ĐỒ ÁN XỬ LÝ ẢNH

Nhóm 5: Nguyễn Quang Minh, Trần Nhật Minh, Lâm Quang Huy

I.Giới thiệu về Pose Recogniton (Estimation)

1. Nhận dạng và ước tính tư thế là gì

Nhận dạng và theo dõi tư thế con người là một nhiệm vụ trong thị giác máy tính, bao gồm việc phát hiện, liên kết và theo dõi các điểm đặc trưng (semantic key points). Ví dụ về các điểm đặc trưng là "vai phải" và "đầu gối trái."

Hiệu suất theo dõi các điểm đặc trưng (semantic keypoints) trong cảnh quay video trực tiếp đòi hỏi nhiều tài nguyên tính toán, điều này đã giới hạn độ chính xác trong việc hiểu tư thế. Với những tiến bộ mới nhất về phần cứng và hiệu quả của mô hình, các ứng dụng có yêu cầu thời gian thực hiện nay trở nên khả thi và kinh tế hơn.

Hiện nay, các mô hình xử lý hình ảnh mạnh mẽ nhất đều dựa trên mạng nơ-ron tích chập (CNN). Do đó, các phương pháp tiên tiến hiện đại thường dựa trên việc thiết kế kiến trúc CNN được tùy chỉnh đặc biệt cho các hệ thống phát hiện tư thế người hoặc đối tượng.

2. Tại sao cần nhận dạng tư thế

Trong bối cảnh hiện đại, nhận dạng tư thế người (Nhận dạng tư thế con người) ngày càng trở nên quan trọng để phát triển nhanh chóng các ứng dụng công nghệ trong y tế, thể thao và giám sát an ninh.

1. **Ứng dụng trong y tế và chức năng hồi phục** : Nhận dạng trợ giúp của người theo dõi và đánh giá chuyển động của bệnh nhân trong quá trình hồi phục sau chấn thương hoặc phẫu thuật. Theo một nghiên cứu, công nghệ này đã được áp dụng để đo phạm vi chuyển động và đánh giá quá trình phục hồi bệnh nhân, đặc biệt là trong các bài tập tại nhà. Sử dụng công nghệ nhận dạng tư thế, hệ thống robot phục hồi chức năng đã được phát triển để hỗ trợ chuyển động của bệnh nhân, cải thiện hiệu quả phục hồi mà không cần đến sự giám sát liên tục của bác sĩ​ [1] .
2. **Thể thao và cải thiện hiệu suất vận động** : Trong lĩnh vực thể thao, nhận được tư cách là người giúp huấn luyện viên và vận động viên tối ưu hóa kỹ thuật và nâng cao hiệu suất. Ví dụ: các thế hệ phân tích hệ thống đã được sử dụng để theo dõi và đánh giá thế giới trong các môn học như thể giáo dục cụ và trượt tuyết, nơi mà thành công phụ thuộc nhiều vào độ chính xác của thế tư​ [2] . Ngoài ra, hệ thống này còn cho phép theo dõi và phân tích hoạt động của vận động viên trong điều kiện thực tế, giúp giảm nguy cơ chấn thương và tối ưu hóa huấn luyện viên.
3. **Giám sát an ninh** : Trong các hệ thống giám sát thông minh, nhận dạng tư thế có thể được sử dụng để theo dõi hành vi và phát hiện các hoạt động bất thường hoặc bạo lực trong đám đông. Nghiên cứu đã chỉ ra rằng việc áp dụng nghiên cứu công nghệ nhận dạng tư thế trong giám sát an ninh giúp giảm chi phí vận hành và nâng cao hiệu quả giám sát cho hệ thống truyền thông phương pháp. Tuy nhiên, việc phát triển khai báo trên quy mô Yêu cầu phải có các thuật toán và hệ thống linh hoạt, có khả năng nhận dạng chính xác trong môi trường phức tạp và với phần cứng hạn chế​ [3] .

Ngoài ra, theo báo cáo của Allied Market Research, thị trường các giải pháp phân tích tư thế người dự kiến ​​sẽ đạt hơn 1,2 tỷ USD vào năm 2027, tăng trưởng mạnh mẽ do nhu cầu từ các lĩnh vực như chăm sóc sức khỏe, an ninh công cộng và giải trí​ [4]. Điều này cho thấy tầm quan trọng và tiềm năng phát triển lớn của công nghệ nhận dạng tư vấn cho người ở tương lai.

3. Các vấn đề thách thức trong nhận dạng tư thế

Trong nhận dạng tư thế người , có nhiều thách thức cần giải quyết để đạt được độ chính xác cao và hiệu suất tốt trong các ứng dụng thực tế. Dưới đây là một số thách thức chính và nguyên nhân:

1. **Occlusion (Chướng ngại vật che khuất)**:
   * Các bộ phận cơ thể có thể bị che khuất bởi các vật thể khác hoặc các phần khác của cơ thể, làm cho việc phát hiện và xác định vị trí chính xác của các khớp trở nên khó khăn. Đây là một vấn đề phổ biến trong các bối cảnh phức tạp như đám đông hoặc khi người dùng di chuyển trong không gian hẹp.
   * Tác động: Occlusion làm giảm hiệu quả của các thuật toán, khiến các mô hình dễ bị lỗi khi cố gắng xác định tư thế chính xác của người dùng​[4], [5]
2. **Biến dạng cơ thể và đa dạng tư thế**:
   * Cơ thể con người có thể chuyển động và thay đổi hình dáng linh hoạt, điều này làm cho việc xây dựng mô hình để dự đoán chính xác tư thế trở nên phức tạp. Đặc biệt, các tư thế khó như co giãn hoặc uốn cong cơ thể đòi hỏi hệ thống phải có khả năng phân tích và nhận diện tốt các tình huống không thường gặp.
   * Tác động: Các mô hình hiện tại thường gặp khó khăn trong việc xử lý các tư thế hiếm gặp hoặc không điển hình, gây giảm hiệu suất tổng thể và làm sai lệch kết quả [6].
3. **Ảnh hưởng của ánh sáng và điều kiện môi trường**:
   * Trong các ứng dụng ngoài trời hoặc môi trường không được kiểm soát, ánh sáng và môi trường thay đổi liên tục, làm thay đổi màu sắc và độ phân giải của hình ảnh. Điều này có thể gây nhiễu và làm giảm chất lượng dữ liệu đầu vào.
   * Tác động: Hệ thống nhận dạng tư thế hoạt động không ổn định, đòi hỏi phải có các phương pháp xử lý trước hoặc kỹ thuật ổn định hóa để cải thiện độ chính xác​[5].
4. **Thiếu dữ liệu đào tạo đa dạng**:
   * Các hệ thống nhận dạng tư thế phụ thuộc vào dữ liệu lớn để đào tạo, tuy nhiên, việc thu thập và dán nhãn các bộ dữ liệu đa dạng từ các bối cảnh khác nhau rất tốn kém. Đặc biệt là các bộ dữ liệu 3D với nhiều góc nhìn và điều kiện khác nhau.
   * Tác động: Thiếu dữ liệu đa dạng dẫn đến việc mô hình không tổng quát hóa tốt, khiến nó hoạt động kém hiệu quả trong các tình huống thực tế ngoài phạm vi của dữ liệu đào tạo​(6).

Các nghiên cứu đã đề xuất nhiều giải pháp như sử dụng các mạng tích chập sâu (CNN) và mạng nơ-ron hồi tiếp (RNN) để cải thiện độ chính xác và tính linh hoạt của hệ thống nhận diện. Tuy nhiên, việc triển khai những giải pháp này trong các ứng dụng thời gian thực và bối cảnh phức tạp vẫn là một thách thức lớn cần được giải quyết​.

4. Các nghiên cứu liên quan

1. **OpenPose: Real-Time Multi-Person 2D Pose Estimation using Part Affinity Fields** [7]**.**

OpenPose sử dụng Part Affinity Fields (PAFs) để xác định mối quan hệ giữa các phần cơ thể con người và kết nối chúng thành cấu trúc xương. PAFs là một dạng trường véc-tơ cho phép xác định mối liên kết giữa các điểm trên cơ thể qua nhiều người, giúp nhận dạng và phân biệt tư thế của từng cá nhân trong một khung hình. Hệ thống này có khả năng nhận dạng tư thế của nhiều người trong cùng một khung hình với độ chính xác cao và trong thời gian thực, ngay cả trong các trường hợp có sự giao thoa giữa các cá nhân. OpenPose đã trở thành một trong những hệ thống phổ biến nhất để nhận dạng tư thế, với nhiều ứng dụng trong các lĩnh vực như thể dục thể thao, tương tác người-máy, và giám sát. Tuy nhiên, OpenPose vẫn gặp khó khăn khi xử lý các tư thế phức tạp, đặc biệt khi các phần cơ thể bị che khuất (occlusion). Hiệu suất của hệ thống cũng giảm khi xử lý hình ảnh có độ phân giải thấp hoặc khi đối mặt với chuyển động nhanh.

1. **CONet: Crowd and occlusion-aware network for occluded human pose estimation** [8]

CONet sử dụng Crowd and Occlusion-aware Head (COHead), một cấu trúc gồm hai nhánh độc lập (dual-branch) để phát hiện tư thế của cả người bị che khuất (occluded) và người che (occluder) trong các khung hình đông người. Mỗi nhánh ước lượng các điểm khớp của từng cá nhân một cách riêng biệt. Cơ chế chú ý (Attention Mechanism) được tích hợp vào các nhánh này để giúp tập trung vào những thông tin quan trọng và thực hiện học phân biệt (differentiated learning), từ đó cải thiện độ chính xác trong việc nhận diện các đặc trưng của từng người trong các khung hình đông đúc. Ngoài ra, mô hình còn sử dụng Interference Point Loss để giải quyết vấn đề can nhiễu khi các điểm khớp của nhiều người chồng chéo lên nhau, giúp phân biệt rõ ràng hơn giữa các điểm nhiễu và điểm thực tế. Khi đánh giá trên bộ dữ liệu CrowdPose, CONet đạt 71.6 AP (Average Precision), cao hơn 1.6 AP so với các phương pháp hiện tại. Mô hình cũng được thử nghiệm trên bộ dữ liệu COCO và cho thấy khả năng tương thích tốt ngay cả trong các tình huống không bị che khuất. Tuy nhiên, CONet đòi hỏi tính toán cao, gây khó khăn khi triển khai trong thời gian thực, và dữ liệu huấn luyện của nó chưa phong phú đủ để xử lý mọi tình huống phức tạp, đặc biệt là trong các trường hợp can nhiễu khó khăn.

1. **SPIN: End-to-End Learning of Human Pose and Shape with Regression and Parameterization**[9]

SPIN sử dụng một mô hình thống nhất để ước lượng tư thế và hình dạng 3D của con người từ hình ảnh 2D bằng cách áp dụng học hồi quy (regression) và tham số hóa hình dạng. Mô hình này dựa trên một hệ thống tham số hóa 3D gọi là SMPL, cho phép tái tạo lại hình dạng 3D của cơ thể người từ dữ liệu 2D. SPIN đã đạt được độ chính xác cao trong việc ước lượng tư thế và hình dạng 3D, ngay cả khi không có thông tin chiều sâu trực tiếp. Phương pháp này đã được đánh giá là tốt hơn các phương pháp khác trên nhiều bộ dữ liệu chuẩn. Tuy nhiên, SPIN vẫn có thể bị ảnh hưởng bởi tư thế của người trong các khung hình bị che khuất (occlusion) hoặc khi có nhiều sự giao thoa giữa các cá nhân. Ngoài ra, SPIN yêu cầu lượng lớn tài nguyên tính toán, do đó gặp khó khăn khi triển khai trong các ứng dụng thời gian thực.

1. **Human Pose Estimation with Iterative Error Feedback** [10]

Phương pháp này sử dụng mô hình học sâu với cơ chế phản hồi lỗi lặp lại (iterative error feedback) để cải thiện độ chính xác của hệ thống ước lượng tư thế. Các lỗi ước lượng ban đầu được mô hình sử dụng để tinh chỉnh và cải thiện các dự đoán trong các bước tiếp theo. Nhờ vậy, phương pháp này cải thiện đáng kể độ chính xác trong việc nhận dạng tư thế người, đặc biệt trong các tư thế phức tạp hoặc khi hình ảnh bị nhiễu. Hệ thống đã được thử nghiệm trên các bộ dữ liệu chuẩn và đạt kết quả tốt hơn so với các phương pháp trước đó. Tuy nhiên, việc sử dụng phản hồi lỗi lặp lại có thể khiến mô hình bị quá tải tính toán (overfitting), đặc biệt khi dữ liệu chứa nhiều nhiễu hoặc không chính xác. Bên cạnh đó, hiệu quả xử lý trong thời gian thực vẫn là một thách thức lớn, đặc biệt khi mô hình phải xử lý nhiều lần phản hồi liên tục.

1. **PoseFlow: Efficient Online Multi-Person 2D Pose Tracking**[11]

PoseFlow sử dụng một thuật toán theo dõi đa người trong thời gian thực, dựa trên việc ước lượng chuyển động giữa các khung hình và tương quan giữa các vị trí tư thế để theo dõi người qua các khung hình. Phương pháp này đã đạt được kết quả tốt trong việc theo dõi nhiều người trong các video và có thể xử lý với độ chính xác tương đối cao trong thời gian thực. Ngoài ra, PoseFlow đã cải thiện hiệu suất theo dõi tư thế người trong các hệ thống giám sát và phân tích video. Tuy nhiên, độ chính xác của PoseFlow giảm khi có nhiều người di chuyển nhanh hoặc khi có sự giao thoa lớn giữa các cá nhân. Hệ thống này cũng phụ thuộc nhiều vào chất lượng của dữ liệu đầu vào, như video, và dễ bị ảnh hưởng bởi điều kiện môi trường thay đổi, chẳng hạn như ánh sáng kém hoặc video có chất lượng thấp.

5. Đóng góp của đề tài: Nhóm sẽ giải quyết bằng phương pháp xử lý ảnh HOG, egde dectection và SVM

**Occlusion (chướng ngại vật che khuất)** là một thách thức quan trọng trong nhận dạng tư thế. Khi các phần cơ thể bị che khuất bởi các vật thể khác hoặc bởi chính các phần khác của cơ thể, việc xác định chính xác các khớp trở nên rất khó khăn. Để giải quyết vấn đề này, sự kết hợp giữa phương pháp xử lý ảnh **HOG (Histogram of Oriented Gradients)** với **SVM (Support Vector Machine)** mang lại hiệu quả cao. Cụ thể, HOG giúp trích xuất các đặc trưng mạnh mẽ từ hình ảnh, cho phép hệ thống nhận diện được hình dạng tổng quát của cơ thể và các khớp dù có sự che khuất. **SVM** sau đó sẽ học và phân loại các đặc trưng này từ dữ liệu huấn luyện để xác định tư thế chính xác, ngay cả khi các phần cơ thể bị che khuất, nhờ vào khả năng phân tách tuyến tính hoặc phi tuyến giữa các lớp dữ liệu.

Với các tư thế phức tạp hoặc biến dạng cơ thể, chẳng hạn như khi uốn cong hoặc giãn cơ, sự kết hợp giữa **HOG**, **Edge Detection**, và **SVM** cũng xử lý tốt. HOG có khả năng mô hình hóa các hình dạng không gian phức tạp của cơ thể, trong khi **Edge Detection** giúp phát hiện các biên và cấu trúc quan trọng. **SVM** sử dụng các đặc trưng hình học và biên này để phân loại tư thế, nhờ vào khả năng tạo ra một siêu phẳng phân tách tối ưu giữa các tư thế khác nhau. Điều này giúp hệ thống dự đoán chính xác các tư thế không điển hình mà không bị nhầm lẫn.

**Điều kiện môi trường và ánh sáng thay đổi** cũng là một yếu tố ảnh hưởng đến độ chính xác của hệ thống nhận dạng tư thế, đặc biệt khi hoạt động ngoài trời. Tuy nhiên, phương pháp **HOG** chủ yếu dựa vào đặc trưng hình học thay vì màu sắc của hình ảnh, giúp giảm thiểu tác động của các yếu tố môi trường. HOG tập trung vào việc trích xuất các đường nét và cấu trúc không gian, cho phép hệ thống hoạt động ổn định ngay cả khi ánh sáng thay đổi. Đồng thời, **SVM** giảm thiểu ảnh hưởng của các biến đổi ngẫu nhiên bằng cách sử dụng các đặc trưng hình học đã được học từ dữ liệu huấn luyện, đảm bảo tính nhất quán trong việc dự đoán tư thế.

Một thách thức quan trọng khác là **sự thiếu hụt dữ liệu đa dạng** trong quá trình huấn luyện. Khi hệ thống không có đủ dữ liệu từ nhiều tình huống thực tế khác nhau, khả năng tổng quát hóa của mô hình sẽ giảm, dẫn đến hiệu suất kém trong các tình huống mới hoặc chưa được huấn luyện. Để khắc phục điều này, **HOG** cung cấp một loạt các đặc trưng mạnh mẽ từ hình ảnh mà không phụ thuộc nhiều vào dữ liệu huấn luyện, giúp mô hình học được cấu trúc và hình dạng cơ thể một cách hiệu quả. Kết hợp với **SVM**, hệ thống sẽ dự đoán chính xác tư thế dựa trên các đặc trưng này, ngay cả trong các tình huống mà dữ liệu không đa dạng hoặc không đầy đủ. Nhờ đó, hệ thống có thể duy trì hiệu suất cao trong nhiều điều kiện thực tế khác nhau.

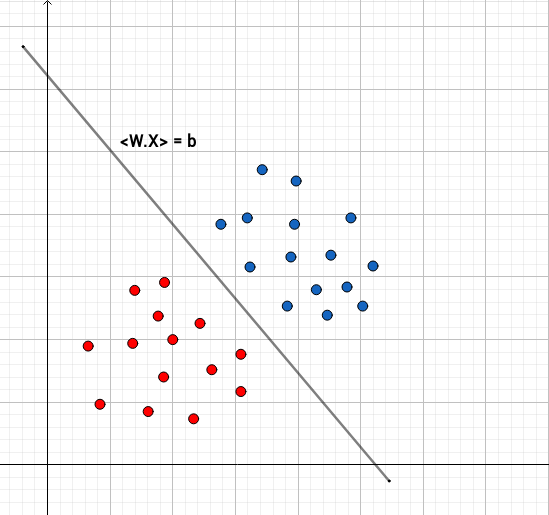
II. Lý thuyết nền tảng

1. Support Vector Machine (SVM) [12]
   1. Định nghĩa

**SVM** là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồi thị dữ liệu là các điểm trong n chiều ( ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (*hyper-plane*) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.. Mục tiêu của SVM là tìm ra một siêu phẳng trong không gian N chiều (ứng với N đặc trưng) chia dữ liệu thành hai phần tương ứng với lớp của chúng. Nói theo ngôn ngữ của đại số tuyển tính, siêu phẳng này phải có lề cực đại và phân chia hai bao lồi và cách đều chúng.

* 1. Cách hoạt động của thuật toán

Ý tưởng của SVM là tìm một siêu phẳng (hyper lane) để phân tách các điểm dữ liệu. Siêu phẳng này sẽ chia không gian thành các miền khác nhau và mỗi miền sẽ chứa một loại dữ liệu.



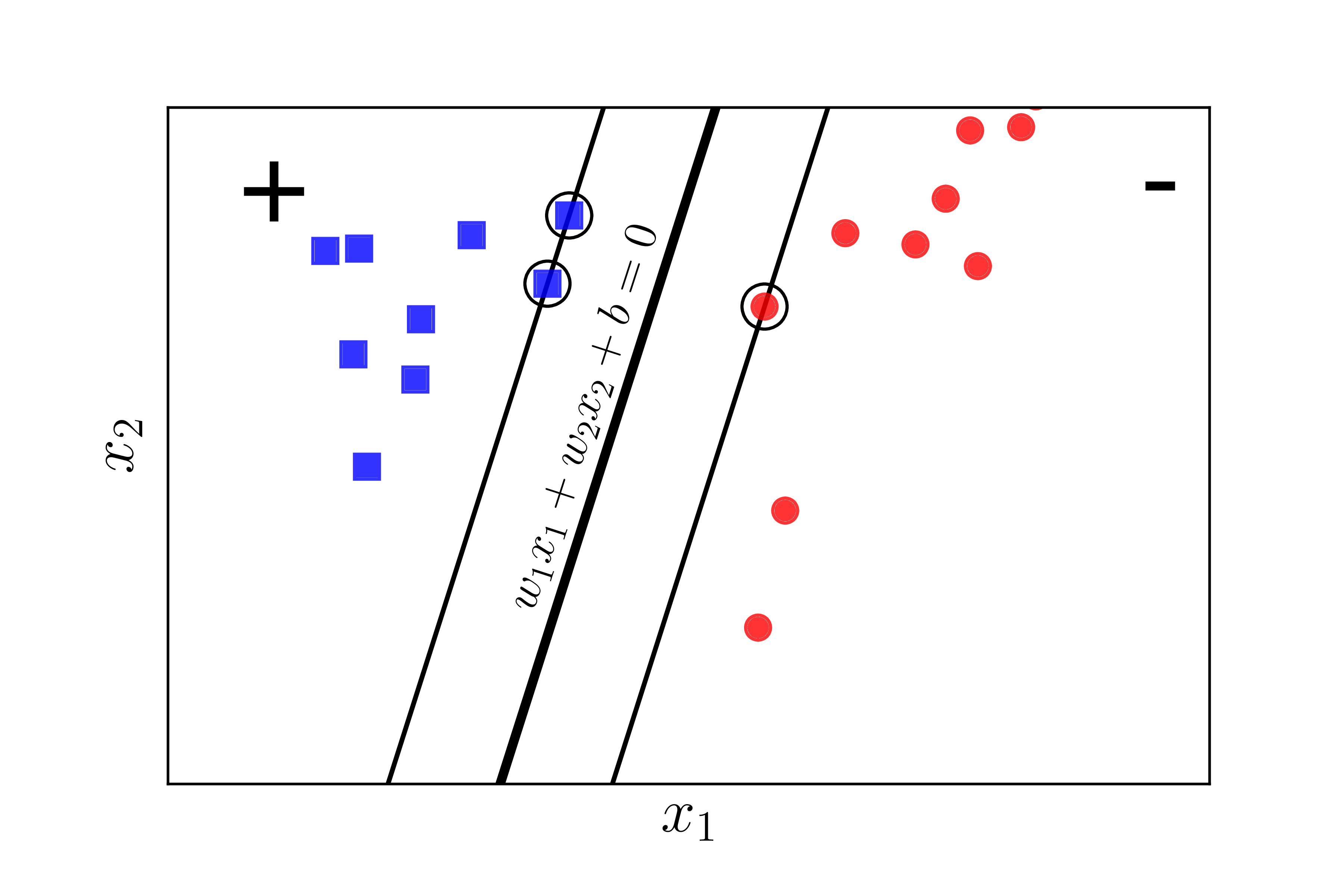
Siêu phẳng được biểu diễn bằng hàm số  ( W và X là các vector <W.X> là tích vô  ).Hay

(  là ma trận chuyển vị). Vấn đề là có rất nhiều siêu phẳng, chúng ta phải chon cái nào để tối ưu nhất ?

* 1. Xây dựng bài toán tối ưu cho SVM

Giả sử rằng các cặp dữ liệu của trainning set là (x1,y1),(x2,y2),…,(xN,yN) với vector thể hiện đầu vào của một điểm dữ liệu và là nhãn của điểm dữ liệu đó. d là số chiều của dữ liệu và N là số điểm dữ liệu. Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi (class 1) hoặc (class 2)

Để dễ hình dung, ta cùng xét trường hợp trong không gian hai chiều dưới đây. Không gian hai chiều để các bạn dễ hình dung, các phép toán hoàn toàn có thể được tổng quát lên không gian nhiều chiều.



Giả sử các điểm vuông xanh thuộc class 1, các điểm tròn đỏ thuộc class -1 và mặt là mặt phân chia giữa hai classes. Hơn nữa, class 1 nằm về phía dương, class -1 nằm về *phía âm* của mặt phân chia. Nếu ngược lại, ta chỉ cần đổi dấu của w và b. Chú ý rằng chúng ta cần đi tìm các hệ số w và b.

Ta quan sát thấy một điểm quan trọng sau đây: với cặp dữ liệu  bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là:

Điều này có thể dễ nhận thấy vì theo giả sử ở trên, luôn cùng dấu với *phía* của  Từ đó suy ra  cùng dấu với  và tư luôn là 1 số không âm

Với mặt phần chia như trên, margin được tính là khoảng cách gần nhất từ 1 điểm tới mặt đó (bất kể điểm nào trong hai classes):

margin = min

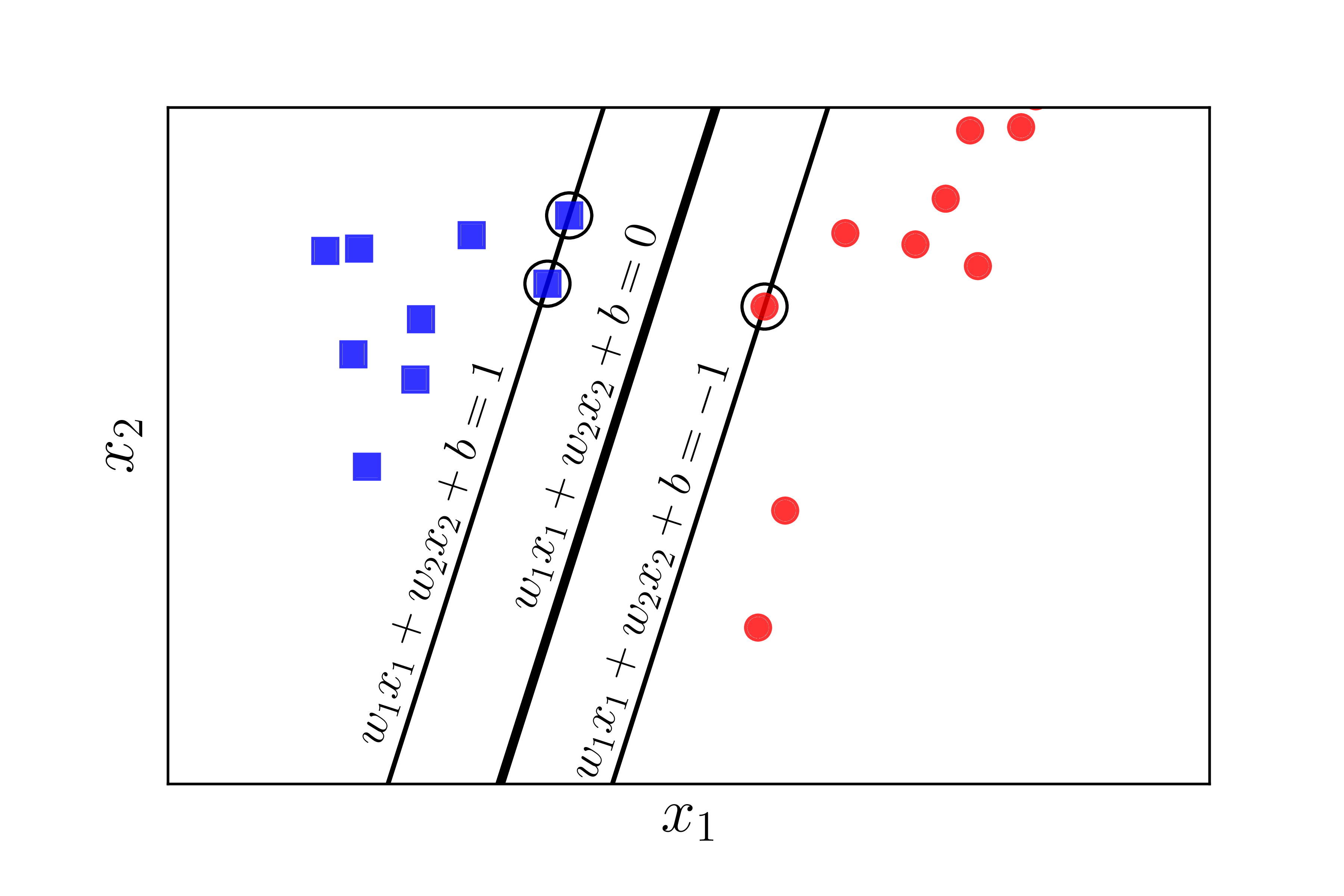
Bài toán tối ưu trong SVM chính là bài toán tìm w và b sao cho margin này đạt giá trị lớn nhất

(w,b) = argmax {min } = argmax {} (1)

Việc giải trực tiếp bài toán này sẽ rất phức tạp, nhưng ta sẽ có cách để đưa nó về bài toán đơn giản hơn.

Nhận xét quan trọng nhất là nếu ta thay vector hệ số w bởi kw và bb bởi kb trong đó kk là một hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức margin không đổi. Dựa trên tính chất này, ta có thể giả sử:

**với những điểm nằm gần mặt phân chia nhất** như hình dưới đây:



Như vậy, với mọi n, ta có:

1. Edge Detection (Phát hiện biên) [13]

2.1. Giới thiệu

Edge detection (phát hiện biên) là một kỹ thuật trong xử lý ảnh nhằm xác định những điểm trong ảnh có sự thay đổi đáng kể về độ sáng, thường là để phát hiện biên của đối tượng. Đối với nhận dạng tư thế người, phát hiện biên giúp làm nổi bật hình dạng hoặc đường viền của cơ thể con người, từ đó phân tích được các bộ phận và tư thế.

Mục đích của phát hiện biên: Để trích xuất thông tin từ bức ảnh, ví dụ: vị trí của các đối tượng trong bức ảnh, hình dạng và kích thước của chúng, làm sắc nét ảnh, tăng cường ảnh, phục vụ cho quá trình nhận dạng ảnh.

A person wearing a cowboy hat

Description automatically generated

Edge detection giúp tìm ranh giới của cơ thể người trong hình ảnh. Từ biên, có thể chuyển thành khung xương đơn giản hơn để phân tích tư thế. Giúp loại bỏ thông tin không cần thiết, tập trung vào biên của cơ thể.

2.2. Các loại biên

1. **Biên kiểu bước nhảy (step edge)**

Cường độ sáng thay đổi một cách đột ngột.

A black and white rectangular object with a white text

Description automatically generated

1. **Biên thoai thoải (ramp edge)**

Cường độ sáng không thay đổi ngay lập tức mà xảy ra trên một khoảng cách nhất định.

A diagram of a black and white gradient

Description automatically generated

1. **Biên kiểu mái nhà (roof edge)**

Thường là mô hình đường thẳng chạy qua một vùng với độ rộng của mái nhà tùy thuộc vào độ dày và độ sắc nét của đường thẳng.

A diagram of a roof edge

Description automatically generated

2.3 . Phát hiện biên tối ưu – Phương pháp Canny [14]

Phát hiện cạnh Canny (Canny edge detector) là một thuật toán bao gồm nhiều giai đoạn để phát hiện một loạt các cạnh trong hình ảnh. Nó được phát triển bởi John F. Canny vào năm 1986. Canny cũng đưa ra một lý thuyết tính toán về phát hiện cạnh giải thích tại sao kỹ thuật này hoạt động. Thuật toán Canny bao gồm các bước: Chuyển đổi ảnh sang màu xám -> Làm mịn ảnh -> Tính gradient thành phần –> Loại bỏ các giá trị không phải cực đại -> Phân ngưỡng kép để xác định biên

1. **Chuyển đổi ảnh sang màu xám**

Dùng phương pháp ánh sáng cảm nhận (Luminosity method): Phương pháp này chính xác hơn vì nó tính toán dựa trên độ nhạy cảm của mắt người với từng kênh màu. Công thức chuyển đổi thường là:

Gray= 0.2989×R+0.5870×G+0.1140×B

Ví dụ sau bước 1:

A person holding his hands up

Description automatically generated -----> A person with his hands up

Description automatically generated

1. **Làm mịn ảnh bằng Gaussian**

Sử dụng lọc **Gaussian** làm mịn ảnh nhằm giảm nhiễu, giúp quá trình phát hiện biên chính xác hơn. Bộ lọc Gaussian là phổ biến nhất, giúp làm mịn mà vẫn giữ được các đặc trưng quan trọng của ảnh.

Kernel Gaussian thường là ma trận 3x3, 5x5 hoặc 7x7:

Như với ma trận 5x5 và hệ số là 1 ta sẽ có ma trận là

Kernel Gaussian =

Ảnh sẽ được làm mịn bằng cách convolution với kernel Gaussian để làm giảm các biên không mong muốn do nhiễu. Cụ thể là nhân ảnh với ma trận kernal gausian

Ví dụ sau khi làm mịn

A person with his hands up

Description automatically generated -----> A person with his hands up

Description automatically generated

1. **Tính gradient thành phần**

Sự rõ ràng của biên được phát hiện bằng cách tính gradient của ảnh. Có thể tính gradient sử dụng toán tử Sobel. Với kernal Sobel là

A pair of squares with numbers

Description automatically generated

Sx Sy

Để tính gradient tại một pixel, ta nhân ảnh với các kernel Sobel tương ứng với các hướng xxx và yyy. Kết quả thu được là hai ma trận: một ma trận thể hiện gradient theo hướng ngang (x), và một ma trận thể hiện gradient theo hướng dọc (y).

Gx = I(x,y) \* Sx

Gy = I(x,y) \* Sy

Sau khi có hai ma trận Gx​ và Gy, ta tính **độ lớn của gradient** tại mỗi pixel bằng công thức sau:

Kết quả này cho biết độ mạnh của sự thay đổi độ sáng tại điểm I(x,y) và có thể được dùng để phát hiện biên (edges).

Hướng của biên: Giá trị θ này cho ta biết hướng của biên tại điểm (x,y).

Khi đã biết hướng của biên, bước tiếp theo là liên hệ hướng này với một hướng có thể lần theo trong ảnh.

Khi xem xét I(xy), chỉ có thể có 4 hướng khi mô tả các pixel lân cận

- 0 độ (theo hướng nằm ngang)

45 độ (theo hướng dương của đường chéo)

90 độ (theo hướng dọc)

135 độ (theo hướng âm của đường chéo).

Vì thế biên được định hướng vào một trong 4 hướng này tùy thuộc vào hướng của biên gần cái nào nhất:

Nếu hướng biên rơi vào trong dải màu vàng (0 đến 22.5 & 157.5 đến 180 độ) thì được coi là 0 độ.

Nếu hướng biên rơi vào dải màu xanh lá cây (22.5 đến 67.5 độ) được coi là 45 độ.

Nếu hướng biên rơi vào dải màu xanh da trời (67.5 đến 112.5 độ) thì được coi là 90 độ

Nếu hướng biên rơi vào dải màu đỏ (112.5 đến 157.5 độ) thì được coi là 135 độ.

A colorful pie chart with numbers

Description automatically generated

Ví dụ chuyển đổi ảnh:

A person with his hands up

Description automatically generated sau khi gradient sẽ cho kết quả

A graph of a person's face

Description automatically generated

1. **Loại bỏ các giá trị không phải cực đại**

Đây là kỹ thuật nhằm làm mảnh đường biên. Thực hiện tìm kiếm để xác định độ lớn gradient có phải là giá trị cực đại địa phương theo hướng gradient hay không. Tại mỗi pixel, thiết lập giá trị pixel bằng 0 nếu độ lớn của nó không lớn hơn độ lớn của 2 lân cận theo hướng gradient và giữ lại pixel có độ lớn cực đại.

Ví dụ :

Nếu góc gradient được làm tròn thành 0 độ (nghĩa là biên theo hướng bắc- nam), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng đông và tây.

Nếu góc gradient được làm tròn thành 90 độ (nghĩa là biên theo hướng đông- tây), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng nam và bắc.

Nếu góc gradient được làm tròn thành 135 độ (nghĩa là biên theo hướng đông bắc- tây nam), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng tây bắc và đông nam.

Nếu góc gradient được làm tròn thành 45 độ (nghĩa là biên theo hướng tây bắc- đông nam), điểm đó được coi là nằm trên biên nếu độ lớn gradient lớn hơn độ lớn tại các pixel theo hướng đông bắc và tây nam.

Minh hoạ quá trình loại bỏ các giá trị không phải cực đại:

Độ lớn biên được biểu diễn bằng màu và số, trong khi hướng biên được biểu diễn bằng các mũi tên.

A grey square with arrows pointing to the center

Description automatically generated with medium confidence

Các pixel biên kết quả được đánh dấu bằng đường viền màu trắng.

A person with his hands up

Description automatically generated------🡪 A person with his hands up

Description automatically generated

1. **Phân ngưỡng kép để xác định biên**

Sử dụng ngưỡng T để giảm số biên giả.

Tất cả các pixel có giá trị nhỏ hơn T được thiết lập giá trị 0

Chọn giá trị T phù hợp rất khó.

Nếu chọn T quá thấp sẽ xuất hiện nhiều biên giả.

Nếu chọn T quá cao sẽ làm mất một số biên thực.

🡪 Giải pháp: Sử dụng 2 ngưỡng T1 cao và T2 thấp (thường chọn T1 = 2. T2)

Nếu pixel có giá trị lớn hơn T1 sẽ được coi là điểm biên ngay lập tức.

Các pixel có giá trị nhỏ hơn T2 sẽ không được coi là điểm biên.

Pixel có giá trị nằm giữa T2 và T1 sẽ được xem xét tiếp: nếu pixel này có kết nối với một trong số các điểm biên khác thì sẽ được coi là điểm biên, ngược lại sẽ không coi là điểm biên.

Ví dụ: ảnh I có giá trị như sau

|  |  |  |
| --- | --- | --- |
| 50 | 120 | 200 |
| 70 | 250 | 180 |
| 40 | 100 | 220 |

Và ta chọn hai ngưỡng Th=150 và Tl=50:

Điểm G(1,3)= 200 (lớn hơn Th): Đây là **biên mạnh**.

Điểm G(2,2)= 250 (lớn hơn Th​): Đây là **biên mạnh**.

Điểm G(2,1)= 70 (giữa Tl ​ và Th): Đây là **biên yếu**, nhưng phải xác nhận với điểm biên mạnh xung quanh.

Điểm G(3,1)=40 (nhỏ hơn Tl​): Đây không phải là biên, loại bỏ.

Các điểm biên yếu mà có ít nhất một điểm biên mạnh xung quanh (lân cận) sẽ được giữ lại. Những điểm này kết nối với biên mạnh và có thể là một phần của biên thực sự.

Các điểm biên yếu không có kết nối với điểm biên mạnh sẽ bị loại bỏ.

A person with his hands up

Description automatically generated---🡪 A person with his fist raised

Description automatically generated

1. **Ví dụ sau các bước**

A collage of images of a machine

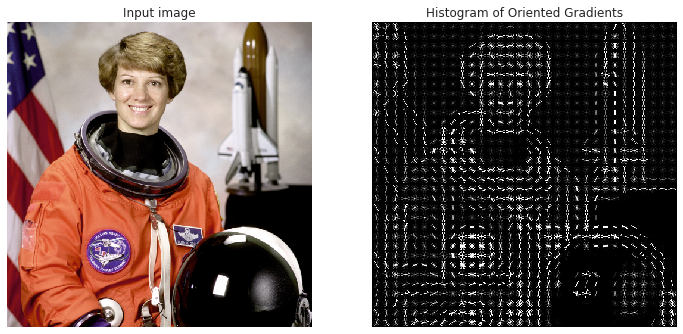
Description automatically generated

1. Histogram of Oriented Gradients [15]
   1. Giới thiệu chung về HOG

HOG là viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “feature descriptor” là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh.

Thuật toán này sẽ tạo ra các bộ mô tả đặc trưng (feature descriptor) nhằm mục đích phát hiện vật thể (object detection). Từ một bức ảnh, ta sẽ lấy ra 2 ma trận quan trọng giúp lưu thông tin ảnh đó là độ lớn gradient (gradient magnitute) và phương của gradient (gradient orientation). Bằng cách kết hợp 2 thông tin này vào một biểu đồ phân phối histogram, trong đó độ lớn gradient được đếm theo các nhóm bins của phương gradient. Cuối cùng ta sẽ thu được vecto đặc trưng HOG đại diện cho histogram. Sơ khai là vậy, trên thực tế thuật toán còn hoạt động phức tạp hơn khi vecto HOG sẽ được tính trên từng vùng cụ bộ như mạng CNN và sau đó là phép chuẩn hóa cụ bộ để đồng nhất độ đo. Cuối cùng vecto HOG tổng hợp từ các vecto trên vùng cục bộ.

Ứng dụng của HOG vào nhận diện khuôn mặt (face detection): Thường chúng ta sẽ nghĩ ngay đến thuật toán Haar Cascde Classifier. Tuy nhiên HOG cũng là một thuật toán rất hiệu quả được áp dụng trong bài toán này. Bởi nó có khả năng biểu diễn các đường nét chính của khuôn mặt dựa trên phương và độ lớn gradient thông qua các vecto trên mỗi cell như hình mô tả bên dưới:



Sự khác nhau cơ bản giữa **Histogram of Oriented Gradient (HOG)** và **Haar Cascade Classifier** trong lĩnh vực nhận diện khuôn mặt:

* **HOG** tập trung vào việc phân tích hình dáng và cấu trúc của đối tượng thông qua gradient, trong khi **Haar Cascade** sử dụng các đặc trưng đơn giản và kỹ thuật cascading để phát hiện khuôn mặt nhanh chóng.
* HOG thường tốt hơn trong việc phát hiện đối tượng với hình dạng rõ ràng, trong khi Haar Cascade là lựa chọn phổ biến cho các ứng dụng thời gian thực do tốc độ phát hiện nhanh hơn.
  1. Các thuật ngữ liên quan:
* **Feature Descriptor**: Bộ mô tả đặc trưng, là một phép biến đổi dữ liệu thành các đặc trưng giúp ích cho phân loại hoặc nhận diện vật thể.
* **Histogram**: Là biểu đồ histogram biểu diễn phân phối của các cường độ màu sắc theo khoảng giá trị.
* **Gradient**: Là đạo hàm của vecto cường độ màu sắc giúp phát hiện hướng di chuyển của các vật thể trong hình ảnh.
* **Local cell**: Ô cục bộ. Trong thuật toán HOG, một hình ảnh được chia thành nhiều cells bởi một lưới ô vuông. Mỗi cell được gọi là một ô cục bộ.
* **Local portion**: Vùng cục bộ. Là một vùng trước trích suất ra từ ô vuông trên hình ảnh. Trong phần trình bày về thuật toán thì vùng cục bộ còn được gọi là block.
* **Local normalization**: Phép chuẩn hóa được thực hiện trên một vùng cục bộ. Thường là chia cho norm chuẩn bậc 2 hoặc norm chuẩn bậc 1. Mục đích của việc chuẩn hóa là để đồng nhất các giá trị cường độ màu sắc về chung một phân phối. Ta sẽ làm rõ hơn trong phần trình bày thuật toán.
* **Gradient direction**: Phương gradient. Là độ lớn góc giữa vecto gradient x và y giúp xác định phương thay đổi cường độ màu sắc hay chính là phương đổ bóng của hình ảnh. Giả sử Gx,Gy lần lượt là giá trị gradient theo lần lượt phương x và y của hình ảnh. Khi đó phương gradient được tính như sau:
* **Gradient magnitude**: Độ lớn gradient. Là chiều dài của vecto gradient theo phương x và phương y. Biểu diễn phân phối histogram của vecto này theo vecto phương gradient sẽ thu được vecto mô tả đặc trưng HOG. Độ lớn gradient được tính như sau:
  1. Xây dựng Vecto HOG cho hình ảnh

Có 5 bước cơ bản để xây dựng một vector HOG cho hình ảnh, bao gồm:

* Tiền xử lý
* Tính gradient
* Tính vector đặc trưng cho từng ô (cells)
* Chuẩn hóa khối (blocks)
* Tính toán vector HOG

1. Tiền xử lý

Để thuận tiện cho việc chia hình ảnh thành các khối , các ô ở các bước sau , nhiệm vụ đầu tiên là cần phải resize các ảnh trong tập dữ liệu về cùng một kích thước (ví dụ như 64x128).

A group of people running on a track

Description automatically generated

1. Tính Gradient

Hai phép nhân chập ảnh gốc với 2 chiều, tương ứng với các toán tử lấy đạo hàm theo hai hướng Ox và Oy. Trong đó, 2 hướng tương ứng đó là:

và

Và nếu có một ảnh input là **I**, ta sẽ có 2 ảnh đạo hàm riêng theo 2 hướng đó, theo công thức:

Khi đó, ta có thể tính được Gradient bao gồm hai thành phần cường độ(Gradient Magnitude) và hướng(Gradient Derection) theo công thức **(\*)**:

**Ví dụ:** Giả sử ta có một điểm ảnh như sau:

A close up of a person's face

Description automatically generated

Chúng ta sẽ áp dụng các công thức trên để tính được gradient của điểm ảnh này:

Cường độ: |G| =

Hướng:

Đối với hình ảnh màu, gradient của ba kênh(red, green và blue) được đánh giá. Độ lớn của gradient tại một điểm ảnh là giá trị lớn nhất của cường độ gradient của ba kênh, và góc là góc tương ứng với gradient tối đa.

Sau bước này, kết quả thu được sẽ là:

A collage of a person running

Description automatically generated

1. Tính vector đặc trưng cho từng ô (cells)

Để tính toán vector đặc trưng cho từng ô (cell), chúng ta cần chia hình ảnh thành các block, mỗi block lại chia đều thành các cell. Để xác định được số block, chúng ta sẽ sử dụng công thức sau:

Trong đó:

Sau khi xác định số block và kích thước mỗi block, cell, để tính toán vector đặc trưng cho từng cell, chúng ta cần:

1. Chia không gian hướng thành p bin(số chiều vector đặc trưng của ô).
2. Rời rạc hóa góc hướng nghiêng tại mỗi điểm ảnh vào trong các bin.

Giả sử góc hướng nghiêng tại pixel ở vị trí (x,y) có độ lớn là alpha(x,y)

Trường hợp rời rạc hóa unsigned-HOG với p=9:

Trường hợp rời rạc hóa signed-HOG với p=18:

Giá trị bin được định lượng bởi tổng cường độ biến thiên của các pixels thuộc về bin đó. Sau khi tính toán đặc trưng ô, ta sẽ nối các vector đặc trưng ô để thu được vector đặc trưng khối. Số chiều vector đặc trưng khối tính theo công thức:

Trong đó:

* n là số ô trong khối

**Ví dụ:** Trong trường hợp này, hình ảnh của chúng ta có kích thước là 64x128, ta sẽ chia mỗi hình ảnh thành các block có kích thước 16x16. Mỗi block sẽ bao gồm 4 cell, mỗi cell có kích thước là 8x8.

A person running in a race

Description automatically generated

Tiếp theo, tiến hành tính toán đặc trưng HOG tại mỗi cell *sử dụng không gian hướng 9 bin*, trường hợp “**unsigned-HOG**”. Hướng gradient sẽ chạy trong khoảng 0 độ đến 180 độ, trung bình 20 độ mỗi bin.

Tại mỗi cell, xây dựng một biểu đồ cường độ gradient bằng cách vote các pixel vào biểu đồ. Trọng số vote của mỗi pixel phụ thuộc hướng và cường độ gradient (được tính toán từ bước 2) của pixel đó. Ví dụ:

A person pointing at a grid

Description automatically generated

A graph with red and blue lines and numbers

Description automatically generated

Như trong hình ảnh trên, đầu tiên là pixel có bao quanh màu xanh lam. Nó có hướng 80 độ và cường độ là 2, vì vậy ta thêm 2 vào bin thứ 5 (hướng 80 độ). Tiếp theo là pixel có bao quanh màu đỏ. Nó có hướng 10 độ và cường độ 4. Vì không có bin 10 độ, nên ta vote cho bin 0 độ và 20 độ, mỗi bin thêm 2 đơn vị. Sau khi vote hết các pixel trong một cell kích thước 8x8 vào 9 bin, ta có thể thu được kết quả như sau:

A graph of a bar graph

Description automatically generated

1. Chuẩn hóa khối (blocks)

Để tăng cường hiệu năng nhận dạng, các histogram cục bộ sẽ được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một khối và sử dụng giá trị đó để chuẩn hóa tất cả các ô trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

Đầu tiên, hãy xem xét ảnh hưởng của việc chuẩn hóa tới các vector gradient trong ví dụ sau:

A collage of images of a person holding a card

Description automatically generated

Hãy nhìn vào các giá trị pixel thực tế và sự thay đổi của vector gradient của ba trường hợp trên trong hình ảnh sau:

A group of images of numbers and equations

Description automatically generated

* Các con số trong các ô là giá trị pixel của các điểm ảnh lân cận điểm ảnh được đánh dấu màu đỏ.
* Delta F là đạo hàm theo riêng hai hướng của điểm ảnh ([Ix, Iy])
* | Delta F| là giá trị cường độ điểm ảnh (Gradient Magnitude), tính theo công thức (\*)

Có nhiều phương pháp có thể được dùng để chuẩn hóa khối. Gọi v là vector cần chuẩn hóa chứa tất cả các histogram của mội khối. ‖v(k)‖ là giá trị chuẩn hóa của v theo các chuẩn k=1, 3 và e là một hằng số nhỏ. Khi đó, các giá trị chuẩn hóa có thể tính bằng một trong những công thức sau:

Ghép các vector đặc trưng khối sẽ thu được vector đặc trưng R-HOG cho ảnh. Số chiều vector đặc trưng ảnh tính theo công thức :

Trong đó :

* n là số khối của hình ảnh

1. Tính toán vector đặc trưng HOG

Với mỗi hình ảnh kích thước 64x128, chia thành các block 16x16 chồng nhau, sẽ có 7 block ngang và 15 block dọc, nên sẽ có 7x15 = 105 blocks.

Mỗi block gồm 4 cell. Khi áp dụng biểu đồ 9-bin cho mỗi cell, mỗi block sẽ được đại diện bởi một vector có kích thước 36x1.

Vì vậy, khi nối tất cả các vector trong một block lại với nhau, ta sẽ thu được vector đặc trưng HOG của ảnh có kích thước 105x36x1 = 3780x1.

III. Phương pháp đề xuất

**Phương Pháp Đề Xuất: Ứng Dụng HOG, Edge Dectection và SVM cho Nhận Dạng Tư Thế Người**

Hệ thống sử dụng HOG, Edge dectection và SVM để nhận dạng và phân loại tư thế người dựa trên video hoặc chuỗi hình ảnh theo từng bước cụ thể sau đây:

**1. Thu thập dữ liệu (Data Collection)**

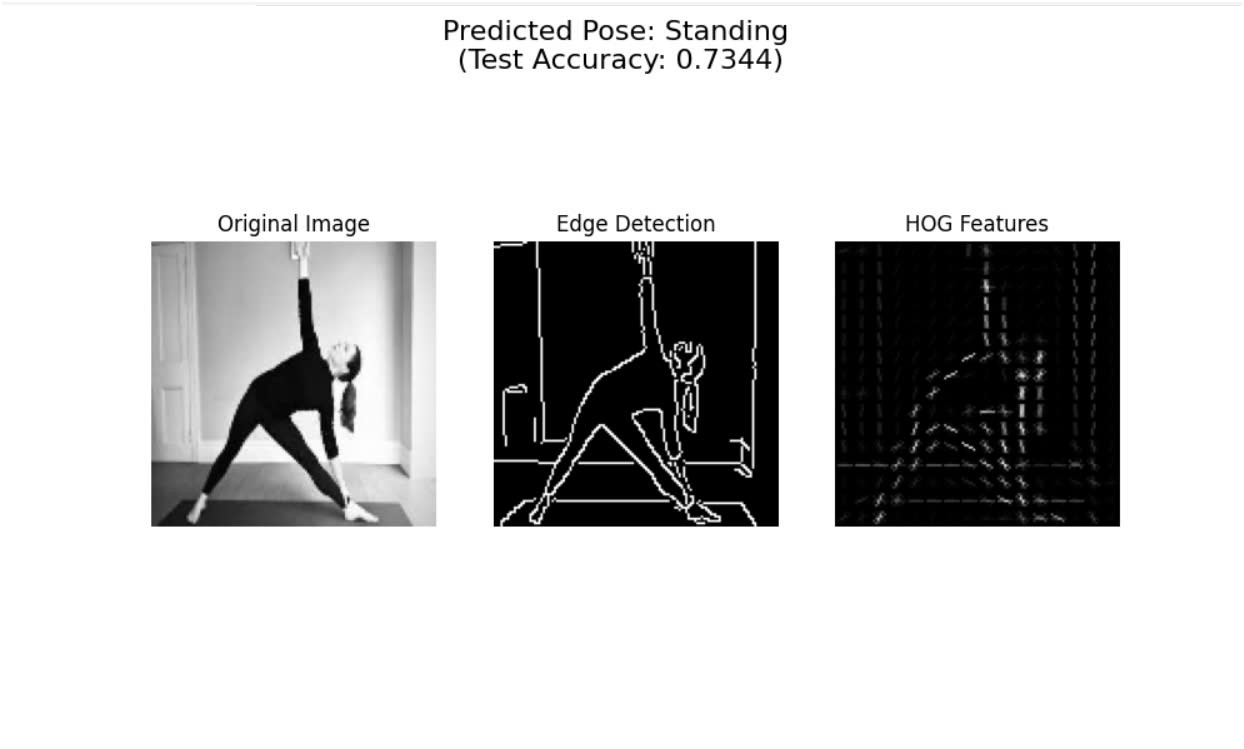
Để xây dựng một hệ thống nhận diện tư thế, bước đầu tiên là chuẩn bị dữ liệu đầu vào. Người dùng thu thập các hình ảnh trong các thư mục con, mỗi thư mục đại diện cho một tư thế cụ thể: "standing" (đứng), "lying" (nằm), và "sitting" (ngồi). Các thư mục này được tổ chức sao cho tên của chúng tương ứng với nhãn mà hệ thống sẽ học. Mỗi ảnh trong tập dữ liệu sẽ được gán nhãn bằng cách lấy tên thư mục của ảnh đó. Khi mô hình huấn luyện, các nhãn này sẽ giúp mô hình hiểu hình ảnh thuộc các tư thế khác nhau để nhận diện chính xác hơn.

Các thư mục này có thể bao gồm nhiều hình ảnh với các kích thước, điều kiện ánh sáng, góc chụp khác nhau, nhằm tạo ra một bộ dữ liệu phong phú, giúp mô hình học được các yếu tố phân biệt đặc trưng cho từng tư thế trong các điều kiện đa dạng.

**2. Tiền xử lý ảnh:**

Khi đọc mỗi hình ảnh từ tập dữ liệu, ảnh được chuyển sang chế độ grayscale (ảnh xám) để giảm bớt độ phức tạp trong phân tích. Ảnh màu thường có ba kênh màu (RGB) chứa nhiều thông tin không cần thiết cho nhận diện tư thế, vì phần lớn thông tin cần thiết nằm ở cấu trúc biên và độ sáng. Chuyển đổi sang grayscale giúp giảm độ phức tạp và tăng hiệu quả mà không làm giảm độ chính xác. Sau đó, ảnh được thay đổi kích thước thành 128x128 pixel. Kích thước này được chọn để cân bằng giữa độ chi tiết và tốc độ xử lý. Chuẩn hóa kích thước ảnh giúp quá trình trích xuất đặc trưng với HOG (Histogram of Oriented Gradients) và Edge Detection được thực hiện ổn định, đồng thời tránh tình trạng kích thước không đồng nhất ảnh hưởng đến mô hình.

**3. Trích xuất đặc trưng với Edge Dectection và HOG (Histogram of Oriented Gradients):**

****

Sau khi chuyển sang grayscale và resize, ảnh được trích xuất đặc trưng thông qua HOG và Edge Detection. Phương pháp HOG cho phép mô tả hình dạng và cấu trúc tổng thể trong ảnh thông qua phân tích gradient, giúp nhận diện các chi tiết và cấu trúc cơ bản dù có sự che khuất. Cụ thể, HOG tính toán gradient của từng pixel theo hướng dọc và ngang để xác định biên và các chi tiết trong ảnh. Các hướng gradient này được tổ chức thành histogram trong từng ô nhỏ (cells), biểu diễn tần suất của các hướng gradient, từ đó tạo ra một vector đặc trưng.

Song song với HOG, Edge Detection (phát hiện biên) cũng được áp dụng để tăng cường khả năng phát hiện biên của các bộ phận cơ thể. Phương pháp này xác định các đường viền của đối tượng trong ảnh, làm nổi bật các chi tiết biên và cấu trúc cơ bản của cơ thể. Sự kết hợp của Edge Detection và HOG cung cấp cho hệ thống một bộ đặc trưng phong phú hơn, mô tả chính xác các khớp và tư thế cơ thể.

**4.Chuẩn bị dữ liệu huấn luyện và kiểm tra**

Sau khi trích xuất đặc trưng HOG và Edge Detection cho mỗi ảnh, dữ liệu được chia thành hai phần: tập huấn luyện và tập kiểm tra theo tỷ lệ 80-20. Tập huấn luyện chứa phần lớn ảnh để mô hình học các mẫu tư thế khác nhau. Tập kiểm tra bao gồm các ảnh chưa từng được mô hình thấy trước đó, giúp đánh giá độ chính xác của mô hình trên dữ liệu mới. Các đặc trưng HOG và biên được kết hợp để tạo thành một vector đặc trưng đầy đủ cho từng ảnh, với mỗi vector sẽ được gán nhãn tương ứng để hệ thống học phân biệt các tư thế.

**5. Xây dựng và huấn luyện mô hình SVM (Support Vector Machine):**

Mô hình SVM (Support Vector Machine) được sử dụng để học và phân loại tư thế dựa trên các đặc trưng HOG (Histogram of Oriented Gradients) và thông tin biên được phát hiện từ ảnh. Khi áp dụng, SVM sẽ phân tích các đặc trưng HOG và các đường biên của các khớp cơ thể trong từng hình ảnh, từ đó đưa ra nhãn tư thế tương ứng, chẳng hạn như "đứng", "ngồi", "chạy", hoặc các tư thế cụ thể khác. Việc sử dụng HOG là rất cần thiết vì nó cung cấp thông tin chi tiết về hình dạng và cấu trúc của cơ thể, giúp cải thiện khả năng phân loại bằng cách trích xuất những đặc trưng mạnh mẽ. Mô hình SVM tập trung vào việc tìm một siêu phẳng tối ưu để phân chia các tư thế khác nhau dựa trên các đặc trưng HOG và Edge Detection của từng khung hình. Với SVM, từng khung hình được xử lý độc lập, và mô hình tìm cách tối ưu để phân loại tư thế một cách chính xác dựa trên các đặc trưng từ từng ảnh mà không cần đến thông tin chuỗi thời gian. Điều này giúp SVM trở thành lựa chọn phù hợp cho các bài toán nhận diện tư thế tĩnh, nơi mỗi ảnh được phân loại độc lập.

**6. Dự đoán tư thế với ảnh mới**

Sau khi hoàn thành huấn luyện, mô hình SVM có khả năng nhận diện tư thế từ các ảnh mới. Quá trình này bao gồm các bước như sau:

Tiền xử lý: Ảnh đầu vào mới sẽ trải qua các bước tiền xử lý tương tự như ảnh trong tập huấn luyện. Đầu tiên, ảnh được chuyển sang chế độ grayscale để giảm độ phức tạp. Sau đó, ảnh được resize về kích thước chuẩn 128x128 để đảm bảo đồng nhất khi tính toán các đặc trưng. Tiếp theo, ảnh sẽ được phân tích biên qua phương pháp Edge Detection, giúp xác định các đường biên và cấu trúc nổi bật. Cuối cùng, đặc trưng HOG (Histogram of Oriented Gradients) được trích xuất từ ảnh, nhằm tạo ra vector đặc trưng đại diện cho cấu trúc của tư thế trong hình.

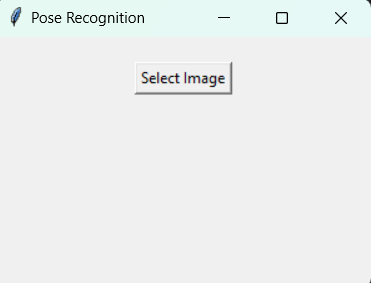
Dự đoán và Kết quả: Vector đặc trưng HOG và thông tin biên từ Edge Detection sẽ được chuyển vào mô hình SVM. Mô hình SVM sẽ phân loại ảnh đầu vào và chọn nhãn tư thế có xác suất cao nhất. Kết quả cuối cùng bao gồm nhãn tư thế được dự đoán và mức độ chính xác của dự đoán (được biểu thị dưới dạng xác suất dự đoán hoặc khoảng cách từ siêu phẳng quyết định trong SVM).

Hiển thị: Kết quả được hiển thị lên màn hình, bao gồm nhãn tư thế dự đoán và mức độ chính xác của mô hình. Ngoài ra, ảnh gốc sẽ được kèm theo các hình ảnh minh họa của quá trình Edge Detection và HOG, giúp người dùng có thể trực quan thấy các đặc trưng biên và cấu trúc đã giúp mô hình đưa ra dự đoán. Điều này cung cấp cái nhìn rõ ràng về cách thức mô hình phân tích và nhận diện tư thế từ các ảnh đầu vào.

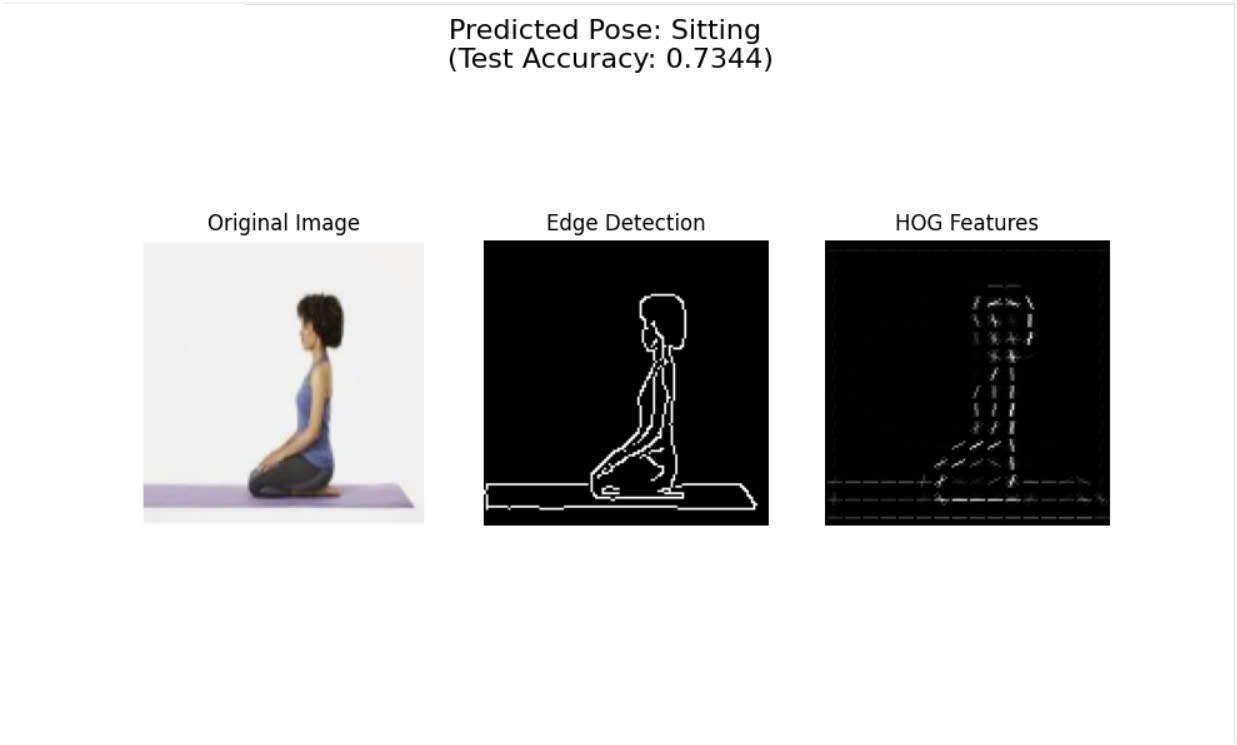
IV. Báo cáo thực nghiệm

Trong đồ án này , sử dụng bộ dữ liệu huấn liệu gồm 3 folder hình ảnh các tư thế như đứng, ngồi, nằm. Mỗi folder gồm 70-80 ảnh để làm dataset. Thời gian xử lý nhận diện rơi vào khoảng 2,7s cho thời gian huấn luyện và 1s cho thời gian dự đoán.

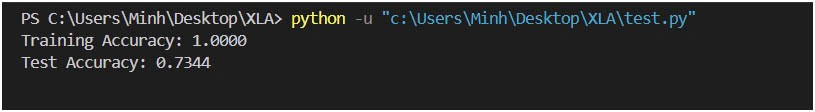
**1. Giao diện hiển thị cho người dùng**

****

**2. Nhận diện tư thế**

****

1. So sánh độ chính xác:



Note : sửa lại báo cáo . đánh giá độ chính xác , tại sao kết quả như này . kết luận . ưu điểm nhược điểm ,

Mỗi trương tách ra 1 trang

Kết luận , buổi sau là báo cáo cuối cùng

IV. References

[1] A. Raza, A. Qadri, I. Akhtar, N. Abdelsamee, and M. Alabdulhafith, “LogRF: An Approach to Human Pose Estimation Using Skeleton Landmarks for Physiotherapy Fitness Exercise Correction,” *IEEE Access*, vol. PP, pp. 1–1, Jan. 2023, doi: 10.1109/ACCESS.2023.3320144.

[2] J. Stenum, K. M. Cherry-Allen, C. O. Pyles, R. D. Reetzke, M. F. Vignos, and R. T. Roemmich, “Applications of Pose Estimation in Human Health and Performance across the Lifespan,” *Sensors*, vol. 21, no. 21, Art. no. 21, Jan. 2021, doi: 10.3390/s21217315.

[3] M. Cormier, A. Clepe, A. Specker, and J. Beyerer, “Where are we with Human Pose Estimation in Real-World Surveillance?,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 591–601. doi: 10.1109/WACVW54805.2022.00065.

[4] “Human Pose-based Estimation, Tracking and Action Recognition with Deep Learning: A Survey,” ar5iv. Accessed: Oct. 09, 2024. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2310.13039

[5] E. Samkari, M. Arif, M. Alghamdi, and M. A. Al Ghamdi, “Human Pose Estimation Using Deep Learning: A Systematic Literature Review,” *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, Art. no. 4, Dec. 2023, doi: 10.3390/make5040081.

[6] S. Dubey and M. Dixit, “A comprehensive survey on human pose estimation approaches,” *Multimed. Syst.*, vol. 29, no. 1, pp. 167–195, Feb. 2023, doi: 10.1007/s00530-022-00980-0.

[7] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” May 30, 2019, *arXiv*: arXiv:1812.08008. Accessed: Oct. 10, 2024. [Online]. Available: http://arxiv.org/abs/1812.08008

[8] X. Bai, X. Wei, Z. Wang, and M. Zhang, “CONet: Crowd and occlusion-aware network for occluded human pose estimation,” *Neural Netw.*, vol. 172, p. 106109, Apr. 2024, doi: 10.1016/j.neunet.2024.106109.

[9] “LPSNet: End-to-End Human Pose and Shape Estimation with Lensless Imaging.” Accessed: Oct. 10, 2024. [Online]. Available: https://arxiv.org/html/2404.01941v3

[10] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human Pose Estimation with Iterative Error Feedback,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 4733–4742. doi: 10.1109/CVPR.2016.512.

[11] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, “Pose Flow: Efficient Online Pose Tracking,” Feb. 2018. doi: 10.48550/arXiv.1802.00977.

[12] T. Vu, “Bài 19: Support Vector Machine,” Tiep Vu’s blog. Accessed: Nov. 14, 2024. [Online]. Available: https://machinelearningcoban.com/2017/04/09/smv/

[13] “Edge detection .pdf.” Accessed: Oct. 22, 2024. [Online]. Available: https://bbau.ac.in/dept/CS/TM/Edge%20detection%20.pdf

[14] “Canny edge detector,” *Wikipedia*. Sep. 21, 2024. Accessed: Oct. 22, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Canny\_edge\_detector&oldid=1246919049

[15] “Tìm hiểu về phương pháp mô tả đặc trưng HOG (Histogram of Oriented Gradients).” Accessed: Oct. 24, 2024. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-phuong-phap-mo-ta-dac-trung-hog-histogram-of-oriented-gradients-V3m5WAwxZO7