

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 1**

# **TÌM HIỂU ỨNG DỤNG NHẬN DẠNG BIỂN BÁO GIAO THÔNG**

*Người hướng dẫn:* **Cô VÕ HOÀNG ANH**

*Người thực hiện:* **HỒNG QUANG VINH - 51303447**

**NGUYỄN TRỌNG CÔNG - 51303246**

**Lớp : 13050303**

**Khoá : 17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2016**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 1**

# **TÌM HIỂU ỨNG DỤNG NHẬN DẠNG BIỂN BÁO GIAO THÔNG**

*Người hướng dẫn:* **Cô VÕ HOÀNG ANH**

*Người thực hiện:* **HÔNG QUANG VINH - 51303447**

**NGUYỄN TRỌNG CÔNG - 51303246**

**Lớp : 13050303**

**Khoá : 17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2016**

## **LỜI CẢM ƠN**

Sau đây nhóm xin chân thành cảm ơn sự hướng dẫn cô Võ Hoàng Anh trong quá trình hướng dẫn làm đồ án. Những kiến thức mà cô giảng dạy đã góp phần cung cấp những kiến thức cần thiết cho chúng em vào quá trình làm đồ án. Xin chân thành cảm ơn cô.

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của cô Võ Hoàng Anh;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Hồng Quang Vinh*

*Nguyễn Trọng Công*

## PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

### Phần xác nhận của GV hướng dẫn

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(kí và ghi họ tên)

### Phần đánh giá của GV chấm bài

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(kí và ghi họ tên)

## TÓM TẮT

Nhận diện đối tượng (object recognition) là một bài toán quan trọng trong lĩnh vực thị giác máy tính. Đây là một hướng nghiên cứu có nhiều ứng dụng trong thực tế như giao thông, thể thao, giao tiếp người máy, an ninh,.. Đồ án này tập trung nghiên cứu các vấn đề về nhận diện đối tượng và chủ yếu vào lĩnh vực nhận diện biển báo giao thông. Đồ án này được thực hiện với các mục đích:

- Tìm hiểu các phương pháp nhận diện đối tượng khác nhau.
- Ứng dụng các thuật toán nhận diện đối tượng vào thực tiễn, đặc biệt là lĩnh vực nhận diện biển báo giao thông.

Các nội dung chính được trình bày trong đồ án gồm:

- Sử dụng rút trích đặc trưng bằng giải thuật HOG (Histogram of Gradient).
- Sử dụng bộ lọc Gabor Filter.
- Sử dụng rút trích đặc trưng bằng giải thuật LBP (Local Binary Pattern).
- Sau đó sử dụng SVM (Support Vector Machine) để phân lớp hình ảnh đó thuộc loại biển báo nào.

Từ khóa: *nhận diện đối tượng (object recogniton), HOG, LBP, Gabor, SVM.*

## MỤC LỤC

LỜI CẢM ƠN .....	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....	iii
TÓM TẮT .....	iv
MỤC LỤC.....	1
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ .....	4
CHƯƠNG 1 – TỔNG QUAN ĐỀ TÀI.....	7
1.1 Giới thiệu đề tài – Bài toán nhận diện đối tượng.....	7
1.2 Mục tiêu đề tài – Bài toán nhận diện biển báo giao thông.....	7
1.3 Phạm vi đề tài.....	9
1.3.1 Giới thiệu bài toán nhận diện biển báo nguy hiểm trong giao thông đường bộ .....	9
1.3.2 Cơ sở dữ liệu sử dụng trong bài toán.....	9
1.4 Bố cục.....	11
CHƯƠNG 2 – CÔNG TRÌNH LIÊN QUAN .....	12
2.1 Giới thiệu .....	12
2.2 Các kỹ thuật được sử dụng.....	12
2.3 Các thông số chính.....	12
CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT .....	14
3.1 Histogram Of Gradient (HOG): .....	14
3.1.1 Giới thiệu thuật toán HOG (Histogram of Oriented Gradients): ....	14
3.1.2 Rút trích đặc trưng .....	15
3.1.3 Chuẩn hoá vector đặc trưng cho từng block .....	19
3.2 Local Binary Pattern (LBP) .....	20
3.2.1 Giới thiệu .....	20
3.2.2 Các biến thể của LBP.....	21
3.2.2.1 LBP đồng dạng .....	21

3.2.2.2 LBP bất biến với phép quay .....	21
3.2.2.3 LBP đồng dạng có khả năng bất biến với phép quay .....	21
3.3 Gabor Filter .....	22
3.4 Support Vector Machine (SVM) .....	23
3.4.1 Giới thiệu .....	23
3.4.2 Các khái niệm cơ bản .....	23
3.4.2.1 Siêu phẳng phân cách .....	23
3.4.2.2 Support Vector .....	27
3.4.2.3 Biên độ Margin .....	28
3.4.3 Phân lớp dữ liệu .....	29
3.4.3.1 Trường hợp dữ liệu có thể phân chia tuyến tính được .....	29
3.4.3.2 Trường hợp dữ liệu không thể chia tuyến tính được .....	30
CHƯƠNG 4 – Ý TƯỞNG GIẢI QUYẾT BÀI TOÁN .....	32
4.1 Mô hình thực hiện .....	32
4.2 Mô tả mô hình .....	32
4.3 Phương pháp đánh giá .....	33
CHƯƠNG 5 – KẾT QUẢ THỰC NGHIỆM .....	35
5.1 Bộ dữ liệu the German Traffic Sign Detection Benchmark (GTSDB) .....	35
5.1.1 HOG .....	35
5.2 Bộ dữ liệu biển báo giao thông Việt Nam .....	35
5.2.1 HOG .....	35
5.2.2 LBP .....	36
5.2.3 LBP (Overlap) .....	36
5.2.4 Gabor moment .....	37
5.2.5 Gabor .....	37
5.2.6 LBP và Gabor .....	38
5.2.7 HOG và Gabor .....	38



5.2.8 HOG, LBP và Gabor.....	39
5.3 So sánh giữa các giải thuật.....	39
CHƯƠNG 6 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	40
6.1 Kết luận .....	40
6.2 Hướng phát triển .....	41
TÀI LIỆU THAM KHẢO.....	42

## DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

### DANH MỤC HÌNH

Hình 1: Ảnh dữ liệu có backgroud phức tạp .....	8
Hình 2: Ảnh dữ liệu bất lợi về ánh sáng .....	8
Hình 3: Ảnh dữ liệu bị che khuất bởi vật thể khác .....	8
Hình 4: Ảnh dữ liệu cùng một nội dung nhưng ảnh khác .....	8
Hình 5: Ảnh dữ liệu cùng một nội dung nhưng ảnh khác .....	9
Hình 6: Hình ảnh biển báo giao thông được chụp trong môi trường cùng với các đối tượng khác nhau .....	10
Hình 7: Hình biển báo giao thông đã được cắt sát với biển báo .....	10
Hình 8: Hình tỷ lệ % đúng khi sử dụng các kỹ thuật rút trích đặc trưng khác nhau .....	12
Hình 9: Bảng thời gian rút trích đặc trưng .....	13
Hình 10: R-HOG và C-HOG .....	14
Hình 11: Hình qui trình rút trích đặc trưng .....	15
Hình 12: Mỗi khối (block) gồm nhiều ô (cell) .....	15
Hình 13: Các khối được xếp chồng lên nhau .....	16
Hình 14: Tính góc và biên độ theo X-gradient và Y-gradient .....	17
Hình 15: Các bước rút trích đặc trưng HOG .....	18
Hình 16: Hình minh hoạ về trường hợp của LBP sau khi quay với góc 15 độ .....	21
Hình 17: Hình ví dụ về ngân hàng bộ lọc Gabor .....	22
Hình 18: Hình biểu diễn phần thực và ảo của tập hợp hàm Gabor .....	22
Hình 19: Đường màu vàng trên hình là đường phân chia đối với tập dữ liệu gồm hai thuộc tính .....	24
Hình 20: Một bộ dữ liệu hai chiều được phân chia tuyến tính .....	25
Hình 21: 2 siêu phẳng phân chia tuyến tính cùng với biên độ của nó .....	26
Hình 22: Đường biểu diễn H1 và H2, đường màu đỏ là khoảng cách Euclidean của hai điểm 1 và 2. Đường màu xanh là khoảng cách Euclidean nhỏ nhất .....	27

Hình 23: Các support vector trong SVM. Các support vector là những hình có viền dày hơn.....	28
Hình 24: Một trường hợp đơn giản trên không gian 2 chiều .....	30

## **DANH MỤC BẢNG**

Bảng 1: Bảng thông tin về các bộ cơ sở dữ liệu sử dụng trong bài toán .....	9
Bảng 2: Bảng tỷ lệ % đúng, sai của giải thuật HOG.....	35
Bảng 3: Bảng tỷ lệ % đúng,sai của giải thuật HOG.....	35
Bảng 4: Bảng tỷ lệ % đúng,sai của giải thuật LBP .....	36
Bảng 5: Bảng tỷ lệ % đúng,sai của giải thuật LBP (Overlap) .....	36
Bảng 6: Bảng tỷ lệ % đúng,sai của giải thuật Gabor moment .....	37
Bảng 7: Bảng tỷ lệ % đúng,sai của giải thuật Gabor .....	37
Bảng 8: Bảng tỷ lệ % đúng,sai của giải thuật LBP và Gabor .....	38
Bảng 9: Bảng tỷ lệ % đúng,sai của giải thuật HOG và Gabor.....	38
Bảng 10: Bảng tỷ lệ % đúng,sai của giải thuật HOG, LBP và Gabor .....	39

## **CHƯƠNG 1 – TỔNG QUAN ĐỀ TÀI**

### **1.1 Giới thiệu đề tài – Bài toán nhận diện đối tượng**

Nhận diện đối tượng trong thị giác máy tính là công việc tìm kiếm các vật thể trong ảnh. Đây là hướng nghiên cứu được quan tâm nhiều trong lĩnh vực công nghệ hiện nay với rất nhiều ứng dụng. Con người có thể nhận biết được các đối tượng từ ảnh một cách rất dễ dàng mặc dù sự khác nhau giữa chúng là vô cùng đa dạng. Tuy nhiên, việc này đối với hệ thống thị giác máy tính vẫn là một thách thức rất lớn. Nguyên nhân chủ yếu là do sự đa dạng về hình dáng, màu sắc của vật thể, đa dạng về diện mạo và tư thế của mỗi loại vật thể, ảnh hưởng về điều kiện ánh sáng, sự che lấp lẫn nhau giữa các vật thể, chất lượng ảnh, .v.v.

Ý tưởng chính trong các phương pháp nhận diện đối tượng là dựa vào đặc trưng cụ thể được rút trích ra từ những đối tượng mẫu. Đặc trưng này được sử dụng cùng với bộ phân lớp hoặc sử dụng thuật toán để nhận ra những đối tượng tương tự với các đối tượng mẫu.

Nhận diện biển báo nguy hiểm trong giao thông đường bộ là một nhánh của nhận diện đối tượng.

### **1.2 Mục tiêu đề tài – Bài toán nhận diện biển báo giao thông**

Nhận diện biển báo nguy hiểm trong ảnh là bài toán có nhiều ứng dụng trong thực tế. Việc phát hiện tốt, chính xác những biển báo nguy hiểm có mặt trong ảnh sẽ tạo tiền đề tốt cho những hướng phát triển như:

- Phục vụ quá trình tham gia giao thông diễn ra đúng với luật giao thông đường bộ.
- Hướng dẫn người tham gia giao thông các hướng di chuyển, xử lý tình huống tiếp theo nếu gặp 1 biển báo nguy hiểm.

Những khó khăn phải đối mặt trong đề tài là:

- Các ảnh thường không đạt chuẩn, background phức tạp.



Hình 1: Ảnh dữ liệu có background phức tạp

- Ảnh hưởng bất lợi về điều kiện ánh sáng.



Hình 2: Ảnh dữ liệu bất lợi về ánh sáng

- Sự che khuất lẫn nhau vật thể và biển báo trong ảnh.



Hình 3: Ảnh dữ liệu bị che khuất bởi vật thể khác

- Sự đa dạng về loại, kiểu dáng, nội dung biển báo.



Hình 4: Ảnh dữ liệu cùng một nội dung nhưng ảnh khác



Hình 5: Ảnh dữ liệu cùng một nội dung nhưng ảnh khác

### 1.3 Phạm vi đề tài

#### 1.3.1 Giới thiệu bài toán nhận diện biển báo nguy hiểm trong giao thông đường bộ

Với một số hình ảnh đầu vào về biển báo nguy hiểm, yêu cầu tính tỉ lệ chính xác khi nhận diện biển báo – có nghĩa là tính tỉ lệ chính xác khi phân các hình ảnh này vào các loại biển báo ương ứng.

#### 1.3.2 Cơ sở dữ liệu sử dụng trong bài toán

Để kiểm tra, đánh giá và tăng độ chính xác với kết quả bài toán, đồ án sử dụng 2 bộ database, mỗi bộ được chụp tại mỗi quốc gia khác nhau với điều kiện ánh sáng, không gian và biển báo khác nhau.

- The German Traffic Sign Detection Benchmark (GTSDB): Đây là bộ ảnh được chụp tại Đức.
- Bộ ảnh được sưu tập (VN): Đây là bộ ảnh được chụp tại Việt Nam.

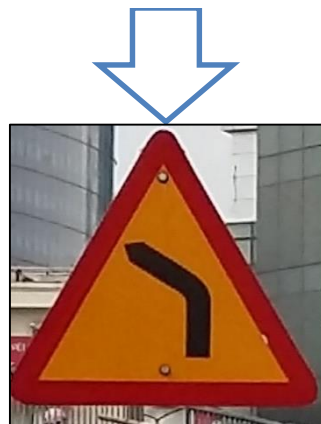
	<b>GTSDB</b>	<b>VN</b>
<b>Số lớp (loại biển báo)</b>	43	22
<b>Số ảnh train</b>	852	896
<b>Số ảnh test</b>	361	383
<b>Tổng ảnh</b>	1213	1279

Bảng 1: Bảng thông tin về các bộ cơ sở dữ liệu sử dụng trong bài toán

Trong phạm vi đề tài này tập trung vào phần nhận diện biển báo nguy hiểm (tức phân các ảnh vào các loại biển báo tương ứng) vì vậy sẽ bỏ qua phần phát hiện biển báo trong ảnh. Do đó tập ảnh train và test sẽ là những ảnh đã được xử lý phần nền, chỉ giữ lại phần biển báo.



Hình 6: Hình ảnh biển báo giao thông được chụp trong môi trường cùng với các đối tượng khác nhau



Hình 7: Hình biển báo giao thông đã được cắt sát với biển báo



## 1.4 Bố cục

Nội dung của đề án này được trình bày trong 6 chương, bao gồm các nội dung về các phương pháp và vấn đề về nhận diện đối tượng. Cùng với việc áp dụng các phương pháp này vào vấn đề nhận diện biển báo giao thông.

**Chương 1 – Tổng quan về đề tài:** Giới thiệu tổng quan về đề tài, bài toán, mục tiêu và phạm vi đề tài.

**Chương 2 – Công trình liên quan:** Giới thiệu về bài báo, công trình liên quan đến đề tài đang thực hiện.

**Chương 3 – Cơ sở lý thuyết:** Tìm hiểu về các kỹ thuật rút trích đặc trưng như Histogram Of Gradient (HOG), Local Binary Pattern (LBP), Gabor Filter – và bộ phân lớp Support Vector Machine (SVM).

**Chương 4 – Ý tưởng giải quyết bài toán:** Mô tả ý tưởng, các bước áp dụng cơ sở lý thuyết để giải quyết bài toán.

**Chương 5 – Kết quả thực nghiệm:** Kết quả đạt được khi giải quyết xong bài toán.

**Chương 6 – Kết luận và hướng phát triển:** Rút ra các kết luận và xác định hướng phát triển tiếp theo cho bài toán.

## CHƯƠNG 2 – CÔNG TRÌNH LIÊN QUAN

### 2.1 Giới thiệu

Đồ án tham khảo bài báo khoa học “On circular traffic sign detection and recognition” của Selcan Kaplan Berkaya công bố vào năm 2015, database sử dụng là The German Traffic Sign Detection Benchmark.

Mục tiêu bài báo là sử dụng các kỹ thuật để phát hiện ra các biển báo trong ảnh, từ đó sử dụng tiếp các kỹ thuật rút trích đặc trưng để nhận diện biển báo đó.

### 2.2 Các kỹ thuật được sử dụng

Tham khảo chính ở phần nhận diện, vì vậy sẽ bỏ qua phần phát hiện biển báo.

Các kỹ thuật được sử dụng trong quá trình nhận diện là Histogram Of Gradient (HOG), Local Binary Pattern (LBP), Gabor Filter – chi tiết các kỹ thuật này sẽ được trình bày ở chương tiếp theo.

### 2.3 Các thông số chính

Tỉ lệ % khi sử dụng các kỹ thuật rút trích đặc trưng để nhận diện biển báo:

Features	All Signs	Speed Limits	Other Prohibitions	Derestriction	Mandatory	Danger	Unique Signs
LBP	93.36	94.17	98.06	91.94	98.92	85.66	99.85
GABOR	93.90	95.17	98.86	98.33	95.98	91.07	99.26
HOG	94.56	94.58	99.86	90.27	99.32	88.20	100.00
HOG + LBP	95.24	95.34	99.86	91.94	99.43	89.31	100.00
HOG + GABOR	97.00	96.49	99.80	98.88	99.83	94.58	99.95
GABOR + LBP	95.17	95.80	99.46	98.61	98.19	91.36	99.65
GABOR + LBP + HOG	97.04	96.88	99.86	97.50	99.83	94.15	99.95

Hình 8: Hình tỷ lệ % đúng khi sử dụng các kỹ thuật rút trích đặc trưng khác nhau

Thời gian trung bình khi sử dụng các kỹ thuật rút trích trên:

Features	Time (ms)
LBP	1.21
GABOR	2.31
HOG	0.02
HOG + LBP	1.24
HOG + GABOR	2.40
GABOR + LBP	3.49
GABOR + LBP + HOG	3.60

Hình 9: Bảng thời gian rút trích đặc trưng

## CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT

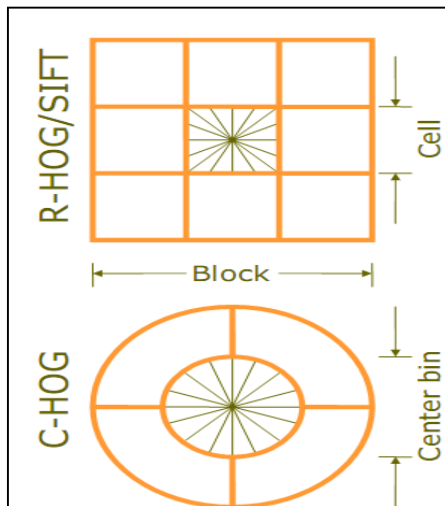
### 3.1 Histogram Of Gradient (HOG):

#### 3.1.1 Giới thiệu thuật toán HOG (Histogram of Oriented Gradients):

Histogram of Oriented Gradients (HOG) là đặc trưng được dùng nhiều trong lĩnh vực phát hiện và nhận diện đối tượng. Kỹ thuật này được đề xuất bởi Bill Triggs và Navel Dalal vào năm 2005 tại viện nghiên cứu INRIA.

Ý tưởng chính trong đặt trưng HOG là hình dạng và trạng thái của đối tượng có thể được đặc trưng bằng sự phân bố về gradient và hướng của cạnh. Đặc trưng này được phát triển dựa trên SIFT, đặc trưng HOG được tính trên cả một vùng. Do sự biến thiên màu sắc trong các vùng là khác nhau, kết quả là mỗi vùng sẽ cho ta một vector đặc trưng của nó, vì vậy để có được đặc trưng của toàn bộ cửa sổ (window) ta phải kết hợp nhiều vùng liên tiếp lại với nhau.

Đặc trưng HOG có một số biến thể thường gặp như: R-HOG, R2-HOG, C-HOG. Các đặc trưng này khác nhau ở cách phân bố và hình dạng của các ô như trong hình.



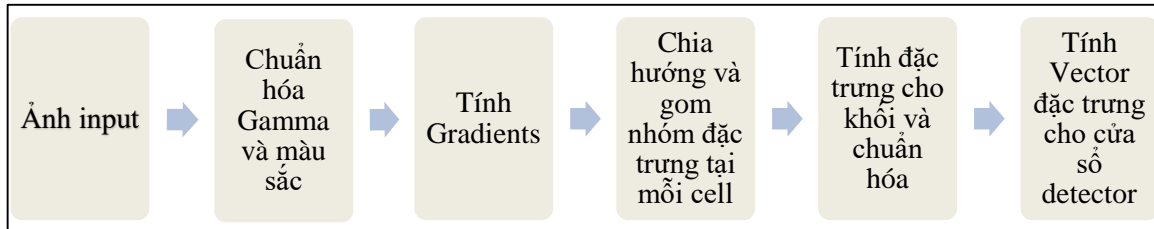
Hình 10: R-HOG và C-HOG

R-HOG: mỗi ô có hình dạng là hình chữ nhật.

C-HOG: mỗi ô có dạng hình tròn.

### 3.1.2 Rút trích đặc trưng

Quy trình rút trích đặc trưng:

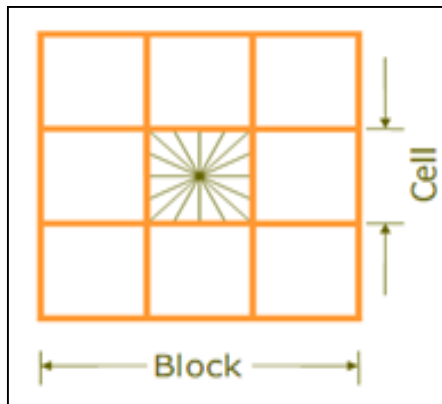


Hình 11: Hình qui trình rút trích đặc trưng

Từ một ảnh cho trước, qua việc rút trích đặc trưng trên ảnh đó cho ta được một vector đại diện cho đối tượng đó. Việc rút ra được các vector đặc trưng như vậy giúp chuyển không gian ảnh sang không gian vector để có thể xử lý và tính toán đơn giản hơn trên máy tính.

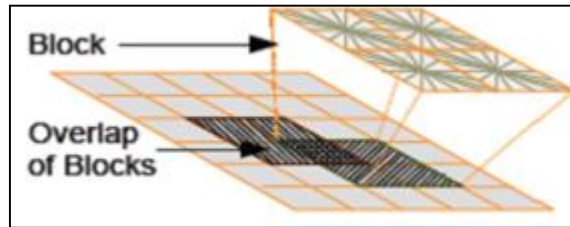
Phương pháp tổng quan để tính toán đặc trưng HOG trên một cửa sổ (window) bất kỳ:

- Phân chia cửa sổ (window) cần tính HOG thành nhiều khối (block). Mỗi khối gồm nhiều ô (cell). Các ô phải có kích thước bằng nhau, số lượng ô trong mỗi khối là bằng nhau.
- Các khối này được đặt chồng lên nhau, khoảng cách giữa hai khối con liên tiếp nhau phải là một hằng số.



Hình 12: Mỗi khối (block) gồm nhiều ô (cell).

Ví dụ trong hình trên ta thấy một khối gồm 9 ô



Hình 13: Các khối được xếp chồng lên nhau

- Sau khi đã phân chia cửa sổ cần tính toán HOG như trên, ta tiến hành rút trích đặc trưng như sau:
  - Tính toán đặc trưng trên từng vùng:
    - Tính toán đặc trưng trên từng ô nhỏ (cell)
    - Tính toán đặc trưng trên từng khối (block) bằng cách kết hợp các ô lại với nhau
  - Thu thập đặc trưng của các vùng trên ảnh
    - Tính và chuẩn hóa vector đặc trưng cho từng block
    - Thu thập các đặc trưng HOG cho các cửa sổ

Sau đây là chi tiết thuật toán rút trích đặc trưng HOG. Để minh họa thuật toán, ta có thể chọn một ví dụ như sau để tính vector đặc trưng HOG và số lượng thành phần của vector đó: giả sử ta có một ảnh với kích thước là 64x128 pixels. Ta chia ảnh này thành các cell với kích thước 8x8 pixels. Và ta chọn kích thước của block là 2x2 cell.

**Bước 1:** Chuyển ảnh trong không gian RGB sang ảnh dạng GRAY SCALE, sau đó tiến hành cân bằng histogram trên ảnh GRAY SCALE để giảm đi ảnh hưởng của sự thay đổi ánh sáng.

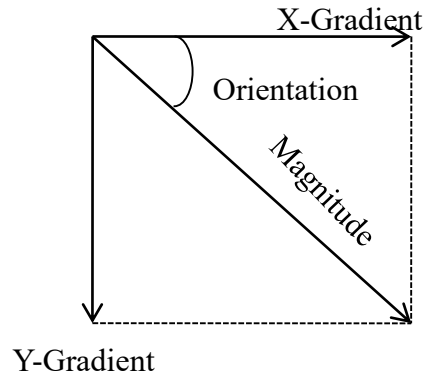
**Bước 2:** Tính sự biến thiên màu sắc tại tất cả các pixel của ảnh GRAY SCALE theo chiều X  $[-1 \ 0 \ 1]$  và theo chiều Y  $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$  và thu được 2 ảnh gradient-x và gradient-y

có kích thước bằng kích thước ảnh GRAY SCALE và cho thấy được sự biến thiên màu sắc nói trên.

**Bước 3:** tiến hành tính góc và hướng biến thiên màu sắc từ 2 ảnh gradient-x và gradient-y như hình vẽ sau :

$$Magnitude = \sqrt{I_x^2 + I_y^2}$$

$$Orientation = \arctan\left(\frac{I_x}{I_y}\right)$$



Hình 14: Tính góc và biên độ theo X-gradient và Y-gradient

Việc lưu trữ chính xác từng giá trị góc (orientation) của từng vị trí  $(x,y)$  không mang lại nhiều kết quả, tốn nhiều chi phí do vậy ta sẽ chia không gian góc ra thành các bin. Việc phân chia bin càng nhỏ sẽ càng làm tăng độ chính xác, thực nghiệm cho thấy kích thước bin khoảng  $20^\circ$  cho kết quả tốt nhất. Do đó từ 0 - 180 ta chia thành 9 bin như sau:

0 - 20, 21 - 40, 41 - 60, 61 - 80, 81 - 100, 101 - 120, 121 - 140, 141 - 160, 161 - 180.

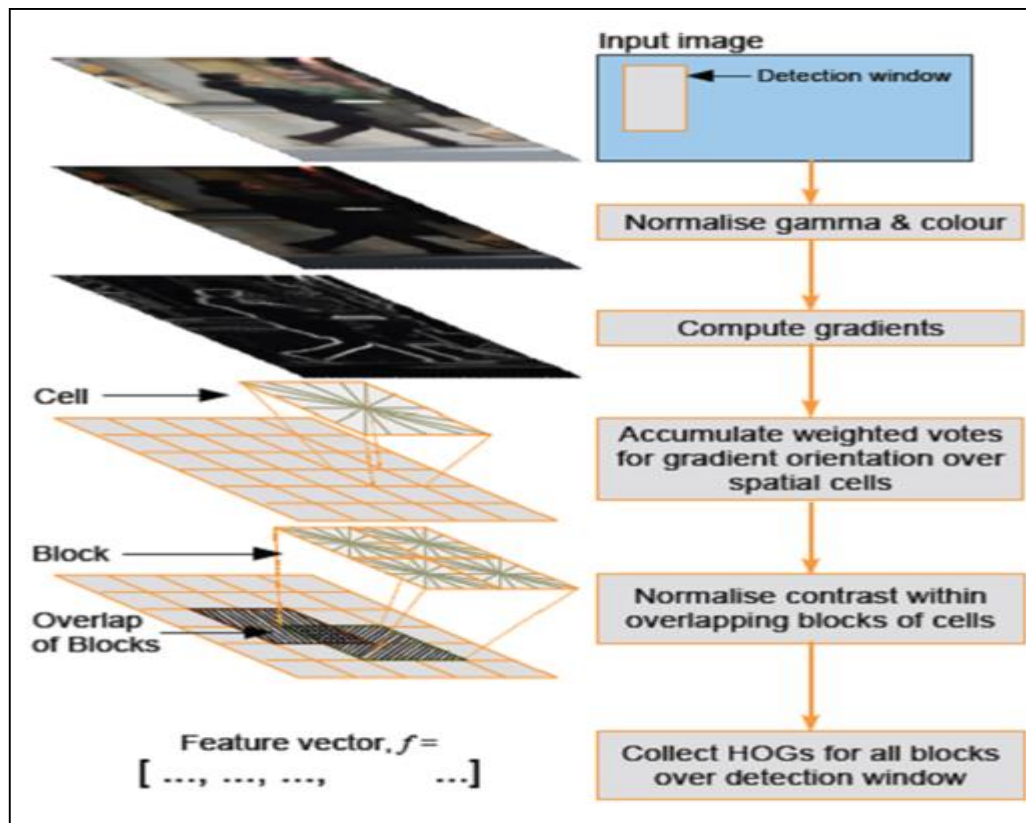
**Bước 4:** Ứng với mỗi bin trên ta tiến hành thống kê biên độ (magnitude) tại từng vị trí. Với mỗi bin, tại vị trí  $(x,y)$  nếu như góc (orientation) thuộc về bin đó thì giá trị của bin đó tại vị trí  $(x,y)$  bằng giá trị biên độ (magnitude), ngược lại giá trị bin tại vị trí  $(x,y)$  bằng 0.

**Bước 5:** Tính toán vector đặc trưng cho từng cell  $(8 \times 8)$ . Vector đặc trưng của mỗi cell sẽ gồm 9 thành phần tương ứng với 9 bin, và giá trị tại thành phần  $i$  bằng tổng giá trị của các điểm trong bin  $i$  mà có tọa độ nằm trong cell đó.

**Bước 6:** Tính toán vector đặc trưng cho từng block (16x16). Ta chỉ cần ghép các vector đặc trưng của từng cell trong block lại với nhau để được vector đặc trưng của một block và chuẩn hóa lại vector đặc trưng của block này. Việc chuẩn hóa đặc trưng trong block sẽ được bàn chi tiết ở phần sau. Như vậy vector đặc trưng của block sẽ gồm  $9 \times 4 = 36$  thành phần.

**Bước 7:** Tính toán vector đặc trưng cho toàn bộ ảnh (64x128). Ta chỉ cần ghép các vector đặc trưng của từng block lại với nhau để được vector đặc trưng của cả cửa sổ (window). Lưu ý rằng các block không đặt tách biệt nhau mà gối lên nhau, cách nhau một khoảng bằng kích thước của cell theo mỗi chiều (hoặc bằng phân nửa kích thước của cell). Như vậy vector đặc trưng của cửa sổ (window) = số block trong window \* 36 =  $((64 - 16)/8 + 1) * ((128 - 16)/8 + 1) * 36 = 7 * 15 * 36 = 3780$  thành phần.

**Bước 8:** Sau cùng ta chuẩn hóa lại vector đặc trưng của cả cửa sổ (window).



Hình 15: Các bước rút trích đặc trưng HOG



### 3.1.3 Chuẩn hoá vector đặc trưng cho từng block

Để chuẩn hóa vector đặc trưng cho cả cửa sổ (window), ta chuẩn hóa đặc trưng cho từng vector của từng block. Ta sử dụng các phương pháp sau để chuẩn hóa vector đặc trưng của từng block.

- L2-norm:

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (1)$$

- L1-norm:

$$f = \frac{v}{(\|v\|_1 + e)} \quad (2)$$

- L1-sqrt:

$$f = \sqrt{\frac{v}{(\|v\|_1 + e)}} \quad (3)$$

Trong đó:

- $v$ : vector đặc trưng ban đầu của một block (chưa chuẩn hóa).
- $\|v_k\|$ :  $k$ -norm của  $v$  với  $k = 1, 2$
- $e$ : Hằng số nhỏ

Theo Dalal và Triggs, L2-norm và L1-sqrt cho cùng kết quả. Trong khi đó L1-norm cho kết quả kém hơn. Tuy vậy việc chuẩn hóa bằng một trong các phương pháp trên vẫn cho kết quả tốt hơn đối với những vector không chuẩn hóa.

## 3.2 Local Binary Pattern (LBP)

### 3.2.1 Giới thiệu

LBP là viết tắt của Local Binary Pattern hay là mẫu nhị phân địa phương được Ojala trình bày vào năm 1996 như là một đơn vị đo độ tương phản cục bộ của ảnh.

### 3.2.2 LBP cơ bản

Thông tin LBP của pixel trung tâm của mỗi khối ảnh sẽ được tính dựa trên thông tin của các pixel lân cận. Tiến hành qua các bước như sau:

- Bước 1: Xác định bán kính.
- Bước 2: Tính giá trị LBP cho pixel ở trung tâm  $(x_c, y_c)$  khối ảnh dựa trên thông tin của các pixel lân cận:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=1}^P s(g_p - g_c) 2^p.$$

Trong đó,  $(g_p)$  là giá trị grayscale của các pixel lân cận,  $(g_c)$  là giá trị grayscale của các trung tâm và  $(s)$  là hàm nhị phân được xác định như sau:  $s(z) = 1$  nếu giá trị  $z \geq 0$ .

Ví dụ:

47	51	65
62	70	70
80	83	78

-23	-19	-5
-8	*	0
10	13	8

0	0	1
0	*	1
1	1	1

$$LBP_{P,R}(x_c, y_c) = 1 * 2^0 + 1 * 2^1 + 1 * 2^2 + 1 * 2^3 + 0 * 2^4 + 0 * 2^5 + 0 * 2^6 + 0 * 2^7 = 15.$$

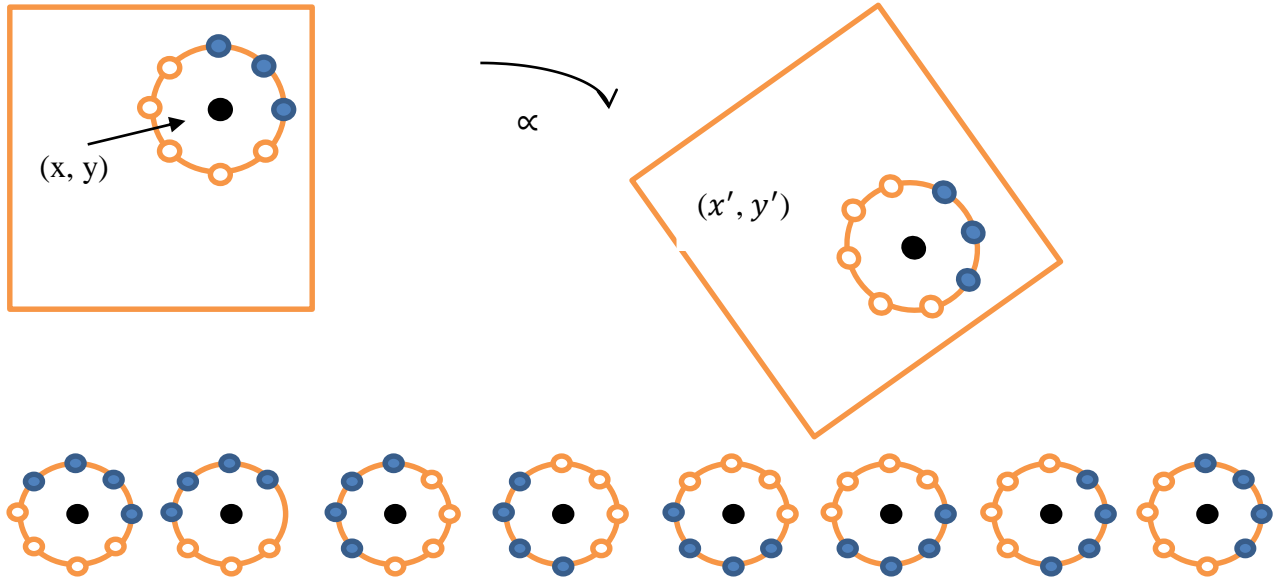
### 3.2.2 Các biến thể của LBP

#### 3.2.2.1 LBP đồng dạng

Một mẫu nhị phân được gọi là đồng dạng khi xét chuỗi xoay vòng thì có nhiều nhất 2 sự biến đổi từ giá trị 0 sang 1 hoặc từ bit 1 sang 0. Ví dụ: 00000000 có 0 sự biến đổi, 01110000 có 2 sự biến đổi, 11000011 có 2 sự biến đổi nên đây là uniform LBP. 11001001 có 4 sự biến đổi và 01010011 có 6 sự biến đổi nên không phải là uniform LBP. Dựa trên sự địa nghĩa này, bảng ánh xạ cho bán kính làm việc P-neighbours sẽ có  $P(P-1) + 3$  nhãn. Có nghĩa là 59 nhãn trong trường hợp làm việc với 8-neighbour.

#### 3.2.2.2 LBP bất biến với phép quay

Giả sử  $I_{\alpha}(x, y)$  là ảnh quay góc ( $\alpha$ ) của ảnh  $I(x, y)$ . Với phép quay này điểm ảnh  $(x, y)$  sẽ nằm tại vị trí  $(x', y')$ . Trong ví dụ này (hình bên dưới): tất cả 8 mẫu đều được ánh xạ về mẫu LBP đầu tiên vì mẫu đầu tiên cho giá trị nhỏ nhất.



Hình 16: Hình minh họa về trường hợp của LBP sau khi quay với góc 15 độ.

#### 3.2.2.3 LBP đồng dạng có khả năng bất biến với phép quay

Kết hợp LBP đồng dạng và LBP bất biến với phép quay có thể tạo nên một dạng biến thể khác của LBP (uniform rotation – invariant LBP). Dựa trên định nghĩa này,

bảng ánh xạ cho bán kính làm việc P-neighbors sẽ có  $P + 2$  nhãn (label). Có nghĩa là 10 nhãn trường hợp làm việc với 8 P-neighbour.

### 3.3 Gabor Filter

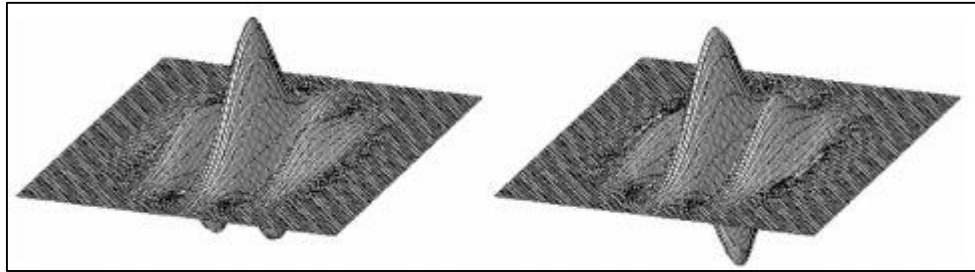
Là một kỹ thuật rút trích đặc trưng được sử dụng phổ biến trong lĩnh vực thị giác máy tính (computer vision). Khi bộ lọc Gabor được áp dụng cho một tấm hình, các điểm ảnh có cùng hướng và tần số cục bộ sẽ nhận các phản hồi lớn nhất. Công thức của bộ lọc Gabor có thể được diễn tả về mặt toán học như sau:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \exp[j2\pi f\tilde{x}]$$

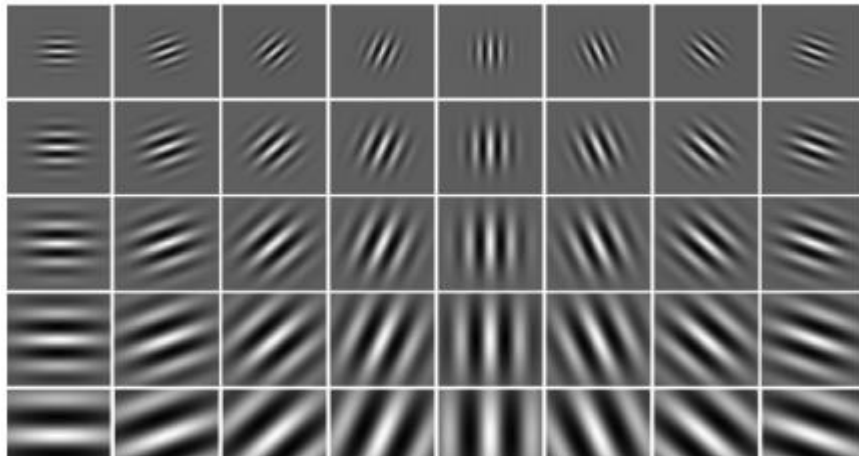
$$\tilde{x} = x\cos\theta + y\sin\theta$$

$$\tilde{y} = -x\sin\theta + y\cos\theta$$

Khi  $\sigma_x$  và  $\sigma_y$  là độ lệch chuẩn của bộ lọc Gaussian trên trục X và Y.  $\theta$  chứng tỏ rằng hướng của bộ lọc Gabor, và nếu  $f$  là một sóng tần số có dạng hình Sin.



Hình 18: Hình biểu diễn phần thực và ảo của tập hợp hàm Gabor



Hình 17: Hình ví dụ về ngân hàng bộ lọc Gabor

Mặt khác để nhận được một vector đặc trưng, một ngân hàng bộ lọc chứa các bộ lọc Gabor với các tần số và góc khác nhau được sử dụng.

Trong phạm vi đề tài này, một ngân hàng bộ lọc Gabor chứa 18 bộ lọc được sử dụng. Các bộ lọc được khởi tạo bởi 6 góc khác nhau với 30 độ được tăng thêm, vd: 0, 30, 60, 90, 120, 150, và 3 phần khác nhau (khoảng 4, 8, 16 pixels/cycle). Khi một ảnh đầu vào  $I(x, y)$  được thực hiện bởi bộ lọc Gabor, đầu ra được tính toán bằng việc xoay tấm ảnh đó với bộ lọc Gabor tương ứng như sau:

$$R_n(x, y) = I_n(x, y) * G_n(x, y)$$

### 3.4 Support Vector Machine (SVM)

#### 3.4.1 Giới thiệu

SVM là một phương pháp trong việc phân loại dữ liệu tuyến tính và không tuyến tính. Bài báo đầu tiên về Support Vector Machine được giới thiệu vào năm 1992 bởi Vladimir Vapnik và hai đồng sự Bernhard Boser và Isabelle Guyon, mặc dù nền móng cơ bản của SVM đã có từ năm 1960 (bao gồm các công việc được thực hiện rất sớm bởi Vapnik và Alexei Chervonenkis trong lý thuyết học thống kê).

Trước khi đi vào tìm hiểu phương pháp SVM, ta phải biết được các khái niệm về siêu phẳng phân chia tuyến tính, support vector...

#### 3.4.2 Các khái niệm cơ bản

##### 3.4.2.1 Siêu phẳng phân cách

Cho trước tập dữ liệu  $D$  gồm  $(X_1, y_1), (X_2, y_2), \dots, (X_{|D|}, y_{|D|})$ .

Trong đó  $X_i$  là một tập các bộ huấn luyện tương ứng với nhãn lớp  $y_i$ . Mỗi  $y_i$  sẽ nhận một trong hai giá trị hoặc là +1 hoặc là -1 ( $y_i \in \{+1, -1\}$ ).

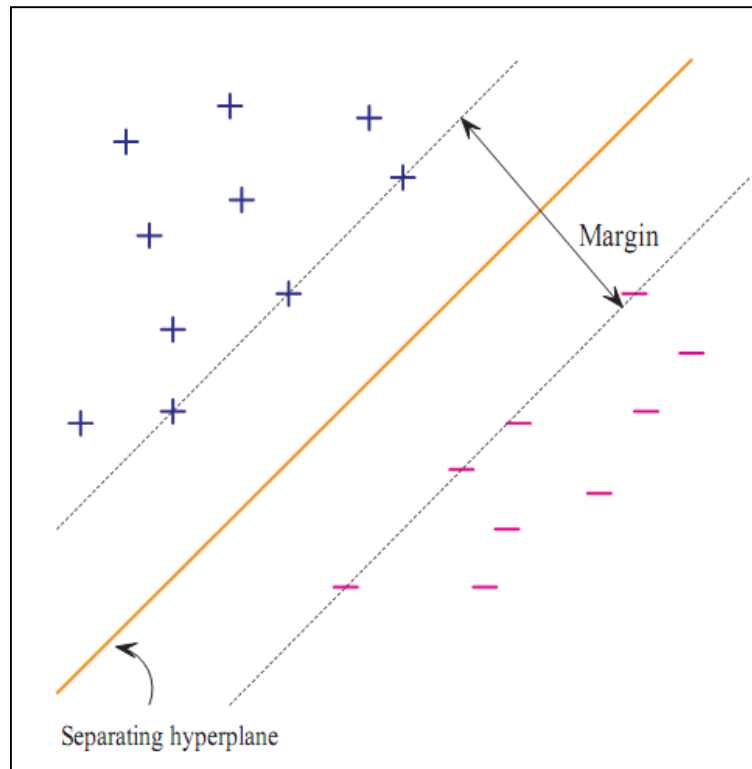
Phương pháp phân lớp SVM sẽ tìm ra đường phân lớp “tốt nhất” để phân chia tập dữ liệu này thành từng lớp tách biệt ra với nhau. Theo tài liệu “Robust Real-time Object Detection” của Paul Viola và Michael Jones, phương trình tổng quát của một đường phân chia như vậy được biểu diễn dưới dạng sau:

$$W \cdot X + b = 0 \tag{1}$$

Trong đó:

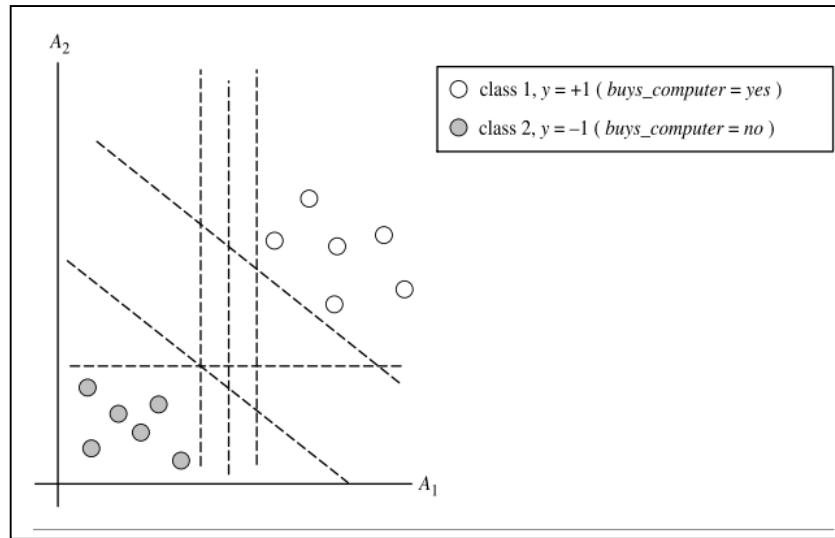
- $W$ : Vector trọng số,  $W = \{w_1, w_2, \dots, w_n\}$ .
- $n$ : Số thuộc tính (hay còn gọi là số chiều của dữ liệu).
- $b$ : Một đại lượng vô hướng, thường được xem như là một độ nghiêng (bias).

Đối với trường hợp dữ liệu hai chiều (hai thuộc tính) thì phương trình trên biểu diễn của đường thẳng phân chia. Nếu dữ liệu là ba chiều thì đường phân chia giữa hai tập sẽ là một mặt phẳng phân cách. Tổng quát cho dữ liệu  $n$  chiều thì sẽ được phân cách bởi một siêu phẳng. Trong bài toán sẽ sử dụng thuật ngữ “siêu phẳng” (hyperplane) để chỉ đến ranh giới quyết định mà muốn tìm kiếm bất chấp số lượng thuộc tính.



Hình 19: Đường màu vàng trên hình là đường phân chia đối với tập dữ liệu gồm hai thuộc tính

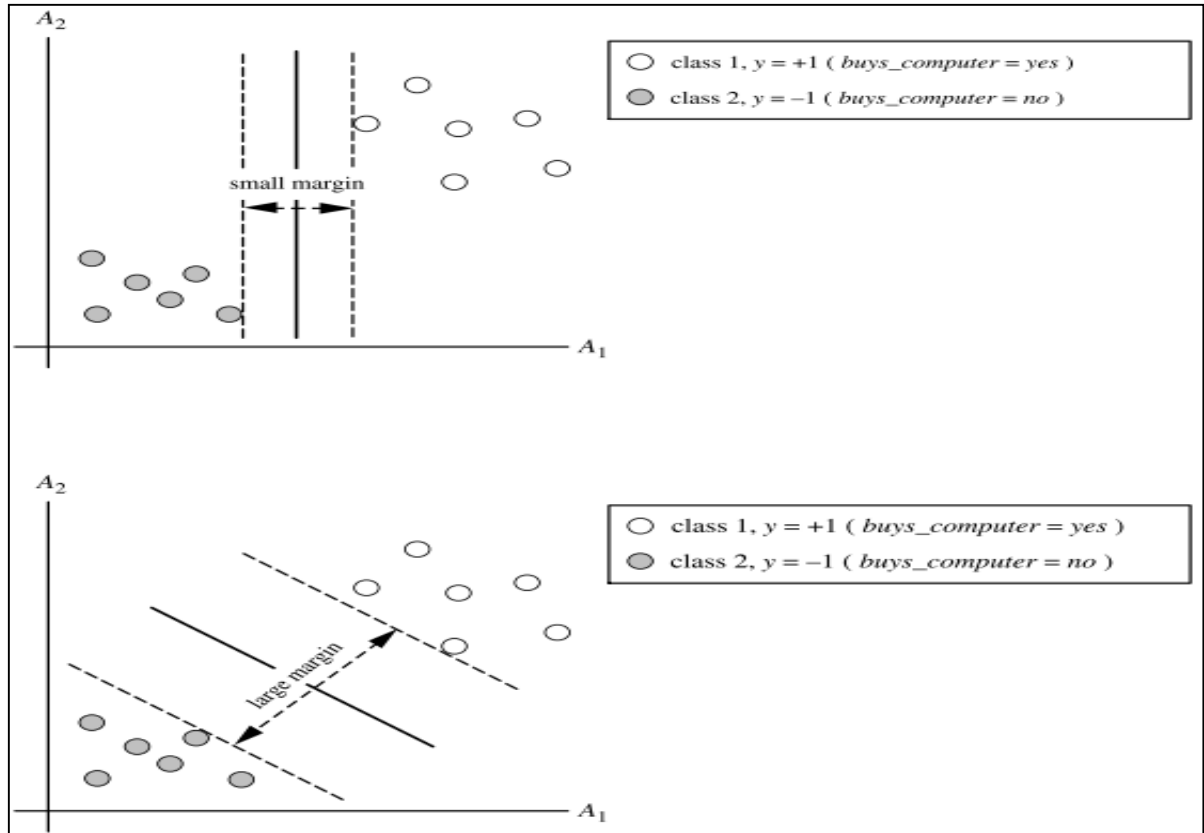
Tuy nhiên trong thực tế ta có thể tìm được vô số những siêu phẳng phân chia trên cùng một tập dữ liệu. Và muốn tìm đường thẳng phân chia sao cho tốt nhất, có nghĩa là có sai sót phân loại bé nhất trên bộ dữ liệu.



Hình 20: Một bộ dữ liệu hai chiều được phân chia tuyến tính

Do đó mục tiêu của phương pháp phân lớp SVM là tìm một siêu phẳng phân cách giữa hai lớp sao cho khoảng cách lề (margin) giữa hai lớp đạt cực đại.

Siêu phẳng có biên độ lớn nhất (maximum marginal hyperplane) sẽ được chọn như là siêu phẳng phân chia tập dữ liệu một cách tốt nhất. Trong hình bên dưới, ta thấy có hai siêu phẳng có thể phân chia được và những biên độ của nó. Trước khi đi vào định nghĩa của biên độ (margin), hãy nhìn vào hình trên một cách trực quan. Cả hai siêu phẳng đều phân tách tất cả những bộ dữ liệu cho trước. Một cách trực quan, siêu phẳng với biên độ lớn hơn sẽ chính xác hơn trong việc phân loại các bộ dữ liệu trong tương lai so với siêu phẳng có biên độ nhỏ hơn. Điều này là lý do tại sao (trong suốt giai đoạn học hay huấn luyện), SVM tìm những siêu phẳng có biên độ lớn nhất, gọi là MMH (maximum marginal hyperlane). Siêu phẳng có biên độ lớn nhất là siêu phẳng có khoảng cách từ nó tới hai mặt bên của nó thì bằng nhau (mặt bên song song với siêu phẳng). Khoảng cách đó thật ra là khoảng cách ngắn nhất từ MMH tới bộ dữ liệu huấn luyện gần nhất của mỗi lớp. Siêu phẳng có biên độ lớn nhất này cho một sự phân loại tốt nhất giữa các lớp.



Hình 21: 2 siêu phẳng phân chia tuyến tính cùng với biên độ của nó

Siêu phẳng phân cách có vai trò quan trọng trong việc phân lớp, nó quyết định xem một bộ dữ liệu sẽ thuộc về lớp nào. Để thực hiện việc phân lớp, SVM chỉ cần xác định xem một bộ dữ liệu nằm về phía nào của siêu phẳng phân cách:

$$D(x) = \text{sign}(W \cdot X + b) \quad (2)$$

- $D(x) < 0$ : bộ dữ liệu sẽ nằm phía dưới siêu phẳng phân cách
- $D(x) = 0$ : bộ dữ liệu sẽ nằm trên siêu phẳng phân cách
- $D(x) > 0$ : bộ dữ liệu sẽ nằm phía trên siêu phẳng phân cách



### 3.4.2.2 Support Vector

Ta có phương trình tổng quát của siêu phẳng:

$$W.X + b = 0$$

Ta xét trên ví dụ sau:

Với bộ dữ liệu huấn luyện có hai thuộc tính  $A_1$  và  $A_2$ :  $X = \{x_1, x_2\}$ , với  $x_1, x_2$  là giá trị của thuộc tính  $A_1, A_2$ .  $W = \{w_1, w_2\}$ . Phương trình siêu phẳng có thể viết lại:

$$w_0 + w_1x_1 + w_2x_2 = 0$$

Trong đó:

- $w_0$  tương đương với hằng số  $b$  trong phương trình tổng quát của siêu phẳng

Vì vậy mỗi điểm nằm trên siêu phẳng phân cách thỏa mãn:

$$w_0 + w_1x_1 + w_2x_2 > 0$$

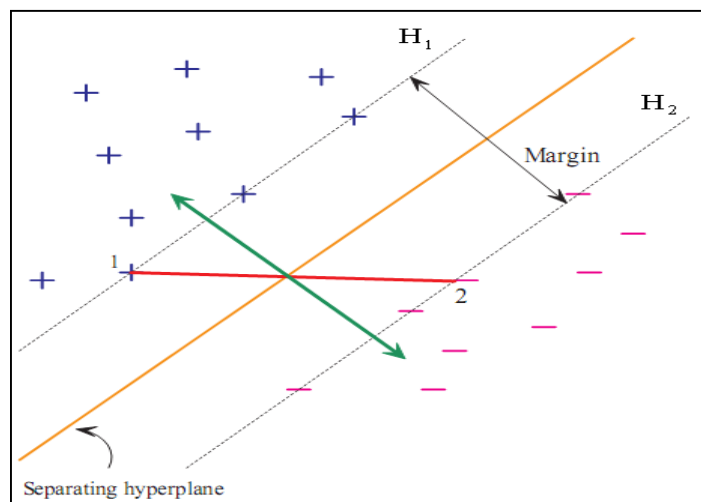
Tương tự, những điểm nằm dưới siêu phẳng phân cách phải thỏa mãn:

$$w_0 + w_1x_1 + w_2x_2 < 0$$

Bằng cách điều chỉnh trọng số  $w_0$  ta có:

$$H_1: w_0 + w_1x_1 + w_2x_2 \geq 1 \text{ với } y_i = +1$$

$$H_2: w_0 + w_1x_1 + w_2x_2 \leq -1 \text{ với } y_i = -1$$



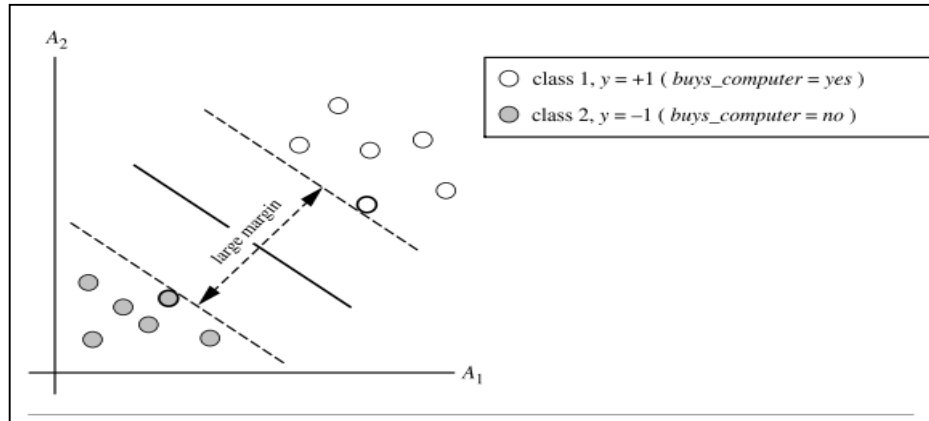
Hình 22: Đường biểu diễn  $H_1$  và  $H_2$ , đường màu đỏ là khoảng cách Euclidean của hai điểm 1 và 2. Đường màu xanh là khoảng cách Euclidean nhỏ nhất

Điều này có nghĩa là nếu bất kì bộ nào nằm tại hoặc trên  $H_1$  đều thuộc về lớp +1, và bất kì bộ nào nằm tại hoặc dưới  $H_2$  đều thuộc về lớp -1. Kết hợp 2 bất đẳng thức trên ta có:

$$y_i(w_0 + w_1x_1 + w_2x_2) \geq 1, \forall i$$

Mỗi bộ huấn luyện nằm tại các mặt biên  $H_1$  hay  $H_2$  thỏa mãn phương trình trên được gọi là support vectors. Support vectors là những bộ gần với siêu phẳng phân chia tuyến tính (MMH) nhất.

Trong hình bên dưới, support vectors là hình tròn có viền dày hơn. Ta thấy rằng các support vectors là những bộ khó phân lớp nhất và cung cấp nhiều thông tin nhất cho việc phân lớp.



Hình 23: Các support vector trong SVM. Các support vector là những hình có viền dày hơn

### 3.4.2.3 Biên độ Margin

Từ các điều trên có thể đưa ra một công thức cho việc tính biên độ lớn nhất.

Khoảng cách từ siêu phẳng phân chia đến mọi điểm tại  $H_1$  là  $\frac{1}{\|W\|}$

Trong đó:

- $\|W\|$  là khoảng cách Euclidean chuẩn của  $W$  là  $\sqrt{W * W}$ . Với  $W = \{w_1, w_2\}$  khi đó  $\sqrt{W * W} = \sqrt{w_1^2 + w_2^2}$ .

Theo định nghĩa, khoảng cách từ siêu phẳng đến  $H_1$  bằng với khoảng cách từ mọi điểm tại  $H_2$  đến siêu phẳng. Vì vậy, kích thước của biên độ cực đại là  $\frac{2}{\|w\|}$ .

### 3.4.3 Phân lớp dữ liệu

#### 3.4.3.1 Trường hợp dữ liệu có thể phân chia tuyến tính được

Việc huấn luyện SVM với mục đích là để tìm ra các support vectors và MMH. MMH là ranh giới phân chia tuyến tính giữa các lớp và vì thế SVM tương ứng có thể được sử dụng để phân lớp dữ liệu mà dữ liệu đó có thể phân chia tuyến tính. Bài toán xem SVM được huấn luyện là SVM tuyến tính.

Sau khi huấn luyện SVM, bài toán sẽ phân loại các bộ mới. Dựa trên công thức Lagrangian ta có:

$$d(X^T) = \sum_{i=1}^l y_i \alpha_i X^T + b_0$$

Trong đó:

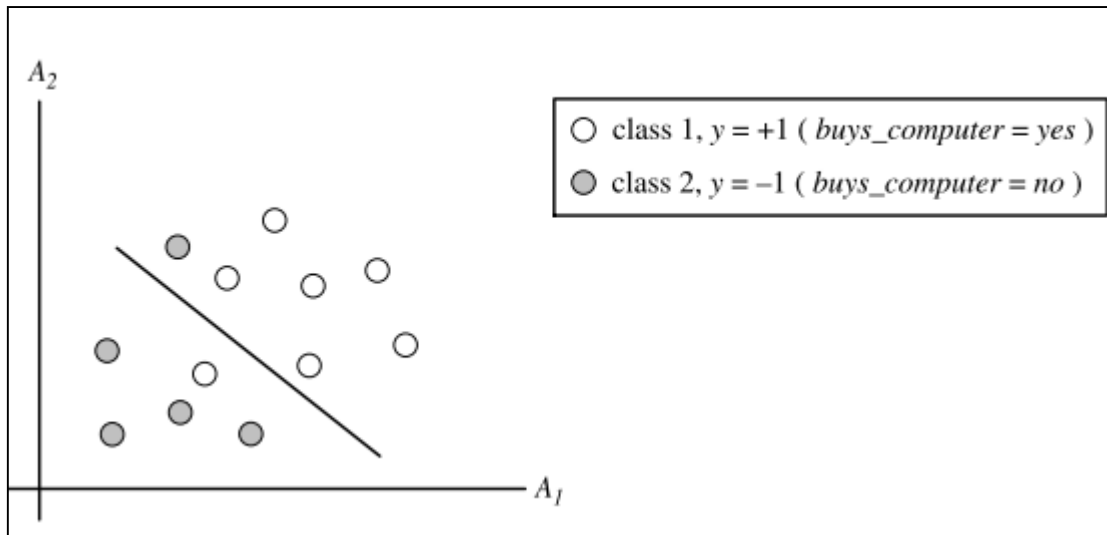
- $y_i$  là nhãn lớp của support vector  $X_i$
- $X^T$  là một bộ test
- $\alpha$  (nhân tử Lagrangian)
- $b_0$  là biến số được xác định bởi sự tối ưu hóa hay các thuật toán SVM
- $l$  là số lượng các support vectors.

MMH có thể được xem như “ranh giới quyết định” trong việc quyết định xem một bộ test bất kỳ sẽ thuộc vào lớp nào. Cho một bộ test  $X^T$ , gán nó vào phương trình trên, và sau đó kiểm tra dấu của kết quả. Từ đó ta sẽ biết được bộ test sẽ rơi vào mặt nào của siêu phẳng. Nếu dấu là dương, thì  $X^T$  rơi vào phía trên của MMH, và SVM đoán rằng  $X^T$  thuộc về lớp +1. Nếu dấu là âm, thì  $X^T$  nằm tại hoặc dưới MMH và nhãn lớp được đoán là -1.

### 3.4.3.2 Trường hợp dữ liệu không thể chia tuyến tính được

Trong phần trên đề cập đến trường hợp SVM phân lớp những dữ liệu có thể phân chia tuyến tính, nhưng nếu dữ liệu không thể phân chia tuyến tính thì sao? Trong trường hợp này không có đường thẳng nào có thể vẽ được để phân chia các lớp này. SVM tuyến tính mà đã học thì không đem lại lời giải khả thi trong trường hợp này.

Tuy nhiên hướng tiếp cận của SVM tuyến tính có thể được mở rộng để tạo ra



Hình 24: Một trường hợp đơn giản trên không gian 2 chiều

SVM không tuyến tính cho việc phân lớp các dữ liệu không thể phân chia tuyến tính (hay gọi tắt là dữ liệu không tuyến tính). Những SVM như vậy có khả năng tìm những ranh giới quyết định không tuyến tính (những mặt không tuyến tính) trong không gian đầu vào.

“Làm thế nào mở rộng tiếp cận tuyến tính?”. Thu được SVM phi tuyến bằng cách mở rộng SVM tuyến tính như sau. Có hai bước chính:

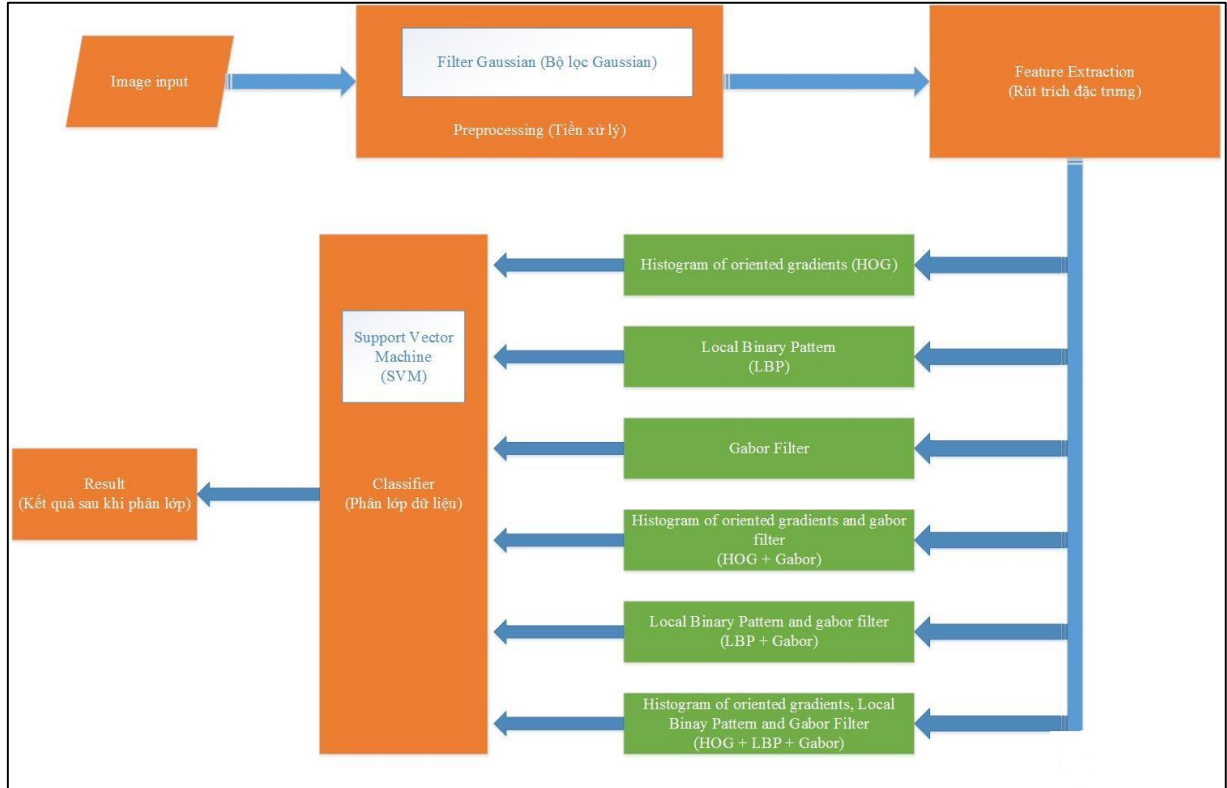
Bước 1: Chuyển dữ liệu nguồn lên một không gian nhiều chiều hơn bằng cách sử dụng ánh xạ phi tuyến. Một vài ánh xạ phi tuyến thông thường có thể được sử dụng để thực hiện bước này

Bước 2: Tìm những siêu phẳng trong không gian mới này. Cuối cùng là lại quay lại vấn đề tối ưu bình phương đã được giải quyết sử dụng công thức SVM tuyến tính.

Siêu phẳng có biên độ lớn nhất được tìm thấy trong không gian mới tương ứng với siêu bề mặt phân chia không tuyến tính trong không gian ban đầu.

## CHƯƠNG 4 – Ý TƯỞNG GIẢI QUYẾT BÀI TOÁN

### 4.1 Mô hình thực hiện



### 4.2 Mô tả mô hình

#### 4.2.1 Thu thập dữ liệu

Tập dữ liệu sử dụng trong bài này gồm:

- Bộ tập dữ liệu do nhóm tự sưu tầm về biển báo nguy hiểm thuộc giao thông đường bộ Việt Nam, gồm có 22 loại biển báo.
- Bộ tập dữ liệu biển báo được chụp tại Đức.

#### 4.2.2 Tiền xử lý

Được áp dụng để cải thiện khả năng nhận dạng hình ảnh, ở bước này sử dụng giải thuật Gaussian để khử nhiễu cho ảnh.

#### 4.2.3 Rút trích đặc trưng

Rút trích đặc trưng là quá trình mô tả một đối tượng được nhận dạng bởi các phép đo có các giá trị tương tự nhau với các đối tượng trong cùng loại, và khác nhau với các đối tượng trong các loại khác nhau. Từ đó tìm ra các đặc trưng phân biệt bất biến. Thường là các đặc trưng mô tả các thuộc tính như hình dạng, màu sắc, và nhiều loại kết cấu là bất biến dịch, sự xoay, và tỷ lệ. Chọn lọc đặc trưng là lựa chọn các đặc trưng mạnh và loại bỏ các đặc trưng dư thừa hoặc không liên quan. Lựa chọn đặc trưng thường được sử dụng khi đầu vào có quá nhiều đặc trưng nhưng lại có quá ít mẫu. Ở phần này, sẽ lần lượt thực hiện từng phương pháp để tìm ra được phương pháp nào đạt hiệu quả tốt nhất.

- Sử dụng giải thuật Histogram of oriented gradients (HOG) để rút trích đặc trưng.
- Sử dụng giải thuật Local Binary Pattern (LBP) để rút trích đặc trưng.
- Sử dụng giải thuật Gabor Filter để rút trích đặc trưng.
- Kết hợp giải thuật HOG và Gabor để rút trích đặc trưng.
- Kết hợp giải thuật LBP và Gabor để rút trích đặc trưng.
- Kết hợp giải thuật HOG, LBP và Gabor để rút trích đặc trưng.

#### 4.2.4 Phân lớp dữ liệu

Nhiệm vụ của phân lớp của một hệ thống đầy đủ là sử dụng các véc tơ đặc trưng được cung cấp bởi rút trích đặc trưng gán đối tượng vào một loại. Vì thường là không thể phân lớp với hiệu suất hoàn hảo nên chỉ ở mức tổng quát là xác định xác suất cho mỗi loại có thể. Từ véc tơ đặc trưng đại diện của dữ liệu đầu vào có thể phát triển bộ phân lớp độc lập.

### 4.3 Phương pháp đánh giá

- Bài toán trong đồ án sử dụng phương pháp đánh giá là Accuracy (thông tin về mức độ chính xác ).
- Công thức tính độ chính xác:

$$\text{Độ chính xác (\%)} = \frac{\text{Số lượng ảnh đúng} \times 100\%}{\text{Tổng số lượng ảnh}}$$

- Đồ án chạy 10 lần với mỗi lần bộ dữ liệu train (70% số lượng ảnh) và test (30% số lượng ảnh) là khác nhau để tránh trường hợp over-fitting (quá vừa dữ liệu).



## CHƯƠNG 5 – KẾT QUẢ THỰC NGHIỆM

### 5.1 Bộ dữ liệu the German Traffic Sign Detection Benchmark (GTSDB)

#### 5.1.1 HOG

Số lần	Tỷ lệ đúng(%)
1	90,61
2	88,64
3	89,2
4	88,37
5	90,58
6	90,03
7	89,47
8	89,75
9	90,86
10	88,64
Trung bình	89,615

Bảng 2: Bảng tỷ lệ % đúng, sai của giải thuật HOG

### 5.2 Bộ dữ liệu biển báo giao thông Việt Nam

#### 5.2.1 HOG

Số lần	Tỷ lệ đúng (%)
1	87,08
2	86,12
3	89,69
4	89,6
5	88,59
6	86,46
7	86,16
8	85,5
9	90,91
10	91,64
Trung bình	88,175

Bảng 3: Bảng tỷ lệ % đúng,sai của giải thuật HOG

### 5.2.2 LBP

Số lần	Tỷ lệ đúng (%)
1	11,6
2	11,88
3	9,6
4	10,5
5	9,8
6	12,6
7	10,8
8	11,8
9	13,8
10	12,8
Trung bình	11,518

Bảng 4: Bảng tỷ lệ % đúng,sai của giải thuật LBP

### 5.2.3 LBP (Overlap)

Số lần	Tỷ lệ đúng(%)
1	48,2
2	48,2
3	48,15
4	49,2
5	53,6
6	52,6
7	51,96
8	49,2
9	58,2
10	53,52
Trung bình	51,283

Bảng 5: Bảng tỷ lệ % đúng,sai của giải thuật LBP (Overlap)

#### 5.2.4 Gabor moment

Số lần	Tỷ lệ đúng(%)
1	7,6
2	8,8
3	7,96
4	11,5
5	10
6	11,6
7	12,8
8	12,6
9	14,4
10	13,6
Trung bình	11,086

Bảng 6: Bảng tỷ lệ % đúng,sai của giải thuật Gabor moment

#### 5.2.5 Gabor

Số lần	Tỷ lệ đúng(%)
1	70,54
2	61,7
3	68,91
4	58,71
5	63,13
6	39,58
7	66,58
8	66,92
9	62,08
10	60
Trung bình	61,815

Bảng 7: Bảng tỷ lệ % đúng,sai của giải thuật Gabor

### 5.2.6 LBP và Gabor

Số lần	Tỷ lệ đúng(%)
1	60,98
2	62,9
3	65,32
4	62,16
5	63,4
6	61,46
7	63,97
8	60,05
9	64,34
10	63,5
Trung bình	62,808

Bảng 8: Bảng tỷ lệ % đúng,sai của giải thuật LBP và Gabor

### 5.2.7 HOG và Gabor

Số lần	Tỷ lệ đúng(%)
1	89,8
2	87,9
3	90,28
4	90,2
5	89,2
6	87
7	86,8
8	86,5
9	91,54
10	91,7
Trung bình	89,092

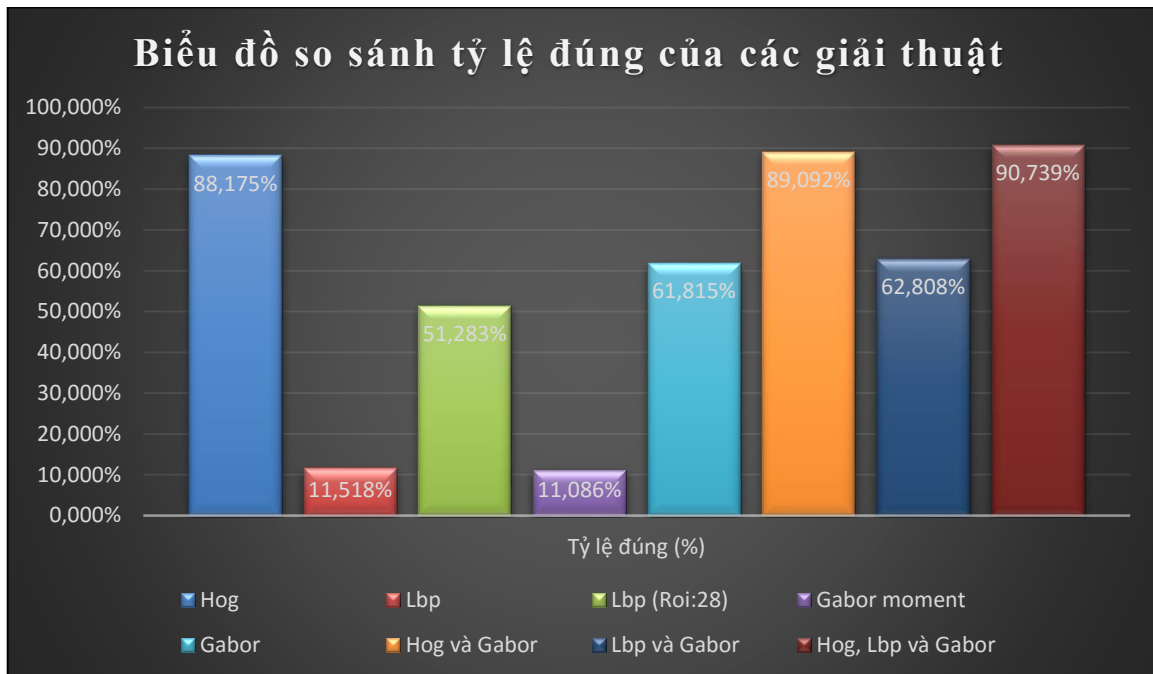
Bảng 9: Bảng tỷ lệ % đúng,sai của giải thuật HOG và Gabor

### 5.2.8 HOG, LBP và Gabor

Số lần	Tỷ lệ sai(%)	Tỷ lệ đúng(%)
1	9,48	90,52
2	9,68	90,32
3	8,82	91,18
4	9,2	90,8
5	9,45	90,55
6	10,9	89,1
7	10,25	89,75
8	10,23	89,77
9	7,48	92,52
10	7,12	92,88
Trung bình	9,261	90,739

Bảng 10: Bảng tỷ lệ % đúng,sai của giải thuật HOG, LBP và Gabor

### 5.3 So sánh giữa các giải thuật



Tỷ lệ phần trăm đúng nhiều nhất (93,02%) là giải thuật kết hợp HOG, LBP và Gabor, đây là giải thuật có số lượng đặc trưng lớn nhất (4376 đặc trưng). Tỷ lệ phần trăm đúng thấp nhất (11,086%) là giải thuật Gabor moment (120 đặc trưng).

## CHƯƠNG 6 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Sau thời gian tìm hiểu đề tài “Nhận diện biển báo giao thông”, với mục tiêu huấn luyện được hệ thống thị giác máy tính phân loại được các đối tượng biển báo trong ảnh, đồ án đã thực hiện đề tài với kết quả đã cho ra tỉ lệ đúng/sai trong việc phân loại các biển báo vào các loại tương ứng.

Đồ án đã đóng góp:

- Bộ dữ liệu biển báo nguy hiểm của giao thông đường bộ Việt Nam.
- Kết quả thực nghiệm các giải thuật rút trích đặc trưng cục bộ như HOG, LBP, Gabor kết hợp bộ phân lớp SVM.

Trong quá trình thực hiện đề tài, đồ án đã thu được các kết quả như sau:

- Tìm hiểu nắm được ý tưởng cũng như cách rút trích đặc trưng đối tượng cơ bản của thuật toán HOG, LBP, Gabor.
- Tìm hiểu về phương pháp SVM, một trong những phương pháp phân loại dữ liệu, hỗ trợ nhận dạng đối tượng.
- Phương pháp nhận diện biển báo có độ chính xác cao, bằng cách áp dụng bộ phân lớp SVM cùng với các giải thuật rút trích đặc trưng cục bộ như HOG, LBP, Gabor.

Bên cạnh đó, đề tài còn nhiều hạn chế:

- Vì thuật toán chưa được tối ưu và cải tiến nên kết quả thu được bên cạnh những trường hợp chính xác vẫn còn nhiều trường hợp sai sót, như nhận diện sai hoặc thiếu sót, tỉ lệ sai vẫn còn khá cao.
- Tốc độ xử lý không đạt thời gian thực do chi phí trong việc tính toán HOG, LBP, Gabor, SVM là tương đối lớn.
- Để cho mô hình SVM được chính xác và hiệu quả thì phụ thuộc rất nhiều vào tập dữ liệu huấn luyện. Tập dữ liệu này phải thu thập đủ lớn và khách quan. Ngoài ra, việc nhận diện biển báo còn bị ảnh hưởng rất nhiều bởi background phía sau,

vì vậy cần phải hạn chế tối đa độ phức tạp của background để có thể thu được kết quả ít sai sót nhất.

## **6.2 Hướng phát triển**

Hướng phát triển của đề tài này là phải khắc phục được các hạn chế nêu ở trên, cụ thể như sau:

- Tối ưu thuật toán xử lý để kết quả đạt được thời gian thực.
- Huấn luyện với một tập mẫu lớn hơn để nâng cao khả năng nhận dạng đối tượng của chương trình, tăng tỉ lệ chính xác.
- Khắc phục được sai sót do ảnh hưởng từ điều kiện ánh sáng và background phức tạp.
- Triển khai được ứng dụng lên nền tảng di động, phục vụ quá trình tham gia giao thông của mọi người.

## **TÀI LIỆU THAM KHẢO**

### **Tiếng Việt**

### **Tiếng Anh**

1. “On circular traffic sign detection and recognition” by Selcan Kaplan Berkaya (2015).