

Nhận xét của giảng viên hướng dẫn

Cần Thơ, ngày tháng năm 2019
Giảng viên hướng dẫn (Thư ký Hội đồng)

TS. Phạm Thế Phi

Nhận xét của giảng viên phản biện 1

Cần Thơ, ngày tháng năm 2019
Giảng viên phản biện 1 (Chủ tịch Hội đồng)

PGS. TS. Đỗ Thanh Nghị

Nhận xét của giảng viên phản biện 2

Cần Thơ, ngày tháng năm 2019
Giảng viên phản biện 2 (Ủy viên Hội đồng)

PGS. TS. Phạm Nguyên Khang

Lời cảm ơn

Đầu tiên, và quan trọng nhất xin gửi lời cảm ơn sâu sắc đến gia đình tôi, những người luôn ủng hộ vô điều kiện và hỗ trợ tôi hết mình trong thời gian 4,5 năm học tập và nghiên cứu tại Trường Đại học Cần Thơ.

Tôi xin được gửi lời cảm ơn đến Ban giám hiệu Trường Đại học Cần Thơ. Đặc biệt, quý Thầy, Cô Khoa Công nghệ thông tin và Truyền thông đã truyền đạt rất nhiều kiến thức chuyên ngành hữu ích để tôi có nền tảng lý thuyết thực hiện đề tài này. Không quên gửi lời cảm ơn đến những người bạn đã luôn ủng hộ tôi thực hiện đề tài này.

Cuối cùng, đặc biệt, tôi xin được gửi lời cảm ơn đến TS. Phạm Thế Phi, đã tận tình hướng dẫn, giúp đỡ lúc tôi gặp khó khăn khi thực hiện đề tài, tôi sẽ ghi nhớ những điều này khi bắt đầu đến với một môi trường mới. Một lần nữa, xin được cảm ơn thầy.

Đề tài này là tất cả tâm quyết, thời gian, công sức và sự cố gắng không ngừng nghỉ mà tôi đã dành ra thực hiện. Nhưng vì thời gian và kiến thức còn giới hạn nên không thể tránh khỏi những thiếu sót trong quá trình thực hiện, tôi hy vọng đề tài này sẽ được hoàn thành tốt hơn nhờ vào những lời góp ý và phản hồi tích cực từ quý Thầy, Cô và các bạn Độc giả.

Cần Thơ, ngày tháng năm 2019

Sinh viên thực hiện

Phạm Hoàng Viện

Mục lục

Lời cảm ơn	iv
Mục lục	vii
Bảng thuật ngữ và từ viết tắt	viii
Danh sách hình vẽ	x
Danh sách bảng	xi
Tóm tắt	xii
Abstract	xiii
1 Giới thiệu	1
1.1 Đặt vấn đề	1
1.2 Những nghiên cứu liên quan	2
1.3 Mục tiêu của đề tài	3
1.4 Đối tượng và phạm vi nghiên cứu	3
1.4.1 Đối tượng nghiên cứu	3
1.4.2 Phạm vi nghiên cứu	3
1.5 Phương pháp nghiên cứu	3
1.5.1 Nghiên cứu lý thuyết	3
1.5.2 Thực nghiệm	3
1.6 Nội dung nghiên cứu	4
1.7 Bố cục luận văn	4
2 Cở sở lý thuyết	5
2.1 Một số loại bệnh trên lá cam	5
2.2 Các mô hình màu	7
2.2.1 Giới thiệu	7
2.2.2 Mô hình màu RGB	7

2.2.2.1	Ưu điểm	8
2.2.2.2	Hạn chế	9
2.2.3	Mô hình màu HSV	9
2.2.3.1	Ưu điểm	9
2.2.3.2	Hạn chế	10
2.3	Các mô hình biểu diễn ảnh	10
2.3.1	Giới thiệu	10
2.3.2	Mô hình raster	10
2.3.3	Mô hình vector	11
2.3.4	Phân loại ảnh	12
2.3.4.1	Ảnh nhị phân	12
2.3.4.2	Ảnh mức xám	12
2.3.4.3	Ảnh màu	12
2.4	Các phương pháp trích chọn đặc trưng	14
2.4.1	Giới thiệu	14
2.4.2	Trích chọn đặc trưng SIFT	14
2.4.2.1	Phát hiện điểm cực trị	14
2.4.2.2	Định vị điểm hấp dẫn	16
2.4.2.3	Xác định hướng cho điểm hấp dẫn	17
2.4.2.4	Mô tả các điểm hấp dẫn	17
2.4.2.5	Mô hình túi đựng từ trực quan	17
2.4.3	Trích chọn đặc trưng Color	18
2.4.4	Trích chọn đặc trưng HOG	19
2.4.4.1	Tiền xử lý	19
2.4.4.2	Tính toán ảnh gradient	20
2.4.4.3	Tính toán biểu đồ gradient các ô	20
2.4.4.4	Chuẩn hóa các khối	23
2.4.4.5	Tính toán véc-tơ đặc trưng	23
2.4.5	Trích chọn đặc trưng GIST	24
2.4.5.1	Tiền xử lý ảnh	24
2.4.5.2	Sinh bộ lọc Gabor	24
2.4.5.3	Tính toán véc-tơ đặc trưng	25
2.4.6	Trích chọn đặc trưng sử dụng mạng nơ-ron tích chập	26
2.4.6.1	Tổng quan mạng nơ-ron tích chập	26
2.4.6.2	Mô hình Residual Networks	28
2.5	Máy học véc-tơ hỗ trợ	29
2.5.1	Giới thiệu	29
2.5.2	Ưu điểm	32
2.5.3	Nhược điểm	32

3 Xây dựng hệ thống	33
3.1 Hệ thống phân loại bệnh trên lá cam	33
3.2 Thu thập và tổ chức dữ liệu	34
3.2.1 Thu thập ảnh	34
3.2.2 Tổ chức dữ liệu	34
3.3 Trích chọn các đặc trưng	34
3.3.1 Trích chọn đặc trưng SIFT	34
3.3.2 Trích chọn đặc trưng Color	35
3.3.3 Trích chọn đặc trưng HOG	35
3.3.4 Trích chọn đặc trưng GIST	35
3.3.5 Trích chọn đặc trưng Residual Networks	35
3.3.6 Tổng hợp véc-tơ đặc trưng	35
3.4 Huấn luyện mô hình	36
3.4.1 Chuẩn bị tập dữ liệu huấn luyện	36
3.4.2 Huấn luyện mô hình	36
4 Thực nghiệm và đánh giá kết quả	37
4.1 Môi trường thực nghiệm	37
4.2 Thực nghiệm phân loại	37
4.2.1 Chuẩn bị tập dữ liệu kiểm thử	37
4.2.2 Kết quả thực nghiệm	37
4.3 Đánh giá kết quả	39
5 Kết luận và hướng phát triển	40
5.1 Kết luận và hạn chế	40
5.1.1 Kết luận	40
5.1.2 Hạn chế	40
5.2 Hướng phát triển	40
Tài liệu tham khảo	42

Bảng thuật ngữ và từ viết tắt

STT	Thuật ngữ	Điễn giải
1	Bit	Đơn vị đo lượng thông tin – Binary digit
2	Byte	Đơn vị lưu trữ dữ liệu trên đĩa cứng
3	Pixel	Là đơn vị nhỏ nhất (điểm ảnh) của một ảnh
4	Raster model	Mô hình biểu diễn ảnh dưới dạng ma trận các pixel
5	Vector model	Mô hình biểu diễn ảnh dưới dạng ma trận các vector
6	MATLAB	Ngôn ngữ hiệu suất cao cho máy tính kĩ thuật
7	RGB	R ed - Đỏ, G reen - Xanh lá cây, B lue - Xanh dương
8	HSV	H ue - Vị trí vùng màu, S aturation - Độ đậm, nhạt vùng màu, V alue - Độ sáng, tối vùng màu
9	SVM	S upport V ector M achines
10	STL	S tandard T emplate L ibrary
11	BSD	B erkeley S oftware D istribution
12	RCA	R adio C orporation A merica
13	LCD	L iquid C rystal D isplay
14	LED	L ight E mitting D iodes
15	PBM	P ortable B itmap
16	PGM	P ortable G raymap
17	PPM	P ortable B itmap

Danh sách hình vẽ

2.1	Mô hình của một chiếc lá	5
2.2	Bệnh Ghẻ nhám	6
2.3	Bệnh Rầy phấn trắng	6
2.4	Bệnh Vàng lá gân xanh	6
2.5	Bệnh Vàng lá thối rẽ	7
2.6	Mô hình màu RGB	8
2.7	Mô hình màu HSV	9
2.8	Mô hình biểu diễn ảnh raster	11
2.9	Mô hình biểu diễn ảnh vector	11
2.10	Mô hình biểu diễn ảnh vector và raster trong thực tế	11
2.11	Ảnh nhị phân	12
2.12	Ảnh mức xám	13
2.13	Quang phổ nhìn thấy ở người	13
2.14	Ảnh màu 24 bits	13
2.15	Mô phỏng tính toán DoG	15
2.16	Quá trình tìm điểm cực trị trong hàm DoG	16
2.17	Quá trình lựa chọn các điểm hấp dẫn	16
2.18	Biểu diễn các vector đặc trưng	17
2.19	Biểu đồ của các từ trực quan	18
2.20	Phát hiện đặc trưng và trích xuất mô tả	18
2.21	Gom cụm các mô tả	19
2.22	Tiền xử lý ảnh trong HOG	20
2.23	Quá trình tính toán ảnh gradient	21
2.24	Chia ảnh thành các block 16×16	22
2.25	Tính toán biểu đồ gradient	22
2.26	Bình chọn điểm ảnh vào biểu đồ	23
2.27	Kiến trúc mạng nơ-ron tích chập	26
2.28	Nhân tích chập ảnh I và cửa sổ trượt K	27
2.29	Lớp pooling giảm kích thước dữ liệu đầu vào	28
2.30	Làm phẳng output của lớp cuối cùng	28

2.31	Làm phẳng lớp cuối cùng và kết nối đến fully-connected	28
2.32	Residual block	29
2.33	Phương pháp tuyến tính với máy học SVM	30
2.34	Phương pháp 1-vs-all cho máy học SVM đa lớp	31
2.35	Phương pháp 1-vs-1 cho máy học SVM đa lớp	31
3.1	Sơ đồ hệ thống phân loại bệnh trên lá cam	33
3.2	Cách tổ chức dữ liệu	34
4.1	Kết quả phân loại của mô hình SVM	38

Danh sách bảng

3.1	Thống kê số lượng mẫu trong tập dữ liệu huấn luyện	36
4.1	Thống kê số lượng mẫu trong tập dữ liệu thử	38
4.2	Thống kê mẫu phân loại đúng và sai trên tập dữ liệu kiểm thử của mô hình SVM	38
4.3	So sánh mô hình KNN và mô hình SVM	39

Tóm tắt

Trong luận văn này, chúng tôi đề xuất để tài ứng dụng máy học véc-tơ hỗ trợ (Support Vector Machines - SVM) trong phân loại bệnh trên lá cam sử dụng các đặc trưng SIFT (Scale-Invariant Feature Transform) kết hợp mô hình túi đựng từ trực quan (Bag of Visual Words - BoVW), Color, HOG (Histogram of Oriented Gradient), GIST và ResNet (Residual Networks). Bước rút trích đặc trưng từ ảnh lá bệnh, được thu thập từ 2 tỉnh Đồng Tháp và Hậu Giang. Chúng tôi đề xuất huấn luyện mô hình máy học SVM sử dụng các đặc trưng SIFT, Color, HOG, GIST và ResNet để thực hiện rút trích dựa trên các đặc trưng khác nhau của từng loại bệnh. Kết quả thực nghiệm trên tập dữ liệu gồm 711 ảnh của 5 loại, trong đó 4 loại là bệnh và 1 loại lá khỏe thường được xuất hiện phổ biến trong các nhà vườn cho thấy máy học SVM sử dụng các đặc trưng SIFT, Color, HOG, GIST và ResNet đạt đến 90.84% độ chính xác trên tập kiểm tra, cao hơn so với mô hình K láng giềng (K-Nearest Neighbors - KNN) có độ chính xác chỉ 42.95%.

Từ khóa—*Phân loại ảnh, phân loại bệnh trên lá cam, các phương pháp trích chọn đặc trưng, SIFT, BoVW, Color, HOG, GIST, ResNet, máy học véc-tơ hỗ trợ (SVM)*

Abstract

In this thesis, we present classification of orange leaf disease applied Support Vector Machines (SVM) using combination Scale-Invariant Feature Transform (SIFT) and Bag-of-Visual-Words (BoVW), Color, Histogram of Oriented Gradient (HOG), GIST and Residual Networks (ResNet). In step features extraction from leaf disease dataset collected at 2 provinces Hau Giang and Dong Thap. Learns the SVM model from combination SIFT and BoVW, Color, HOG, GIST and ResNet to found difference the features of each kind of leaf disease. The numerical test result real dataset with 711 images of 5 kind, inside, 4 kind is leaf disease and 1 kind is leaf healthy popular appearance in garden houses show that our SVM model achieves an accuracy 90.84%, more than when comparison with K-Nearest Neighbors (KNN) model achieves an accuracy 42.95% only.

Keyword—*Image classification, classification of orange leaf disease, feature extraction methods, SIFT and BoVW, Color, HOG, GIST, ResNet, Support Vector Machines (SVM)*

Chương 1

Giới thiệu

1.1 Đặt vấn đề

Nhắc đến cái tên Đồng bằng sông Cửu Long (ĐBSCL), không thể không nhắc đến sông Mê Kông, một trong những con sông lớn nhất trên Thế giới bắt nguồn từ cao nguyên Thanh Tạng chảy qua Trung Quốc, Lào, Myanma, Thái Lan, Campuchia và chảy vào Việt Nam chia thành hai sông là sông Tiền (Tiền Giang) và sông Hậu (Hậu Giang) rồi tỏa ra thành chín nhánh đổ ra biển (Biển Đông) thành chín cửa: Tiểu, Đại, Ba Lai, Hàm Luông, Cổ Chiên, Cung Hầu, Định An, Ba Thắc và Trần Đề. Chín nhánh này như chín con rồng uốn lượn nén ở Việt Nam sông Mê Kông được đặt tên là sông Cửu Long. Qua nhiều năm, hai cửa sông Ba Thắc và Trần Đề bị bùn đất bồi lấp và biến mất, do đó sông Cửu Long giờ chỉ còn bảy cửa.

Đồng bằng sông Cửu Long gồm thành phố Cần Thơ (được coi là thủ phủ miền Tây) và 12 tỉnh Long An, Tiền Giang, Bến Tre, Vĩnh Long, Trà Vinh, Hậu Giang, Sóc Trăng, Đồng Tháp, An Giang, Kiên Giang, Bạc Liêu và Cà Mau với tổng diện tích 41,000 km² và tổng dân số là 20 triệu dân. Theo thống kê từ Cục trồng trọt (tháng 03/2019) tổng diện tích trồng cây ăn trái ở các tỉnh phía Nam đạt hơn 569,000 ha (chiếm 60% diện tích cả nước), sản lượng ước đạt 6,6 triệu tấn (chiếm 67% sản lượng cả nước), thống kê 14 loại cây ăn trái được trồng hơn 10,000 ha thì trong đó cam, bưởi đạt diện tích 44,000 ha. Đặc biệt, cam là một trong những loại quả được sử dụng nhiều nhất trên Thế giới được trồng nhiều ở các tỉnh Đồng Tháp, Hậu Giang và thành phố Cần Thơ, vì nó giàu chất chống oxy hóa và chất phytocchemical. Theo các nhà khoa học Anh: “Bình quân trong một trái cam có chứa khoảng 170 mg phytocchemicals bao gồm các chất dưỡng da và chống lão hóa”. Giá trị dinh dưỡng trong mỗi quả cam gồm: Mỗi 100 gr quả cam có chứa 87,6 g nước, 1.104 microgram Carotene – một loại vitamin chống oxy hóa, 30 mg vitamin C, 10,9 g chất tinh bột, 93 mg kali, 26 mg canxi, 9 mg magnesium, 0,3 g chất xơ, 4,5 mg natri, 7 mg Chromium, 20 mg phốt pho, 0,32 mg sắt và giá trị năng lượng là 48 kcal [1].

Theo đó, tình hình dịch bệnh trên cây ăn trái xuất hiện, gây thiệt hại nặng nề cho các chủ nhà vườn. Cụ thể tại tỉnh Đồng Tháp, theo trang Đồng Tháp Online (07/2018) “Theo

thống kê của Phòng Nông nghiệp và Phát triển nông thôn huyện Lai Vung, dịch bệnh vàng lá thối rễ xuất hiện trên cây có múi từ năm 2016 đến nay, đã gây thiệt hại 260 ha diện tích vườn cam, quýt tỷ lệ thiệt hại 50 – 80%” [2]. Cùng thời điểm, trang Vĩnh Long Online đưa tin “Ở các tỉnh Vĩnh Long, Trà Vinh, Hậu Giang, v.v. số lượng vườn cây có múi bị nhiễm bệnh khá lớn. Theo các ngành chức năng tỉnh Hậu Giang, toàn tỉnh có hàng ngàn hécta vườn cây có múi bị bệnh vàng lá (chủ yếu là cam sành); loại bệnh này giống như bị ung thư nên dù nông dân chữa trị nhiều cách vẫn không khỏi được” [3].

Theo Thông báo tình hình sinh vật gây hại chủ yếu 7 ngày của Cục bảo vệ thực vật (từ ngày 09-15/03/2019) ở cây ăn quả có múi, “Bệnh Greening (vàng lá gân xanh) diện tích nhiễm bệnh 2.055 ha (tăng 105 ha so với kỳ trước, giảm 642 ha so với cùng kỳ năm trước), nặng 87 ha. Tập trung chủ yếu tại các tỉnh Tiền Giang, Hậu Giang, Vĩnh Long, Kiên Giang, Bình Phước, Nghệ An.” cho thấy tình hình sinh vật gây hại đang diễn biến phức tạp và đa dạng gây khó khăn trong quá trình nhận dạng bằng mắt thường dẫn đến tình trạng nhận dạng nhầm hoặc khó nhận dạng ở thời gian đầu phát bệnh cho các chủ nhà vườn.

Vì vậy, tìm một phương pháp nhanh, tự động, ít tốn kém và chính xác để phân loại các trường hợp bệnh trên cây ăn quả (cam) có ý nghĩa thực tế rất lớn với nền kinh tế nước nhà.

1.2 Những nghiên cứu liên quan

Các nghiên cứu về xử lý ảnh trước đây được dùng như một cơ chế phát hiện bệnh [4][5]. Từ cuối những năm 1970, công nghệ xử lý hình ảnh dựa trên máy tính được áp dụng trong nghiên cứu kỹ thuật nông nghiệp trở nên phổ biến [4][6].

Trong bài báo [7] trình bày nghiên cứu trước đây trong phát hiện và phân loại bệnh trên lá. Chuyển đổi ảnh từ không gian ảnh màu (RGB) sang không gian màu độc lập với thiết bị (CIELAB). Sử dụng kỹ thuật gom cụm K-means [Diday et al., 1979] để phân đoạn ảnh. Tính toán các đặc trưng kết cấu từ ảnh đã được phân đoạn. Trích các đặc trưng thông qua mạng thần kinh được đào tạo trước (pre-trained neural network). Mô hình mạng thần kinh được dùng để phát hiện và phân loại các bệnh trên lá.

Trong luận văn này, chúng tôi đề xuất đề tài ứng dụng máy học véc-tơ hỗ trợ trong phân loại bệnh trên lá cam sử dụng các đặc trưng SIFT [Lowe, 2004] kết hợp mô hình túi đựng từ trực quan (BoVW), Color [Michael, 1991], HOG [Dalal & Triggs, 2005], GIST [Oliva & Torralba, 2001], ResNet [He et al., 2015] và huấn luyện mô hình máy học SVM [Vapnik, 1995]. Thực hiện rút trích các đặc trưng (được đề xuất) từ ảnh bệnh và huấn luyện mô hình máy học SVM để phân loại. Kết quả thực nghiệm trên tập dữ liệu gồm 711 ảnh của 5 loại, trong đó 4 loại lá bệnh và 1 loại lá khỏe lần lượt là bệnh Ghé nhám, Rầy phấn trắng, Vàng lá gân xanh, Vàng lá thối rễ và Lá khỏe được thu thập tại tỉnh Đồng Tháp và tỉnh Hậu Giang.

1.3 Mục tiêu của đề tài

Mục tiêu của đề tài *Ứng dụng máy học véc-tơ hỗ trợ trong phân loại bệnh trên lá cam* bao gồm ba mục tiêu.

- Thứ nhất, xây dựng được một mô hình phân loại.
- Thứ hai, độ chính xác chấp nhận được.
- Cuối cùng là thời gian phân loại không quá lâu.

1.4 Đối tượng và phạm vi nghiên cứu

1.4.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là các hình ảnh lá cam bệnh được chụp từ camera:

- Thời gian chụp khi lá cam bắt đầu nhiễm bệnh hoặc đang bị nhiễm bệnh.
- Ảnh chụp trực diện lá cam bị bệnh.

1.4.2 Phạm vi nghiên cứu

Trong đề tài, chúng tôi tìm hiểu về đặc điểm màu sắc, hình dạng của bốn loại bệnh (Ghẻ nhám, Rầy phấn trắng, Vàng lá gân xanh và Vàng lá thối rẽ). Phản kỹ thuật, tìm hiểu các phương pháp trích chọn đặc trưng SIFT kết hợp túi đựng từ trực quan (BoVW), Color, HOG, GIST, ResNet và mô hình máy học SVM.

1.5 Phương pháp nghiên cứu

1.5.1 Nghiên cứu lý thuyết

Chúng tôi nghiên cứu nguyên nhân cơ bản, cũng như hiện tượng một số loại bệnh trên lá cam. Các mô hình màu, mô hình biểu diễn ảnh, phương pháp trích chọn đặc trưng ảnh. Mô hình máy học SVM được chúng tôi nghiên cứu nhằm mục đích xây dựng mô hình phân loại bệnh.

1.5.2 Thực nghiệm

Bắt đầu từ việc thu thập hình ảnh bệnh trên lá cam, sau đó xây dựng các mô đun trích chọn đặc trưng, nối các véc-tơ đặc trưng của từng mô đun thành một véc-tơ có 103124 thành phần và cuối cùng là huấn luyện mô hình phân lớp SVM. Tiếp theo, chúng tôi tiến hành chạy thực nghiệm, so sánh kết quả với mô hình máy học KNN và đánh giá kết quả của hệ thống.

1.6 Nội dung nghiên cứu

Các lý thuyết về các mô hình màu (RGB, HSV); các mô hình biểu diễn ảnh (raster, vector); các kỹ thuật trích đặc trưng (SIFT, Color, HOG, GIST và ResNet); mô hình máy học SVM của thư viện scikit-learn [Cournapeau, 2007]. Ngôn ngữ lập trình Python [Rossum, 1991], thư viện mã nguồn mở OpenCV [Bradski & Kaehler, 2012], thư viện scikit-image [8], thư viện Pyleargist [Grisel, 2009] và thư viện Keras [Chollet, 2015].

1.7 Bố cục luận văn

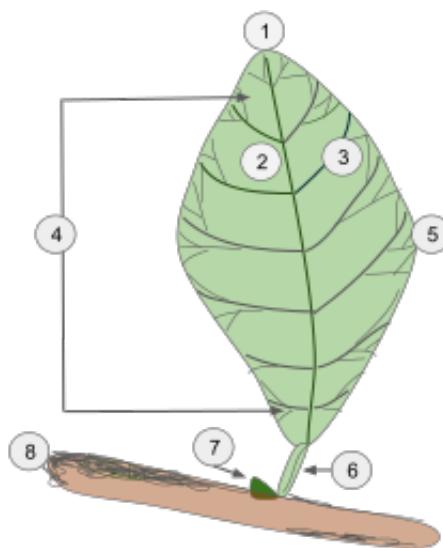
Phần còn lại của luận văn được tổ chức như sau. Chúng tôi trình bày các khái niệm về kỹ thuật được dùng xây dựng hệ thống trong chương 2. Các bước xây dựng một hệ thống nhận dạng và phân loại được trình bày trong chương 3. Kết quả thực nghiệm sẽ được trình bày trong chương 4 trước khi kết luận và hướng phát triển được trình bày trong chương 5.

Chương 2

Cở sở lý thuyết

2.1 Một số loại bệnh trên lá cam

Trong phần này, chúng tôi giới thiệu mô hình của một chiếc lá để có thể phân loại các bệnh chính xác nhờ đó giúp cho mô hình huấn luyện được chính xác hơn. Tiếp theo, chúng tôi sẽ giới thiệu qua các đặc điểm của từng loại bệnh (Ghẻ nhám, Rầy phấn trắng, Vàng lá gân xanh và Vàng lá thối rễ). Mô hình của một chiếc lá gồm 8 thành phần: (1) Chóp lá, (2) Gân chính, (3) Gân phụ, (4) Phiến lá, (5) Rìa lá, (6) Cuốn lá, (7) Chồi và (8) Thân, được mô tả trong hình 2.1.



Hình 2.1: Mô hình của một chiếc lá (Nguồn: wikipedia.org)

Trong đề tài, chúng tôi dùng các ảnh chụp bị nhiễm bốn loại bệnh:

- *Ghẻ nhám*: Nguyên nhân, do nấm. Hiện tượng, vết bệnh đầu tiên là những chấm nhỏ mất màu, trong mờ nhô ra ở mặt dưới lá, sau đó biến thành các mụn nhỏ như mụn ghẻ, màu nâu (hình 2.2).



Hình 2.2: Bệnh Ghẻ nhám

- *Rầy phấn trắng*: Nguyên nhân, do rầy đẻ trứng. Hiện tượng, rầy phấn trắng đẻ trứng ở mặt dưới lá, rải rác thành vòng tròn hình xoắn ốc và được phủ bởi những lông sáp trắng mịn, mỗi vòng xoắn có khoảng 15 đến 25 trứng (hình 2.3).



Hình 2.3: Bệnh Rầy phấn trắng

- *Vàng lá gân xanh*: Nguyên nhân, do vi khuẩn. Hiện tượng, phiến lá hép, nhọn như hình tai thỏ, khoảng các lá ngắn, lá vàng nhưng gân chính và gân phụ vẫn xanh (hình 2.4).



Hình 2.4: Bệnh Vàng lá gân xanh

- *Vàng lá thối rẽ*: Nguyên nhân, tuyển trùng, rệp sáp đất và nhện hại rẽ. Hiện tượng, gân lá màu vàng nhạt, phiến lá ngã vàng cam và dễ rụng (hình 2.5).



Hình 2.5: Bệnh Vàng lá thối rẽ

2.2 Các mô hình màu

2.2.1 Giới thiệu

Phần này giới thiệu về các mô hình màu, vậy màu sắc được định nghĩa như thế nào? Màu sắc là kết quả tương quan giữa ánh sáng (vật lý) trong môi trường và hệ thống thị giác của con người.

Có hai khái niệm cần phân biệt rõ mô hình màu và không gian màu [9]:

- Mô hình màu là một mô hình toán học trừu tượng mô tả cách màu sắc có thể được biểu thị bằng các bộ số. Ví dụ, RGB, HSV.
- Không gian màu là một tổ chức đặc biệt của màu sắc, có thể có các không gian màu khác nhau cho một mô hình màu cụ thể. Ví dụ, sRGB và Adobe RGB là không gian màu của mô hình màu RGB.

2.2.2 Mô hình màu RGB

Mô hình màu RGB [RCA, 1953] là một trong những phương pháp biểu diễn màu được sử dụng rộng rãi nhất trong đồ họa máy tính. Nó sử dụng hệ thống tọa độ với ba màu (phốt pho) chính, trong khuôn khổ tài liệu này chúng tôi sẽ dùng “kênh màu” thay cho màu. Mỗi kênh màu chính có thể lấy giá trị cường độ từ thấp đến cao (0 – 1). Trộn ba kênh màu cơ bản này ở các cường độ khác nhau sẽ tạo ra màu sắc khác nhau. Tập hợp tất cả các màu nhận được từ sự kết hợp tuyến tính của các kênh màu đỏ, lục và lam tạo thành mô hình màu RGB hình khối .

Trong khối màu RGB như hình 2.6 góc tọa độ tương ứng với màu đen, trong khi góc chéo đối diện là màu trắng. Đường chéo kết nối màu đen và trắng là màu xám. Mô hình

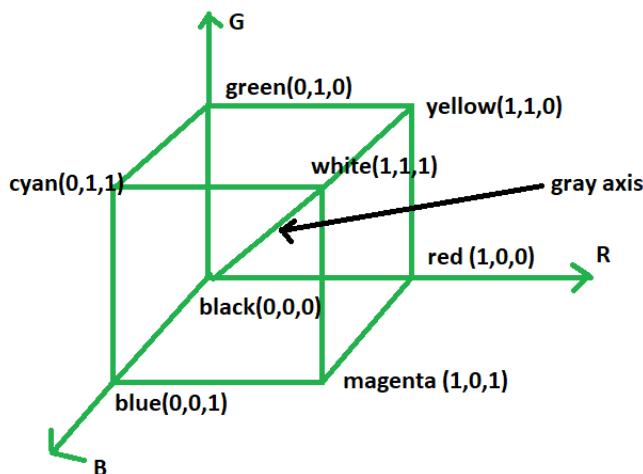
màu RGB được dùng cho việc hiển thị lên màu sắc trong các ống tia âm cực, màn hình LCD [Heilmeyer, 1986], LED [Holonyack, 1962] và màn hình plasma [Bitzer và Slottow, 1964]. Mỗi điểm ảnh trong màn hình có thể được lưu trong bộ nhớ máy tính như các giá trị độc lập, nó được biến đổi thành các cường độ sáng và gửi đến màn hình. Mỗi điểm ảnh được biểu diễn bằng 3 bytes. Tương ứng 1 byte (1 byte = 8 bits) cho mỗi kênh màu (đỏ, xanh lá cây, xanh dương). Do mã máy là dạng nhị phân (0, 1) nên ta dễ dàng tính được số lượng màu được hiển thị lên một màn hình LCD.

$$3(\text{bytes}) \times 8(\text{bits}) = 24(\text{bits})$$

Số lượng màu được hiển thị trên một màn hình LCD:

$$2^{24} = 16.777.216(\text{color})$$

Khi biểu diễn bằng hệ số thập phân (decimal) giá trị nằm trong phạm vi 0 đến 255. Ví dụ, RGB(0,0,0) hoặc rgb(255,255,255). Biểu diễn bằng hệ số thập lục phân (hexadecimal) sẽ bắt đầu bằng dấu “#” chỉ định các giá trị đỏ, lục và lam bằng cách sử dụng 1 đến 4 chữ số hex. Ví dụ, #RGB, #rrrrggggbbbb, sử dụng “#ffff” để biểu diễn màu tối đa hoặc “0” là không có màu.



Hình 2.6: Mô hình màu RGB (Nguồn: wikipedia.org)

2.2.2.1 Ưu điểm

Hầu như mọi ứng dụng phổ biến đều tương thích với mô hình màu RGB như Microsoft Office, Adobe Creative Suite (Photoshop, Illustrator, InDesign, v.v) và các trình soạn thảo khác.

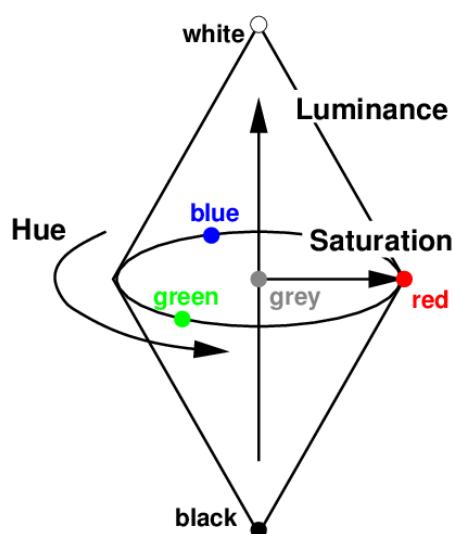
2.2.2.2 Hạn chế

Một trong những hạn chế chính của mô hình màu RGB là nó không thực sự cho ra màu sắc tốt khi in, việc này dẫn đến sự sai lệch về màu sắc trong các tài liệu được in từ Microsoft Office. Các thiết bị sử dụng loại màn hình LED như Smartphone thì tọa độ màu không được nhất quán. Thậm chí trên cả màn hình TV [Farnsworth, 1927] và một số loại màn hình khác. Vì vậy, giữa các thiết bị khác nhau sẽ có sự sai lệch về màu sắc là khác nhau.

2.2.3 Mô hình màu HSV

Ở phần trước 2.2.2 chúng tôi đã nói qua mô hình màu RGB. Không giống như RGB, mô hình màu HSV [Smith, 1978] gần hơn với cách con người cảm nhận màu sắc (bằng mắt).

Mô hình màu HSV gồm có ba thành phần chính Hue, Saturation và Value như hình 2.7. *Hue* là thành phần màu được biểu thị từ 0° đến 360° . Đỏ ($0^\circ - 60^\circ$), Vàng ($61^\circ - 120^\circ$), Lục ($121^\circ - 180^\circ$), Lục lam ($181^\circ - 240^\circ$), Lam ($241^\circ - 300^\circ$) và Đỏ ($301^\circ - 360^\circ$). *Saturation* mô tả lượng màu xám (đạm nhạt) của một màu cụ thể, có giá trị 0 đến 100%, giảm giá trị này về 0 sẽ tạo ra nhiều xám hơn và tạo hiệu ứng mờ dần. Đôi khi, độ được biểu diễn dưới phạm vi 0 đến 1, tương ứng 0 (xám) và 1 (màu chính). *Value* mô tả độ sáng hay cường độ của màu, có giá trị từ 0 đến 100%, trong đó 0 (đen hoàn toàn) và 100% (sáng nhất). Một số bộ chọn màu như trong Adobe Photoshop sử dụng thuật ngữ HSB, đó là sự thay thế *B* (brightness) cho *V* (value).



Hình 2.7: Mô hình màu HSV (Nguồn: researchgate.net)

2.2.3.1 Ưu điểm

Mô hình màu HSV có một số ưu điểm tốt hơn so với các mô hình màu khác. Ví dụ, chọn một màu duy nhất trong mô hình màu ABC, nó bỏ qua phần lớn sự phức tạp của màu,

nó liên quan về mặt nhận thức đối với tốc độ tính toán, khi các màu này trở nên phức tạp sẽ khó khăn trong việc tính toán. Trong một số ứng dụng đồ họa đều dựa trên mô hình màu RGB. Tuy nhiên, đều cho phép chuyển sang mô hình màu HSV giúp linh hoạt hơn trong lựa chọn màu sản phẩm. Các nhà thiết kế sử dụng mô hình màu HSV khi chọn màu cho sơn hoặc mực vì nó được thể hiện tốt trong mắt người dùng so với mô hình màu RGB.

2.2.3.2 Hạn chế

Saturation và Value đã bị giới hạn, do đó thang đo saturation cũng có phạm vi rộng lớn để chứa luôn value. Tương tự RGB, mô hình màu HSV có sự phụ thuộc vào thiết bị hiển thị. Ví dụ: nó có thể chuyển từ màu trắng sang màu lục (xanh lá cây), đó là nhờ sự kết hợp của saturation và value. Màu vàng bão hòa và màu xanh bão hòa có thể được gán cùng một value, nhưng có sự khác biệt lớn về màu sắc đối các hệ thống trong việc kiểm soát giao diện.

2.3 Các mô hình biểu diễn ảnh

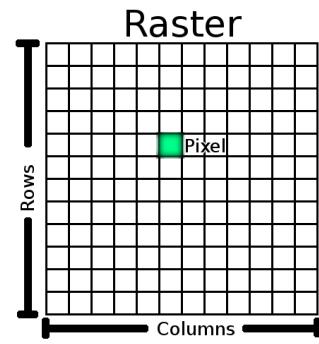
2.3.1 Giới thiệu

Ảnh trên máy tính là kết quả thu nhận theo các phương pháp số hóa được nhúng trong các thiết bị kỹ thuật khác nhau camera, scanner, v.v.. Quá trình lưu trữ gồm hai mục đích tiết kiệm bộ nhớ và giảm thời gian xử lý. Việc lưu trữ thông tin trong bộ nhớ có ảnh hưởng rất lớn đến việc hiển thị và in ấn. Việc lưu trữ ảnh được xem như là một tập hợp các điểm ảnh (pixel) cùng kích thước, nếu sử dụng càng nhiều điểm ảnh thì bức ảnh càng đẹp, càng mịn và thể hiện rõ các chi tiết trong ảnh, người ta gọi đặc điểm này là độ phân giải. Việc lựa chọn độ phân giải thích hợp phụ thuộc vào nhu cầu sử dụng và đặc trưng của mỗi ảnh cụ thể. Trên cơ sở đó, ảnh thường được biểu diễn theo hai mô hình cơ bản là mô hình raster và mô hình vector.

2.3.2 Mô hình raster

Đây là cách biểu diễn thông dụng nhất hiện nay, ảnh được biểu diễn dưới dạng ma trận các điểm ảnh như hình 2.8. Thường được thu nhận qua các thiết bị camera, scanner, v.v. tùy theo yêu cầu mà mỗi điểm ảnh được biểu diễn bằng một hay nhiều bit.

Một trong những hướng nghiên cứu cơ bản trên mô hình biểu diễn này là các kỹ thuật nén ảnh, các kỹ thuật nén ảnh được chia hai khuynh hướng nén bảo toàn và nén không bảo toàn thông tin. Nếu nén bảo toàn có khả năng phục hồi bảo toàn dữ liệu ban đầu, thì nén không bảo toàn chỉ có khả năng phục hồi trong một sai số cho phép nào đó. Theo mô hình này có thể dễ dàng thấy một số ảnh được biểu diễn với phần mở rộng như BMP, TIF, GIF, v.v. Mô hình raster thuận lợi cho việc hiển thị và in ấn.

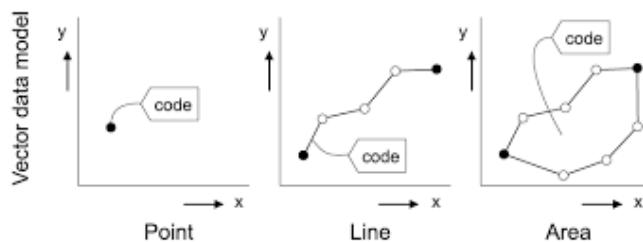


Hình 2.8: Mô hình biểu diễn ảnh raster (Nguồn: docs.qgis.org)

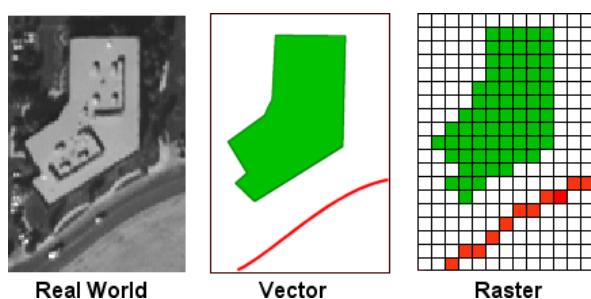
2.3.3 Mô hình vector

Mô hình vector được sử dụng cho việc biểu diễn ảnh ngoài mục đích tiết kiệm không gian lưu trữ, dễ dàng cho hiển thị, in ấn, sao chép, di chuyển và tìm kiếm.

Trong mô hình này, sử dụng hướng giữa các vector của điểm ảnh lân cận như hình 2.9 để mã hóa và tái tạo ảnh. Ban đầu ảnh vector được thu nhận trực tiếp từ các thiết bị kỹ thuật số digital hoặc được chuyển đổi từ ảnh raster thông qua các chương trình số hóa. Do công nghệ phần cứng đa phần chỉ hỗ trợ cho ảnh raster nên các nghiên cứu về biểu diễn ảnh vector đều tập trung từ chuyển đổi ảnh raster. Có thể thấy sự khác biệt giữa hai cách biểu diễn ảnh trong thực tế ở hình 2.10.



Hình 2.9: Mô hình biểu diễn ảnh vector (Nguồn: deltauniv.edu.eg)

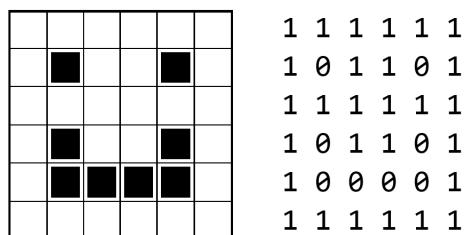


Hình 2.10: Mô hình vector và raster trong thực tế (Nguồn: geography.hunter.cuny.edu)

2.3.4 Phân loại ảnh

2.3.4.1 Ảnh nhị phân

Như tên gọi của nó, ảnh nhị phân (1 bit) một điểm ảnh chỉ chứa một giá trị là 0 hoặc 1. Trong đó, giá trị 0 đại diện cho màu đen và giá trị 1 đại diện cho màu trắng như hình 2.11, còn gọi là ảnh đơn sắc. Ảnh thu được chỉ bao gồm màu trắng và đen nên còn được gọi là ảnh trắng đen, nó có định dạng PBM (bitmap). Tùy vào độ phân giải của ảnh, có thể biểu diễn bằng một ma trận có kích thước khác nhau, hình 2.11 được biểu diễn bằng một ma trận kích thước 35×35 .



Hình 2.11: Ảnh nhị phân (Nguồn: thimbleprojects.org)

2.3.4.2 Ảnh mức xám

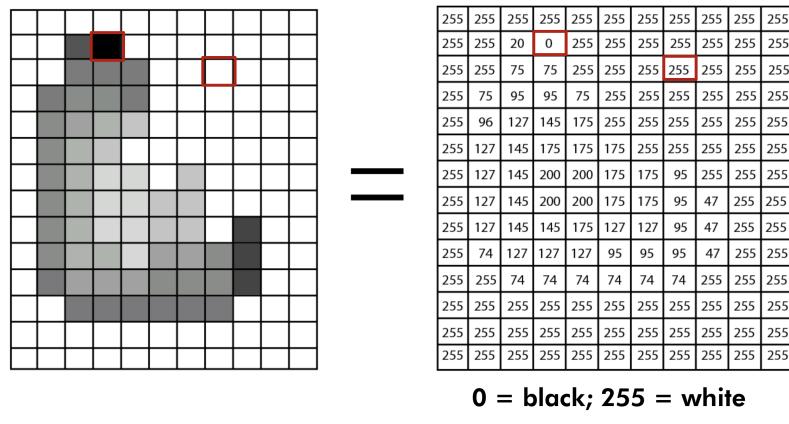
Khác với ảnh nhị phân được biểu diễn bằng 1 bit, ảnh mức xám hay ảnh xám (gray-level) được biểu diễn 8 bits trên mỗi điểm ảnh giá trị nằm trong phạm vi 0 đến 255. Trong đó, 0 (màu đen), 127 (màu xám) và 255 (màu trắng).

Định dạng ban đầu được sử bởi các mô hình của hệ điều hành UNIX [Thompson, Ritchie và McIlroy, 1969] và màu Macintosh [Apple, 1993] đầu tiên. Ảnh mức xám có định dạng PGM khác với ảnh nhị phân (PBM), định dạng này không hỗ trợ mặc định trên hệ điều hành Windows. Muốn xem được trên Windows cần phải có trình xem ảnh hoặc bộ công cụ xử lý ảnh như MATLAB [MathWorks, 1984]. Ảnh mức xám được biểu diễn bằng mô hình raster với một ma trận hai chiều, có giá trị nằm trong khoảng 0 đến 255 (hình 2.12).

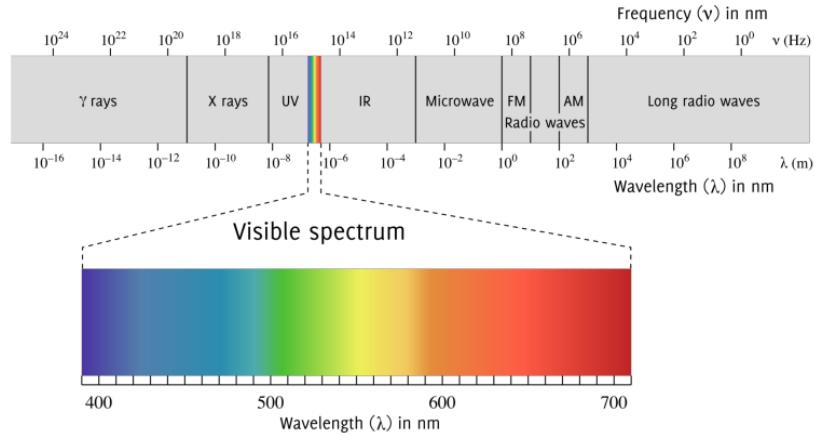
2.3.4.3 Ảnh màu

Ảnh màu là sự phát triển của ảnh mức xám. Có thể biết được nhiều thông tin hơn dựa vào màu sắc, các thông tin này có thể đơn giản hóa dùng cho việc phân tích hình ảnh [7]. Ví dụ, xác định đối tượng hoặc trích xuất đặc trưng của một ảnh dựa trên màu sắc. Màu sắc được xác định dựa vào bước sóng có giá trị lớn hơn các bước sóng còn lại. Màu sắc nhìn thấy được trong khoảng 400nm (tím) và 700 (đỏ) trên phổ điện từ như hình 2.13.

Ban đầu ảnh được biểu diễn bằng 16 bits trên mỗi điểm ảnh, tương ứng với 65,356 các màu khác nhau, còn gọi là định dạng màu cao. Giá trị 16 bits được chia thành ba kênh màu (đỏ, xanh lá cây, xanh dương), số lượng bit được chia tương ứng 5 bits, 6 bits và 5 bits.

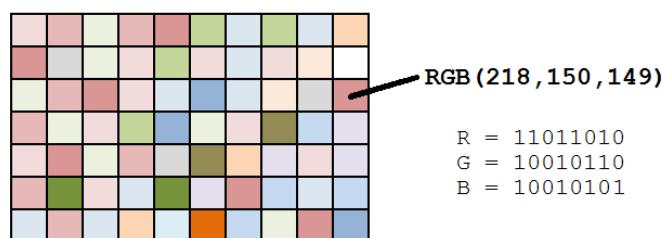


Hình 2.12: Ảnh mức xám (Nguồn: edtech.engineering.utoronto.ca)



Hình 2.13: Quang phổ nhìn thấy ở người (Nguồn: inf.edu.ac.uk)

Như chúng tôi đã giới thiệu ở phần 2.2.2 mô hình màu RGB được biểu diễn bằng 24 bits trên mỗi điểm ảnh. Tương tự như cách biểu diễn 16 bits, giá trị 24 bits được chia tương ứng cho ba kênh màu RGB. Có thể thấy trong hình 2.14 mỗi kênh màu được biểu diễn bằng một giá trị 8 bits. Đối với ảnh biểu diễn bằng giá trị 24 bits được sử dụng phổ biến nhất hiện nay, nó có định dạng PPM.



Hình 2.14: Ảnh màu 24 bits (Nguồn: towardsdatascience.com)

2.4 Các phương pháp trích chọn đặc trưng

2.4.1 Giới thiệu

Trích chọn đặc trưng là cơ sở của tra cứu ảnh dựa vào nội dung. Theo nghĩa rộng, các đặc trưng có thể bao gồm cả các đặc trưng dựa vào văn bản và các đặc trưng trực quan như màu sắc, kết cấu và hình dạng. Trích chọn đặc trưng giúp giảm lượng dữ liệu phải xử lý, trong khi vẫn mô tả chính xác toàn bộ dữ liệu gốc. Quá trình trích chọn đặc trưng là hữu ích, vì nó làm giảm lượng dữ liệu dư thừa nhưng không làm mất thông tin quan trọng hoặc có liên quan giúp tăng thời gian nhận dạng. Tùy vào mục đích xây dựng mô hình nhận dạng mà chọn các kỹ thuật trích chọn đặc trưng sao cho phù hợp.

2.4.2 Trích chọn đặc trưng SIFT

Đặc trưng SIFT (Scale-Invariant Feature Transform) [Lowe, 2004] là giải thuật trong nghành Khoa học máy tính, dùng để nhận dạng và miêu tả những điểm đặc trưng cục bộ trong ảnh, được ứng dụng rộng rãi trong nhận dạng đối tượng (object recognition), mô hình hóa 3D (3D modeling), v.v.. Đặc trưng cục bộ SIFT không bị thay đổi trước những biến đổi tỉ lệ ảnh, tịnh tiến, phép quay, không bị thay đổi một phần đối với phép biến đổi hình học affine (thay đổi góc nhìn) và mạnh với những thay đổi tỷ lệ về độ sáng, nhiễu và sự che khuất.

Điểm hấp dẫn (keypoint) là điểm ảnh hấp dẫn trên ảnh, có nghĩa là điểm đó có các đặc trưng bất biến. Trong bài báo khoa học [10] giải thuật SIFT trải qua bốn giai đoạn tính toán chính:

1. Phát hiện điểm cực trị.
2. Định vị điểm hấp dẫn.
3. Xác định hướng cho điểm hấp dẫn.
4. Mô tả các điểm hấp dẫn.

2.4.2.1 Phát hiện điểm cực trị

Các điểm hấp dẫn với đặc trưng SIFT tương thích với các cực trị cục bộ của bộ lọc difference-of-Gaussian (DoG) ở các tỉ lệ khác nhau. Định nghĩa không gian tỉ lệ của một ảnh là hàm $L(x, y, k\sigma)$ là kết quả từ việc nhân chập biến tỉ lệ $G(x, y, k\sigma)$ với ảnh đầu vào $I(x, y)$:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \quad (2.1)$$

khi đó, $*$ là phép nhân tích chập giữa x và y và

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{(-x^2+y^2)/2\sigma^2} \quad (2.2)$$

Để phát hiện các điểm hấp dẫn, ta đi tìm cực trị của hàm DoG được định nghĩa:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (2.3)$$

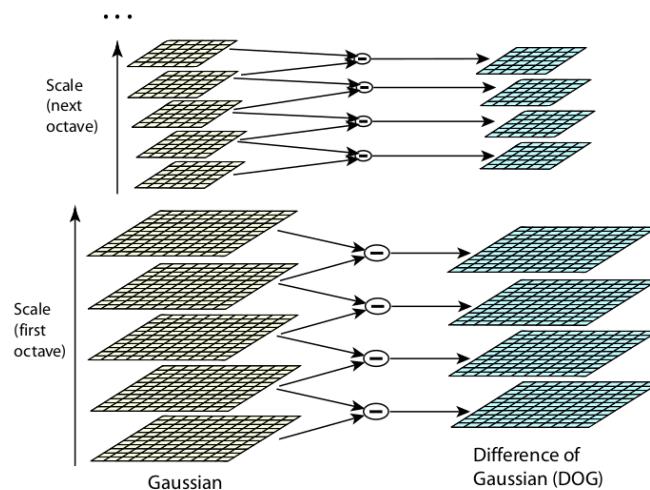
Giá trị của hàm DoG được tính xấp xỉ dựa vào giá trị scale-normalized Laplacian of Gaussian ($\sigma^2 \nabla^2 G$) thông qua các phương trình (2.1), (2.2) và (2.3).

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (2.4)$$

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (2.5)$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \partial^2 G \quad (2.6)$$

Như vậy bước đầu tiên của giải thuật SIFT phát hiện các điểm hấp dẫn với bộ lọc Gaussian ở các tỉ lệ khác nhau và các ảnh DoG từ sự khác nhau của các ảnh kè mờ như hình 2.15.

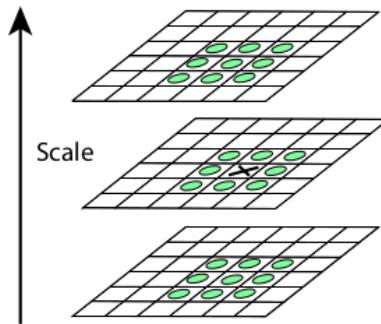


Hình 2.15: Mô phỏng tính toán DoG (Nguồn:[10])

Các ảnh cuộn được nhóm thành các octave (mỗi octave tương ứng với giá trị gấp đôi của σ). Giá trị của k được chọn sao cho số lượng ảnh mờ cho mỗi octave là cố định. Điều này đảm bảo cho số lượng các ảnh DoG cho mỗi octave không thay đổi.

Các điểm hấp dẫn được xác định là các cực đại cực tiểu của ảnh DoG qua các tỉ lệ. Mỗi điểm ảnh trong DoG được so sánh với 8 điểm láng giềng của nó ở cùng tỉ lệ đó và 9 láng giềng kè ở các tỉ lệ ngay trước và sau nó như hình 2.16. Nếu điểm ảnh đó đạt giá trị cực tiểu

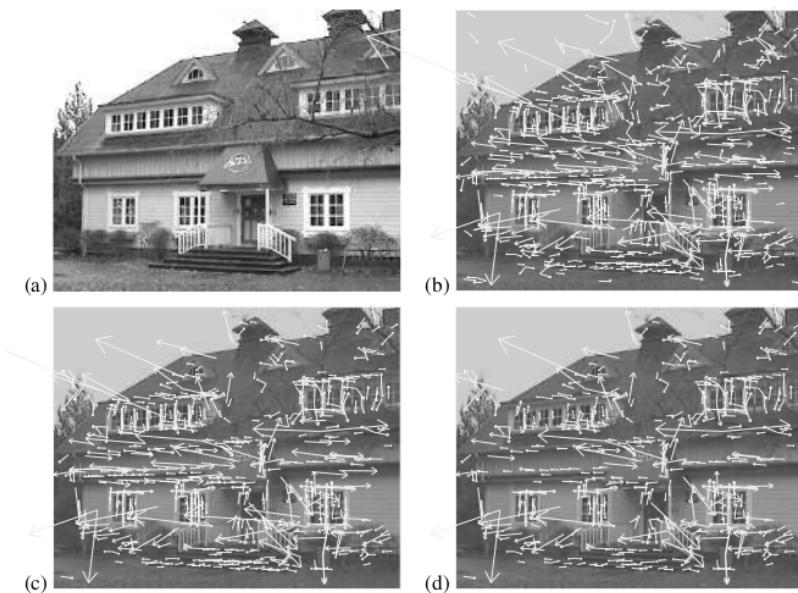
hoặc cực đại thì được chọn làm điểm hấp dẫn ứng viên.



Hình 2.16: Quá trình tìm điểm cực trị trong hàm DoG (Nguồn:[10])

2.4.2.2 Định vị điểm hấp dẫn

Loại bỏ các điểm hấp dẫn có độ tương phản thấp và các điểm hấp dẫn dọc theo các cạnh cũng được loại bỏ do không giữ được tính ổn định khi ảnh bị nhiễu như hình 2.17 (c) sau khi đã chọn được các điểm hấp dẫn ứng viên từ hình 2.17 (b). Các điểm còn lại sẽ được xác định hướng trong hình 2.17 (d).



Hình 2.17: Quá trình lựa chọn các điểm hấp dẫn (Nguồn:[10]). (a) Ảnh gốc. (b) Các điểm hấp dẫn được phát hiện. (c) Sau khi loại bỏ các điểm hấp dẫn có độ tương phản thấp. (d) Sau khi loại bỏ các điểm hấp dẫn dọc theo cạnh.

2.4.2.3 Xác định hướng cho điểm hấp dẫn

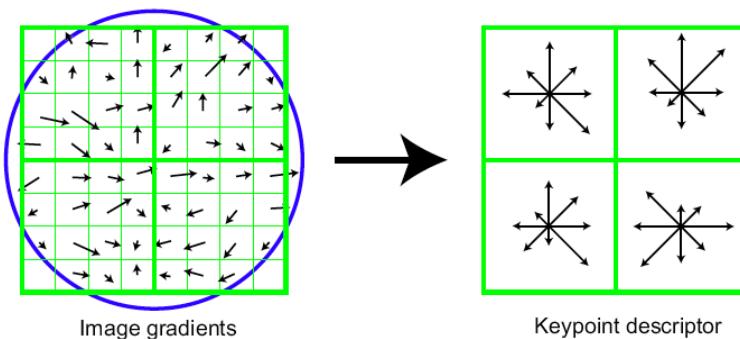
Để xác định hướng cho điểm hấp dẫn, bằng cách tính toán biểu đồ gradient trong vùng lảng giềng của điểm hấp dẫn. Độ lớn và hướng của điểm hấp dẫn được xác định theo công thức.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2.7)$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1)) / (L(x+1,y) - L(x-1,y))) \quad (2.8)$$

2.4.2.4 Mô tả các điểm hấp dẫn

Điểm hấp dẫn sau khi được xác định hướng sẽ được biểu diễn dưới dạng các véc-tơ $4 \times 4 \times 8 = 128$ chiều.



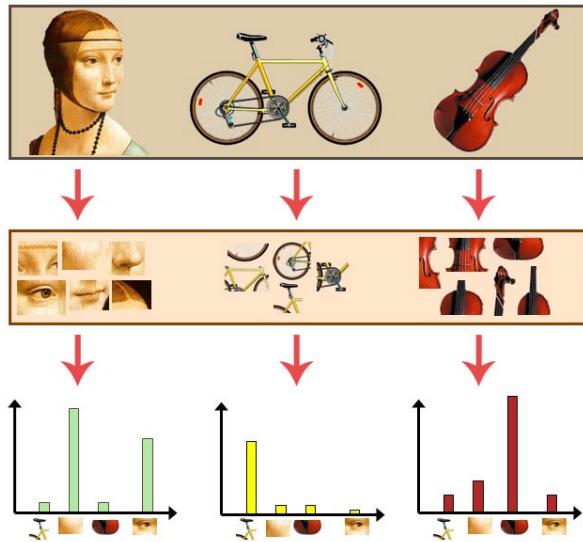
Hình 2.18: Biểu diễn các vector đặc trưng (Nguồn:opencv-python-tutroals)

2.4.2.5 Mô hình túi đựng từ trực quan

Mô hình túi đựng từ trực quan (Bag of Visual Words - BoVW) là một khái niệm quan trọng trong chuyên ngành Khoa học máy tính, dựa trên mô hình túi đựng từ (Bag of Words - BoW) trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). Mô hình túi đựng từ tính toán tần số xuất hiện của từ trong tài liệu, sử dụng một biểu đồ biểu diễn tần số của từ (hình 2.19). Túi đựng từ trực quan cũng được áp dụng nguyên lý trên nhưng được sử dụng trong phân loại nội dung hình ảnh, tức là, thay thế các từ (words) thành các đặc trưng của ảnh.

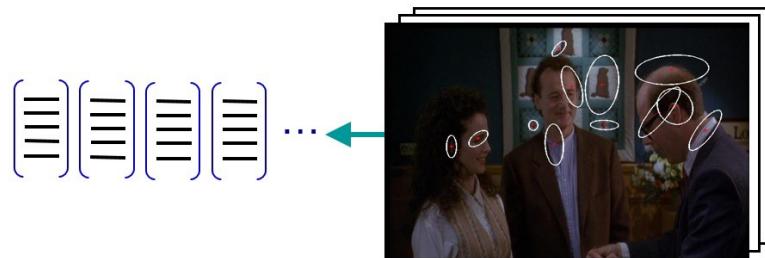
Túi đựng từ trực quan được hiểu như là một tập các đặc trưng của một ảnh được trích chọn từ giải thuật SIFT (keypoint và descriptor) được trình bày ở phần 2.4.2, sử dụng các keypoint và descriptor này để xây dựng từ vựng.

Để xây dựng một mô hình túi đựng từ trực quan. Đầu tiên, phát hiện các đặc trưng và trích xuất các mô tả (hình 2.20) tương ứng bằng một giải thuật trích chọn đặc trưng (SIFT,



Hình 2.19: Biểu đồ của các từ trực quan (Nguồn: towardsdatascience.com)

KAZE, v.v.). Tiến hành gom cụm (hình 2.21) các mô tả bằng các giải thuật gom cụm (K-Means, DBSCAN, v.v.) tâm của từng cụm sẽ được sử dụng như là các từ vựng của từ điển trực quan. Tạo biểu đồ tần số dựa trên các từ vựng và tần số của các từ trong ảnh, các biểu đồ này chính là túi đựng từ trực quan (BoVW).

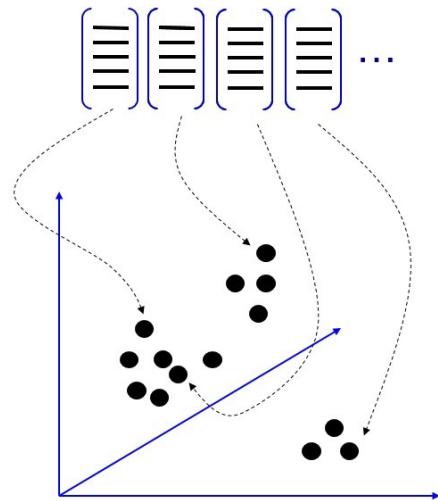


Hình 2.20: Phát hiện đặc trưng và trích xuất mô tả (Nguồn: towardsdatascience.com)

2.4.3 Trích chọn đặc trưng Color

Đặc trưng Color là một trong những đặc trưng phổ biến và được sử dụng nhiều trong các hệ thống tìm kiếm ảnh, được giới thiệu bởi [Michael, 1991]. Đây là phương pháp rút trích đặc trưng có nhiều ưu điểm như đơn giản, tốc độ tìm kiếm tương đối nhanh, tuy nhiên độ chính xác còn hạn chế. Một số lược đồ màu được sử dụng phổ biến như lược đồ màu RGB, HSI và HSI cải tiến. Tuy nhiên, lược đồ màu RGB được sử dụng phổ biến. Trong phần này chúng tôi lựa chọn sử dụng lược đồ màu RGB.

Lược đồ màu của một hình ảnh sẽ đại diện cho sự phân bố của các thành phần màu sắc trong hình ảnh đó. Lược đồ màu sẽ được tính bằng cách rời rạc hóa từng màu trong một



Hình 2.21: Gom cụm các mô tả (Nguồn: towardsdatascience.com)

ảnh, sau đó đếm số điểm ảnh của mỗi màu. Khi mà số lượng màu là có hạn, thuận tiện hơn là cách chuyển đổi ba 3 kênh màu (đỏ, xanh lá cây, xanh dương) thành một biến giá trị duy nhất. Một cách khác để tính lược đồ màu của ảnh RGB là phân ra làm 3 lược đồ riêng biệt h_R , h_G và h_B . Khi đó, mỗi lược đồ được tính bằng cách đếm kênh màu tương ứng trong mỗi điểm ảnh.

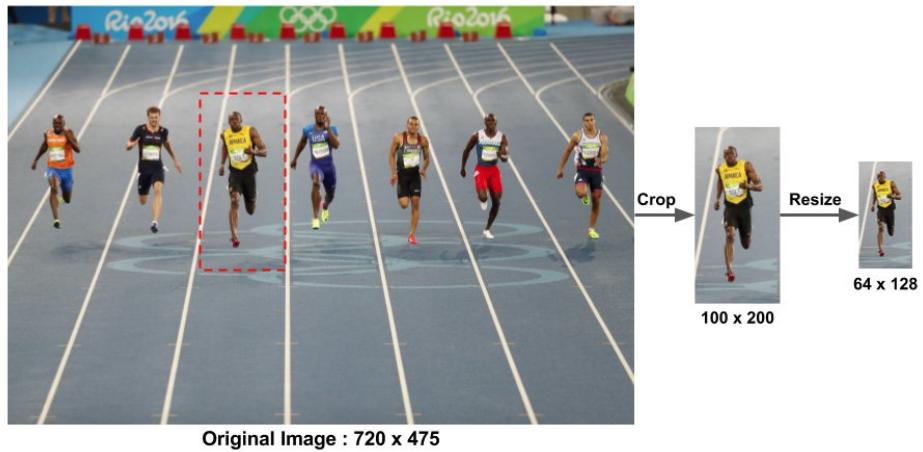
2.4.4 Trích chọn đặc trưng HOG

Đặc trưng HOG (Histogram of Oriented Gradient) [11] là một loại mô tả đặc trưng (feature descriptor) 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. Đặc trưng HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh. Một mô tả đặc trưng chuyển đổi một ảnh có kích thước $w \times h \times 3$ (channel) nhận được một vector đặc trưng có độ dài n .

Véc-tơ đặc trưng của thuật toán này khi được đưa vào mô hình máy học SVM sẽ cho ra kết quả tốt [12]. Thực hiện 5 bước cơ bản để có thể xây dựng một véc-tơ đặc trưng HOG.

2.4.4.1 Tiết xử lý

Chuẩn hóa tỉ lệ các ảnh đầu vào theo tỉ lệ (1:2). Ví dụ, 100×200 , 128×256 , như hình 2.22 một ảnh với kích thước 720×475 , chọn một đối tượng trong ảnh, tiến hành cắt đối tượng với kích thước 100×200 và resize ảnh thành kích thước 64×128 để tính toán mô tả đặc trưng HOG.



Hình 2.22: Tiền xử lý ảnh trong HOG (Nguồn:[12])

2.4.4.2 Tính toán ảnh gradient

Để nhận được ảnh gradient, thực hiện nhân chập (convolution) ảnh gốc I với hai bộ lọc ảnh d_x (vertical gradients) và d_y (horizontal gradients).

$$g_x = I * d_x \quad (2.9)$$

$$g_y = I * d_y \quad (2.10)$$

Tiếp tục tính toán cường độ và hướng gradient của ảnh:

$$\begin{aligned} g &= \sqrt{g_x^2 + g_y^2} \\ \theta &= \arctan \frac{g_y}{g_x} \end{aligned} \quad (2.11)$$

Hình 2.23 minh họa công thức (2.9), (2.10) và (2.11).

2.4.4.3 Tính toán biểu đồ gradient các ô

Để tính toán véc-tơ đặc trưng cho từng ô (cell), chia hình ảnh thành các khối (block), mỗi khối (block) lại được chia thành các ô (cell) nhỏ hơn.

$$n_{block} = \left(\frac{w_{image} - w_{block} * w_{cell}}{w_{cell}} + 1 \right) * \left(\frac{h_{image} - h_{block} * h_{cell}}{h_{cell}} + 1 \right) \quad (2.12)$$

trong đó:

- $w_{image}, w_{block}, w_{cell}$ lần lượt là chiều rộng của ảnh, khối và ô.
- $h_{image}, h_{block}, h_{cell}$ lần lượt là chiều cao của ảnh, khối và ô.



Hình 2.23: Quá trình tính toán ảnh gradient (Nguồn:[12]). (Left) Giá trị tuyệt đối g_x . (Center) Giá trị tuyệt đối g_y , (Right) Giá trị cường độ và hướng gradient

Sau khi xác định số khối và kích thước khối (block), ô (cell) để tính véc-tơ đặc trưng cho từng ô (cell), thực hiện hai bước sau:

- Chia không gian hướng thành b bin (số chiều véc-tơ đặc trưng của ô).
- Rời rạc hóa gốc hướng nghiêng tại mỗi điểm ảnh trong các bin.

Giả sử gốc nghiêng tại vị trí (x, y) có độ lớn $\alpha(x, y)$.

- Trường hợp rời rạc hóa unsigned, $b = 9$.

$$B(x, y) = \text{round}\left(\frac{b * \alpha(x, y)}{\pi}\right) \bmod b \quad (2.13)$$

- Trường hợp rời rạc hóa signed, $b = 18$.

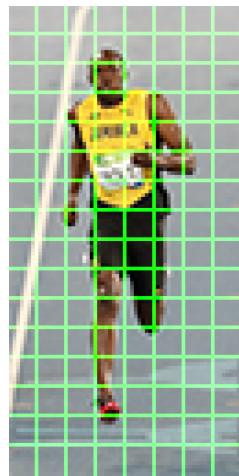
$$B(x, y) = \text{round}\left(\frac{b * \alpha(x, y)}{2\pi}\right) \bmod b \quad (2.14)$$

Giá trị bin được định lượng bởi tổng cường độ biến thiên của các điểm ảnh (pixel) thuộc về bin đó. Tập các véc-tơ của ô sẽ được nối thành véc-tơ đặc trưng khối theo công thức:

$$\text{size}_{block} = n * \text{size}_{cell} \quad (2.15)$$

trong đó:

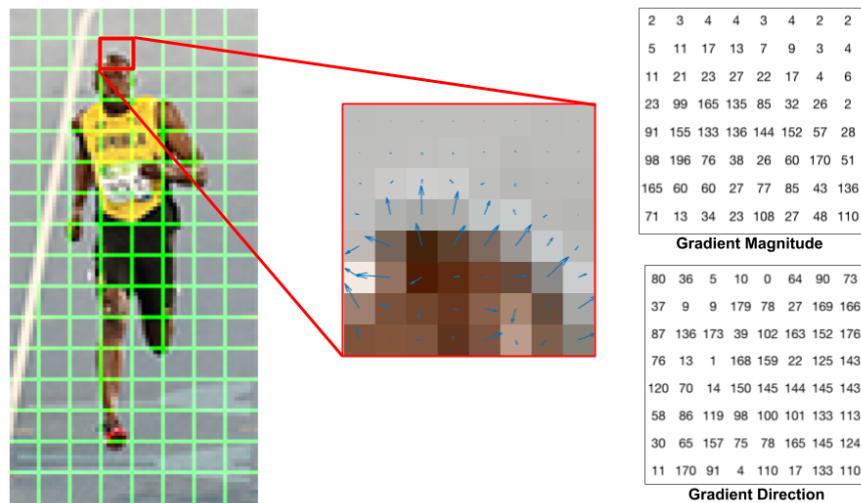
- n là số khối
- size_{cell} là số chiều véc-tơ đặc trưng ô



Hình 2.24: Chia ảnh thành các block 16×16 (Nguồn:[12])

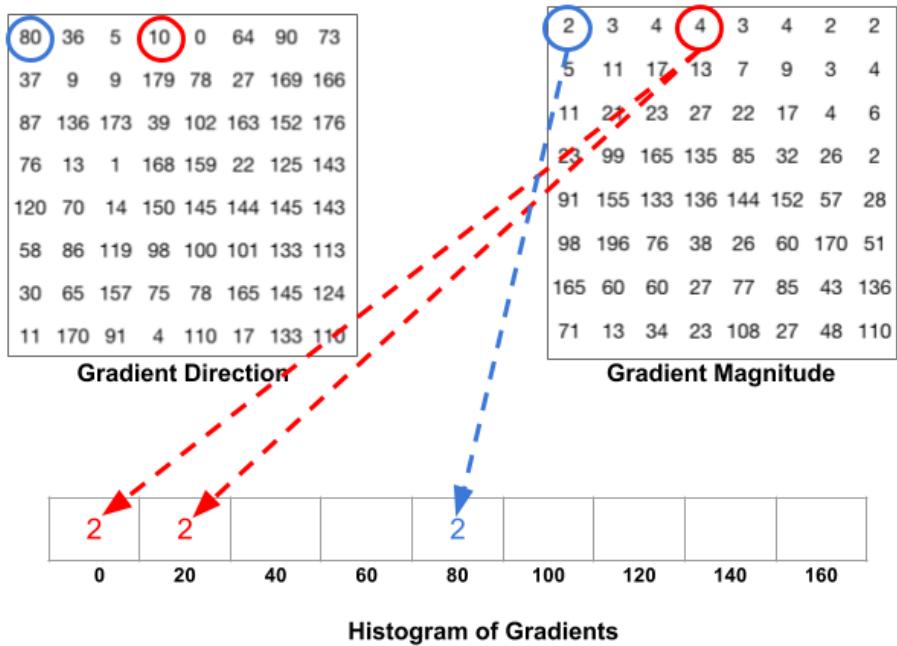
Như hình 2.22 có kích thước 64×128 sẽ được chia thành các khối có kích thước 16×16 . Mỗi khối gồm 4 ô, mỗi ô có kích thước 8×8 như hình 2.24. Tính toán đặc trưng HOG tại mỗi cell ở trung tâm hợp rìa rạc hóa với $b = 9$. Hướng gradient chạy trong khoảng $0 - 180^\circ$, trung bình 20° mỗi bin.

Tại mỗi ô, xây dựng một biểu đồ cường độ gradient bằng cách bình chọn điểm ảnh vào biểu đồ. Trọng số bình chọn của mỗi điểm ảnh phụ thuộc vào hướng và cường độ gradient của điểm ảnh đó, được mô tả trong hình 2.25.



Hình 2.25: Tính toán biểu đồ gradient (Nguồn:[12])

Tại vị trí điểm ảnh đầu tiên, sẽ có hướng 80° và giá trị cường độ là 2, sẽ được thêm vào vị trí bin thứ 5. Tại vị trí pixel thứ 4 có hướng 10° và giá trị cường độ là 4, do không có bin với hướng 10° nên giá trị cường độ được chia 2 và thêm vào bin 1 và 2 trong hình 2.26.



Hình 2.26: Bình chọn điểm ảnh vào biểu đồ (Nguồn:[12])

2.4.4.4 Chuẩn hóa các khối

Do gradient của ảnh nhạy cảm với độ sáng, việc bình thường hóa biểu đồ (histogram) để gradient không bị ảnh hưởng từ những thay đổi ánh sáng trong ảnh. Gọi I_{01} là một vector màu $[128, 64, 32]$, độ dài của véc-tơ I_{01} là $\sqrt{128^2 + 64^2 + 32^2} = 146.64$, chia từng phần tử $[128, 64, 32]$ cho 146.64 nhận được một véc-tơ chuẩn hóa $[0.87, 0.43, 0.22]$. I_{02} là một véc-tơ màu $[256, 128, 64]$ thực hiện các bước tiếp theo tương tự như I_{01} ban đầu ta vẫn nhận được một véc-tơ chuẩn hóa giống I_{01} .

Để chuẩn hóa 1 biểu đồ (histogram) có số bin là $b = 9$, không thể áp dụng như cách chuẩn hóa một véc-tơ (3×1) đã nêu. Chuẩn hóa một khối kích thước 16×16 tương ứng mỗi khối sẽ có 4 biểu đồ (histograms) tức là, tương ứng 4 ô sẽ là 1 biểu đồ, mỗi 1 histogram có kích thước 9×1 nối 4 biểu đồ lại tạo thành một véc-tơ phần tử 36×1 , véc-tơ phần tử này có thể được chuẩn hóa theo cách một véc-tơ 3×1 được chuẩn hóa.

2.4.4.5 Tính toán véc-tơ đặc trưng

Với mỗi ảnh kích thước 64×128 , chia thành các khối 16×16 chồng nhau, sẽ có 7 khối theo chiều ngang và 15 khối theo chiều dọc. Như vậy số khối nhận được là $7 \times 5 = 105$. Mỗi khối gồm 4 ô, khi áp dụng biểu đồ có số bin là $b = 9$ cho mỗi ô, mỗi khối sẽ được đại diện bởi một véc-tơ có kích thước 36×1 . Khi nối của tất cả khối lại với nhau, thu được véc-tơ đặc trưng HOG có kích thước $105 \times 36 \times 1 = 3780 \times 1$.

2.4.5 Trích chọn đặc trưng GIST

Đặc trưng GIST là đặc trưng toàn cục biểu diễn nội dung của một ảnh được giới thiệu bởi [Oliva & Torralba, 2004]. Ý tưởng chính là xây dựng một mô tả ở mức hấp dẫn của ngữ cảnh trong ảnh, mà không yêu cầu một hình thức phân đoạn nào, nghiên cứu đã đề xuất một tập hợp các đặc trưng quan trọng như tính tự nhiên (degree of naturalness), tính mở rộng (degree of expansion), độ nhám (degree of roughness), tính cởi mở (degree of openness), độ chắc chắn (degree of ruggedness). Tập hợp này cho phép trình bày cấu trúc không gian của một ngữ cảnh trong ảnh. Đặc trưng mô tả toàn cục GIST hiện nay đã cho kết quả tốt trong việc tìm kiếm ảnh [Douze et al., 2009], phân lớp ảnh, nhận dạng chữ viết tay [Do & Pham, 2015]. Quá trình trích chọn đặc trưng GIST được thực hiện như sau:

2.4.5.1 Tiết xử lý ảnh

Chuẩn hóa kích thước ảnh về $n \times m$ sao cho có thể chia ảnh thành $w \times h$ vùng bằng nhau. Phép biến đổi Fourier được sử dụng khi trích đặc trưng GIST, vì thế kích thước của ảnh phải được chuẩn hóa về dạng $n \times m = 2^n \times 2^n$. Tiếp theo, tách ảnh thành ba kênh màu riêng biệt (đỏ, xanh lá cây, xanh dương) đối với ảnh màu để thực hiện phép biến đổi cho từng kênh màu. Cuối cùng, biến đổi ảnh mức xám mục đích cho ra ảnh có giá trị mức xám thấp được mở rộng và các giá trị mức xám cao bị nén lại.

2.4.5.2 Sinh bộ lọc Gabor

Bộ lọc Gabor được sử dụng rộng rãi khi phân tích dữ liệu. Trong lĩnh vực xử lý ảnh, phương pháp này thường được dùng để trích đặc trưng hay phân đoạn ảnh kết cấu và có độ chính xác cao. Trong các ứng dụng phân tích kết cấu ảnh, thông thường bộ lọc Gabor được sử dụng theo một thang lọc. Mỗi thang lọc thực hiện cho một tần số và đồng thời lọc theo nhiều ứng dụng khác nhau. Trong việc trích chọn đặc trưng này, chúng tôi sử dụng 20 bộ lọc Gabor bao gồm 3 thang lọc và 8 hướng. Trong đó, thang 1 và 2 gồm 8 bộ lọc và thang 3 gồm 4 bộ lọc. Công thức sinh dãy bộ lọc gabor được thể hiện như sau:

$$G(u, v) = e^{-10*0.35*(\frac{f_r(u,v)}{Nf_s}-1)^2 - 2\pi p\delta_\theta^2} \quad (2.16)$$

trong đó,

- $f_r(u, v)$ là giá trị chuẩn hóa bán kính tần số, được tính theo công thức (2.17) và (2.18):

$$f_r(f_x, f_y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2} \quad (2.17)$$

$$f_r(u, v) = shift(f_r(f_x, f_y)) \quad (2.18)$$

- N : kích thước của bộ lọc (bằng kích thước ảnh)
- f_s : tần số ứng với từng thang $f_s = k^{-s} f_{max}$ với, $f_{max} = 0.3$ là tần số cực đại
- p : băng thông của bộ lọc $p = \frac{16*O_s^2}{N^2}$ với, O_s là số hướng ứng với từng thanh s
- δ_θ : hướng của bộ lọc với, $\theta = n \frac{\pi}{O} n = 0, 1, \dots, O - 1$

2.4.5.3 Tính toán véc-tơ đặc trưng

Biến đổi Fourier rời rạc – Discrete Fourier Transform (DFT). Phép biến đổi Fourier được dùng nhiều trong xử lý tín hiệu số. Vì ảnh kỹ thuật số là một phần của tín hiệu số nên phải dùng dạng khác của biến đổi Fourier, đó là biến đổi Fourier rời rạc. Biến đổi DFT được triển khai hai chiều của một ảnh kích thước $n \times m$, xác định bởi một cặp biến đổi:

- Biến đổi Fourier rời rạc thuận (DFT), chuyển đổi từ miền không gian $f(x, y)$ sang miền tần số $F(x, y)$.
- Biến đổi Fourier rời rạc ngược (IDFT), chuyển đổi ảnh từ miền tần số $F(x, y)$ sang miền không gian $f(x, y)$

Áp dụng bộ lọc Gabor. Trước tiên thực hiện phép biến đổi Fourier thuận trên từng ảnh ứng với từng thành phần màu đã được tiền xử lý để chuyển đổi ảnh về miền tần số (2.19). Gọi $i = 1, 2, \dots, 20$ là số bộ lọc Gabor được áp dụng lên từng ảnh màu bằng cách nhân giá trị của bộ lọc Gabor với giá trị phức của từng ảnh tại vị trí (u, v) tương ứng công thức (2.20). Công thức (2.21) thực hiện phép biến đổi Fourier ngược lên từng ảnh để chuyển đổi miền không gian, dựng lại ảnh và trích đặc trưng.

$$F_{R,G,B}(u, v) = DFT[f_{R,G,B}(x, y)] \quad (2.19)$$

$$G_{R,G,B}(u, v) = Gabor_i(u, v) * F_{R,G,B}(u, v) \quad (2.20)$$

$$\begin{aligned} g_{R,G,B} &= IDFT[G_{R,G,B}(u, v)] \\ f_{R,G,B}(x, y) &= \sqrt{g_{R,G,B}_{real}(x, y)^2 + g_{R,G,B}_{imaginary}(x, y)^2} \end{aligned} \quad (2.21)$$

Chia vùng và trích đặc trưng. Sau khi ảnh được áp dụng qua 20 bộ lọc Gabor và phép biến đổi Fourier ngược sẽ tiến hành trích đặc trưng. Để trích đặc trưng, trước hết cần chia ảnh thành 16 vùng phân biệt có kích thước bằng nhau, rồi trích đặc trưng trên mỗi vùng dựa vào công thức (2.22). Ghép vector đặc trưng của từng vùng lại với nhau để nhận được vector đặc trưng của ảnh. Công thức (2.23) xác định số chiều của vector đặc trưng.

$$d_i = \frac{\sum \text{Giá trị các điểm ảnh thuộc vùng } i}{\sum \text{Số lượng điểm ảnh của vùng } i} \quad (2.22)$$

$$n_D = n_R * n_{Gabor} * W^2 \quad (2.23)$$

trong đó,

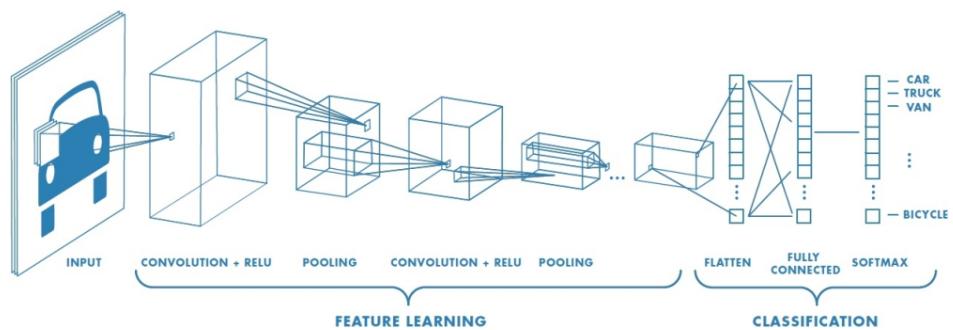
- n_D là số chiều của vector đặc trưng
- n_R là số thành phần màu sử dụng để biểu diễn cho mỗi điểm ảnh
- n_{Gabor} là số lượng bộ lọc áp dụng lên mỗi kênh màu
- W^2 là số vùng phân biệt có kích thước bằng nhau được chia ra từ ảnh

2.4.6 Trích chọn đặc trưng sử dụng mạng nơ-ron tích chập

2.4.6.1 Tổng quan mạng nơ-ron tích chập

Mạng nơ-ron tích chập (Convolutional Neural Network - CNNs) là một trong những mô hình học sâu tiên tiến, được sử dụng nhiều trong các bài toán nhận dạng các đối tượng trong ảnh.

Kiến trúc mạng nơ-ron tích chập chung thông thường sẽ có cấu trúc hai phần. Đầu tiên, phần học đặc trưng (feature learning). Thứ hai, là phần phân lớp (classification). Một ảnh đầu vào (input layer) sẽ được nhân chập với lớp convolutional (convolutional layer), đầu ra của lớp convolutional sẽ đi qua hàm kích hoạt (activation) theo sau là một lớp pooling (pooling layer) mục đích giảm kích thước dữ liệu tại phần học đặc trưng. Sau khi qua lớp convolutional và lớp pooling, kết quả tại phần học đặc trưng sẽ được làm phẳng thành một véc-tơ cột, dùng lớp fully connected (fully connected layer) để kết nối các đặc điểm của ảnh để nhận được một lớp đầu ra (output layer) được mô tả trong hình 2.27.



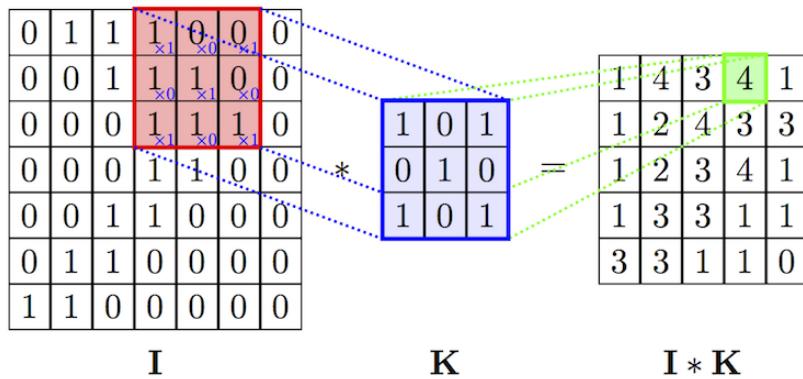
Hình 2.27: Kiến trúc mạng nơ-ron tích chập (Nguồn: anhvnn.wordpress.com)

- *Lớp convolutional* (convolutional layer) còn được biết đến như là một cửa sổ trượt (sliding window), nhân (kernel) hay bộ lọc (filter) như hình 2.28. Một ảnh nhị phân (trắng đen) I được biểu diễn dưới dạng một ma trận 2 chiều, một cửa sổ trượt K được biểu diễn bằng một ma trận 2 chiều (kích thước 3×3). Thực hiện phép tích chập ($I * K$) giữa ảnh I và cửa sổ trượt K bằng cách di chuyển cửa sổ trượt theo số bước trượt (stride – thường được dùng để giảm kích thước của ảnh đầu vào, sau phép nhân tích chập) được chỉ định từ trái sang phải và từ trên xuống dưới. Kết quả thu được một ma trận I' có kích thước nhỏ hơn ma trận I , tức là ($I' < I$). Để nhận được một ma trận I' có cùng kích thước với I ta thực hiện nhân tích chập với tham số *padding* = k (zero-padding) sẽ thêm k véc-tơ giá trị 0 vào viền ngoài của ma trận ảnh I . Để dàng tính toán lớp đầu ra bằng công thức (2.24).

$$\frac{N_I - F_K + 2P}{S} + 1 \quad (2.24)$$

trong đó,

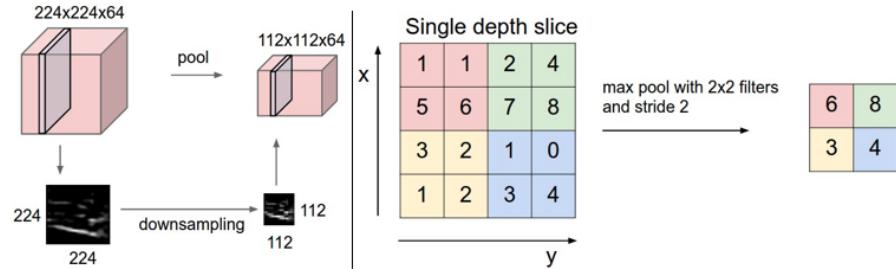
- N_I là kích thước ảnh đầu vào
- F_K là kích thước cửa sổ trượt
- P là giá trị của zero-padding
- S là giá trị của bước trượt



Hình 2.28: Nhân tích chập ảnh I và cửa sổ trượt K (Nguồn: anhvnn.wordpress.com)

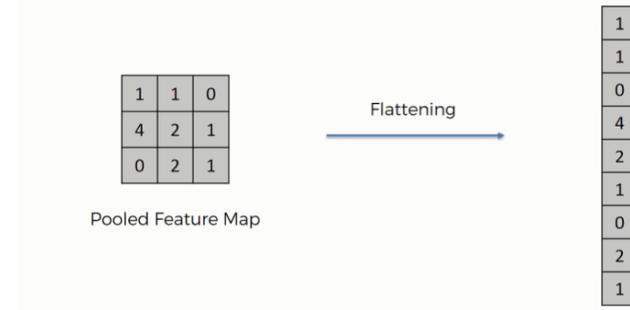
- *Lớp pooling* (pooling layer). Lớp pooling thường được dùng giữa các lớp convolutional, giảm kích thước dữ liệu đầu vào (ảnh I) nhưng giữ được các thông tin quan trọng làm giảm việc tính toán cho mô hình phân lớp. Sau khi kết quả đầu ra của convolutional đi qua hàm activation, mỗi một activation sẽ được biến đổi bằng hàm kích hoạt phi tuyến ReLU như hình 2.27, nếu sử dụng max-pooling kích thước 2×2 (pooling-size), *stride* = 2 thì nhận được một ảnh có kích thước bằng một nữa ảnh đầu vào như hình 2.29. Max-pooling được sử dụng thông dụng hiện nay sẽ lấy giá

trị cao nhất trong pooling-size kích thước 2×2 , còn average-pooling sẽ lấy giá trị trung bình.

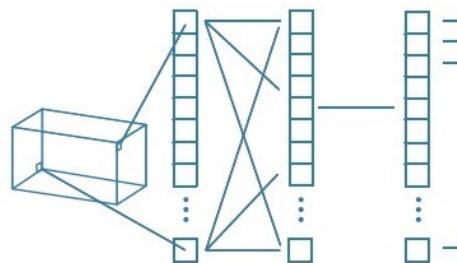


Hình 2.29: Lớp pooling giảm kích thước dữ liệu đầu vào (Nguồn: anhvnn.wordpress.com)

- *Lớp fully-connected* (fully-connected layer). Sau khi qua các lớp convolutional và lớp pooling thì mô hình đã học được tương đối nhiều đặc điểm của ảnh, đầu ra cuối cùng này sẽ là được làm phẳng (hình 2.30) thành một véc-tơ cột trước khi dùng lớp fully-connected để kết nối các đặc điểm của ảnh để ra được output của model (hình 2.31).



Hình 2.30: Làm phẳng output của lớp cuối cùng (Nguồn: nttuan8.com)



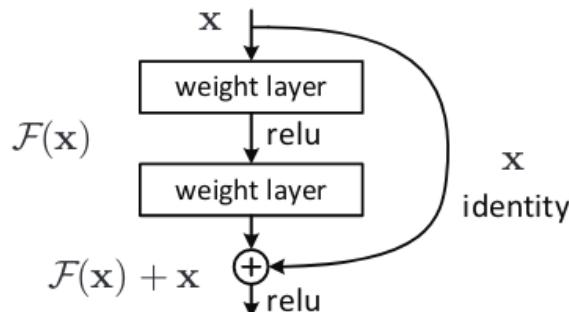
Hình 2.31: Làm phẳng và kết nối đến fully-connected (Nguồn: anhvnn.wordpress.com)

2.4.6.2 Mô hình Residual Networks

Để tăng độ chính xác khi huấn luyện cho các mô hình mạng nơ-ron tích chập, bằng cách tăng số lượng layer, nhưng nếu số lượng layer lớn (trên 50) sẽ gặp tình trạng gradient

của hàm matsu quá nhỏ hoặc quá lớn do thuật toán lan truyền ngược được gọi lần lượt là vanishing gradients đối với mạng nơ-ron sâu (Deep Neural Networks) và exploding gradients đối với mạng nơ-ron hồi quy (Recurrent Neural Networks) làm cho mô hình khó hội tụ. Mô hình Residual Networks (ResNet) [He et al., 2015] [13] được ra đời để giải quyết các vấn đề của các mô hình máy học truyền thống.

Mô hình ResNet là mô hình chiến thắng trong cuộc thi *ImageNet Large Scale Visual Recognition Challenge 2015* (ILAVRC2015). Sử dụng đến 152 layers – sâu hơn 8 lần so với mô hình VGG, chỉ số lỗi nhận được trên tập dữ liệu kiểm tra của ImageNet là 3.57%. Để giải quyết vấn đề vanishing và exploding bằng cách thêm một *batch normalization* trước mỗi activation cho mỗi layer có nhiệm vụ chuẩn hóa dữ liệu đầu ra, các hệ số trở nên cân bằng giúp mô hình dễ dàng hội tụ. Một vấn đề nữa được nêu trong [13] là vấn đề bão hòa độ chính xác của mô hình (degradation), tức là, độ chính xác của mô hình không tăng khi thêm nhiều layer hơn hoặc không giảm khi ít layer, đây không phải là hiện tượng overfitting mà do mô hình càng sâu thì càng khó huấn luyện, vấn đề được tác giả và cộng sự của ông giải quyết bằng cách cứ mỗi 2 layers cộng input với output, được gọi là *residual block*. Kiến trúc của ResNet gồm nhiều *residual block* được tính theo công thức $H(x) = F(x) + x$ mô hình sẽ dễ dàng hội tụ hơn như trong hình 2.32.



Hình 2.32: Residual block (Nguồn: anhvnn.wordpress.com)

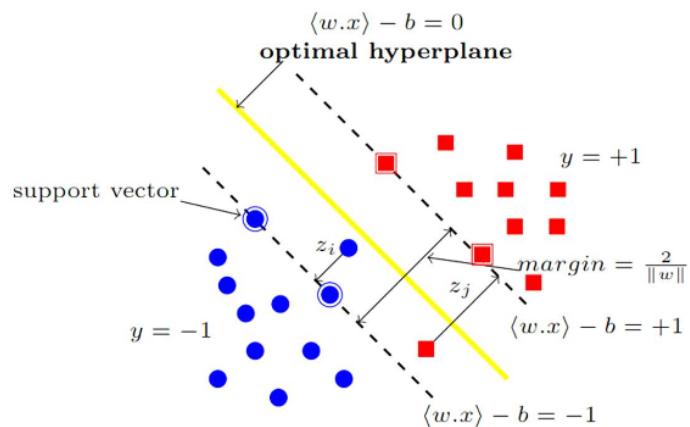
2.5 Máy học véc-tơ hỗ trợ

2.5.1 Giới thiệu

Máy học véc-tơ hỗ trợ (Support Vector Machines - SVM) [Vapnik, 1995] là một mô hình phân lớp nhị phân tuyến tính (học có giám sát), phân lớp được xác định bởi một siêu phẳng tách biệt, thuật toán đi tìm một siêu phẳng tối ưu.

Xét ví dụ phân lớp nhị phân tuyến tính đơn giản [14] được mô tả như hình 2.33 với m phần tử x_1, x_2, \dots, x_n trong không gian n chiều với nhãn (lớp) của các phần tử tương ứng là y_1, y_2, \dots, y_n có giá trị là 1 (lớp dương) hay -1 (lớp âm). Giải thuật SVM [Vapnik, 1995] tìm

một siêu phẳng tối ưu (xác định bởi véc-tơ pháp tuyến w và độ lệch của siêu phẳng với tọa độ b) để tách dữ liệu ra hai lớp. Giải thuật SVM tìm siêu phẳng tách biệt hai lớp ra xa nhất có thể (siêu phẳng tối ưu) dựa trên hai siêu phẳng hỗ trợ song song của hai lớp. Siêu phẳng hỗ trợ của lớp $+1$ ($w \cdot x - b = +1$) là siêu phẳng của phần tử x_p thuộc lớp $y_p = +1$ nằm về phía bên phải của nó, tức là: $w \cdot x_p - b \geq +1$. Tương tự siêu phẳng hỗ trợ của lớp -1 ($w \cdot x - b = -1$) là các siêu phẳng mà các phần tử x_n thuộc lớp $y_n = -1$ nằm về phía bên trái của siêu phẳng của nó, tức là: $w \cdot x_n - b \leq -1$. Những phần tử nằm ngược phia với siêu phẳng hỗ trợ được coi như lỗi. Khoảng cách được biểu diễn bởi $z_i \geq 0$ (với x_i nằm đúng phia của siêu phẳng hỗ trợ của nó thì khoảng cách lỗi tương ứng $z_i = 0$, ngược lại $z_i > 0$ là khoảng cách từ điểm x_i đến siêu phẳng hỗ trợ tương ứng của nó). Khoảng cách giữa hai siêu phẳng hỗ trợ gọi là lề $= 2/\|w\|$, trong đó $\|w\|$ là độ lớn ($2 - \text{norm}$) của vector pháp tuyến w . Siêu phẳng tối ưu (nằm giữa hai siêu phẳng hỗ trợ) cần tìm phải được cực đại hóa lề (lề càng lớn, mô hình phân lớp càng an toàn) và cực tiểu lỗi. Vấn đề tìm siêu phẳng tối ưu của giải thuật SVM dẫn đến việc giải quyết bài toán quy hoạch toàn phương (2.42).



Hình 2.33: Phương pháp tuyến tính với máy học SVM (Nguồn: NCKH)

$$\min_{\alpha} \frac{1}{2} \sum_{i=0}^m \sum_{j=0}^m y_i y_j \alpha_i \alpha_j K \langle x_i, x_j \rangle - \sum_{i=1}^m \alpha_i$$

với ràng buộc :

$$\begin{cases} \sum_{i=1}^m y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \quad \forall i = 1, 2, \dots, m \end{cases} \quad (2.25)$$

Trong đó C là hằng số thường dùng để điều chỉnh độ rộng của lề phân hoạch và tổng khoảng cách lỗi; $K \langle x_i, x_j \rangle$ là hàm nhân tuyến tính $K \langle x_i, x_j \rangle = \langle x_i \bullet x_j \rangle$.

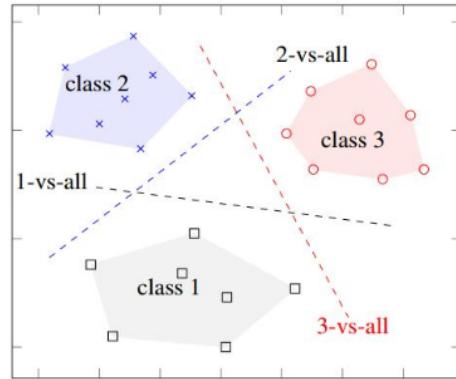
Giải bài toán huy hoạch toàn phương (2.25) thu được $\#LC$ phần tử x_i tương ứng với $x_i > 0$, được gọi là vector hỗ trợ. Chỉ cần $\#LC$ hỗ trợ này ta có thể dựng lại được siêu phẳng phân lớp. Mô hình SVM thực hiện phân lớp phần tử mới x bằng (2.26)

$$predict(x) = sign \left(\sum_{i=1}^{\#LC} y_i \alpha_i K \langle x, x_i \rangle - b \right) \quad (2.26)$$

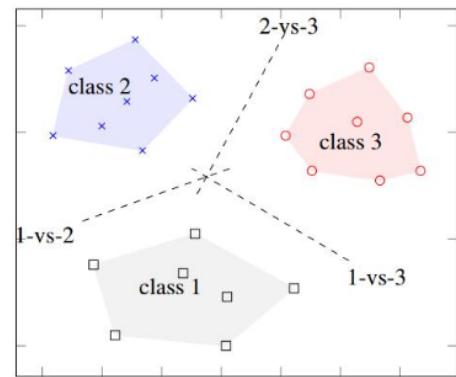
Máy học vector hỗ trợ có thể sử dụng các hàm nhân khác nhau để giải quyết lớp các bài toán phân lớp phi tuyến [Cristianini & Shawe-Taylor, 2000]. Để xử lý các vấn đề phân lớp phi tuyến, không cần bất kỳ thay đổi nào hơn từ giải thuật mà chỉ cần thay thế hàm nhân tuyến tính trong (2.25) và (2.26) bằng các hàm nhân khác. Hàm phi tuyến phổ biến là hàm cơ sở bán kính (Radial Basic Function – RBF): $K \langle x_i, x_j \rangle = e^{-\gamma \|x_i - x_j\|^2}$

Mô hình máy học SVM cho kết quả cao, ổn định, chịu đựng nhiễu tốt và phù hợp với các bài toán phân lớp, hồi quy. Nhiều ứng dụng thành công của SVM đã được công bố trong nhiều lĩnh vực như nhận dạng ảnh, phân loại văn bản và sinh-tin học [Guyon, 1999].

Để giải quyết vấn đề phân lớp đa lớp (số lớp $c \geq 3$), giải thuật SVM thường mở rộng theo 2 phương pháp đơn giản là 1-vs-all [Vapnik, 1995] và 1-vs-1 [Krebel, 1999].



Hình 2.34: Phương pháp 1-vs-all cho máy học SVM đa lớp (Nguồn: [14])



Hình 2.35: Phương pháp 1-vs-1 cho máy học SVM đa lớp (Nguồn: [14])

Phương pháp 1-vs-all (hình 2.34) xây dựng c mô hình SVM nhị phân, mô hình c_i tách lớp c_i (lớp dương) ra khỏi các lớp khác (âm). Phương pháp 1-vs-1 (hình 2.35) xây dựng

$c(c - 1)/2$ mô hình SVM nhị phân, mỗi mô hình tách một cặp 2 lớp. Việc phân lớp dựa vào bình phương chọn khoảng cách đến các siêu phẳng thu được từ SVM nhị phân.

2.5.2 Ưu điểm

Điểm nổi bật của giải thuật SVM là đồng thời cực tiểu hóa lỗi phân lớp và cực đại hóa khoảng cách lề giữa các lớp, từ đó tối ưu hóa hiệu suất phân lớp kể cả khi không gian đặc trưng có số chiều lớn. Ngoài ra, thời gian phân lớp của giải thuật SVM cũng khá nhanh.

2.5.3 Nhược điểm

Đối với giải thuật SVM để đạt được kết quả phân loại tốt cần chọn hàm nhân phù hợp. Ngoài ra, đối với bài toán phân lớp đa lớp, giải thuật SVM phải lặp đi lặp lại quá trình huấn luyện vì SVM chỉ giải quyết bài toán phân lớp nhị phân. Do đó cần rất nhiều thời gian để giải thuật SVM huấn luyện mô hình.

Chương 3

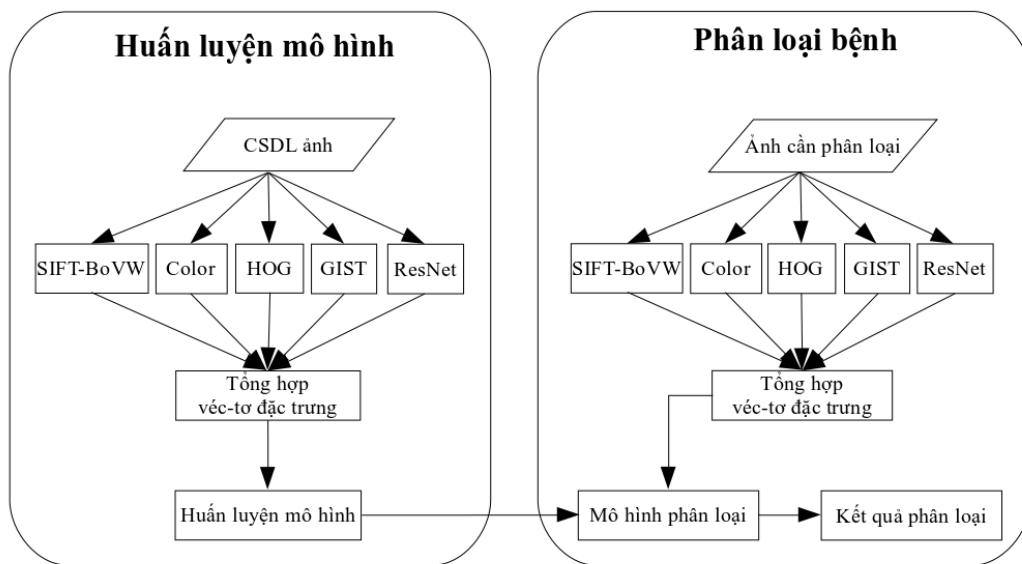
Xây dựng hệ thống

3.1 Hệ thống phân loại bệnh trên lá cam

Hệ thống phân loại bệnh trên lá cam bao gồm 2 mô đun chính: mô đun huấn luyện mô hình và mô đun phân loại bệnh lá cam như hình 3.1.

Mô đun huấn luyện mô hình thực hiện các bước: cơ sở dữ liệu ảnh được trải qua các giải thuật trích đặc trưng (SIFT-BoVW, Color, HOG, GIST, ResNet), tổng hợp véc-tơ đặc trưng (từ các giải thuật trích đặc trưng), huấn luyện mô hình.

Mô đun phân loại nhận đầu vào là một ảnh được trải qua các giải thuật trích đặc trưng (SIFT-BoVW, Color, HOG, GIST, ResNet), tổng hợp véc-tơ đặc trưng (từ các giải thuật trích đặc trưng), sử dụng mô hình phân loại (được huấn luyện bởi mô đun huấn luyện mô hình) để phân loại.



Hình 3.1: Sơ đồ hệ thống phân loại bệnh trên lá cam

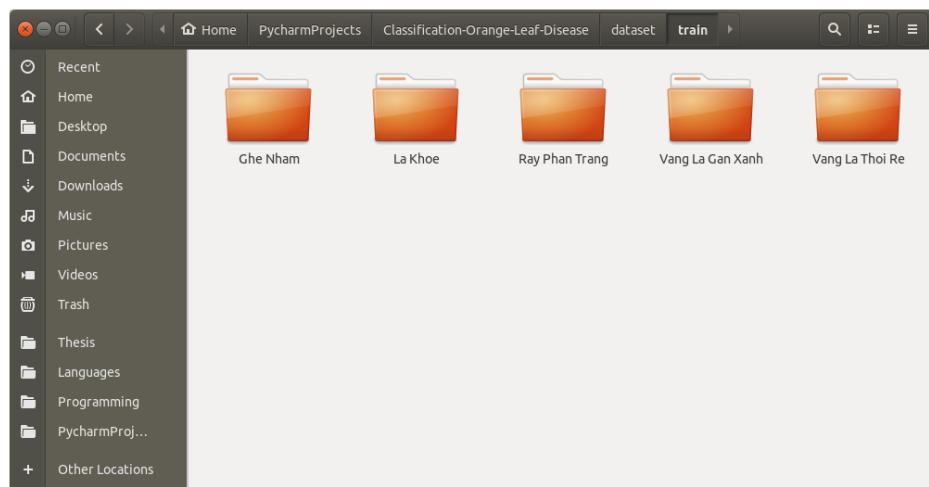
3.2 Thu thập và tổ chức dữ liệu

3.2.1 Thu thập ảnh

Ảnh lá cam được thu thập từ máy ảnh Panasonic DMC-FH1 độ phân giải 3-Megapixel, độ nhạy sáng (ISO) 400 và tiêu cự 28mm, tại 2 tỉnh Đồng Tháp và Hậu Giang. Một ảnh hợp lệ chỉ chứa một lá cam bệnh, nên không chứa các đối tượng khác, thời gian chụp khi lá cam bắt đầu nhiễm bệnh hoặc đang nhiễm bệnh, chụp trực diện ở phần mặt trên của lá, riêng bệnh rầy phấn trắng do đặc điểm gây bệnh nên sẽ được chụp thêm ở phần mặt dưới của lá.

3.2.2 Tổ chức dữ liệu

Tập các ảnh được tổ chức dưới dạng các thư mục, một thư mục tương ứng là tập hợp các ảnh của một loại bệnh và tên thư mục cũng là tên lớp (hình 3.2). Mỗi lớp có số lượng ảnh gần ngang bằng nhau, cụ thể, lớp Ghé nhám là 142, Lá khỏe là 143, Rầy phấn trắng là 141, Vàng lá gân xanh là 142 và Vàng lá thối rẽ là 143 ảnh.



Hình 3.2: Cách tổ chức dữ liệu

3.3 Trích chọn các đặc trưng

3.3.1 Trích chọn đặc trưng SIFT

Dựa trên phần cơ sở lý thuyết được giới thiệu ở chương 2, tiến hành xây dựng tập các mô tả của điểm hấp dẫn (keypoint), khởi tạo các tham số mặc định từ giải thuật SIFT, một ảnh đầu vào sau khi qua các bước tính toán sẽ nhận được một véc-tơ 128 thành phần (bins). Bước tiếp theo, chúng tôi áp dụng giải thuật K-Means để gom cụm các mô tả, trong luận văn

này chúng tôi tiến hành chọn số lượng cụm là 400 (clusters). Mô tả trích chọn ở giải thuật SIFT của mỗi ảnh đầu vào sau khi lượng hóa véc-tơ sẽ có 400 thành phần.

3.3.2 Trích chọn đặc trưng Color

Tiếp theo thực hiện trích chọn các đặc trưng Color nhận được một véc-tơ 512 thành phần cho mỗi ảnh, tức là, $8 \times 8 \times 8 = 512$. Tiến hành trích chọn dựa trên ba kênh màu, chia biểu đồ (histogram) có giá trị cường độ trong khoảng 0 đến 255 thành 8 bins cho từng kênh màu (đỏ, xanh lá canh, xanh dương).

3.3.3 Trích chọn đặc trưng HOG

Ảnh đầu vào được đưa về kích thước 80×80 trước khi được tính toán gradient ảnh và biểu đồ gradient, ảnh được chia thành 100 blocks, 8 là kích thước pixel trên mỗi cell (1×1), kích thước bin trên mỗi cell là 9 (bins). Mỗi ảnh đầu vào sẽ nhận được một véc-tơ có 900 thành phần, tức là, $100 \times 9 \times 1 \times 1 = 900$.

3.3.4 Trích chọn đặc trưng GIST

Trước tiên ảnh cần được chuẩn hóa về kích thước 128×128 , – Tiền xử lý, tách 3 kênh màu (đỏ, xanh lá cây, xanh dương), biến đổi tỷ lệ lôgarit (giảm độ chênh lệch tỷ lệ các điểm ảnh tối so với các điểm ảnh sáng), thêm các điểm ảnh biên, lọc trắng ảnh (cân bằng năng lượng phổ), lọc chuẩn hóa độ tương phản, xóa các điểm ảnh biên, – Sinh 20 bộ lọc Gabor để áp dụng lên từng ảnh (đỏ, xanh lá cây, xanh dương, đã được tiền xử lý và chuyển về miền tần số), – Chia ảnh thành 16 vùng riêng biệt bằng nhau, tính giá trị trên mỗi vùng bằng cách lấy tổng giá trị của các điểm ảnh trên mỗi vùng chia cho số điểm ảnh của vùng, thực hiện lần lượt 20 bộ lọc trên 16 vùng của ảnh đỏ, xanh lá cây, xanh dương thu được véc-tơ 960 thành phần, tức là, $16 \times 20 \times 3 = 960$.

3.3.5 Trích chọn đặc trưng Residual Networks

Ảnh đầu vào được chuẩn hóa về kích thước 224×224 trước khi nhận được một véc-tơ đặc trưng có 100352 thành phần tại lớp fully-connected sau khi đã dropout. Bộ tham số của mô hình được huấn luyện sẵn từ tập dữ liệu ImageNet sau đi đã loại bỏ lớp soft-max.

3.3.6 Tổng hợp véc-tơ đặc trưng

Ảnh bệnh sau khi trải qua các bước trích chọn đặc trưng, mỗi giải thuật trích chọn đặc trưng sẽ nhận được một véc-tơ đặc trưng có thành phần (chiều) khác nhau (SIFT–400, Color–512, HOG–900, GIST–960 và Residual Networks–100352), tổng hợp (nối) các véc-tơ tương ứng này ta nhận được một véc-tơ có 103124 thành phần.

3.4 Huấn luyện mô hình

3.4.1 Chuẩn bị tập dữ liệu huấn luyện

Tập dữ liệu gồm 569 ảnh, gồm 5 loại, trong đó 4 loại lá bệnh và 1 loại lá khỏe được tổ chức thành các thư mục (lớp). Sau khi qua bước rút trích các đặc trưng tạo ra dữ liệu dạng bảng mỗi dòng tương ứng là ảnh lá bệnh có 103124 thành phần (chiều) và nhãn là loại bệnh trong hệ thống, dữ liệu đã được xáo trộn ngẫu nhiên trước khi đưa vào huấn luyện. Bảng 3.1 thống kê số lượng mẫu trong từng lớp.

Bảng 3.1: Thống kê số lượng mẫu trong tập dữ liệu huấn luyện

STT	Tên loại bệnh	Nhãn trong hệ thống	Số lượng mẫu
1	Ghẻ nhám	0	112
2	Lá khỏe	1	115
3	Rầy phấn trắng	2	111
4	Vàng lá gân xanh	3	115
5	Vàng lá thối rẽ	4	116
Tổng cộng:		5	569

3.4.2 Huấn luyện mô hình

Trong luận văn này, chúng tôi đề xuất huấn luyện mô hình SVM phân loại bệnh do những ưu điểm vượt trội trong vấn đề phân loại ảnh so với mô hình K láng giềng gần nhất (K-Nearest Neighbour - KNN) [15]. Mô hình SVM được chúng tôi sử dụng hàm nhân Linear vì số chiều của véc-tơ lớn. Chúng tôi cũng điều chỉnh siêu tham số γ của hàm nhân Linear và hằng số C (tham số dung hòa lỗi và độ rộng của lề SVM) để có được kết quả cao nhất dựa trên nghi thức kiểm tra chéo từ tập dữ liệu huấn luyện, trong phần tiếp theo chúng tôi cũng sẽ sử dụng nghi thức này để tìm bộ tham số tốt nhất cho mô hình KNN nhằm so sánh hiệu quả so với mô hình SVM. Bộ tham số chúng tôi tìm được là $\gamma = 0.0001$ và $C = 0.01$.

Chương 4

Thực nghiệm và đánh giá kết quả

4.1 Môi trường thực nghiệm

Để đánh giá hiệu quả của mô hình phân loại được đề xuất, chúng tôi cài đặt chương trình nhận dạng và phân loại ảnh bệnh trên lá cam bằng ngôn ngữ lập trình Python [Rossum, 1991], rút trích các đặc trưng SIFT, Color bằng thư viện OpenCV [Bradski & Kaehler, 2012], HOG bằng thư viện scikit-image [8], GIST bằng thư viện Pyleargist [Grisel, 2009] và đặc trưng ResNet bằng thư viện Keras [Chollet, 2015]. Thư viện scikit-learn [Cournapeau, 2007] được sử dụng để huấn luyện mô hình phân loại bệnh. Trong phần 4.2 thực nghiệm phân loại, chúng tôi cũng muốn so sánh mô hình đề xuất SVM với mô hình KNN để kiểm chứng như trong bài nghiên cứu [15] là đúng sự thật.

Thực nghiệm được thực hiện trên máy tính cá nhân, cài đặt hệ điều hành Linux Ubuntu 18.04 LTS, bộ vi xử lý Intel® Core i5-3337U, 1.8GHz, 4 nhân và bộ nhớ RAM 8 GB.

4.2 Thực nghiệm phân loại

4.2.1 Chuẩn bị tập dữ liệu kiểm thử

Mô hình sau khi được huấn luyện ở phần 3.4.2 được sử dụng để phân loại trên tập dữ liệu kiểm thử, bước rút trích các đặc trưng tạo ra tập dữ liệu dạng bảng, có 142 dòng, mỗi dòng tương là ảnh lá bệnh có 103124 chiều, đã được xáo trộn trước khi đưa vào mô hình phân loại. Bảng 4.1 thống kê số lượng mẫu của từng lớp trong tập dữ liệu kiểm thử.

4.2.2 Kết quả thực nghiệm

Chúng tôi sử dụng tập dữ liệu kiểm 142 dòng để tiến hành kiểm thử dựa trong mô hình đã được huấn luyện trước đó. Bảng 4.2 thống kê số nhãn trong mỗi lớp, kết quả phân loại đúng và sai của mô hình. Kết quả cho thấy bệnh vàng lá thối rẽ cho kết quả phân loại chính xác tuyệt đối do bệnh có thể dễ dàng nhận dạng bằng màu vàng nổi bậc trên toàn bộ phiến

Bảng 4.1: Thống kê số lượng mẫu trong tập dữ liệu thử

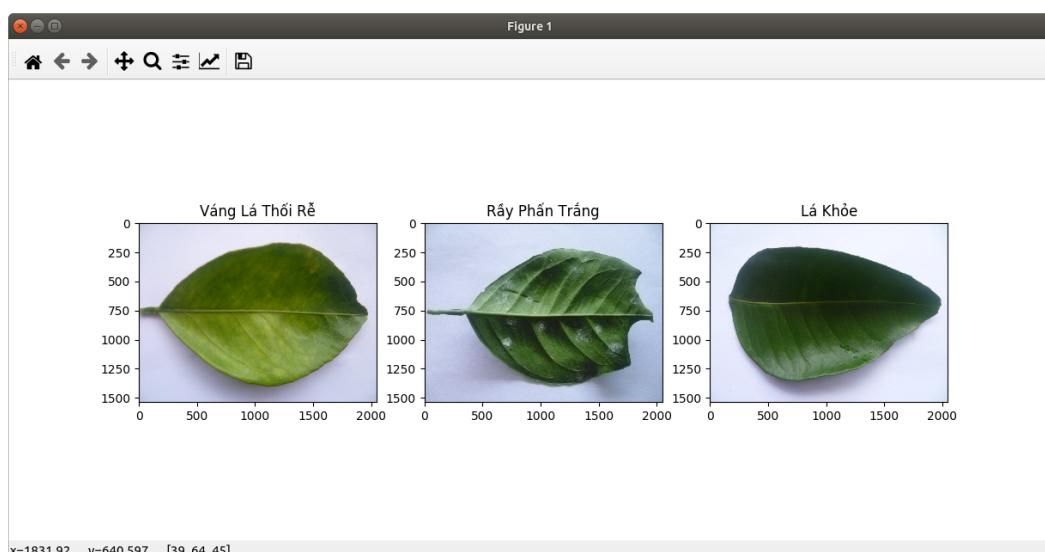
STT	Tên loại bệnh	Nhãn trong hệ thống	Số lượng nhãn/lớp
1	Ghẻ nhám	0	30
2	Lá khỏe	1	28
3	Rầy phấn trắng	2	30
4	Vàng lá gân xanh	3	27
5	Vàng lá thối rẽ	4	27
Tổng cộng:		5	142

lá, khác với bệnh rầy phấn trắng, bệnh khó nhận biết với đóm nhỏ màu nâu ở phần mặt trên của lá với bệnh ghẻ nhám hoặc lá khỏe, để nhận biết chính xác thì cần phải chụp ở phần dưới của mặt lá vì có nhiều lông sáp hình dạng tròn.

Bảng 4.2: Thống kê mẫu phân loại đúng và sai trên tập dữ liệu kiểm thử của mô hình SVM

STT	Tên loại bệnh	Mẫu kiểm thử	Phân loại đúng	Phân loại sai
1	Ghẻ nhám	30	26 (86.66%)	4 (13.33%)
2	Lá khỏe	28	27 (96.42%)	1 (3.57%)
3	Rầy phấn trắng	30	25 (83.33%)	5 (16.66%)
4	Vàng lá gân xanh	27	24 (88.88%)	3 (11.11%)
5	Vàng lá thối rẽ	27	27 (100%)	0 (0%)
Tổng cộng:		142	129 (90.84%)	13 (9.15%)

Hình 4.1 là kết quả phân loại với 3 ảnh là các loại bệnh khác nhau, được phân loại chính xác với tổng thời gian phân loại là 1.6 phút, tương ứng mỗi hình cho thời gian phân loại là 32 giây, đạt mục tiêu đề ra của đề tài là phân loại bệnh trên lá cam.



Hình 4.1: Kết quả phân loại của mô hình SVM

Chúng tôi cũng tiến hành huấn luyện mô hình máy học KNN để so sánh với mô hình máy học SVM do chúng tôi đề xuất, bộ tham số của hai mô hình cũng được tìm dựa trên nghiệm kiểm tra chéo từ tập dữ liệu huấn luyện. Mô hình máy học KNN phân lớp dữ liệu dựa trên khoảng cách Euclidean, tức là một điểm dữ liệu mới sẽ được gán nhãn dựa trên K láng giềng gần nó nhất. Bảng 4.3 cho thấy trong trường hợp này mô hình máy học SVM cho kết quả tốt hơn mô hình KNN, điều này là đúng như đã được chứng minh trong [15] của tác giả J. Kim và các cộng sự.

Bảng 4.3: So sánh mô hình KNN và mô hình SVM

ID	Mô hình	Bộ tham số	Độ chính xác (%)
1	SVM	$C = 0.01$, $kernel = \text{linear}$, $\gamma = 0.0001$	90.84
2	KNN	$neighbors = 2$, $weights = \text{uniform}$	42.95

4.3 Đánh giá kết quả

Trong phần thực nghiệm, có thể thấy mô hình máy học SVM sử dụng đặc trưng SIFT kết hợp mô hình túi đựng từ trực quan (BoVW), Color, HOG, GIST và ResNet đạt độ chính xác 90.84% là kết quả chấp nhận được, thời gian phân loại là 32.2 giây cũng không quá lâu, đúng với mục tiêu đề tài đã đề ra ở chương 1.

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận và hạn chế

5.1.1 Kết luận

Luận văn thực hiện ứng dụng máy học véc-tơ hỗ trợ trong phân loại bệnh trên lá cam sử dụng các đặc trưng SIFT- (BoVW), Color, HOG, GIST và ResNet. Bước rút trích đặc trưng và phân loại, đạt mục tiêu đề ra thời gian không quá lâu và độ chính xác chấp nhận được. Kết quả thực nghiệm trên tập dữ liệu gồm 711 ảnh của 5 loại, trong đó 4 loại là bệnh thường xuất hiện trong các nhà vườn tại 2 tỉnh (Đồng Tháp và Hậu Giang) và 1 loại lá khỏe cho thấy phương pháp SVM-SIFT-(BoVW)-Color-HOG-GIST-ResNet đạt đến 90.84% độ chính xác trên tập kiểm tra, cao hơn so với mô hình KNN sử dụng cùng đặc trưng là 42.95%.

5.1.2 Hạn chế

Trong đề tài này, tuy độ chính xác là chấp nhận được, nhưng cần kiểm chứng trên tập dữ liệu lớn hơn để tránh trường hợp học vẹt làm cho độ chính xác ngoài thực tế không giống với kết quả thực nghiệm. Mô hình phân loại chỉ giới hạn ở 4 loại bệnh là một hạn chế, vì trong thực tế còn rất nhiều loại bệnh. Do vào thời điểm thực hiện đề tài (10/2019) , thực hiện thu thập mẫu ở 2 tỉnh Đồng Tháp và Hậu Giang chỉ xuất hiện 4 loại bệnh, vì một số loại bệnh xuất hiện theo mùa nên còn nhiều hạn chế.

5.2 Hướng phát triển

Trong tương lai gần, chúng tôi dự định thu thập thêm mẫu của một số loại bệnh mới và tăng thêm số lượng mẫu cho các loại bệnh đã được thực hiện trong đề tài này, mục đích cung cấp thêm kết quả thực nghiệm trên tập dữ liệu lớn hơn, đa dạng các loại bệnh hơn và so sánh hiệu quả của mô hình máy học SVM với các mô hình máy học khác.

Tài liệu tham khảo

- [1] (2019). Quả cam – vị thuốc bôn mùa, [Online]. Available: <https://hongngochospital.vn/qua-cam-vi-thuoc-bon-mua/>.
- [2] M. Thức. (2019). Bệnh trên cây có múi thiệt hại 260ha, [Online]. Available: <http://www.baodongthap.vn/giam-ngheo-ve-thong-tin-2018/benh-tren-cay-co-mui-thiet-hai-260ha-78516.aspx>.
- [3] H. Tân. (2019). Phát triển vườn cây có múi ở đbscl: Còn nhiều nỗi lo, [Online]. Available: <http://www.baovinhlong.com.vn/nhip-song-dong-bang/201807/phat-trien-vuon-cay-co-mui-o-dbscl-con-nhieu-noi-lo-2902014/#.Xc-MJdFS80h>.
- [4] S. Weizheng, W. Yachun, C. Zhanliang, and W. Hongda, “Grading method of leaf spot disease based on image processing,” in *2008 international conference on computer science and software engineering*, IEEE, vol. 6, 2008, pp. 491–494.
- [5] M. Ei-Helly, A. Rafea, S. Ei-Gamal, and R. A. E. Whab, “Integrating diagnostic expert system with image processing via loosely coupled technique,” *Central Laboratory for Agricultural Expert System (CLAES)*, 2004.
- [6] K. Moshashai, M. Almasi, S. Minaei, and A. Borghei, “Identification of sugarcane nodes using image processing and machine vision technology,” *International Journal of Agricultural Research*, vol. 3, no. 5, pp. 357–364, 2008.
- [7] D. Al Bashish, M. Braik, and S. Bani-Ahmad, “Detection and classification of leaf diseases using k-means-based segmentation and,” *Information Technology Journal*, vol. 10, no. 2, pp. 267–275, 2011.
- [8] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, “Scikit-image: Image processing in python,” *PeerJ*, vol. 2, e453, Jun. 2014, ISSN: 2167-8359. DOI: [10.7717/peerj.453](https://doi.org/10.7717/peerj.453). [Online]. Available: <https://doi.org/10.7717/peerj.453>.
- [9] R. C. Gonzalez and R. E. Woods, “Digital image processing,” in *Digital Image Processing*. 2002.

- [10] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” 2005.
- [12] S. Mallick. (2019). Histogram of oriented gradients, [Online]. Available: <https://www.learnopencv.com/histogram-of-oriented-gradients/>.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.
- [14] P. N. Khang, T. N. M. Thư, and Đ. T. Nghị, “Điểm danh bằng mặt người với đặc trưng gist và máy học véctơ hỗ trợ,” *PROCEEDING of Publishing House for Science and Technology*, 2019.
- [15] J. Kim¹, B. Kim, and S. Savarese, “Comparing image classification methods: K-nearest-neighbor and support-vector-machines,” in *Proceedings of the 6th WSEAS international conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics*, vol. 1001, 2012, pp. 48 109–2122.