|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH** |
| **BÁO CÁO MÔN HỌC**  **TRÍ TUỆ NHÂN TẠO VÀ ỨNG DỤNG**  **PHÂN LOẠI CẢM XÚC KHUÔN MẶT**  Ngành: Robot và trí tuệ nhân tạo  Lớp: 22DRTA1  **GIẢNG VIÊN :**  **ThS. Ngô An Thuyên**   |  |  |  | | --- | --- | --- | | **Sinh viên thực hiện:** | **MSSV:** | **Lớp:** | | Nguyễn Văn Đạt | 2286300010 | 22DRTA1 | | Huỳnh Long | 2286300028 | 22DRTA1 | | Nguyễn Chấn Huy | 2286300020 | 22DRTA1 |   *TP. Hồ Chí Minh, ngày tháng 8 năm 2025* | |

|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH** |
| **BÁO CÁO MÔN HỌC**  **MẠNG MÁY TÍNH TRONG CÔNG NGHIỆP**  **Hệ thống điều khiển và giám sát nhà thông minh sử dụng giao tiếp Modbus TCP/IP**  Ngành: Robot và trí tuệ nhân tạo  Lớp: 22DRTA1  **GIẢNG VIÊN :**  **TS. Phạm Quốc Thiện**   |  |  |  | | --- | --- | --- | | **Sinh viên thực hiện:** | **MSSV:** | **Lớp:** | | Nguyễn Văn Đạt | 2286300010 | 22DRTA1 | | Huỳnh Long | 2286300028 | 22DRTA1 | | Nguyễn Chấn Huy | 2286300020 | 22DRTA1 |   *TP. Hồ Chí Minh, ngày tháng 8 năm 2025* | |

**Mục lục**

[Tóm tắt 3](#_Toc204755730)

[I. Mục tiêu 4](#_Toc204755731)

[II. Các bước thực hiện 4](#_Toc204755732)

[III. Thuật toán sử dụng 5](#_Toc204755733)

[**1. SVM** 5](#_Toc204755734)

[**a. Linear kernel** 7](#_Toc204755735)

[**b. Polynomial kernel** 7](#_Toc204755736)

[**c. RBF (Radial Basis Function) kernel** 7](#_Toc204755737)

[**d. Hinge Loss** 10](#_Toc204755738)

[**2. HOG** 11](#_Toc204755739)

[**Khái niệm** 11](#_Toc204755740)

[**Nguyên lý hoạt động** 11](#_Toc204755741)

[**Ưu điểm** 12](#_Toc204755742)

[**Ứng dụng** 12](#_Toc204755743)

[**3. Local Binary Pattern (LBP)** 12](#_Toc204755744)

[**Khái niệm** 12](#_Toc204755745)

[**Nguyên lý hoạt động** 12](#_Toc204755746)

[**Công thức** 12](#_Toc204755747)

[**Ưu điểm** 13](#_Toc204755748)

[**Ứng dụng** 13](#_Toc204755749)

[**4. Landmark** 13](#_Toc204755750)

[**Khái niệm** 13](#_Toc204755751)

[**Nguyên lý hoạt động** 13](#_Toc204755752)

[**Cấu trúc điểm** 13](#_Toc204755753)

[**Ưu điểm** 13](#_Toc204755754)

[**Ứng dụng** 13](#_Toc204755755)

[IV. Triển khai 14](#_Toc204755756)

[**1. Cấu trúc dataset** 14](#_Toc204755757)

[**2. Lưu đồ thuật toán** 14](#_Toc204755758)

[V. Kết quả và đánh giá 16](#_Toc204755759)

[**1. Bảng kết quả huấn luyện** 16](#_Toc204755760)

[**2. Confusion matrix** 17](#_Toc204755761)

[**3. Kết quả nhận diện** 18](#_Toc204755762)

[VI. Kết luận 19](#_Toc204755763)

[VII. Tài liệu tham khảo 21](#_Toc204755764)

# **Tóm tắt**

Đề tài tập trung vào xây dựng một hệ thống nhận diện cảm xúc khuôn mặt nhằm nâng cao khả năng tương tác người – máy trong các robot dịch vụ. Hệ thống khai thác các đặc trưng hình ảnh như HOG (Histogram of Oriented Gradients), LBP (Local Binary Pattern) và Landmark facial points để trích xuất thông tin khuôn mặt, kết hợp với thuật toán phân loại SVM (Support Vector Machine) sử dụng các kernel khác nhau (linear, polynomial, RBF) nhằm phân loại cảm xúc người dùng.

Phần lý thuyết trình bày chi tiết cơ sở của các thuật toán sử dụng, bao gồm đặc điểm của từng loại kernel trong SVM, cơ chế hoạt động và ứng dụng của HOG, LBP và Landmark trong xử lý ảnh khuôn mặt. Dữ liệu huấn luyện được xây dựng có cấu trúc rõ ràng, qua đó nhóm triển khai hệ thống nhận diện cảm xúc theo lưu đồ thuật toán cụ thể.

Kết quả thực nghiệm thể hiện qua bảng số liệu huấn luyện, ma trận nhầm lẫn và tỷ lệ nhận diện cho thấy hệ thống có độ chính xác cao, phù hợp ứng dụng thực tế trong các robot dịch vụ để nhận diện và phản hồi cảm xúc người dùng một cách tự nhiên và thân thiện hơn.

# **I. Mục tiêu**

Mục tiêu chính của đề tài là xây dựng một hệ thống có khả năng nhận diện cảm xúc của người dùng thông qua hình ảnh khuôn mặt, từ đó tích hợp vào robot dịch vụ để cải thiện khả năng tương tác giữa con người và máy móc. Việc robot có thể hiểu được cảm xúc người dùng sẽ giúp tăng cường trải nghiệm sử dụng, tạo cảm giác thân thiện và gần gũi hơn trong quá trình giao tiếp. Đây là bước quan trọng trong việc đưa robot dịch vụ đến gần hơn với nhu cầu thực tế của xã hội hiện đại.

Trước hết, đề tài tập trung vào việc tìm hiểu và áp dụng các kỹ thuật trích xuất đặc trưng khuôn mặt nhằm phục vụ quá trình nhận diện cảm xúc. Các kỹ thuật như HOG (Histogram of Oriented Gradients), LBP (Local Binary Pattern) và Landmark facial points được lựa chọn vì khả năng mô tả hình dạng, kết cấu và vị trí các điểm đặc trưng trên khuôn mặt – những yếu tố có tính quyết định trong việc biểu hiện và phân biệt cảm xúc. Việc kết hợp các phương pháp này giúp mô hình học được đầy đủ các đặc trưng cần thiết để nhận diện biểu cảm khuôn mặt một cách chính xác và ổn định.

Tiếp theo, đề tài tiến hành nghiên cứu và triển khai mô hình phân loại cảm xúc dựa trên thuật toán SVM (Support Vector Machine), một phương pháp học máy phổ biến và hiệu quả cao trong các bài toán phân loại. Các loại kernel khác nhau như linear, polynomial và RBF được thử nghiệm nhằm đánh giá ảnh hưởng của từng kernel đến độ chính xác của mô hình. Việc thử nghiệm đa dạng kernel cũng giúp nhóm nghiên cứu lựa chọn được cấu hình tối ưu nhất cho bài toán cụ thể về phân loại cảm xúc từ khuôn mặt.

Một mục tiêu quan trọng khác là xây dựng một quy trình hoàn chỉnh – hay còn gọi là pipeline – cho toàn bộ hệ thống nhận diện cảm xúc. Pipeline này bao gồm các bước chính từ xử lý dữ liệu đầu vào, trích xuất đặc trưng, huấn luyện mô hình cho đến phân loại cảm xúc đầu ra. Cách tiếp cận này không chỉ giúp hệ thống hoạt động trơn tru mà còn tạo nền tảng cho việc mở rộng và tùy biến trong các ứng dụng khác nhau sau này.

Bên cạnh đó, đề tài chú trọng vào việc đánh giá hiệu suất của hệ thống bằng các chỉ số định lượng như độ chính xác, ma trận nhầm lẫn và biểu đồ phân phối kết quả. Đây là cơ sở để kiểm chứng tính khả thi và độ tin cậy của hệ thống trong môi trường thực tế. Việc đánh giá kỹ lưỡng cũng giúp nhận diện được những điểm hạn chế cần cải thiện trong các bước tiếp theo.

Cuối cùng, hướng phát triển dài hạn của đề tài là tích hợp hệ thống nhận diện cảm xúc vào các robot dịch vụ. Với khả năng nhận biết và phân tích cảm xúc người dùng, robot có thể tự động điều chỉnh hành vi, lời nói hoặc cách tương tác sao cho phù hợp với trạng thái cảm xúc hiện tại của con người. Điều này không chỉ nâng cao hiệu quả giao tiếp mà còn tạo ra những trải nghiệm mang tính cá nhân hóa, góp phần xây dựng thế hệ robot dịch vụ thông minh và nhân văn hơn.

# **II. Các bước thực hiện**

Trong quá trình triển khai đề tài, nhóm thực hiện tiến hành theo một quy trình gồm nhiều bước chính, bắt đầu từ khâu thu thập và xử lý dữ liệu hình ảnh khuôn mặt. Dữ liệu được lấy từ các bộ dữ liệu công khai có sẵn hoặc có thể được tự tạo, bao gồm các hình ảnh được gán nhãn cảm xúc như vui, buồn, tức giận, ngạc nhiên, sợ hãi, ghê tởm và trung tính. Nhằm đảm bảo độ chính xác và tính công bằng trong huấn luyện mô hình, dữ liệu được tiền xử lý để chuẩn hóa kích thước ảnh, đồng thời cân bằng số lượng mẫu giữa các lớp cảm xúc, tránh hiện tượng mất cân bằng dữ liệu gây sai lệch trong phân loại.

Sau khi đã có dữ liệu đầu vào đạt chuẩn, bước tiếp theo là trích xuất đặc trưng khuôn mặt – một khâu quan trọng để mô hình có thể nhận biết được các đặc điểm cảm xúc tiềm ẩn trong từng ảnh khuôn mặt. Trong đó, phương pháp HOG (Histogram of Oriented Gradients) được sử dụng để thu thập thông tin về các cạnh và hướng cạnh – yếu tố phản ánh hình dạng tổng thể và các đường nét khuôn mặt. Bên cạnh đó, LBP (Local Binary Pattern) giúp trích xuất các đặc trưng về kết cấu bề mặt da – rất nhạy với các thay đổi nhỏ trong biểu cảm khuôn mặt. Ngoài ra, các điểm Landmark trên khuôn mặt cũng được xác định, bao gồm các điểm như khóe mắt, miệng, chân mày – những vùng có xu hướng thay đổi rõ rệt khi cảm xúc con người thay đổi. Việc kết hợp ba phương pháp trích xuất đặc trưng này mang lại một bộ đặc trưng đa chiều, vừa bao quát vừa chi tiết, góp phần tăng độ chính xác cho hệ thống.

Tiếp theo là bước huấn luyện mô hình phân loại cảm xúc. Nhóm nghiên cứu lựa chọn thuật toán SVM (Support Vector Machine) làm mô hình chính do khả năng phân loại mạnh mẽ và hiệu quả trong không gian đặc trưng cao chiều. Trong quá trình huấn luyện, các loại kernel khác nhau như tuyến tính (linear), phi tuyến bậc hai (polynomial) và Gaussian RBF (Radial Basis Function) được thử nghiệm để tìm ra cấu hình tối ưu nhất cho bài toán phân loại cảm xúc. Đồng thời, nhóm áp dụng kỹ thuật đánh giá chéo (cross-validation) cùng quá trình tinh chỉnh tham số (hyperparameter tuning) nhằm tăng cường độ chính xác, giảm thiểu hiện tượng overfitting và cải thiện hiệu suất tổng thể của mô hình.

Sau khi có mô hình tối ưu, đề tài tiến hành xây dựng một pipeline hệ thống hoàn chỉnh cho bài toán nhận diện cảm xúc. Pipeline này bao gồm chuỗi các bước từ ảnh đầu vào được đưa qua khâu tiền xử lý, tiếp đó là trích xuất đặc trưng, sau đó là phân loại bằng mô hình SVM và cuối cùng là xuất ra nhãn cảm xúc dự đoán. Một lưu đồ thuật toán tổng thể được thiết kế rõ ràng để đảm bảo hệ thống vận hành mạch lạc, dễ dàng bảo trì, nâng cấp hoặc tích hợp vào các nền tảng robot dịch vụ trong tương lai.

Cuối cùng, hiệu quả của hệ thống được đánh giá thông qua các chỉ số như độ chính xác và ma trận nhầm lẫn (confusion matrix). Việc phân tích kết quả giúp nhóm nhận diện rõ điểm mạnh và điểm yếu của từng cấu hình mô hình, đồng thời đưa ra so sánh giữa các phương pháp trích xuất đặc trưng và các loại kernel SVM. Qua đó, đề tài không chỉ dừng lại ở việc xây dựng một hệ thống nhận diện cảm xúc hiệu quả mà còn đóng góp thêm những đánh giá có giá trị thực tiễn, phục vụ cho việc triển khai trong môi trường thực tế.

# **III. Thuật toán sử dụng**

## **1. SVM**

Support Vector Machine (SVM) là một thuật toán học máy dùng để phân loại dữ liệu bằng cách tìm ra hyperplane tối ưu trong không gian đặc trưng. Hyperplane này có nhiệm vụ phân tách các điểm dữ liệu thuộc các lớp khác nhau, đồng thời tối đa hóa khoảng cách (margin) giữa hyperplane và các điểm dữ liệu gần nhất của mỗi lớp. Mục tiêu của SVM là tạo ra một ranh giới quyết định vững chắc, giúp phân loại chính xác các dữ liệu mới.

Hyperplane là một không gian con có số chiều thấp hơn không gian dữ liệu gốc một đơn vị, đóng vai trò như ranh giới phân loại. Trong không gian một chiều, hyperplane là một điểm; trong không gian hai chiều, nó là một đường thẳng; trong không gian ba chiều, nó là một mặt phẳng. Tổng quát hơn, trong không gian n chiều, hyperplane có số chiều là n-1. SVM tìm kiếm hyperplane phù hợp nhất để phân tách dữ liệu dựa trên phương trình tổng quát:

Trong đó w là vectơ trọng số, x là vectơ đặc trưng của dữ liệu, và b là độ lệch.

SVM không chỉ tìm một hyperplane bất kỳ, mà còn tìm hyperplane có margin lớn nhất, tức khoảng cách xa nhất đến các điểm dữ liệu gần nhất của mỗi lớp. Việc tối đa hóa margin giúp mô hình tổng quát hóa tốt hơn trên dữ liệu mới, giảm nguy cơ overfitting. Hai đường biên hỗ trợ hyperplane có phương trình:

Khoảng cách giữa hai đường này chính là margin, được tính bằng:

Do đó, để tối đa hóa margin, SVM thực chất đang tối thiểu hóa độ lớn của vectơ trọng số w.



Hình 1: Mô tả phân loại dữ liệu bằng Hyperplane

Các điểm dữ liệu gần hyperplane nhất, quyết định vị trí và hướng của hyperplane tối ưu, được gọi là support vectors. Chỉ các điểm này ảnh hưởng đến việc tối ưu hóa, trong khi các điểm dữ liệu khác không có tác động đáng kể đến ranh giới phân loại.

Khi dữ liệu có thể phân tách hoàn toàn bằng một hyperplane, SVM sẽ tìm ra hyperplane tối ưu với margin lớn nhất mà không có điểm dữ liệu nào nằm giữa hoặc bị phân loại sai. Tuy nhiên, trong thực tế, nhiều bài toán có dữ liệu không thể phân tách tuyến tính. Để giải quyết vấn đề này, SVM sử dụng kernel trick.

Kernel trick giúp SVM xử lý dữ liệu phi tuyến tính bằng cách ánh xạ dữ liệu từ không gian ban đầu sang một không gian có số chiều cao hơn, nơi nó có thể phân tách tuyến tính. Thay vì tính toán trực tiếp trong không gian mới, kernel trick sử dụng các hàm kernel để tính toán tích vô hướng giữa các cặp điểm dữ liệu một cách hiệu quả. Một số hàm kernel phổ biến gồm:

**a. Linear kernel**: Phù hợp với dữ liệu có thể phân tách tuyến tính.

Trong đó, và là hai vectơ dữ liệu trong không gian đầu vào, còn là chuyển vị của . Linear kernel không thực hiện bất kỳ phép biến đổi phi tuyến nào, do đó hyperplane quyết định vẫn là tuyến tính trong không gian ban đầu. Kernel này thường được sử dụng khi dữ liệu có thể phân tách tuyến tính.

**b. Polynomial kernel**: Xét đến các quan hệ bậc cao giữa các đặc trưng.

Trong đó, **c** là một hằng số không âm, thường được chọn là 0 hoặc 1. Tham số này đóng vai trò điều chỉnh mức độ ảnh hưởng của phần tuyến tính trong đa thức. Khi c=0c = 0c=0, hàm kernel trở thành dạng thuần đa thức; ngược lại, khi c>0c > 0c>0, mô hình có thêm một thành phần tuyến tính, giúp tăng khả năng điều chỉnh linh hoạt cho các ranh giới phân loại.

Tham số **d** biểu thị **bậc của đa thức**, là yếu tố then chốt quyết định mức độ phức tạp của ranh giới phân loại mà mô hình có thể học được. Bậc đa thức càng cao, mô hình càng có khả năng học được các mối quan hệ phi tuyến tính phức tạp giữa các đặc trưng đầu vào. Tuy nhiên, nếu ddd quá lớn, mô hình sẽ dễ rơi vào tình trạng **overfitting** – tức là học quá sát với dữ liệu huấn luyện, nhưng lại kém hiệu quả khi áp dụng lên dữ liệu mới.

Việc sử dụng **hàm kernel đa thức** cho phép mô hình SVM mở rộng không gian đặc trưng một cách hiệu quả mà không cần tính toán trực tiếp các đặc trưng bậc cao. Điều này giúp mô hình **xem xét không chỉ các đặc trưng ban đầu mà còn cả các tương tác phức tạp giữa chúng**, từ đó tăng khả năng phân loại chính xác trên các bài toán có tính phi tuyến cao. Đây là một trong những điểm mạnh cốt lõi của kernel trick nói chung và kernel đa thức nói riêng trong học máy.

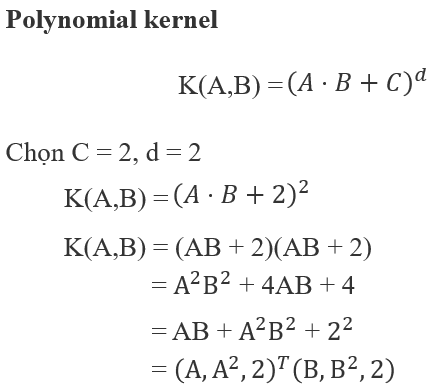
**c. RBF (Radial Basis Function) kernel**:

Tạo ra ranh giới phi tuyến tính phức tạp theo phân bố Gaussian.

Các tham số trong công thức:

* là bình phương khoảng cách Euclidean giữa hai vectơ.

(gamma) là một tham số dương, quyết định mức độ ảnh hưởng của một điểm dữ liệu lên mô hình. Khi nhỏ, mô hình có xu hướng trở thành tuyến tính. Khi lớn, mô hình trở nên phức tạp hơn và có thể dễ bị overfitting.



Hình 2: Mô tả cách tính tích vô hướng của các cặp điểm dữ liệu trong một không gian có chiều thấp hơn

RBF kernel còn được gọi là Gaussian kernel, giúp ánh xạ dữ liệu vào một không gian có số chiều rất cao, thậm chí vô hạn chiều, làm cho các bài toán phi tuyến tính có thể phân tách bằng một hyperplane.

Các hàm kernel này giúp SVM giải quyết bài toán phân loại phi tuyến tính bằng cách tính toán tích vô hướng trong một không gian đặc trưng có số chiều cao hơn mà không cần thực sự thực hiện phép biến đổi đó một cách tường minh. Việc lựa chọn kernel phù hợp và điều chỉnh các tham số của nó có ảnh hưởng quan trọng đến hiệu suất của mô hình SVM.

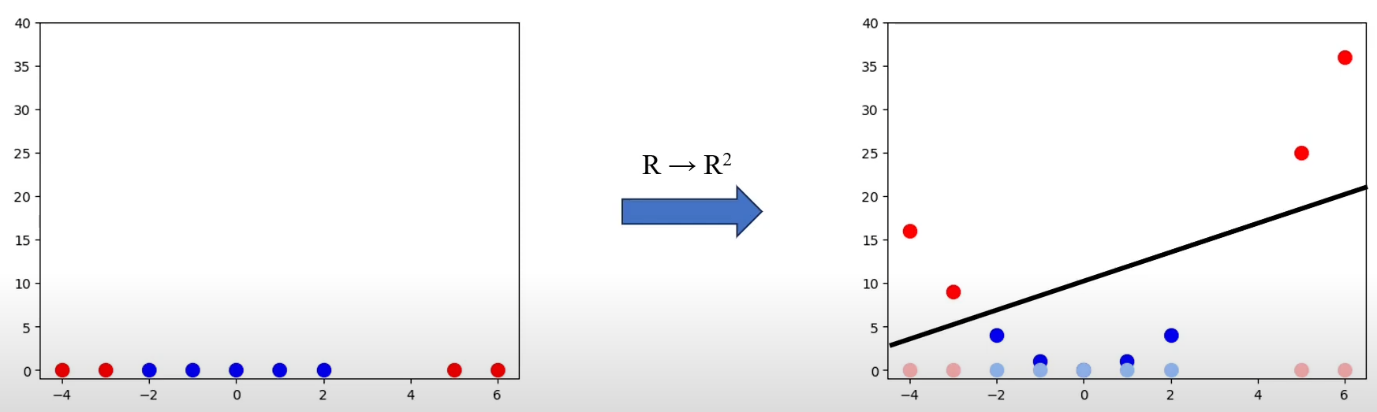


Hình 3: SVM tuyến tính



Hình 4: SVM không tuyến tính

Việc chọn hàm kernel phù hợp có vai trò quan trọng trong hiệu suất của SVM trên dữ liệu phi tuyến tính.



Hình 5: Mô tả tác dụng của kernel

Trong nhiều trường hợp, dữ liệu có thể chứa nhiễu hoặc điểm ngoại lai, khiến việc tìm một hyperplane phân tách hoàn toàn trở nên không khả thi. Để xử lý điều này, SVM sử dụng soft margin, cho phép một số điểm dữ liệu nằm trong vùng margin hoặc bị phân loại sai.

Tham số C giúp điều chỉnh sự cân bằng giữa:

* C lớn: Giảm lỗi phân loại nhưng có nguy cơ overfitting.
* C nhỏ: Cho phép một số lỗi để tăng tính tổng quát hóa.

SVM tìm hyperplane tối ưu bằng cách giải một bài toán tối ưu hóa có ràng buộc. Mục tiêu là tối thiểu hóa:

Dưới ràng buộc rằng tất cả điểm dữ liệu được phân loại đúng.

Trong soft margin SVM, hàm mục tiêu trở thành:

Trong đó là biến trễ, đại diện cho lượng lỗi phân loại.

**d. Hinge Loss**: Một hàm mất mát phổ biến trong SVM là Hinge Loss, với công thức:

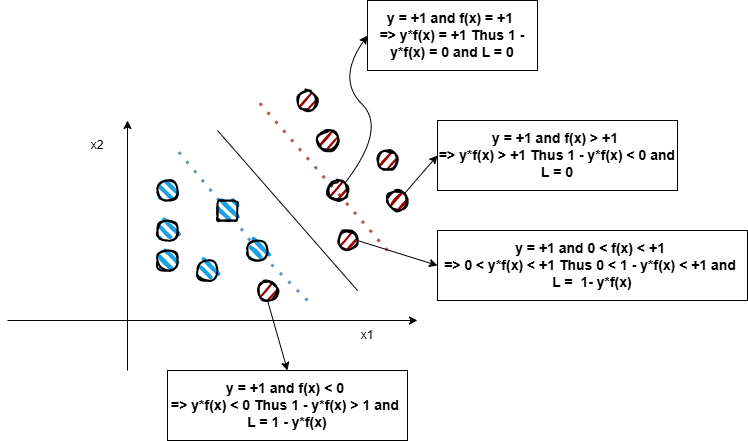
Với:

y là giá trị thật của đầu ra (+1 hoặc -1).

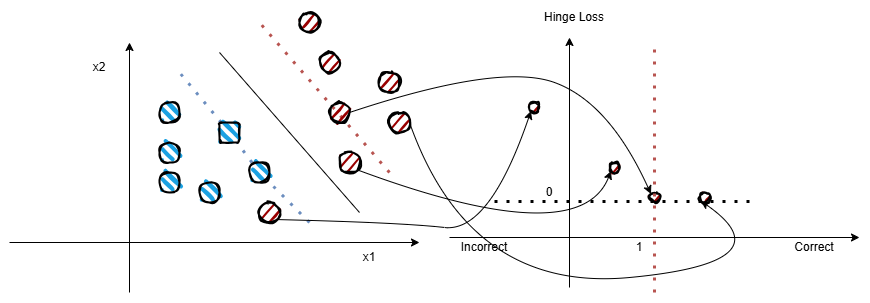
L 0.

Trong mô hình Soft Margin SVM, biến slack được định nghĩa là:

Biến thể hiện mức độ vi phạm margin của một điểm dữ liệu, đồng thời cũng chính là giá trị của Hinge Loss đối với điểm đó. Nếu một điểm dữ liệu nằm đúng phía của hyperplane và bên ngoài vùng margin, giá trị sẽ bằng 0. Ngược lại, nếu điểm đó nằm trong vùng margin hoặc bị phân loại sai, giá trị sẽ tăng lên.



Hình 6: Mô tả hàm hinge loss



Hình 7: Mô tả hàm hinge loss

Dựa vào định nghĩa trên, hàm mục tiêu của Soft Margin SVM có thể được viết lại dưới dạng tổng Hinge Loss như sau:

Trong đó:

* là thành phần điều chỉnh độ phức tạp của mô hình, đảm bảo SVM tìm được một hyperplane có margin lớn nhất.
* C là tham số điều chỉnh mức độ ưu tiên giữa việc tối đa hóa margin và giảm lỗi phân loại.
* là tổng Hinge Loss trên toàn bộ dữ liệu huấn luyện, giúp đo lường mức độ vi phạm margin của các điểm dữ liệu.

Hàm mục tiêu này thể hiện sự cân bằng giữa việc tối ưu hóa margin và hạn chế lỗi phân loại, đảm bảo SVM có khả năng tổng quát hóa tốt trên tập dữ liệu mới.

Bài toán tối ưu hóa này có thể được giải bằng các phương pháp như Lagrangian multipliers hoặc SMO (Sequential Minimal Optimization).

## **2. HOG**

HOG (Histogram of Oriented Gradients) là một phương pháp trích xuất đặc trưng hình ảnh dựa trên việc phân tích sự phân bố của các gradient định hướng. Kỹ thuật này tập trung vào mô tả cấu trúc, hình dạng và các đường biên trong ảnh, thông qua việc ghi lại cách các hướng cạnh thay đổi trong các vùng nhỏ của ảnh. HOG đặc biệt hiệu quả trong việc nhận diện các đối tượng có hình dạng rõ ràng và được sử dụng phổ biến trong các hệ thống thị giác máy tính.

Quy trình xử lý ảnh bằng HOG gồm nhiều bước tuần tự. Trước tiên, ảnh đầu vào được chuyển sang thang độ xám để đơn giản hóa thông tin màu sắc, sau đó tính toán gradient theo hai hướng cơ bản là ngang (x) và dọc (y) thông qua các toán tử như Sobel. Tiếp theo, ảnh được chia thành các ô nhỏ gọi là *cells* (thường có kích thước 8×8 pixel). Với mỗi ô, một histogram gradient định hướng được tạo ra, phản ánh tần suất xuất hiện của các hướng cạnh trong phạm vi ô đó. Thông thường, histogram bao gồm 9 hướng từ 0° đến 180°, đủ để biểu diễn các cạnh cơ bản.

Để tăng độ bền vững với nhiễu và thay đổi ánh sáng, các ô tiếp tục được nhóm thành *blocks* (ví dụ: 2×2 ô liền kề), và các histogram trong block được chuẩn hóa. Việc chuẩn hóa giúp đảm bảo rằng đặc trưng đầu ra không bị ảnh hưởng mạnh bởi độ sáng tổng thể hay độ tương phản cục bộ. Cuối cùng, tất cả các histogram chuẩn hóa từ toàn ảnh được nối lại thành một vector đặc trưng duy nhất – chính là đầu ra của HOG, dùng làm đầu vào cho các thuật toán học máy.

HOG có khả năng mô tả rất tốt hình dạng và các đường biên trong ảnh. Nhờ cơ chế chuẩn hóa theo khối, HOG chống nhiễu khá hiệu quả và giữ được tính ổn định khi ảnh bị thay đổi độ sáng hoặc có biến động nhỏ về vị trí đối tượng. Nhờ đó, phương pháp này trở thành lựa chọn lý tưởng cho các bài toán nhận diện hình dạng, đặc biệt là trong điều kiện không hoàn hảo về ánh sáng hoặc độ nhiễu cao.

HOG được sử dụng rộng rãi trong nhiều bài toán thị giác máy tính, điển hình nhất là phát hiện người trong ảnh (Human Detection) và nhận diện khuôn mặt. Ngoài ra, HOG còn có thể áp dụng để nhận diện tư thế, phân tích cử chỉ, hoặc trích xuất biểu cảm khuôn mặt, nhờ khả năng nắm bắt tốt cấu trúc hình học tổng thể của đối tượng.

## **3. Local Binary Pattern (LBP)**

Local Binary Pattern (LBP) là một phương pháp trích xuất đặc trưng tập trung vào mô tả kết cấu bề mặt của ảnh. Khác với các kỹ thuật phát hiện biên hay hình dạng, LBP dựa trên so sánh cường độ sáng giữa điểm ảnh trung tâm và các điểm ảnh lân cận. Nhờ khả năng phản ánh các chi tiết kết cấu nhỏ và ổn định với thay đổi về ánh sáng, LBP được ứng dụng hiệu quả trong các tác vụ như nhận diện khuôn mặt, nhận dạng biểu cảm và phân tích kết cấu vật liệu.Nguyên lý hoạt động:

Thuật toán LBP được triển khai bằng cách duyệt qua từng điểm ảnh trong ảnh xám. Với mỗi điểm, thuật toán sẽ chọn một tập hợp điểm ảnh xung quanh nó (thường là 8 điểm tạo thành một vòng tròn nhỏ quanh tâm) theo một bán kính xác định. Sau đó, giá trị cường độ của từng điểm lân cận sẽ được so sánh với giá trị của điểm trung tâm. Nếu giá trị của điểm lân cận **lớn hơn hoặc bằng** điểm trung tâm, gán giá trị là **1**; ngược lại, gán là **0**. Kết quả là một dãy nhị phân gồm các số 0 và 1.

Dãy nhị phân này được sắp theo thứ tự cố định và **chuyển thành một số thập phân duy nhất** – gọi là **mã LBP** – đại diện cho mẫu kết cấu tại điểm ảnh đó. Quá trình này được thực hiện trên toàn bộ ảnh, sau đó histogram thống kê tần suất xuất hiện của từng mã LBP sẽ được xây dựng. Histogram này đóng vai trò là vector đặc trưng đầu ra, mang thông tin kết cấu cục bộ của khuôn mặt hoặc đối tượng.

**Công thức:**

Trong đó:

* gc ​: giá trị điểm ảnh trung tâm
* gp ​: giá trị điểm ảnh lân cận
* s(x)= Hàm bước, trả về 1 nếu x lớn hơn hoặc bằng 0, ngược lại 0.

LBP là một thuật toán đơn giản, tính toán nhanh và có khả năng chống nhiễu tốt, đặc biệt là trong điều kiện ánh sáng thay đổi. Nhờ tính cục bộ và không phụ thuộc vào cường độ tuyệt đối, LBP phù hợp cho các bài toán nhận diện trong môi trường thực tế, nơi ánh sáng không ổn định hoặc bị tác động bởi yếu tố ngoại cảnh.

Trong lĩnh vực nhận diện khuôn mặt, LBP thường được sử dụng như một bước trích xuất đặc trưng đầu vào cho các thuật toán phân loại. Ngoài ra, nó còn được ứng dụng trong phân tích kết cấu bề mặt, nhận dạng mô hình vật liệu, giám sát an ninh và nhận diện biểu cảm khuôn mặt – nơi mà các thay đổi nhỏ trên da, như nếp nhăn hoặc co giãn cơ mặt, cần được ghi nhận chi tiết.

## **4. Landmark**

Landmarks (các điểm đặc trưng trên khuôn mặt) là tập hợp những điểm có vị trí cố định, dùng để định danh các bộ phận quan trọng như mắt, mũi, miệng, cằm, viền hàm… Mỗi điểm landmark được biểu diễn bằng một cặp tọa độ (x,y) trên ảnh đầu vào. Một trong những cấu trúc phổ biến nhất là bộ **68 điểm landmark** do thư viện Dlib cung cấp, giúp mô tả hình dạng và vị trí các bộ phận trên khuôn mặt một cách chi tiết và có hệ thống.

Quy trình trích xuất landmarks thường bắt đầu bằng việc **xác định vị trí khuôn mặt** trong ảnh. Bước này có thể được thực hiện bằng nhiều phương pháp khác nhau như **HOG kết hợp SVM**, **mạng nơ-ron tích chập (CNN)** hoặc thuật toán cổ điển như **Haar Cascade**. Sau khi phát hiện được vùng khuôn mặt, một **mô hình học máy đã huấn luyện** sẽ được sử dụng để dự đoán vị trí chính xác của các điểm landmark. Kết quả là một vector đặc trưng gồm các tọa độ (x,y) cho từng điểm – ví dụ, với 68 điểm landmark, vector đầu ra sẽ có **136 phần tử** (68 điểm × 2 tọa độ).

Bộ landmarks 68 điểm có cách tổ chức rất trực quan, giúp phân tích rõ ràng từng vùng khuôn mặt. Cụ thể:

* **Điểm 1–17**: Viền hàm, giúp xác định hình dáng khuôn mặt.
* **Điểm 18–27**: Chân mày trái và phải.
* **Điểm 28–36**: Sống mũi và cánh mũi.
* **Điểm 37–48**: Vùng mắt trái và phải.
* **Điểm 49–68**: Miệng – bao gồm cả môi trên, môi dưới và vùng xung quanh.

Landmarks cung cấp thông tin hình học chính xác về khuôn mặt, có khả năng mô tả biểu cảm, vị trí, trạng thái các bộ phận như mắt và miệng. Nhờ đó, chúng rất hữu ích trong các bài toán phân tích biểu cảm khuôn mặt, ước lượng tư thế đầu, và cả các tác vụ cần căn chỉnh khuôn mặt để chuẩn hóa dữ liệu.

Một trong những ứng dụng phổ biến nhất của landmarks là **phân tích biểu cảm khuôn mặt**, bằng cách theo dõi sự thay đổi hình học của các vùng mắt, miệng và chân mày. Ngoài ra, landmarks còn giúp **phát hiện trạng thái mắt** (như mở hay nhắm), **phân tích khẩu hình miệng**, và làm cơ sở cho các bài toán chuyên sâu hơn như **face alignment** (căn chỉnh khuôn mặt), hoặc **head pose estimation** (ước lượng tư thế đầu) – cực kỳ quan trọng trong robot tương tác và các hệ thống AI nhận thức hành vi người dùng.

# **IV. Triển khai**

## **1. Cấu trúc dataset**

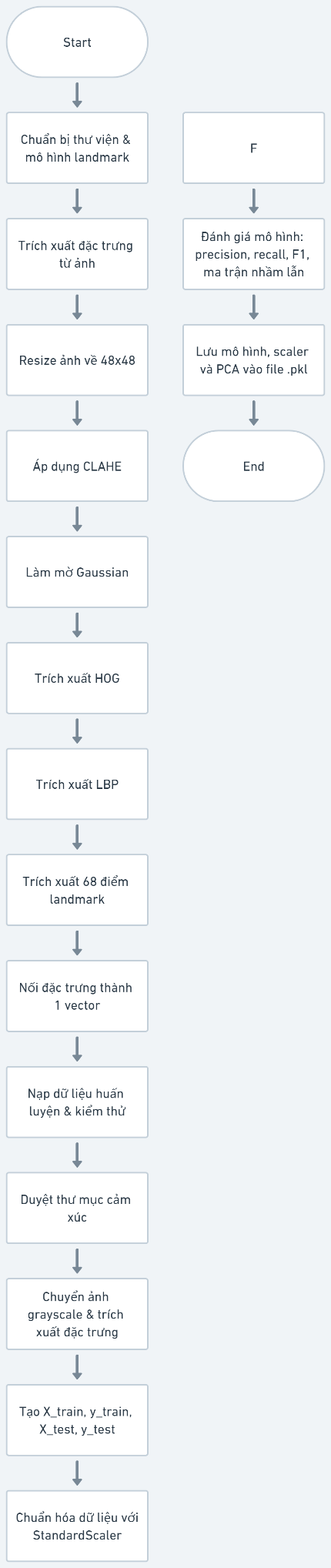
A diagram of a dataset

AI-generated content may be incorrect.

Hình 8: Cấu trúc dataset

Trong folder dataset được dùng để huấn luyện và đánh giá model có tổng cộng 35887 ảnh trong đó có 28709 ảnh được dùng cho quá trình huấn luyện SVM và 7178 ảnh dùng cho việc đánh giá. Dataset gồm 7 lớp: angry, disgust, sad, happy, fear, neutral và surprise tương đồng cho cả hai thư mục Train và Test, mỗi thư mục chỉ chứa các ảnh biểu thị cảm xúc theo tên của thư mục đó và tên của thư mục được xem là nhãn của các ảnh trong thư mục.

## **2. Lưu đồ thuật toán**



Hình 9: Lưu đồ giải thuật

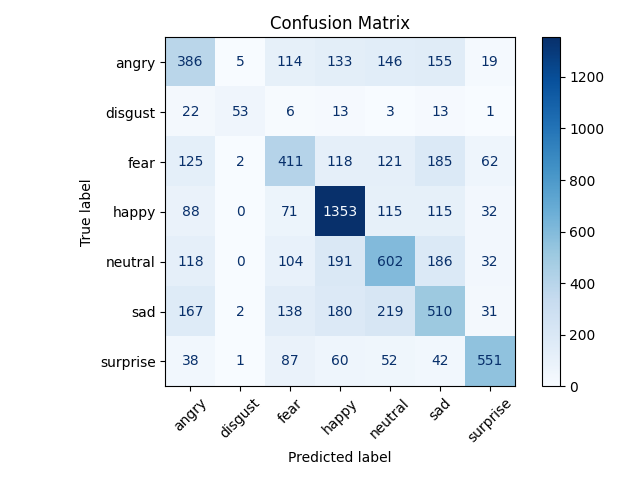
# **V. Kết quả và đánh giá**

## **1. Bảng kết quả huấn luyện**

Sau khi thực hiện quá trình huấn luyện và kiểm thử mô hình nhận diện cảm xúc, nhóm thu được các chỉ số đánh giá như sau: độ chính xác tổng thể (**Accuracy**) đạt **54%**; các chỉ số **Precision**, **Recall**, và **F1-score** được tính cho từng cảm xúc cụ thể. Trong đó, các cảm xúc như *happy* và *surprise* đạt kết quả cao nhất với F1-score khoảng **0.71**, trong khi các cảm xúc tiêu cực như *angry*, *fear*, và *sad* có F1-score dao động quanh mức **0.41–0.42**. Chi tiết kết quả được thể hiện trong bảng dưới đây.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** |
| **Angry** | 0.41 | 0.40 | 0.41 |
| **Disgust** | 0.84 | 0.48 | 0.61 |
| **Fear** | 0.44 | 0.40 | 0.42 |
| **Happy** | 0.66 | 0.76 | 0.71 |
| **Neutral** | 0.48 | 0.49 | 0.48 |
| **Sad** | 0.42 | 0.41 | 0.42 |
| **Surprise** | 0.76 | 0.66 | 0.71 |
| **Accuracy** | 0.54 | 0.54 | 0.54 |
| **Macro avg** | 0.57 | 0.51 | 0.54 |
| **Weighted avg** | 0.54 | 0.54 | 0.54 |

## **2. Confusion matrix**



Hình 10: Confusion matrix

Dựa trên ma trận nhầm lẫn, có thể nhận thấy mô hình phân loại cảm xúc hoạt động tốt nhất với các cảm xúc có biểu hiện khuôn mặt rõ ràng như **"happy"** và **"surprise"**, với số mẫu được dự đoán đúng lần lượt là **1353** và **551**, chiếm tỷ lệ cao so với tổng số mẫu thực tế. Điều này cho thấy mô hình học được các đặc trưng nổi bật của những biểu cảm tích cực như nụ cười hoặc ánh mắt ngạc nhiên.

Ngược lại, các cảm xúc tiêu cực như **"fear"**, **"sad"**, và **"angry"** có tỷ lệ nhận diện thấp, thường bị nhầm lẫn lẫn nhau. Ví dụ, cảm xúc *fear* có đến **185 mẫu bị dự đoán nhầm là "neutral"**, và *sad* có tới **167 mẫu bị nhầm là "angry"**. Điều này phản ánh sự tương đồng trong biểu hiện khuôn mặt giữa các trạng thái cảm xúc tiêu cực và cũng là một điểm yếu thường thấy trong các hệ thống nhận diện sử dụng đặc trưng cổ điển như HOG, LBP và Landmark.

Cảm xúc **"disgust"** là một trường hợp đặc biệt khi có rất ít mẫu đúng (53), nhưng lại có độ chính xác dự đoán (precision) khá cao, cho thấy mô hình chỉ dự đoán "disgust" khi thật sự chắc chắn – tuy nhiên do mất cân bằng dữ liệu, *recall* lại rất thấp.

## **3. Kết quả nhận diện**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 11: Kết quả và thời gian nhận diện

A screenshot of a computer

AI-generated content may be incorrect.

Hình 12: Kết quả và thời gian nhận diện

# **VI. Kết luận**

Qua quá trình triển khai hệ thống nhận diện cảm xúc người dùng trong robot dịch vụ, nhóm đã thực hiện huấn luyện mô hình phân loại dựa trên các đặc trưng khuôn mặt (HOG, LBP, Landmark) kết hợp với thuật toán SVM. Dựa vào kết quả đánh giá bằng bảng thông số và ma trận nhầm lẫn (Confusion Matrix), có thể đưa ra một số nhận xét và kết luận như sau:

1. **Hiệu suất tổng thể của hệ thống**:

Mô hình đạt **độ chính xác tổng thể (Accuracy) là 54%**, với điểm **F1-score trung bình cộng (Macro avg) là 0.54** và **trung bình có trọng số (Weighted avg) cũng là 0.54**. Đây là kết quả khá tốt đối với một hệ thống phân loại nhiều lớp (7 cảm xúc) trong điều kiện đặc trưng không quá phức tạp và chỉ sử dụng các kỹ thuật cổ điển như HOG, LBP và Landmark.

1. **Hiệu quả theo từng cảm xúc**:

Kết quả đánh giá cho thấy mô hình nhận diện cảm xúc hoạt động hiệu quả nhất đối với các nhóm cảm xúc có biểu hiện rõ ràng như "happy" (vui vẻ) và "surprise" (ngạc nhiên), với F1-score đạt khoảng 0.71. Những cảm xúc này thường đi kèm với các chuyển động cơ mặt đặc trưng và dễ nhận biết như nụ cười rộng, mắt mở to hay chân mày nâng cao, giúp mô hình dễ dàng phân biệt được trên hình ảnh đầu vào.

Ngược lại, các cảm xúc như "fear" (sợ hãi), "sad" (buồn) và "angry" (tức giận) có F1-score thấp hơn, dao động trong khoảng 0.41–0.42. Đây là nhóm cảm xúc tiêu cực có những biểu hiện khuôn mặt khá tương đồng, khiến mô hình gặp khó khăn trong việc phân biệt rõ ràng. Các điểm đặc trưng như mắt co lại, trán nhăn hay miệng mím chặt có thể xuất hiện ở nhiều cảm xúc cùng lúc, làm giảm độ chính xác của mô hình.

Đáng chú ý, cảm xúc "disgust" (ghê tởm) là nhóm có số lượng mẫu ít trong tập dữ liệu, dẫn đến sự mất cân bằng trong huấn luyện. Điều này thể hiện qua Recall thấp (0.48) – mô hình bỏ sót nhiều trường hợp, trong khi Precision lại khá cao (0.84) – khi mô hình dự đoán là "disgust", thì thường đúng. Điều này cho thấy mô hình có xu hướng “chắc ăn” khi phân loại cảm xúc này, nhưng không tự tin để đưa ra nhiều dự đoán, từ đó làm giảm khả năng bao phủ toàn bộ các trường hợp thực tế.

1. **Phân tích từ ma trận nhầm lẫn**:

Khi phân tích sâu vào ma trận nhầm lẫn, có thể nhận thấy rằng mô hình gặp khó khăn trong việc phân biệt giữa các cảm xúc tiêu cực như "angry", "fear", và "sad". Những cảm xúc này thường bị nhầm lẫn lẫn nhau do đặc điểm khuôn mặt tương đồng như lông mày cau lại, ánh mắt căng thẳng hoặc biểu cảm môi không rõ rệt. Ví dụ, cảm xúc "fear" thường bị mô hình đoán nhầm thành "neutral" đến 185 lần, hoặc thành "sad" khoảng 62 lần, cho thấy mô hình chưa đủ nhạy để phân biệt được những biểu cảm có sắc thái gần nhau.

Ở chiều ngược lại, cảm xúc "happy" được mô hình nhận diện rất tốt, với 1353 mẫu được phân loại chính xác. Điều này cho thấy mô hình đã học được đặc trưng rõ rệt của biểu cảm nụ cười – yếu tố thường rất nổi bật trên khuôn mặt và ít bị nhầm lẫn với các cảm xúc khác.

Ngoài ra, các cảm xúc như "neutral" và "surprise" cũng cho thấy hiệu quả phân loại khá cao, với lần lượt 602 và 551 mẫu đúng. Đối với "neutral", sự ổn định trong biểu cảm giúp mô hình dễ nhận diện hơn. Trong khi đó, "surprise" có những đặc trưng đặc biệt như mắt mở to, miệng há rộng – giúp mô hình phân biệt tốt với các cảm xúc khác.

1. **Đánh giá tổng thể**:

Hệ thống nhận diện cảm xúc được xây dựng cho thấy tiềm năng ứng dụng thực tế cao trong các robot dịch vụ, đặc biệt nhờ khả năng nhận diện chính xác những cảm xúc dễ nhận biết như vui (happy) và ngạc nhiên (surprise) – hai trạng thái thường xuyên xuất hiện trong các tình huống tương tác giữa người dùng và robot. Những cảm xúc này không chỉ phổ biến mà còn có biểu hiện khuôn mặt rõ ràng, giúp hệ thống dễ dàng phản hồi phù hợp, góp phần tạo nên trải nghiệm thân thiện và tự nhiên hơn cho người sử dụng.

Tuy nhiên, để mở rộng phạm vi và tăng độ chính xác đối với các cảm xúc khó phân biệt như sợ hãi (fear), buồn (sad) và tức giận (angry) – vốn có biểu hiện khuôn mặt tương đối giống nhau – hệ thống cần tiếp tục được cải thiện. Trước hết, việc tăng cường và cân bằng dữ liệu huấn luyện là rất cần thiết, giúp mô hình tránh thiên lệch và học tốt hơn các mẫu ít gặp. Bên cạnh đó, áp dụng các mô hình học sâu (deep learning) như mạng nơ-ron tích chập (CNN) sẽ giúp hệ thống tự động trích xuất đặc trưng tốt hơn, vượt qua giới hạn của các phương pháp thủ công như HOG hay LBP.

Cuối cùng, hệ thống có thể đạt độ chính xác và tính linh hoạt cao hơn nếu kết hợp thêm thông tin đa kênh như chuyển động khuôn mặt từ video hoặc giọng nói, nhằm hiểu rõ hơn trạng thái cảm xúc theo ngữ cảnh. Đây là hướng phát triển tiềm năng để đưa hệ thống từ mức độ nguyên mẫu lên thành giải pháp tích hợp hoàn chỉnh trong robot dịch vụ tương tác thông minh.

# **VII. Tài liệu tham khảo**

1. Pramod, O. (n.d.). *Support Vector Machine (SVM)*. Medium. Truy cập tại: <https://medium.com/@ompramod9921/support-vector-machine-svm-68bf201b4aa2>
2. TPoint Tech. (n.d.). *Machine Learning – Support Vector Machine Algorithm*. Truy cập tại: <https://www.tpointtech.com/machine-learning-support-vector-machine-algorithm>
3. YouTube – *SVM - Support Vector Machine Algorithm Explained*. Truy cập tại: <https://www.youtube.com/watch?v=RnapcMi9lqA>
4. YouTube – *Emotion Recognition using Machine Learning*. Truy cập tại: <https://www.youtube.com/watch?v=QdmUKu7LqtI>
5. YouTube – *Facial Emotion Recognition using Python and OpenCV*. Truy cập tại: <https://www.youtube.com/watch?v=hribPP5ccdc&t=538s>