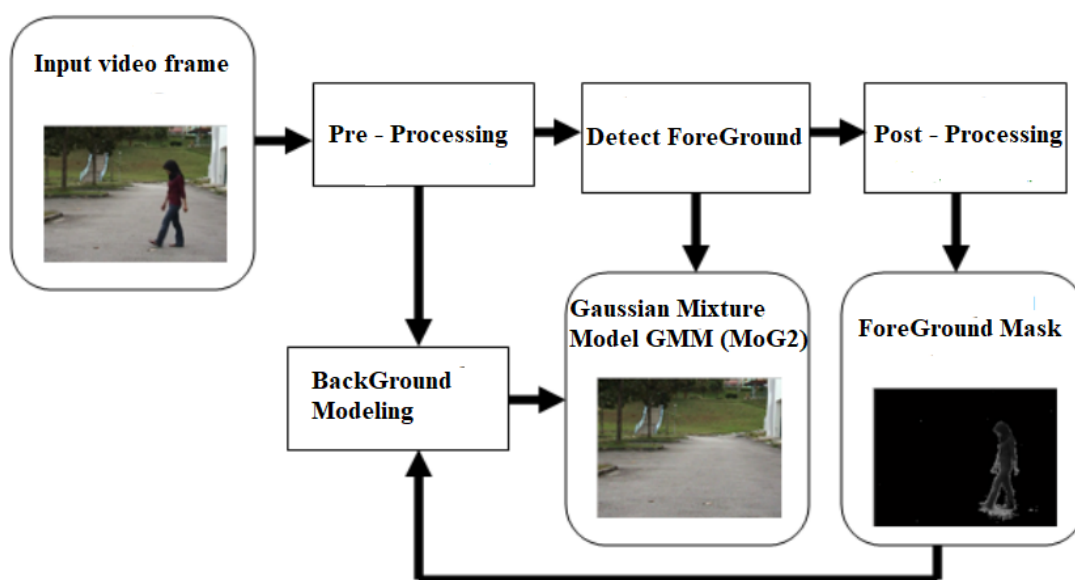
CHƯƠNG 3 : THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1 Giới thiệu chương

- Ở chương này chúng ta sẽ tìm hiểu :

- + Quy trình trừ nền và lưu đồ thuật toán trừ nền sử dụng phương pháp **Gaussian Mixture Model (GMM)**
- + Phân tích chi tiết lưu đồ thuật toán trừ nền của **Gaussian Mixture Model (GMM)**:
 - Khởi tạo mô hình Gauss
 - Huấn luyện mô hình GMM cho từng Frame (Khung hình)
- + Tìm hiểu về hàm **cv2.createBackgroundSubtractorMOG2()** thư viện hỗ trợ việc xây dựng GMM trong python
- + Giới thiệu một số bài toán ứng dụng trong xử lý video bằng phương pháp trừ nền, trình bày lưu đồ thuật toán và hướng giải quyết các bài toán trong mục tiêu đã đề ra ở chương 1

3.2 Sơ đồ khối của quy trình trừ nền sử dụng GMM



Hình 3.1 Quy trình trừ nền sử dụng Gaussian Mixture Model

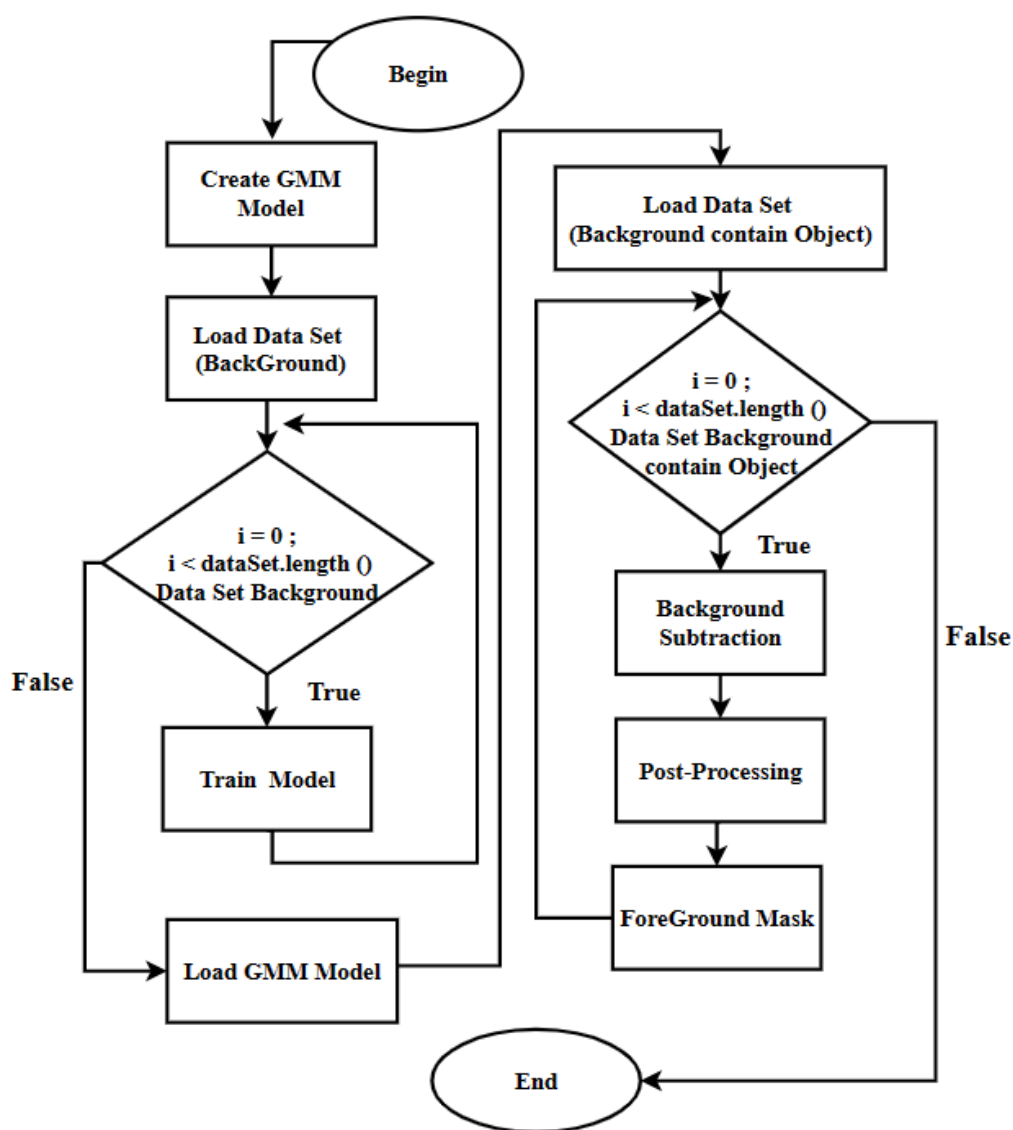
- + **Input video frame**: Đọc vào từng frame của video
- + **Pre-Processing (khởi tiên xử lý)**: Được sử dụng trong giai đoạn đầu xử lý để giảm nhiễu là cần thiết trước khi tạo mô hình nền
- + **BackGround Modelling (Mô hình nền)**: Đây là trọng tâm của bất kỳ thuật toán trừ nền nào nó sẽ dựng lên 1 mô hình nền
- + **Gaussian Mixture Model (MoG2)**: Sau khi dựng lên một mô hình nền thì khối này sẽ là nền (Background) mục đích để khi đọc vào frame tiếp theo thì khối Detect Foreground sẽ so sánh với mô hình này và tiến hành phát hiện (detect) và đưa qua khối Post-Processing
- + **Detect ForeGround** : (Xác thực dữ liệu bằng cách thực hiện trừ frame hiện tại với mô

hình nền trước đó (Khối Gaussian Mixture Model GMM (MoG2)) để đưa ra những pixel không xác định trong khung hình video ở mô hình nền, những pixel này sẽ xuất hiện ở mặt nạ tiền cảnh (Foreground Mask) dưới dạng nhị phân) có thể hiểu là : Phát hiện vật thể chuyển động

+ **Post – Processing (Khối hậu xử lý)**: Hậu xử lý với mục đích là khử nhiễu trước khi đưa ra mặt nạ tiền cảnh (Foreground Mask) mục đích để detect đối tượng đạt được chính xác nhất

+ **Foreground Mask (Mặt nạ tiền cảnh)**: Đưa ra mặt nạ tiền cảnh và quá trình trừ nền đã thành công

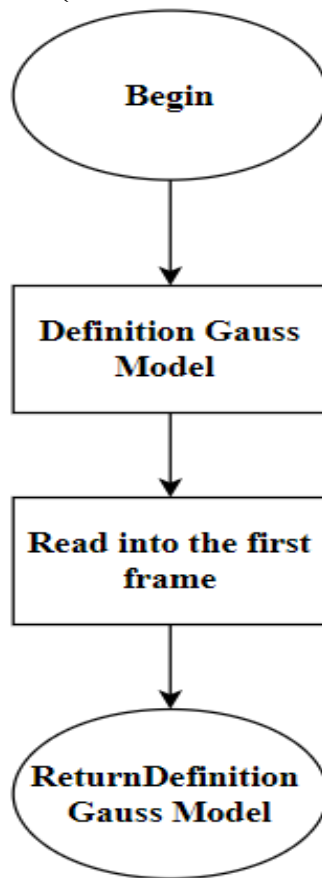
3.3 Lưu đồ thuật toán trừ nền sử dụng GMM



Hình 3.2 Lưu đồ thuật toán của quá trình trừ nền

3.3.1 Phân tích lưu đồ thuật toán từng khối

a) Khởi tạo mô hình Gauss (Create GMM Model)

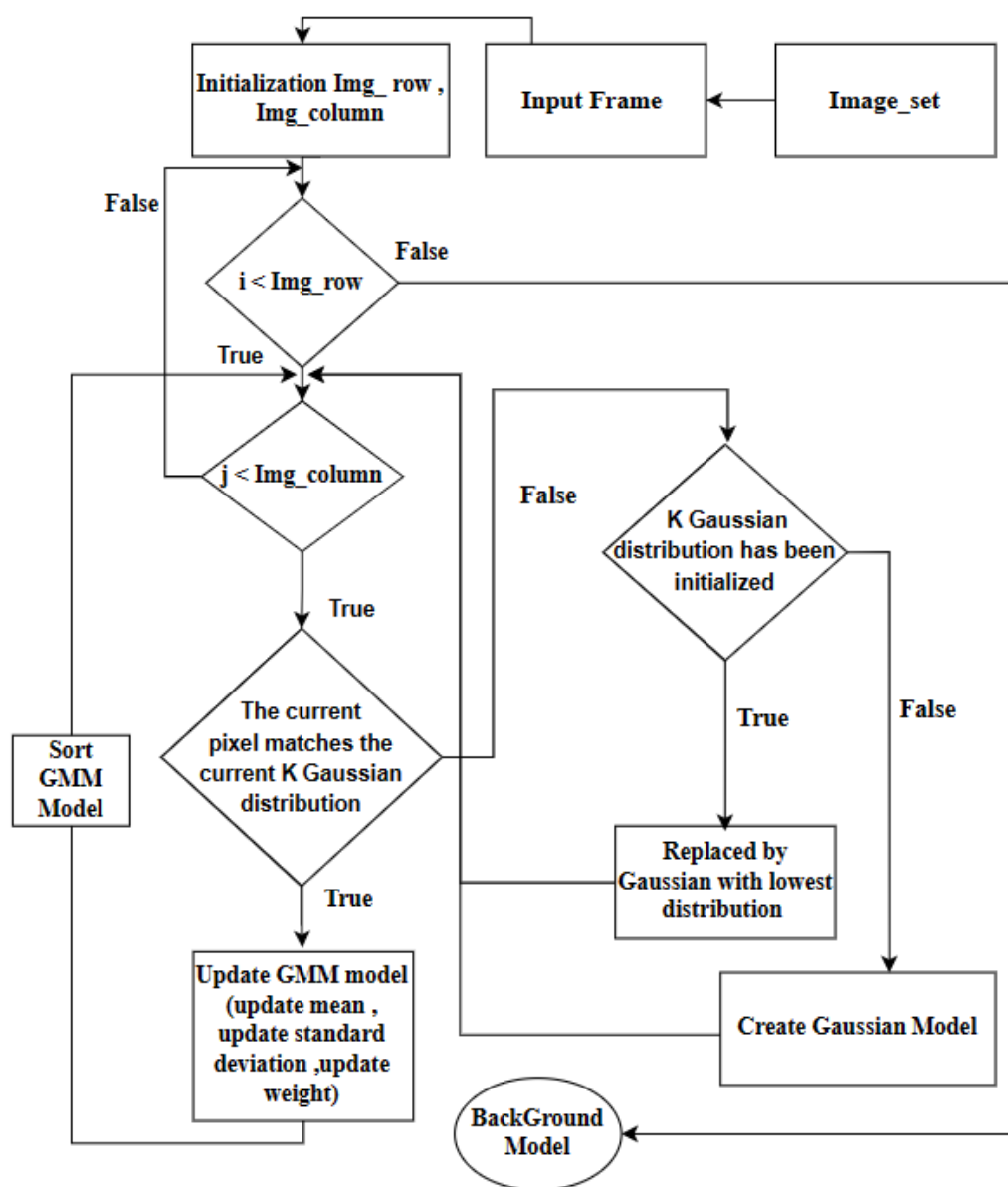


Hình 3.3 Lưu đồ thuật toán của quá trình khởi tạo mô hình Gauss

- Các khối bao gồm :

- + Khối Definition Gauss Model : khối này dùng để định nghĩa các thông số Gauss ban đầu ví dụ như : Kích thước ảnh (số điểm ảnh (Pixel) , số lượng mô hình Gauss trên 1 frame) , Khởi tạo số lượng mô hình trên một điểm ảnh (Pixel) , Khởi tạo ngưỡng () , khởi tạo giá trị trung bình , độ lệch chuẩn , trọng số , tỷ lệ học tập .
- + Khối Read into the first Frame : khối này dùng để đọc vào frame đầu tiên khởi tạo nên GMM

b) Huấn luyện mô hình (Train Model)



Hình 3.4 Lưu đồ thuật toán của quá trình huấn luyện mô hình Gauss

3.4 Sử dụng hỗn hợp Gaussian 2 (MoG2) trong OpenCV

- Phương pháp MOG2 dựa trên các nghiên cứu với mục tiêu giải quyết một trong những hạn chế mà gặp phải: số lượng phân phối được sử dụng cố định. Bằng cách sử dụng một biến lượng phân phối Gaussian, ánh xạ từng pixel, MOG2 đạt được sự thể hiện tốt hơn về độ phức tạp của màu sắc trong mỗi khung hình.
- Khi triển khai OpenCV, MOG2 có ba tham số đầu vào có thể thay đổi để hiệu chỉnh cho từng video khác nhau. Các tham số và giá trị mặc định của chúng có thể được nhìn thấy trên Hình 3.5

```
cv2.createBackgroundSubtractorMOG2(  
                                     history=200,  
                                     varThreshold=16,  
                                     detectShadows=True  
                                   )
```

Hình 3.5 Hàm khởi tạo MOG2 với các tham số mặc định trong OpenCV

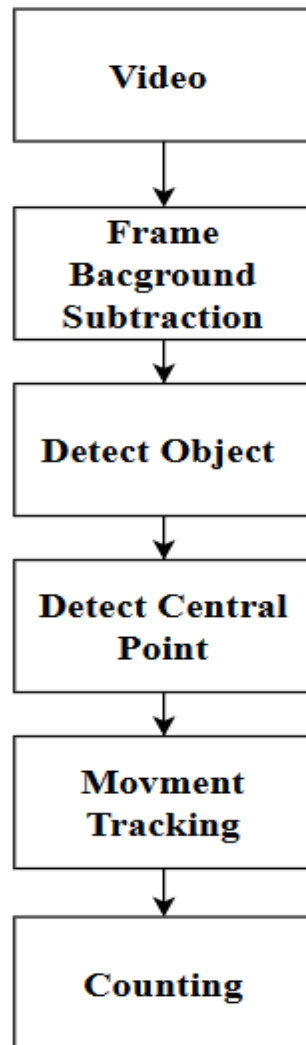
- + Chức năng tham số lịch sử (history = 200) sẽ biểu thị số lượng khung hình được sử dụng để mô hình hóa nền ở đây là mặc định sẽ dùng 200 khung hình để mô hình hóa nền. [16]
- + Chức năng tham số (varThreshold) sẽ thể hiện tương quan giá trị trọng số của các pixel trên khung hình hiện tại với các giá trị trên mô hình nền. Các giá trị thấp hơn trên tham số này có xu hướng tạo ra các đối tượng sai [16]
- + Chức năng tham số (detectShadows) bật hoặc tắt tính năng phát hiện bóng. Kích hoạt tham số này làm tăng thời gian xử lý[16]

3.5 Các bài toán thực hiện

3.5.1 Phát hiện xe ô tô chuyển động trên đường và thực hiện theo dõi (tracking) và đếm số lượng xe trên đường sử dụng mô hình GMM

- Bài toán phát hiện xe chuyển động là bài toán được áp dụng mô hình GMM để xử lý . Đồng thời sử dụng một số thuật toán để tiến hành phát hiện (detect) được xe tiến hành theo dõi (tracking) và đếm số lượng xe

- Sơ đồ thuật toán:

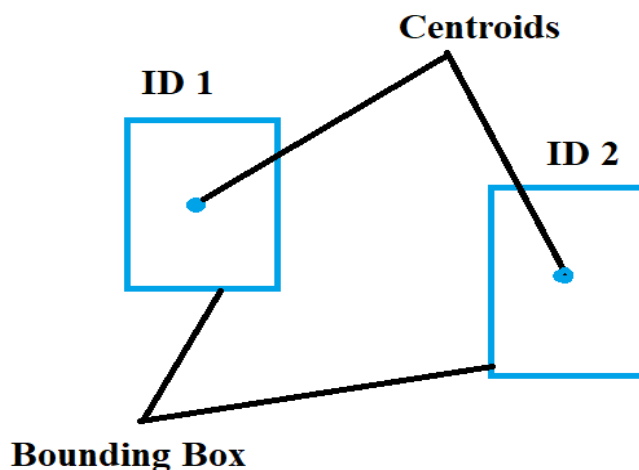


Hình 3.6 Sơ đồ thuật toán hiện theo dõi (tracking) và đếm số lượng xe trên đường

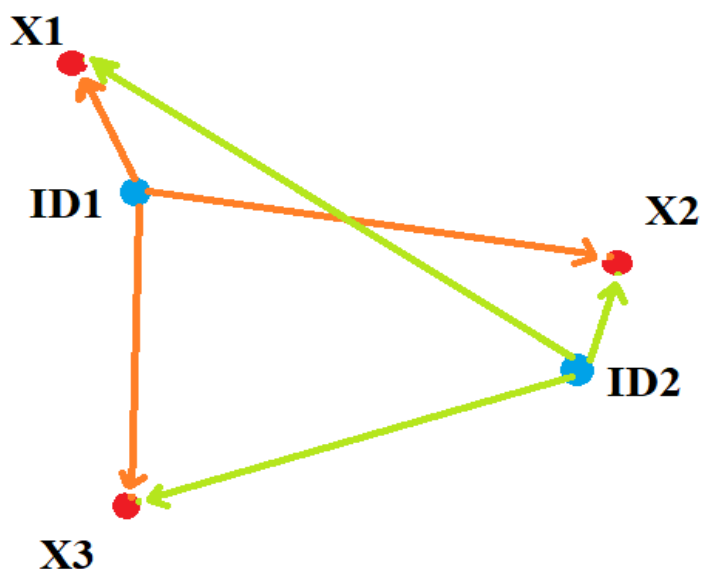
- Khối Frame Background subtraction: Đọc các khung hình sau khi đã trừ nền
- Khối Detect Object : Phát hiện đối tượng chuyển động dựa vào vùng tham chiếu trên khung hình
- Khối Detect Central Point : Xác định tâm của đối tượng
- Khối Movement Tracking : Theo dõi tâm đã đánh dấu
- Khối Counting: Xác định hướng của tâm để tiến hành đếm tùy thuộc vào hướng di chuyển

3.5.1.2 Thuật toán theo dõi (Tracking) và đếm số lượng xe :

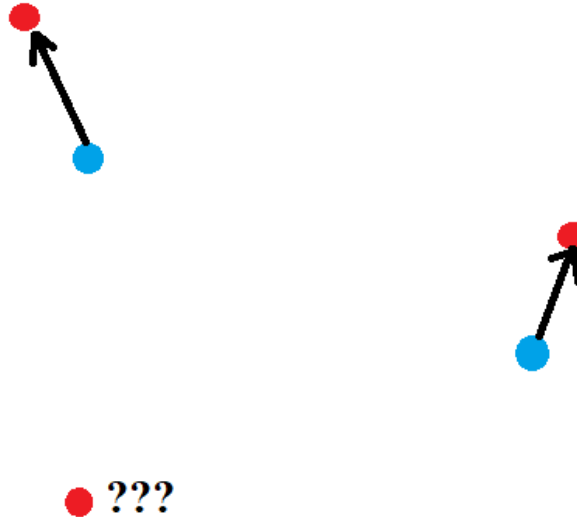
- Bước 1 : Phát hiện các đối tượng trong khung hình hiện tại và vẽ bounding box sau đó tìm trọng tâm của đối tượng



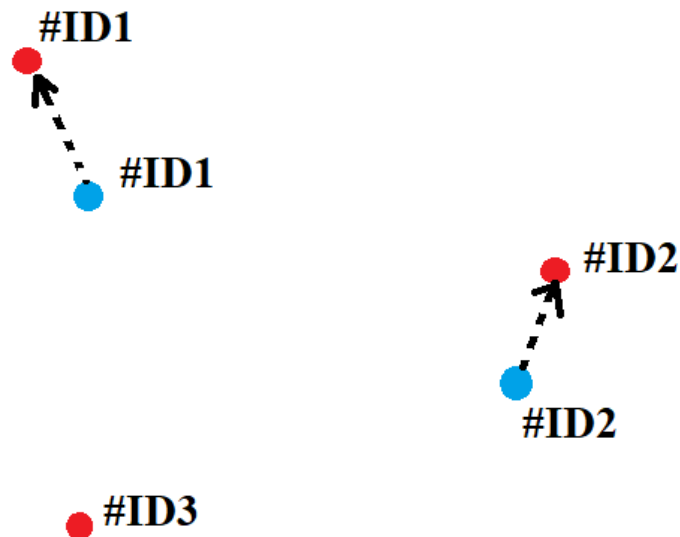
- Bước 2: Tính khoảng cách Euclidean giữa trọng tâm của các đối tượng cũ và đối tượng mới có trong khung hình hiện tại, như hình dưới ta thấy ID1 ID2 là 2 ID có trong khung hình trước đó, và khung hình tiếp theo nó phát hiện ra 3 đối tượng X1,X2,X3, ý tưởng của thuật toán này xét khoảng cách từ ID của khung hình trước đó đến vị trí của 3 đối tượng trong khung hình hiện tại nếu khoảng cách nào có ngắn hơn thì đối tượng đó sẽ thuộc về ID trước đó
 + Giả sử khi ID1 có khoảng cách gần X1 hơn X2 và X3 nên vị trí X1 ở frame hiện tại chính là ID1 của frame trước đó, tương tự X2 sẽ là của ID2



- Bước 3 : Từ bước tính khoảng cách ở bước 2 thì X3 nó không thuộc ID1 và ID2 nên khi đó X3 sẽ được coi là đối tượng có ID mới và nó sẽ khởi tạo nên một đối tượng mới . được mô phỏng như hình dưới



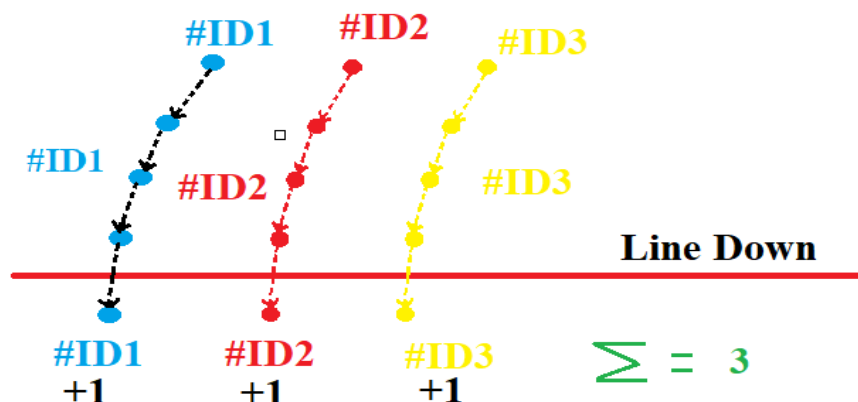
- Bước 4 : Cập nhật lại danh sách và vị trí các đối tượng cần theo dõi và lưu vào danh sách



- Bước 5: Xóa các đối tượng biến mất sau một thời gian nhất định

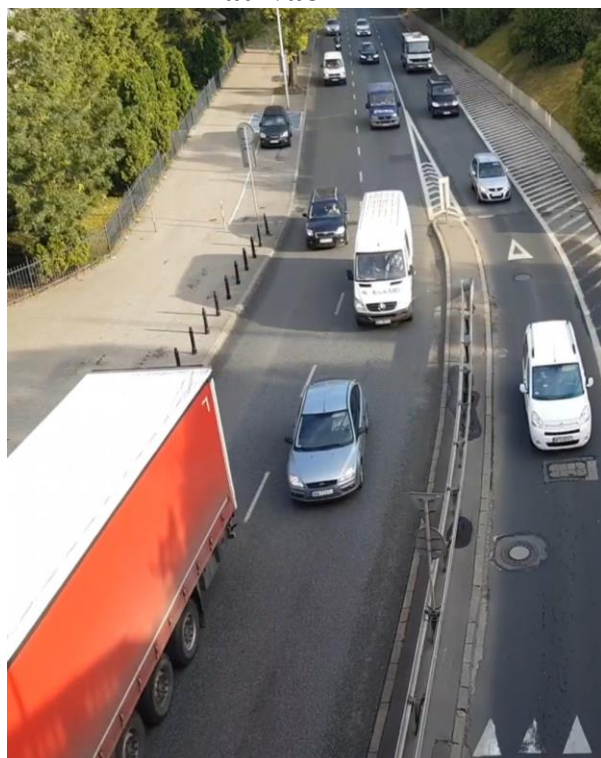
- Bước 6: Thực hiện vẽ một đường ranh giới (Line Down) để tiến hành đếm , cứ nếu ID vượt qua đường ranh giới thực hiện đếm lên một. Mô phỏng như hình dưới:
- Giả sử có 3 ID vượt qua đường ranh giới (Line Down) thì có tổng:

$$\sum ID = 1 + 1 + 1 = 3 \Rightarrow \text{Có 3 đối tượng.}$$

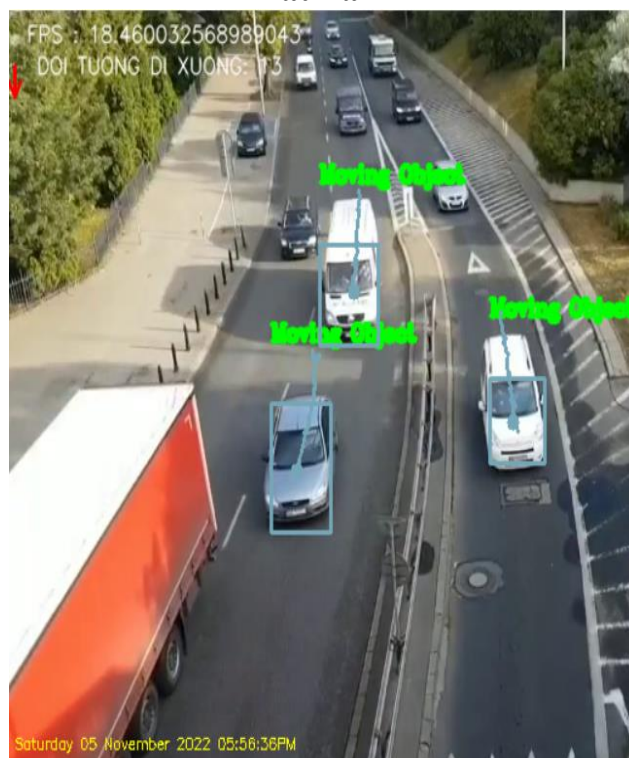


3.4.1.3 Kết quả mô hình

Đầu vào



Đầu ra

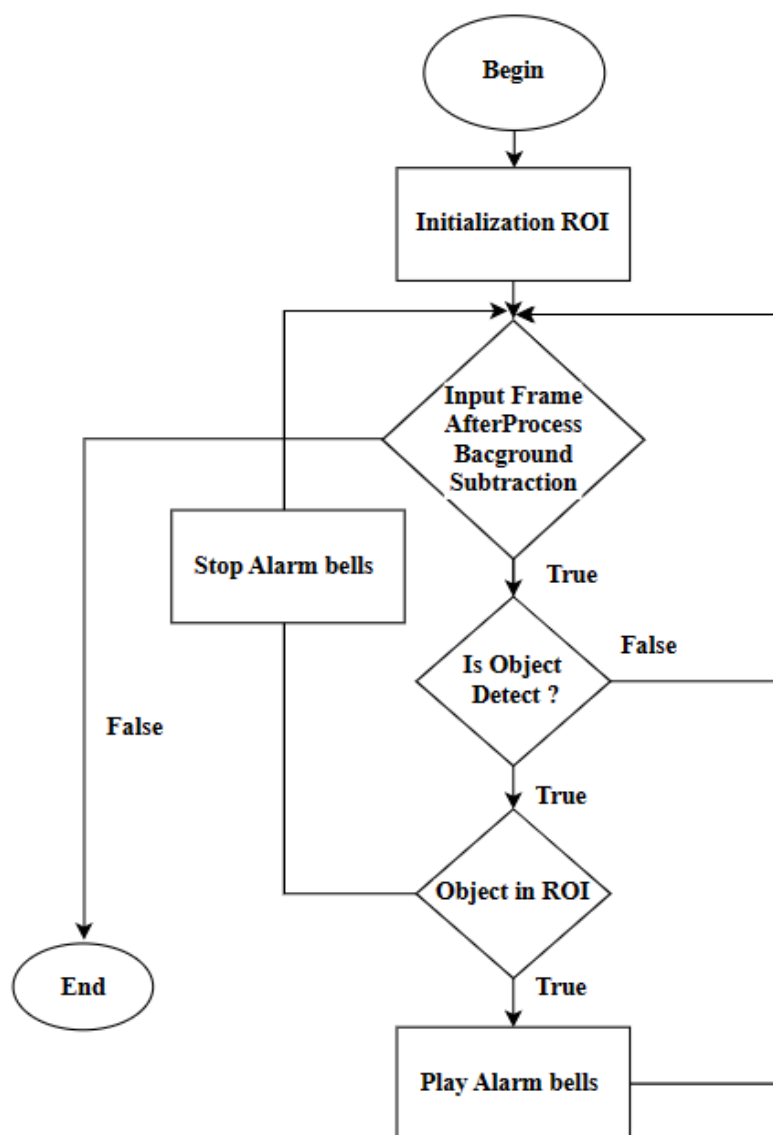


Hình 3.7 Kết quả hoạt động của mô hình GMM đối với bài toán phát hiện chuyển động , theo dõi và đếm xe

3.5.2 Phát hiện trộm di chuyển vào trong nhà và thực hiện báo động

- Bài toán phát hiện trộm chuyển động vào ban đêm là bài toán được áp dụng mô hình GMM để xử lý. Đồng thời sử dụng một số thuật toán để tiến hành detect được đối tượng và tiến hành phát âm thanh cảnh báo

3.5.2.1 Lưu đồ thuật toán phát hiện cảnh báo khi có trộm vào nhà:



Hình 3.8 Lưu đồ thuật toán phát hiện cảnh báo khi có trộm vào nhà

3.5.2.2 Kết quả mô hình:



Hình 3.9 Kết quả hoạt động của mô hình GMM đối với bài toán phát hiện trộm vào nhà , phát chuông cảnh báo

3.6 Đánh giá mô hình

Bảng 3.1 Bảng đánh giá kết quả mô hình khi thực hiện thử nghiệm trên 15 video dựa vào định tính

STT video test	Chỉ số ma trận nhầm lẫn	FN (False Negative)	FP (False Positive)	TP (True Positive)
Video 1		0	1	1
Video 2		0	1	1
Video 3		1	1	1
Video 4		0	1	1
Video 5		1	1	1
Video 6		0	1	1
Video 7		0	0	1
Video 8		0	1	1
Video 9		0	0	1
Video 10		0	0	1
Video 11		0	1	1
Video 12		1	0	1
Video 13		1	0	1
Video 14		0	1	1
Video 15		1	0	1

3.6.1 Nhận xét đánh giá

Ta Thấy sau khi thử nghiệm với 15 video khác nhau thì mô hình cho kết quả là :

+ FN = 5 $\Rightarrow 5/15 = 30\% \Rightarrow$ Nghĩa là cứ trong 15 video để xử lý thì có 5 video chứa một vài đối tượng chuyển động mà hệ thống không phát hiện được

+ FP = 9 $\Rightarrow 9/15 = 60\% \Rightarrow$ Nghĩa là cứ trong 15 video để xử lý thì có 9 video tại một vài vùng không có đối tượng chuyển động mà nhận nhầm thành có đối tượng chuyển động

+ TP = 15 $\Rightarrow 15/15 = 100\% \Rightarrow$ Nghĩa là tất cả video được đưa vào để xử lý thì tất cả đều có thể phát hiện được chuyển động

3.6.2 Kết Luận

+ Đối với chỉ số FN (**False Negative**) : Chiếm tỷ lệ 30% trong 15 video được đưa vào dự đoán ban đầu và trong thực tế tất cả 15 video này đúng là đang chuyển động thật và FN chiếm 30% tức 5 video trong tổng số 15 video dự đoán có chứa đối tượng chuyển động bị bỏ sót .Nguyên nhân dẫn đến chỉ số này chiếm tỷ lệ lên đến 30% là do các tác động từ bên trong video:Như diện tích đối tượng nhỏ hoặc nhiều đối tượng chuyển động khít gần nhau sẽ bị bỏ sót đi một vài đối tượng

+ Đối với chỉ số FP (**False Positive**) : Chiếm tỷ lệ 60% trong 15 video được đưa vào dự đoán ban đầu và trong thực tế tất cả 15 video này đúng là đang chuyển động thật và FP chiếm 60% tức 9 video trong tổng số 15 video dự đoán có chứa một vài vùng không chuyển động bị nhận nhầm thành có chuyển động .Nguyên nhân dẫn đến chỉ số này chiếm tỷ lệ cao là do các tác động từ môi trường bên ngoài: Như cường độ ánh sáng thay đổi đột ngột , hoặc tác động do camera bị lung lay dẫn đến nền chuyển động thì thuật toán lại không thể hiện tốt trong trường hợp này

+ Đối với chỉ số TP (**True Positive**) : Chiếm tỷ lệ 100% trong 15 video được đưa vào dự đoán ban đầu và trong thực tế tất cả 15 video này đúng là chứa đối tượng đang chuyển và hệ thống cũng phát hiện chuyển động tuy nhiên khi thời gian thay đổi hoặc video gặp sự cố bất ngờ thì nó sẽ xảy ra hai trường hợp là FN và FP như được nêu ra ở trên.

Kết Luận Và Hướng Phát Triển

1. Kết quả đạt được:

- Nhóm đã thực hiện trừ nền bằng phương pháp MoG2 sử dụng hàm có sẵn trong thư viện Python với mục đích để tăng độ chính xác và hiệu suất của thuật toán từ đó nâng cao tính ứng dụng vào thực tiễn của đề tài .
- Tuy nhóm sử dụng hàm có sẵn của thư viện OpenCV trong Python nhưng vẫn đảm bảo nắm rõ về mặt lý thuyết , bản chất , quy trình thực hiện của phương pháp MOG2 này.
- Qua chương này nhóm đã đạt được những kết quả như sau:
 - Hiểu được quy trình trừ nền , thuật toán trừ nền bằng phương pháp GMM , và cách khởi tạo nên một mô hình GMM này thông qua hàm **cv2.createBackgroundSubtractorMOG2(200, 16, True)**.
 - Về bài toán phát hiện chuyển động , theo dõi và đếm số lượng xe . Từ phương pháp MoG2 nhóm đã xây dựng thành công hệ thống phát hiện đối tượng chuyển động trong video , theo dõi được xe đang di chuyển , và đếm được số lượng xe đang di chuyển trên đường :
 - Về khả năng phát hiện chuyển động, hệ thống hoạt động và bắt được hầu hết các đối tượng chuyển động
 - Về khả năng theo dõi , hệ thống hầu như đã theo vết đối tượng
 - Về khả năng đếm xe: hệ thống đạt kết quả đếm được xe
 - Tuy nhiên, về khả năng phát hiện thì khi xe đến gần hệ thống mới phát hiện được tốt và thực hiện theo vết đối tượng , còn khi xe ở xa thì hệ thống chưa phát hiện và theo vết được , và đôi khi phát hiện sai do nền chuyển động . Và khả năng đếm xe ở một số trường hợp xe bị bỏ sót
 - Về bài toán phát hiện chuyển động , cảnh báo khi có trộm vào nhà. Từ phương pháp trừ nền bằng MoG2 nhóm cũng đã thực hiện xây dựng thành công hệ thống phát hiện đối tượng chuyển động trong video vào ban đêm , phát hiện được trộm đang di chuyển , và phát chuông cảnh báo:
 - Về khả năng phát hiện chuyển động, hệ thống hoạt động tốt , phát hiện đối tượng khi đối tượng di chuyển vào trong khu vực cấm
 - Về khả năng cảnh báo : Phát ra chuông cảnh báo khi đối tượng (Trộm) đi vào nhà
 - Tuy nhiên, về khả năng phát hiện đối tượng còn gặp một số vấn đề về nhiễu dẫn đến khung phát hiện không bao được hết đối tượng mà bị tách thành nhiều khung khác nhau có thể là do chỉ tay, chân hoặc đầu của đối tượng hoạt động và thân đối tượng không hoạt động nên hệ thống sẽ đưa về nền nên xảy ra hiện tượng như vậy và một điều gặp phải là đối tượng ra khỏi rồi mà chuông vẫn còn reo cảnh báo

2. Hướng phát triển :

- Dựa trên phương pháp MoG2 để xử lý video , trong tương lai, hệ thống có thể cải tiến theo các hướng như sau:
- Cải thiện độ phát hiện chuyển động chính xác cho hệ thống :
 - + Đầu tiên ta cần cải thiện xử lý khung (frame) đầu vào trước khi trừ nền bằng MoG2 vì đây là bước tiền xử lý rất quan trọng trong quá trình xây dựng và cập nhật lại mô hình nền .
 - + Thứ hai là ta cần phải thực hiện các bước ở phía hậu xử lý ví dụ như áp dụng các thuật toán chọn ngưỡng cho phù hợp với từng khung (Frame), xử lý giảm nhiễu ở mặt nạ đầu ra sau quá trình trừ nền bằng MoG2 một cách hiệu quả
- Đối với bài toán phát hiện chuyển động , theo dõi và đếm số lượng xe có thể thêm chức năng:
 - + Phát hiện xe vi phạm giao thông: Bằng cách giám sát tốc độ của xe có thể cắt và lưu biển số của xe đó vào hệ thống
 - + Đối với chức năng đếm xe thì cần cải thiện đếm xe theo hai chiều (số lượng xe đi lên , số lượng xe đi xuống) thay vì đếm số lượng xe đi một chiều ở video demo
- Đối với bài toán phát hiện chuyển động ban đêm, cảnh báo khi có trộm vào nhà:
 - + Có thể phân biệt được đâu là trộm thật và đâu là chủ nhà để nâng cao tính ứng dụng thực tiễn của bài toán này

Phụ Lục

- Bài toán bài toán phát hiện chuyển động , theo dõi và đếm xe :

Object:

```
from random import randint

class Object:
    tracks = []

    def __init__(self, i, xi, yi, max_age):
        self.i = i
        self.x = xi
        self.y = yi
        self.tracks = []
        self.R = randint(0, 255)
        self.G = randint(0, 255)
        self.B = randint(0, 255)
        self.done = False
        self.state = '0'
        self.age = 0
        self.max_age = max_age
        self.dir = None

    def getRGB(self):
        return self.R, self.G, self.B

    def getTracks(self):
        return self.tracks

    def getId(self):
        return self.i

    def getState(self):
        return self.state

    def getDir(self):
        return self.dir

    def getX(self):
        return self.x

    def getY(self):
        return self.y

    def updateCoords(self, xn, yn):
        self.age = 0
        self.tracks.append([self.x, self.y])
        self.x = xn
        self.y = yn
```

```

def setDone(self):
    self.done = True

def timedOut(self):
    return self.done

def going_DOWN(self, mid_start):
    if len(self.tracks) >= 2:
        if self.tracks[-1][1] > mid_start and self.tracks[-2][1] <=
mid_start: # nếu vượt qua ranh giới
            self.dir = 'down'
            return True
        else:
            return False
    else:
        return False

def age_one(self):
    self.age += 1
    if self.age > self.max_age:
        self.done = True
    return True

```

Hàm (Function) phát hiện chuyển động , theo dõi và đếm xe:

```

import time

import cv2
import numpy as np

import Object

# argument parsing
def Detect_Tracking(vd):
    cap = cv2.VideoCapture(vd)
    # Print the capture properties to console, height, width and FPS
    print('Height: ', cap.get(4))
    print('Width: ', cap.get(3))
    print('Frame per Seconds: ', cap.get(5))

    cnt_up = 0
    cnt_down = 0
    w = cap.get(3)
    h = cap.get(4)

    # Entry / exit lines
    line_down = int((h / 2))

    pt1 = [0, line_down]
    pt2 = [w, line_down]
    pts_L1 = np.array([pt1, pt2], np.int32)

```

```

pts_L1 = pts_L1.reshape((-1, 1, 2))

up_limit = int(1 * (h / 5))
down_limit = int(4 * (h / 5))

line_down_color = (255, 0, 255)

# background subtraction
fgbg = cv2.createBackgroundSubtractorMOG2(200, 16, True)

# Structuring elements for morphographic filters
kernelOp = np.ones((6, 6), np.uint8)
kernelOp2 = np.ones((5, 5), np.uint8)
kernel_cl = np.ones((22, 22), np.uint8)

# Variables
font = cv2.FONT_HERSHEY_SIMPLEX
object = []
pid = 1

# lấy fps
prev_frame_time = 0

while cap.isOpened():
    # đọc hình ảnh từ video
    ret, frame = cap.read()
    new_frame_time = time.time()
    fps = 1 / (new_frame_time - prev_frame_time)
    prev_frame_time = new_frame_time
    FPS = fps
    # FPS = cv2.VideoCapture.get(5)
    # top_left, bottom_right = (0, 100), (460, 0)
    for i in object:
        i.age_one()

    # áp dụng background subtraction

    fgmask2 = fgbg.apply(frame)
    try:
        ret, imBin2 = cv2.threshold(fgmask2, 254, 255,
cv2.THRESH_BINARY)
        mask2 = cv2.morphologyEx(imBin2, cv2.MORPH_OPEN, kernelOp)
        mask2 = cv2.morphologyEx(mask2, cv2.MORPH_CLOSE,
kernel_cl)
        mask2 = cv2.erode(mask2, kernelOp2)
    except:
        print("tổng xe trong video là : " + str(cnt_down +
cnt_up))
        print('END of File')
        break

    # Contours

```

```

        contours0, _ = cv2.findContours(mask2, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        for cnt in contours0:
            area = cv2.contourArea(cnt)
            if area > 900:
                M = cv2.moments(cnt)
                cx = int(M['m10'] / M['m00'])
                cy = int(M['m01'] / M['m00'])
                x, y, w, h = cv2.boundingRect(cnt)
                new = True
                if cy in range(up_limit, down_limit):
                    for i in object:
                        if abs(cx - i.getX()) <= w and abs(cy -
i.getY()) <= h:
                            # đối tượng gần với đối tượng đã được
phát hiện trước đó
                            new = False
                            i.updateCoords(cx, cy) # cập nhật tọa độ
trong đối tượng
                            if i.going_DOWN(line_down):
                                cnt_down += 1
                                a = str(cnt_down)
                                b = time.strftime("%c")
                                with open("Data.txt", 'a') as f:
                                    f.write("ID: " + a + "--DOWN-->" + str(b)
+ "\n")
                                break
                            if i.getState() == '1':
                                if i.getDir() == 'down' and i.getY() >
down_limit:
                                    i.setDone()
                                elif i.getDir() == 'up' and i.getY() <
up_limit:
                                    i.setDone()
                            if i.timedOut():
                                # xóa người khỏi danh sách
                                index = object.index(i)
                                object.pop(index)
                                del i # giải phóng bộ nhớ
                            if new:
                                p = Object.Object(pid, cx, cy, 5)
                                object.append(p)
                                pid += 1
                            # vẽ contour
                            for i in object:
                                color_contour = i.getRGB()
                                cv2.circle(frame, (cx, cy), 5, color_contour,
-2)
                                img = cv2.rectangle(frame, (x, y), (x + w, y +
h), color_contour, 2)
                                cv2.putText(img, "Moving Object", (x, y - 40),
cv2.FONT_HERSHEY_COMPLEX, 0.5,

```

```

                                (0, 255, 0),
                                2, cv2.LINE_AA)
                                # LINE_AA là giao diện như : màu sắc, độ dày,
loại đường

                                # tracking code
                                for i in object:
                                    if len(i.getTracks()) >= 2:
                                        pts = np.array(i.getTracks(),
np.int32)
                                        pts = pts.reshape((-1, 1, 2))
                                        frame = cv2.polylines(frame, [pts],
False, color_contour, 2)
                                # display info
                                str_down = ' DOI TUONG DI XUONG: ' + str(cnt_down)
                                cv2.line(frame, (10, 35), (10, 55), (0, 0, 255), 2)
                                cv2.line(frame, (10, 55), (5, 45), (0, 0, 255), 2)
                                cv2.line(frame, (10, 55), (15, 45), (0, 0, 255), 2)
                                FPS = "FPS : " + str(FPS)

                                # ranh giới để phân biệt đối tượng đi đang đi lên hay đi xuố
ng

                                frame = cv2.polylines(frame, [pts_L1], False, line_down_color,
thickness=3)

                                cv2.putText(frame, FPS, (20, 20), font, 0.5, (255, 255, 255),
1, cv2.LINE_AA)
                                cv2.putText(frame, str_down, (20, 40), font, 0.5, (255, 255,
255), 1, cv2.LINE_AA)

                                cv2.imshow('Original Video', frame) # display original video
                                cv2.imshow('Masked Video', mask2) # display B & W video
                                # cv2.imshow('roi' , roi)

                                # press ESC to exit
                                k = cv2.waitKey(30) & 0xff
                                if k == 27:
                                    break

                                cap.release()
                                cv2.destroyAllWindows()

```

- Bài toán bài toán phát hiện trộm chuyển động ,phát chuông cảnh báo:


```

import time

import cv2
import numpy as np
import pygame
from pygame import mixer

def phat_hien_trom():
    mixer.init()
    mixer.music.load('Tieng-coi-xe-can-h-sat-www_tiangdong_com.mp3')
    backSub = cv2.createBackgroundSubtractorMOG2()

    top_left, bottom_right = (200, 100), (700, 680)

    cap = cv2.VideoCapture('9014172086922407300.mp4')
    kernelOp = np.ones((6, 6), np.uint8)
    kernelOp2 = np.ones((5, 5), np.uint8)
    kernel_cl = np.ones((22, 22), np.uint8)

    while True:
        ret, frame = cap.read()
        fgMask = backSub.apply(frame)
        # Khu nhieu
        ret, imBin2 = cv2.threshold(fgMask, 254, 255,
cv2.THRESH_BINARY)
        fgMask = cv2.morphologyEx(imBin2, cv2.MORPH_OPEN, kernelOp)
        fgMask = cv2.morphologyEx(fgMask, cv2.MORPH_CLOSE, kernel_cl)
        fgMask = cv2.erode(fgMask, kernelOp2)

        contours, ret = cv2.findContours(fgMask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

        # khu vực cấm
        cv2.rectangle(frame, top_left, bottom_right, (255, 255, 0), 2)

        for i in range(len(contours)):
            # x là điểm cuối cùng bên trái
            # y là điểm cuối cùng bên phải
            # x + với rộng, y + với cao

            (x, y, w, h) = cv2.boundingRect(contours[i])
            cx = x + w / 2
            cy = y + h / 2

            trong_vong_canh_bao = top_left[0] < cx < bottom_right[0]
and top_left[1] < cy < bottom_right[1]
            area = cv2.contourArea(contours[i])
            if area < 600:
                continue
            if trong_vong_canh_bao:
                img = cv2.rectangle(frame, (x, y), (x + w, y + h), (0,
255, 0), 5)

                mixer.music.play()

```

```

        object.append([(x, y), (x + w, y + h)])
        cv2.putText(frame, "WARNING !!!", (200, 100),
cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 4)
        cv2.putText(img, "TROM DI CHUYEN", (x, y - 20),
cv2.FONT_HERSHEY_COMPLEX, .5,
                    (0, 255, 0),
                    2, cv2.LINE_AA)
        with open("Data_trom.txt", 'a') as f:
            b = time.strftime("%c")
            f.write("DOI_TUONG_VAO_NHA_LUC : " + b + "\n")
            print(object)
    else:
        pygame.mixer.stop()
        break
    cv2.imshow('Camera', frame)
    cv2.imshow('Masked Video', fgMask)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cv2.destroyAllWindows()

```

File main điều khiển luồng cho hai bài toán trên

```

from Human_Detection_and_Tracking import Detect_Tracking
from night_deetection_tracking import phat_hien_trom
if __name__ == '__main__':
    try:
        while 1:
            print("|-----|")
            print("| --- Mời bạn chọn video để thực hiện demo --- | ")
            print("| 1. video đối tượng chuyển động trên đường | ")
            print("| 2. video chuyển động vào trời tối | ")
            print("| 0. thoát lựa chọn | ")
            print("|-----|")
            case = int(input(" Nhập vào lựa chọn của bạn : "))
            if case == 1:
                Detect_Tracking("giaothongtrenduong.mp4")
                break
            elif case == 2:
                phat_hien_trom()
                break
            else:
                print("Nhập không đúng lựa chọn")
    except:
        print("END")

```

Link source code của đề án Github: [PYTHON_XU_LY_ANH/XULYANH/PBL4_AI](https://github.com/PYTHON_XU_LY_ANH/XULYANH/PBL4_AI) at [main · LHHT-DISCOVERY/PYTHON_XU_LY_ANH](https://github.com/LHHT-DISCOVERY/PYTHON_XU_LY_ANH) (github.com)

Tài liệu tham khảo

- [1] [Phân phối chuẩn – Wikipedia tiếng Việt](#)
- [2] [16.1. Ước lượng MLE cho phân phối Gaussian đa chiều — Deep AI KhanhBlog \(phamdinhhkhanh.github.io\)](#)
- [3] [Gaussian Mixture Model | Brilliant Math & Science Wiki](#)
- [4] [\(PDF\) Adaptive Background Mixture Models for Real-Time Tracking \(researchgate.net\)](#)
- [5] [\(PDF\) Adaptive background mixture models for real-time tracking \(researchgate.net\)](#)
- [6] [\(PDF\) Background modelling and background subtraction performance for object detection \(researchgate.net\)](#)
- [7] [Machine Learning cơ bản \(machinelearningcoban.com\)](#)
- [8] [deepai-book/KMeans.ipynb at f03b06c685fa72d19fe9be77a5dfce7ac328730b · adminvsrm/deepai-book \(github.com\)](#)
- [9] [Object tracking using background subtraction | Hands-On GPU-Accelerated Computer Vision with OpenCV and CUDA \(packtpub.com\)](#)
- [10] [Background modeling methods in video analysis: A review and comparative evaluation - ScienceDirect](#)
- [11] [1. Xác suất — Deep AI KhanhBlog \(phamdinhhkhanh.github.io\)](#)
- [12] [\(11\) Gaussian Mixture Model | Object Tracking - YouTube](#)
- [13] [\(11\) Phép Giãn - Xử lý ảnh - YouTube](#)
- [14] [taboGss/Gaussian-Mixture-Model-algorithm-VHDL: Gaussian Mixture Model algorithm for background subtraction in VHDL \(github.com\)](#)
- [15] [Background Subtraction — OpenCV 3.0.0-dev documentation](#)

[16] Background Subtraction OpenCV Python | MOG2 and KNN (hackthedeveloper.com)

[17] Motion Analysis and Object Tracking — OpenCV 2.4.13.7 documentation

[18] Background Subtraction: MOG2 y GMG - Función de Detección (1library.co)

[19] (11) quarter DIP Gaussian Mixture Models for Background Subtraction - YouTube

[20] 1810.02835.pdf (arxiv.org)