

MỤC LỤC

MỤC LỤC	1
DANH SÁCH BẢNG	3
DANH SÁCH HÌNH ẢNH	3
DANH SÁCH TỪ VIẾT TẮT	5
CHƯƠNG 1. TỔNG QUAN VỀ QUANG PHỔ VÀ MÁY ĐO PHỔ CẬN HỒNG NGOẠI	7
1.1. Tổng quan về quang phổ	7
1.1.1. Quang phổ là gì?	7
1.1.2. Ứng dụng của quang phổ.	7
1.2. Máy đo phổ cận hồng ngoại NIR (Near Infrared)	7
1.2.1. Giới thiệu.	7
1.2.2. Một số đặc điểm khi sử dụng máy đo quang phổ	8
1.2.3. Cấu tạo.	9
1.2.4 Cơ chế hoạt động	11
1.2.5. Cảnh báo và điều kiện sử dụng	14
1.3. Kết chương.	15
CHƯƠNG 2. CÁC KỸ THUẬT HỌC MÁY	16
2.1. Tổng quan về học máy.	16
2.1.1. Phân loại học máy.	16
2.1.2. Học có giám sát và các bước giải quyết	18
2.2. Các thuật toán phân loại và thư viện hỗ trợ	20
2.2.1. Thư viện Scikit Learn.	20
2.2.2. Thuật toán KNN (K Nearest Neighbor Classifier)	20
2.2.3. Thuật toán Naive Bayes	22
2.2.3. Thuật toán Random Forest	23
2.2.4. Thuật toán SVM (Support Vector Machine)	25

2.3. Kết chương.....	26
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ.	27
3.1. Giao diện máy khách.....	27
3.1.1. Thư viện PyQt	27
3.1.2. Thư viện Matplotlib.	28
3.1.3. Giao diện.	29
3.2. Xây dựng máy chủ.....	33
3.2.1. Thư viện Numpy.....	33
3.2.2. Thư viện Pandas.....	33
3.2.3. Thư viện requests.	34
3.3. Huấn luyện mô hình.....	34
3.3.1. Thu thập dữ liệu.....	34
3.3.2. Mô tả dữ liệu thô.	37
3.3.3. Thống kê dữ liệu.....	39
3.3.4. Tiền xử lý dữ liệu và trích chọn đặc trưng.	40
3.3.5. Triển khai học máy trên máy chủ.....	43
3.4. Đánh giá kết quả.....	44
3.4.1. Đánh giá thuật toán KNN.....	44
3.4.2. So sánh với các thuật toán phân loại khác.....	44
3.4.3. Thảo luận.	45
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	46
TÀI LIỆU THAM KHẢO.....	47

DANH SÁCH BẢNG

Bảng 1.1: Thông số kỹ thuật DLP NIRscan Nano EVM.

Bảng 3.1: Độ chính xác với thuật toán KNN khi K thay đổi.

Bảng 3.2: Độ chính xác với thuật toán SVM.

Bảng 3.3: Độ chính xác với thuật toán Naive Bayes.

Bảng 3.4: Độ chính xác với thuật toán Random Forest.

DANH SÁCH HÌNH ẢNH

Hình 1.1: Máy đo phổ cận hồng ngoại NIR.

Hình 1.2: Cấu tạo của máy đo phổ cận hồng ngoại NIR.

Hình 1.3: Sơ đồ khối DLP NIRscan Nano.

Hình 1.4: Động cơ DLP NIRscan Nano.

Hình 1.5: Góc nhìn trên cùng của mô-đun chiếu sáng.

Hình 2.1: AlphaGo chơi cờ vây với Lee Sedol. AlphaGo là một ví dụ của Reinforcement learning.

Hình 2.2: Ví dụ về một bộ cơ sở dữ liệu của chữ số viết tay.

Hình 2.3: KNN cho bài toán phân lớp.

Hình 2.4: Ví dụ về cây quyết định.

Hình 2.5: Thuật toán SVM.

Hình 3.1: Import PyQt5 và các thư viện cần thiết.

Hình 3.2: Giao diện máy khách.

Hình 3.3: Mở giao diện

Hình 3.4: Dữ liệu được đọc và ghi ra màn hình.

Hình 3.5: Dữ liệu được vẽ dưới dạng biểu đồ.

Hình 3.6: So sánh sự biến thiên của phổ cường độ giữa nhiều file dữ liệu của các loại quả khác nhau.

Hình 3.7: Kết quả trả về từ máy chủ.

Hình 3.8: Phần mềm DLP NIRscan Nano GUI v2.1.0.

Hình 3.9: Thu thập dữ liệu từ quả Thanh Long.

Hình 3.10: Thu thập dữ liệu từ quả Chuối.

Hình 3.11: Một file dữ liệu có chứa đầy đủ cả ba định dạng.

Hình 3.12: Một file dữ liệu chỉ có chỉ số bước sóng và phổ cường độ.

Hình 3.13: Quang phổ của độ hấp thụ.

Hình 3.14: Phổ cường độ.

Hình 3.15: Đạo hàm bậc 1 của phổ NIR của các loại quả.

Hình 3.16: Đạo hàm bậc 2 của phổ NIR của các loại quả.

Hình 3.17: Quá trình gửi yêu cầu từ máy khách và trả lại kết quả từ máy chủ.

DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
NIR	Near Infrared
DLP	Digital Light Processing
DMD	Digital Micromirror Devices
ADC	Analog-Digital Converter
SNR	Signal-to-noise ratio
KNN	K-nearest neighbors

MỞ ĐẦU

1. Tổng quan về đề tài

Ngày nay Công nghệ thông tin đã trở thành một phần tất yếu của cuộc sống con người, không những thế nó còn góp phần phát triển kinh tế trong tất cả các lĩnh vực. Việc ứng dụng Công nghệ thông tin trong các ngành công nghiệp hay nông nghiệp nhằm nâng cao tính hiệu quả trong công việc sản xuất.

Hoạt động phân loại nông sản ở nhiều nơi sản xuất còn thô sơ, do đó một hệ thống cần thiết để phân loại nông sản nhằm giúp giảm thiểu sức lao động, thời gian và kinh phí cho nông dân.

2. Mục đích và ý nghĩa của đề tài.

2.1. Mục đích.

Hỗ trợ hiệu quả trong việc phân loại trái cây sử dụng phổ cường độ của chúng. Mục đích xa hơn là nghiên cứu thêm về phát hiện chất độc hại trong các loại trái cây cũng như các nông sản khác.

2.2. Ý nghĩa.

Nâng cao hiệu quả trong việc phân loại các loại trái cây sử dụng phổ cường độ.

3. Phương pháp thực hiện.

- Phương pháp phân tích tổng hợp từ nguồn tài liệu trên Internet.
- Phương pháp phân tích thiết kế hệ thống.
- Phương pháp thử nghiệm, đánh giá kết quả.

4. Bố cục của đồ án.

Đồ án bao gồm các nội dung sau:

Chương 1: TỔNG QUAN VỀ QUANG PHỔ VÀ MÁY ĐO PHỔ CẬN HỒNG NGOẠI

Chương 2: NGHIÊN CỨU VỀ CÁC KỸ THUẬT HỌC MÁY (MACHINE LEARNING)- HỌC TỰ ĐỘNG

Chương 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

Kết luận và hướng phát triển.

CHƯƠNG 1. TỔNG QUAN VỀ QUANG PHỔ VÀ MÁY ĐO PHỔ CẬN HỒNG NGOẠI.

1.1. Tổng quan về quang phổ.

1.1.1. Quang phổ là gì?

Quang phổ là một kỹ thuật rất hiệu quả để nhận biết và mô tả đặc tính của vật liệu. Nói chung, nó sẽ đo sự thay đổi của độ hấp thụ hoặc phát xạ cho các bước sóng khác nhau của ánh sáng.

Phương pháp này sẽ xác định cách ánh sáng sẽ tương tác với vật liệu và tạo ra phổ như là một hàm của cường độ ánh sáng phản xạ về phía cảm biến. Phổ này được coi là đặc điểm riêng của vật liệu này.

1.1.2. Ứng dụng của quang phổ.

Quang phổ được sử dụng trong các lĩnh vực phân tích y sinh học, là một phương pháp nhằm tìm ra các quy luật liên hệ giữa tính chất vật lý và hóa học với các quang phổ phát xạ hay hấp thụ của chúng. Ứng dụng quang phổ trong việc tìm lại tính chất của hệ vật chất từ quang phổ quan sát được.

1.2. Máy đo phổ cận hồng ngoại NIR (Near Infrared).

1.2.1. Giới thiệu.

Nhà sản xuất: Texas Instruments.

Tên sản phẩm: DLP NIRscan Nano.

Quang phổ là một kỹ thuật mạnh mẽ để nhận biết và mô tả các vật liệu vật lý thông qua các biến đổi trong sự hấp thụ hoặc phát xạ của các bước sóng ánh sáng khác nhau của một mẫu.

Phổ kế đo sự thay đổi độ hấp thụ ánh sáng của vật liệu. DLP NIRscan Nano EVM [Hình 1.1] là một mô-đun đánh giá hoàn chỉnh để thiết kế một máy quang phổ cầm tay cận hồng ngoại có hiệu suất cao, giá cả phải chăng. Công cụ linh hoạt này chứa mọi thứ mà một nhà thiết kế cần để bắt đầu phát triển máy quang phổ dựa trên DLP ngay lập tức.

Công nghệ DLP cho phép máy phân tích quang phổ cầm tay sử dụng trong thực phẩm, dược phẩm, dầu khí, y tế, an ninh và các ngành công nghiệp mới nổi khác để cung cấp các mức hiệu suất phòng thí nghiệm trong lĩnh vực này.[1]

EVM chứa thiết bị micromirror kỹ thuật số DLP2010NIR, bộ điều khiển kỹ thuật số DLPC150 và các thành phần quản lý năng lượng tích hợp DLPA2005. Công nghệ này tập hợp một bộ các thành phần cung cấp giải pháp hệ thống quang phổ hiệu quả và hấp dẫn cho:

- Máy phân tích quá trình di động.
- Máy quang phổ siêu di động.

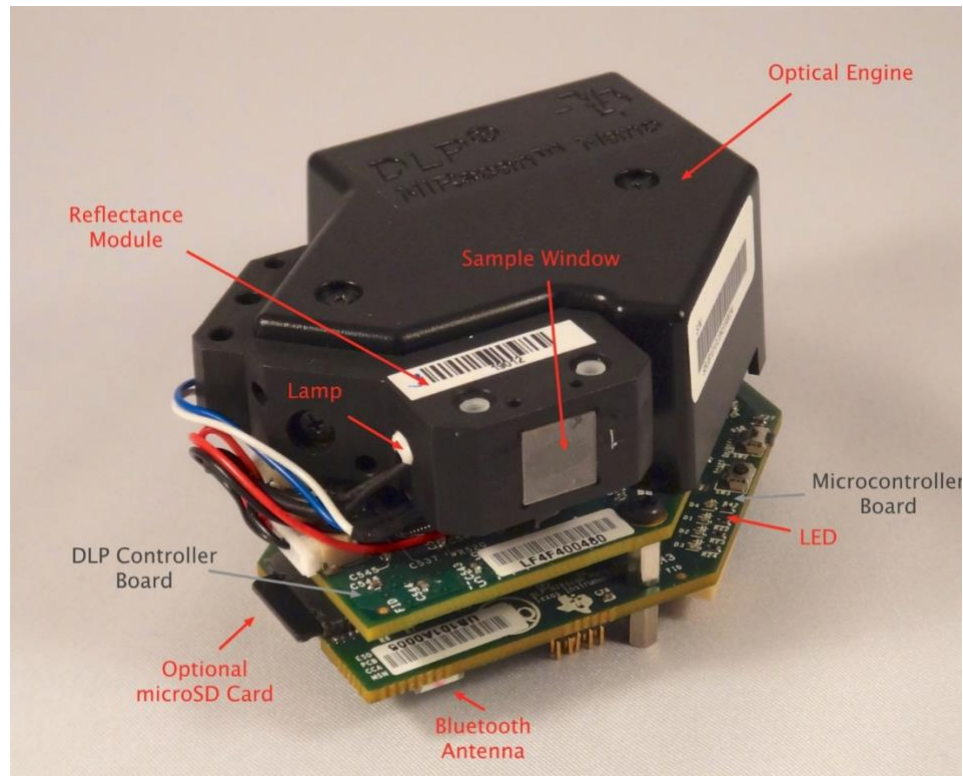


Hình 1.1: Máy đo phổ cận hồng ngoại NIR [1].

1.2.2. Một số đặc điểm khi sử dụng máy đo quang phổ.

- Máy đo quang phổ giúp xác định nhiều loại lượng nguyên tố có trong vật liệu.
- Phân tích được mẫu chất rắn, bột, lỏng mà không cần xử lý hóa học hay gia công mẫu trước.
- Phân tích được nhiều loại vật liệu.
- Phân tích chính xác có thể từ vài ppm đến 100%.
- Gọn nhẹ, có thể di chuyển dễ dàng.

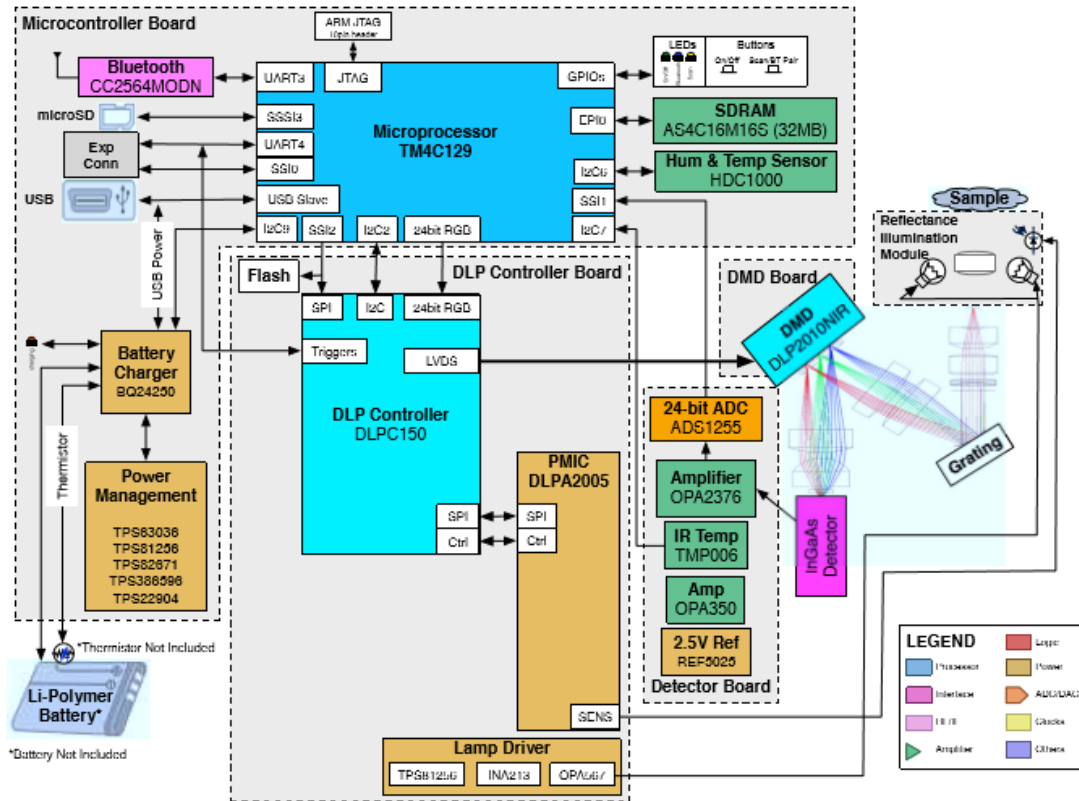
1.2.3. Cấu tạo.



Hình 1.2: Cấu tạo của máy đo phổ cận hồng ngoại NIR [1].

- DLP NIRscan Nano EVM là máy quang phổ EVM hoàn chỉnh sử dụng công nghệ DLP [Hình 1.2]. Gói EVM bao gồm:
 - Máy đo quang phổ kế cận hồng ngoại được tối ưu hóa cho phạm vi bước sóng từ 900 đến 1700nm
 - Mô-đun chiếu sáng phản chiếu với hai đèn hồng ngoại tích hợp.
 - Khe cắm đầu vào 1,8 mm × 0,025 mm. Để cho phép dung sai cơ học, khe quá đầy trong trực tiếp giao. Do đó, hình ảnh 1,69 mm × 0,025 mm từ khe được hiển thị trên DMD.
 - Ống kính chuẩn trực.
 - Bộ lọc độ dài 885nm.
 - Cách tử nhiễu xạ phản xạ.
 - Thấu kính hội tụ.

- DLP2010NIR DMD (WVGA 0,2 inch, pixel trực giao 854×480 , NIR được tối ưu hóa).
- Bộ thu thập quang học.
- Bộ phát hiện pixel đơn 1mm chưa hoàn chỉnh InGaAs.
- Hệ thống con điện tử với bốn thiết bị điện tử thẻ bao gồm [Hình 1.3]:
 1. Thẻ vi điều khiển.
 - Bộ vi xử lý Tiva TM4C1297 để điều khiển hệ thống hoạt động ở tần số 120 MHz
 - Bộ nhớ SDRAM 32 MB để lưu trữ các mô hình
 - Quản lý năng lượng bằng pin lithium polymer hoặc mạch sạc pin lithium-ion bằng mô-đun Bluetooth bq24250.
 - CC2564MODN Năng lượng thấp cho kết nối Bluetooth 4.0
 - Đầu nối micro-USB cho kết nối USB.
 - Khe cắm thẻ nhớ microSD để lưu trữ dữ liệu ngoài.
 - Cảm biến nhiệt độ và độ ẩm HDC1000.
 2. Thẻ điều khiển DLP.
 - Bộ điều khiển DLP150 DLP
 - Mạch quản lý nguồn tích hợp DLPA2005 cho các bộ nguồn của bộ điều khiển DMD và DLP
 - Trình điều khiển đèn hiện tại không đổi dựa trên OPA567 và được giám sát bởi INA213
 3. Bảng phát hiện.
 - Mạch khuếch đại vi sai thấp
 - Bộ chuyển đổi tương tự -digital (ADC) ADS1255 30 kSPS với SPI
 - Cảm biến nhiệt điện TMP006 cho cảm biến và đo nhiệt độ môi trường
 - InGaAs Hamamatsu G12180-010A photodiode không được lọc 1 mm
 4. Thẻ DMD
 - DLP2010NIR gắn micromirror hồng ngoại kỹ thuật số.



Hình 1.3: Sơ đồ khối DLP NIRscan Nano [1].

1.2.4 Cơ chế hoạt động.

Động cơ quang phổ kế DLP NIRscan Nano EVM được gắn trên đỉnh của hệ thống con điện tử. Cấu hình là một kiến trúc phân tán sau với một mô-đun mẫu phản xạ có thể tháo rời. Các mô-đun phản xạ bao gồm hai đèn dây tóc vonfram băng rộng đầu ống kính.

Trong quá trình quét, mẫu hấp thụ một lượng ánh sáng NIR cụ thể và phản xạ khuếch tán ánh sáng không hấp thụ vào hệ thống. Lượng ánh sáng hấp thụ ở mỗi bước sóng phụ thuộc vào cấu tạo phân tử của vật liệu, và đặc trưng cho vật liệu đó, dấu vân tay hóa học.

Ánh sáng phản xạ khuếch tán từ mẫu được thu thập bởi ống kính thu thập và tập trung vào động cơ quang thông qua khe đầu vào. Kích thước khe được chọn để cân bằng độ phân giải bước sóng với SNR của máy quang phổ. Máy quang phổ này sử dụng khe 25 μm rộng 1,8 mm. Ánh sáng đi qua khe được chuẩn trực bởi bộ thấu

kính đầu tiên, đi qua bộ lọc thông sóng dài 885nm, và sau đó chiếu vào một tấm lưới phản xạ.

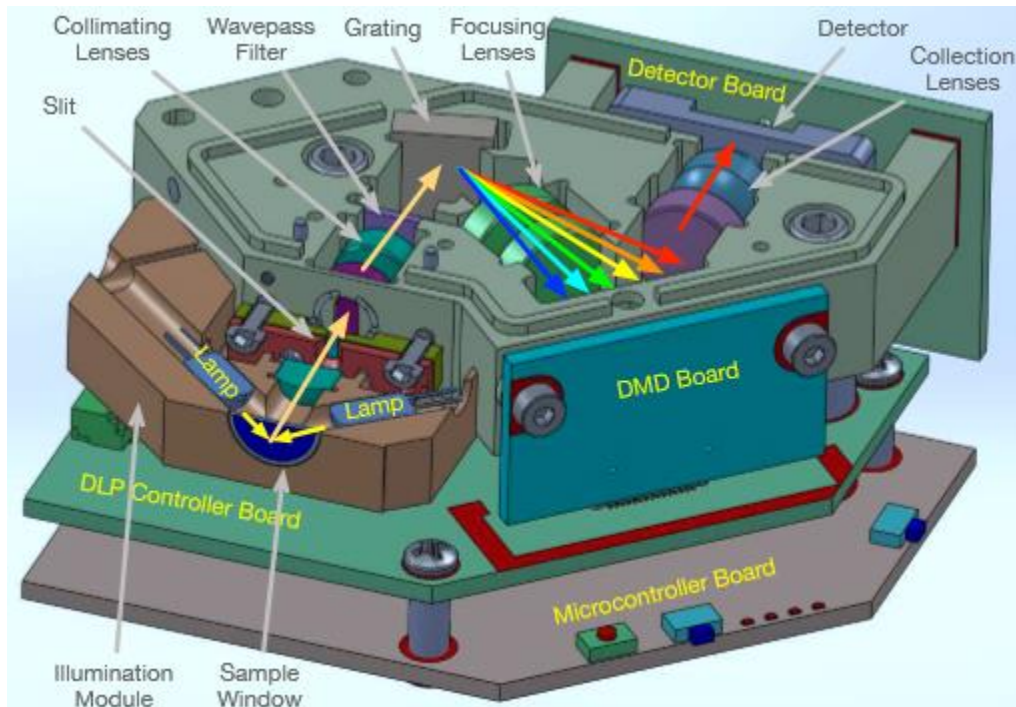
Cách tử này, kết hợp với thấu kính hội tụ, phân tán ánh sáng thành các bước sóng cấu thành của nó. Các thấu kính lấy nét tạo thành hình ảnh của khe tại DLP2010NIR DMD. Các bước sóng khác nhau của hình ảnh khe này được trải theo chiều ngang trên DLP2010NIR DMD.

Hệ thống quang học hình ảnh các bước sóng 900nm đến một đầu của DMD và 1700nm ở đầu kia, với tất cả các bước sóng khác được phân tán liên tục ở giữa. Khi các cột DMD cụ thể được chọn là bật hoặc nghiêng về vị trí 17° dương, năng lượng được phản ánh bởi các cột được chọn sẽ được dẫn qua hệ thống quang học thu thập đến máy dò InGaAs pixel đơn. Tất cả các cột DMD khác được chọn là tắt hoặc nghiêng về vị trí 17° âm, chuyển hướng các bước sóng không được chọn xuống dưới cùng của động cơ ánh sáng và tránh xa đường quang của máy dò để không ảnh hưởng đến phép đo bước sóng đã chọn.

Để cho phép dung sai cơ học ở vị trí khe, góc cách tử và vị trí DMD, hình ảnh khe DLP NIRscan Nano trên DMD được lấp đầy trong trục phân tán 10% ở mỗi đầu và được lấp đầy trong trục trục giao.

Điều này dẫn đến khoảng $(1700 - 900\text{nm}) / (854 * 0,8 \text{ pixels}) = 1,17\text{nm}$ mỗi pixel trên DMD. Trong quá trình sản xuất, hiệu chuẩn được thực hiện giữa các bước sóng và vị trí cột của chúng trên DMD. Do số lượng cột DMD thường không chia hết cho số nhóm bước sóng mong muốn, nên DLP NIRscan Nano duy trì hằng số chiều rộng cột trong quá trình quét, nhưng bước qua mảng DMD bằng một lượng khác với chiều rộng của cột. Số lượng bước này phụ thuộc vào chiều rộng và số lượng mẫu mong muốn (điểm bước sóng). [1].

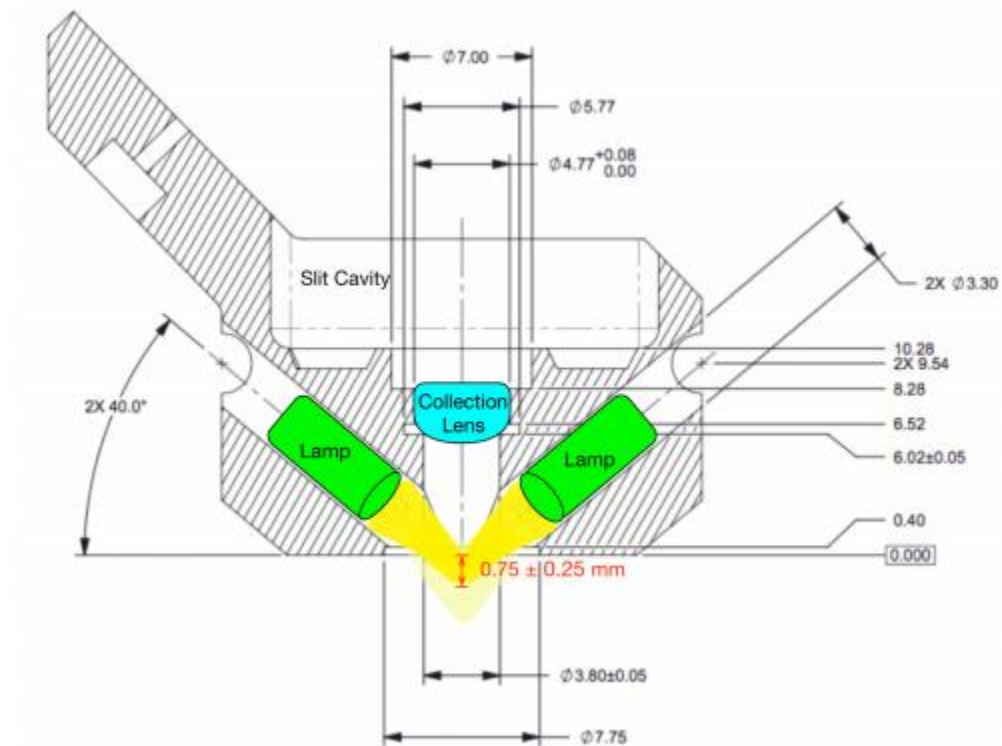
Khi chúng ta thực hiện các phép đo với máy quang phổ, máy quang phổ sẽ phát ra đèn hồng ngoại trên vật liệu. Tùy thuộc vào vật liệu, phản ứng của nó ở các bước sóng khác nhau là khác nhau, cảm biến InGaAS thu thập ánh sáng và chuyển đổi cường độ này thành tín hiệu điện, sau đó thành các giá trị số nhờ ADC. Giá trị này được gọi là cường độ.



Hình 1.4: Động cơ DLP NIRscan Nano [1].

Mô-đun phản xạ Nano DLP NIRscan hoạt động bằng cách chiếu sáng mẫu được thử ở một góc sao cho các phản xạ gương không được thu thập, trong khi thu thập và tập trung các phản xạ khuếch tán vào khe. Các đèn chiếu sáng được chỉ định là đèn kết thúc thấu kính vì đầu trước của bóng đèn thủy tinh được tạo thành một thấu kính hướng nhiều ánh sáng hơn từ dây tóc đến vùng thử nghiệm mẫu.

Các hình chữ nhật màu xanh lá cây đại diện cho đèn kết thúc ống kính. Các hình nón màu vàng đậm là ánh sáng được chiếu ra từ đèn. Mỗi đèn tạo ra một chùm ánh sáng ở góc 40 độ giao nhau qua cửa sổ sapphire ở khoảng 0,75 mm. Có khoảng dung sai khoảng 0,25 mm đối với giao điểm của chùm tia do dung sai cơ học của khung máy, các biến thể của đầu thấu kính từ đèn sang đèn, các biến thể của hình dạng đèn và vị trí của đèn [Hình 1.4]. Đèn kết thúc thấu kính tập trung chùm sáng ở khoảng cách 3 mm so với đèn và tạo kích thước điểm bao phủ của sổ mẫu sapphire.



Hình 1.5: Góc nhìn trên cùng của mô-đun chiếu sáng [1].

Ống kính thu thập tập hợp ánh sáng từ vùng có đường kính 2,5 mm ở cửa sổ mẫu. Kích thước của vùng thu thập được khớp với kích thước điểm chiếu sáng danh nghĩa được tạo bởi các đèn cuối thấu kính. Điều này đòi hỏi mẫu phải được đặt trực tiếp vào cửa sổ sapphire, nơi hai đường dẫn nguồn sáng góc giao nhau với hình nón tầm nhìn của thấu kính [Hình 1.5]. Nếu mẫu được dịch chuyển ra xa khỏi cửa sổ, mẫu có thể không nhận đủ ánh sáng để hệ thống thực hiện quét chính xác.

1.2.5. Cảnh báo và điều kiện sử dụng.

Mở hoặc tháo rời động cơ quang học sẽ ảnh hưởng đến hệ thống Nano NIRscan. Việc tháo vỏ trên động cơ quang học cho phép bụi và vết bẩn thu thập trên quang học ảnh hưởng đến hiệu suất của nó. Ngoài ra, việc tháo nắp có thể di chuyển quang học, khe và máy dò ra khỏi căn chỉnh yêu cầu sắp xếp lại và hiệu chuẩn lại nhà máy. Loại bỏ khe, máy dò InGaAs và DLP2010NIR sẽ yêu cầu hệ thống phải được sắp xếp lại và hiệu chỉnh lại tại nhà máy sản xuất.[1]

Bảng 1.1. Thông số kỹ thuật DLP NIRscan Nano EVM.

Thông số	Min	Typ	Max	Đơn vị
Bước sóng được hỗ trợ	900		1700	nm
Độ phân giải quang		10		nm
Năng lượng đèn		1.4		W
Nhiệt độ	0	25	50	°C

Để thu được dữ liệu chuẩn, cần để thiết bị ở nhiệt độ phòng cùng với ánh sáng thích hợp. Những điều kiện trên có thể ảnh hưởng đến các giá trị phổ khi thu thập dữ liệu.

Texas Instrument cung cấp cho chúng ta một ứng dụng đã được tạo và bao gồm thư viện kết nối và xử lý dữ liệu, tôi sẽ khai thác thư viện này để tạo một ứng dụng cho dự án. Nghĩa là, để tích hợp trực tiếp cơ chế học máy vào ứng dụng GUI (Giao diện người dùng đồ họa) của máy quang phổ.

1.3. Kết chương.

Chương này cung cấp cho người đọc lý thuyết cơ bản về quang phổ và những ứng dụng của nó. Ngoài ra, phân tích và tìm hiểu thiết kế của máy đo quang phổ cận hồng ngoại NIRscan Nano, giúp chúng ta hiểu về cách thức, điều kiện để thu thập dữ liệu.

CHƯƠNG 2. CÁC KỸ THUẬT HỌC MÁY

2.1. Tổng quan về học máy.

2.1.1. Phân loại học máy.

a. Học có giám sát (Supervised Learning): phương pháp này bao gồm việc truyền tới thuật toán dữ liệu được dán nhãn cho phép máy tính học và xây dựng một mô hình để dự đoán các nhãn này.

Đầu ra của một hàm có thể là một giá trị liên tục (gọi là hồi qui), hay có thể là dự đoán một nhãn phân loại cho một đối tượng đầu vào (gọi là phân loại). Nhiệm vụ của chương trình học có giám sát là dự đoán giá trị của hàm cho một đối tượng bất kì là đầu vào hợp lệ, sau khi đã xem xét một số ví dụ huấn luyện (nghĩa là, các cặp đầu vào và đầu ra tương ứng). Để đạt được điều này, chương trình học phải tổng quát hóa từ các dữ liệu sẵn có để dự đoán được những tình huống chưa gặp phải theo một cách hợp lý.[2]

b. Học không giám sát (unsupervised machine learning): Phương pháp này liên quan đến việc cho phép máy tính phân loại các nhóm dữ liệu được liên kết. Nói cách khác, tùy thuộc vào máy tính để ghi nhận dữ liệu.

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng. Những thuật toán loại này được gọi là Unsupervised learning vì không giống như Supervised learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào. Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B. Cụm không giám sát được đặt tên theo nghĩa này.[3]

c. Học bán giám sát (Semi-Supervised Learning): Các bài toán khi chúng ta có một lượng lớn dữ liệu nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.

Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được (ảnh y học chẳng hạn). Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ internet.[3]

d. Học tăng cường (Reinforcement Learning): Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance). Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.[3]



Hình 2.1: AlphaGo chơi cờ vây với Lee Sedol. AlphaGo là một ví dụ của Reinforcement learning [3].

AlphaGo gần đây nổi tiếng với việc chơi cờ vây [Hình 2.2] thắng cả con người. Cờ vây được xem là có độ phức tạp cực kỳ cao với tổng số nước đi là xấp xỉ 10^{171} , so với cờ vua là 10^{120} và tổng số nguyên tử trong toàn vũ trụ là khoảng 10^{80} !! Vì vậy, thuật toán phải chọn ra 1 nước đi tối ưu trong số hàng nhiều tỉ tỉ lựa

chọn, và tất nhiên, không thể áp dụng thuật toán tương tự như IBM Deep Blue (IBM Deep Blue đã thắng con người trong môn cờ vua 20 năm trước). Về cơ bản, AlphaGo bao gồm các thuật toán thuộc cả Supervised learning và Reinforcement learning. Trong phần Supervised learning, dữ liệu từ các ván cờ do con người chơi với nhau được đưa vào để huấn luyện. Tuy nhiên, mục đích cuối cùng của AlphaGo không phải là chơi như con người mà phải thậm chí thắng cả con người. Vì vậy, sau khi học xong các ván cờ của con người, AlphaGo tự chơi với chính nó với hàng triệu ván chơi để tìm ra các nước đi mới tối ưu hơn. Thuật toán trong phần tự chơi này được xếp vào loại Reinforcement learning.[3]

2.1.2. Học có giám sát và các bước giải quyết.

Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách toán học, Supervised learning là khi chúng ta có một tập hợp biến đầu vào $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ và một tập hợp nhãn tương ứng $\mathbf{Y}=\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$ trong đó $\mathbf{x}_i, \mathbf{y}_i$ là các vector. Các cặp dữ liệu biết trước $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{X} \times \mathbf{Y}$ được gọi là tập *training data* (dữ liệu huấn luyện). Từ tập training data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y:

$$\mathbf{Y}_i \approx f(\mathbf{x}_i), \forall i=1, 2, \dots, N$$

Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu \mathbf{x} mới, chúng ta có thể tính được nhãn tương ứng của nó $\mathbf{y}=f(\mathbf{x})$. [3]

Thuật toán supervised learning còn được tiếp tục chia nhỏ ra thành hai loại chính:

Classification (Phân loại)

Một bài toán được gọi là classification nếu các nhãn của dữ liệu đầu vào được chia thành một số hữu hạn nhóm. Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không. Hai ví dụ phía trên được chia vào loại này.

Regression (Hồi quy)

Nếu nhãn không được chia thành các nhóm mà là một giá trị thực cụ thể. Ví dụ: một căn nhà rộng x m², có y phòng ngủ và cách trung tâm thành phố z km sẽ có giá là bao nhiêu?[3]



Hình 2.2: Ví dụ về một bộ cơ sở dữ liệu của chữ số viết tay [3].

Ví dụ trong nhận dạng các chữ viết tay. Chúng ta có ảnh [hình 2.4] của hàng nghìn ví dụ từ mỗi chữ số và được viết từ nhiều người khác nhau. Sau khi đưa bức ảnh này vào thuật toán và chỉ ra nó biết mỗi đến một hàm số mà đầu vào sẽ là một chữ số. Sau khi nhận được bức ảnh mới mà mô hình chưa từng nhìn thấy bao giờ. Từ đó, nó sẽ dự đoán bức ảnh trong đó chứa những chữ số như nào.

Để có thể xây dựng mô hình dự đoán, chúng ta nên thực hiện các bước sau:

Bước 1: Thu thập dữ liệu:

Bước này liên quan đến việc thu thập dữ liệu và ghi nhãn dữ liệu này cho các bước đào tạo. Nghĩa là, chúng ta nên thu thập dữ liệu và sau đó đưa ra một tên hoặc giá trị đại diện cho phép máy tính học và làm điều tương tự trên dữ liệu mới.

Bước 2: Trích chọn đặc trưng:

Bước này là khó khăn nhất trong thế giới của học máy. Thật vậy, bước này bao gồm việc chọn các giá trị đại diện hoặc các đặc điểm cụ thể giúp phân loại mọi thứ. Nhưng tại sao có quá nhiều vấn đề? Ví dụ: Khi bạn nhìn vào một con mèo và một con chó, làm thế nào bạn có thể tạo ra sự khác biệt từ một hình ảnh? Nó có vẻ rất dễ dàng đối với con người, nhưng đối với máy tính, chúng ta nên xác định rằng đó là nhờ vào mắt, lông, ... và tất cả những gì hình thành nên các đặc trưng cho phép xây dựng mô hình dự đoán.

Bước 3: Huấn luyện mô hình:

Dựa trên dữ liệu đầu vào là các vec-tơ đặc trưng được gán nhãn, dùng thuật toán học máy để xây dựng mô hình.

Bước 4: Kiểm tra đánh giá mô hình.

Trong bước này, chúng ta sẽ kiểm tra mô hình, chúng ta sẽ cung cấp dữ liệu đầu vào và so sánh nhãn dự đoán bởi mô hình với nhãn thực sự (nhãn đúng) của dữ liệu, và tính tỷ lệ phần trăm dự đoán đúng. Nói chung, độ chính xác lớn hơn 90% được coi là chấp nhận được.

2.2. Các thuật toán phân loại và thư viện hỗ trợ.

2.2.1. Thư viện Scikit Learn.

Scikit Learn là thư viện miễn phí được viết bằng python cho máy học. Nó được phát triển bởi nhiều người đóng góp, đặc biệt là trong thế giới học thuật, bởi các viện nghiên cứu và giáo dục đại học của Pháp như Inria và Télécom ParisTech.

Thư viện này được bao gồm trong gói Anaconda, là một bộ công cụ phân tích dựa trên python trong các lĩnh vực khác nhau như Khoa học dữ liệu, Xử lý tín hiệu, Xử lý hình ảnh, ... [4]

Tại sao nên dùng scikit-learn?

- Hỗ trợ hầu hết các thuật toán của machine learning một cách đơn giản, hiệu quả mà chúng ta không cần phải mất công ngồi cài đặt lại.
- Có tài liệu hướng dẫn sử dụng.
- Độ tin cậy cao do scikit-learn được xây dựng bởi các chuyên gia hàng đầu.
- Có nguồn dữ liệu phong phú: iris, digit, ...

2.2.2. Thuật toán KNN (*K Nearest Neighbor Classifier*)

KNN là một trong những thuật toán học có giám sát đơn giản nhất mà hiệu quả trong một vài trường hợp trong học máy. Khi huấn luyện, thuật toán này không học một điều gì từ dữ liệu huấn luyện (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. KNN có thể áp dụng được vào cả hai loại của bài toán học có giám sát là phân lớp và hồi quy. KNN còn được gọi là một thuật toán instance-based hay memory-based learning.[5]

Thuật toán này được trình bày dưới dạng các bước như sau:

Bước 1: Chọn số lượng hàng xóm K (số lượng hàng xóm).

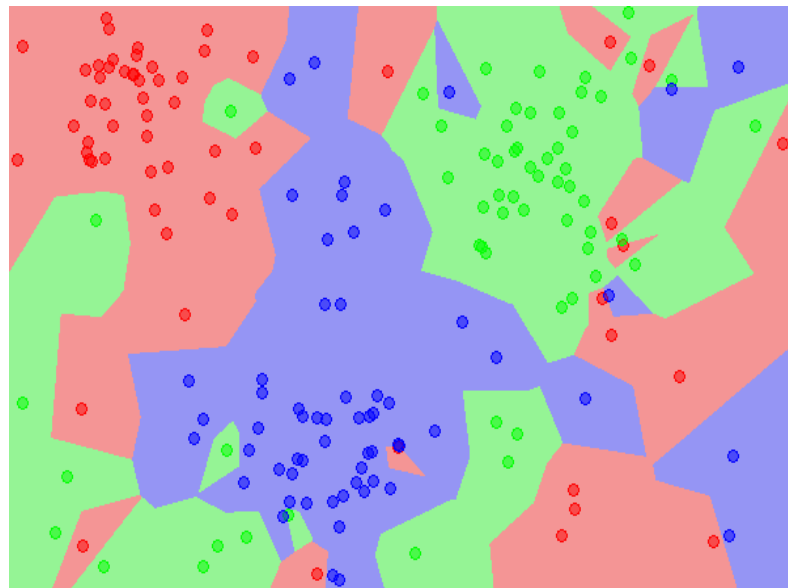
Bước 2: Cô lập một vòng tròn bán kính xung quanh điểm dữ liệu mới nhỏ nhất sao cho số lượng hàng xóm là bằng K.

Bước 3: Giữa các hàng xóm, chúng ta sẽ đếm số lượng điểm dữ liệu trong mỗi danh mục.

Bước 4: Chúng ta đánh giá khoảng cách Euclide giữa các điểm lân cận và điểm dữ liệu mới, nếu tổng các khoảng cách này là nhỏ nhất, điểm mới được gán cho danh mục tương ứng. Công thức tính khoảng cách Euclide:

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (1.1)$$

Khoảng cách Euclid giữa hai điểm p và q là chiều dài đoạn thẳng **pq**. Trong hệ tọa độ Descartes nếu $p = (p_1, p_2, \dots, p_n)$ và $q = (q_1, q_2, \dots, q_n)$ là hai điểm trong không gian Euclide n chiều, thì khoảng cách từ p đến q được tính như công thức (1.1)



Hình 2.3: KNN cho bài toán phân lớp [5].

Hình trên đây là một ví dụ về KNN trong bài toán phân lớp với $K = 1$. Đây là bài toán Classification với 3 lớp: Đỏ, Lam, Lục. Mỗi điểm dữ liệu mới sẽ được gán nhãn theo màu của điểm mà nó thuộc về. Trong hình này, có một vài vùng nhỏ

xem lẫn vào các vùng lớn hơn khác màu. Ví dụ có một điểm màu Lục ở gần góc 11 giờ nằm giữa hai vùng lớn với nhiều dữ liệu màu Đỏ và Lam. Điểm này rất có thể là nhiễu. Dẫn đến nếu dữ liệu test rơi vào vùng này sẽ có nhiều khả năng cho kết quả không chính xác.[5]

Mặc dù có hạn chế nhạy cảm với nhiễu, KNN vẫn là một giải pháp đầu tiên nên nghĩ tới khi giải quyết một bài toán phân lớp. Khi giải quyết các bài toán học máy nói chung, không có mô hình đúng hay sai, chỉ có các mô hình cho độ chính xác khác nhau với độ phức tạp khác nhau. Chúng ta luôn cần một mô hình đơn giản để giải quyết bài toán, sau đó dần dần tìm cách tăng chất lượng của mô hình.

Ưu điểm của KNN

- Độ phức tạp tính toán của quá trình huấn luyện là bằng 0.
- Việc dự đoán kết quả của dữ liệu mới rất đơn giản (sau khi đã xác định được các điểm lân cận).
- Không cần giả sử về phân phối của các lớp.

Nhược điểm của KNN

- KNN rất nhạy cảm với nhiễu khi K nhỏ.
- KNN là một thuật toán mà mọi tính toán đều nằm ở khâu kiểm thử, trong đó việc tính khoảng cách tới từng điểm dữ liệu trong tập huấn luyện tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.

2.2.3. Thuật toán Naive Bayes.

Naive Bayes là một thuật toán phân loại cho các vấn đề phân loại nhị phân (hai lớp) và đa lớp. Kỹ thuật này dễ hiểu nhất khi được mô tả bằng các giá trị đầu vào nhị phân hoặc phân loại. Nó được gọi là Naive Bayes bởi vì việc tính toán xác suất cho mỗi giả thuyết được đơn giản hóa để làm cho phép tính của họ có thể thực hiện được. [6]

Phương pháp này dựa trên toán học xác suất mà chúng ta đã học, công thức được thể hiện là:

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)} \quad (1.2)$$

Trong đó A,B là hai biến cố. với $P(B) > 0$. Trong bài toán phân lớp chúng ta muốn xác định giá trị $P(B/A)$ là xác suất để giả thiết B là đúng với chứng cứ A thuộc vào lớp C với điều kiện ra đã biết các thông tin mô tả A. $P(B|A)$ là một xác suất hậu nghiệm (posterior probability hay posteriori probability) của B với điều kiện A.

2.2.3. Thuật toán Random Forest.

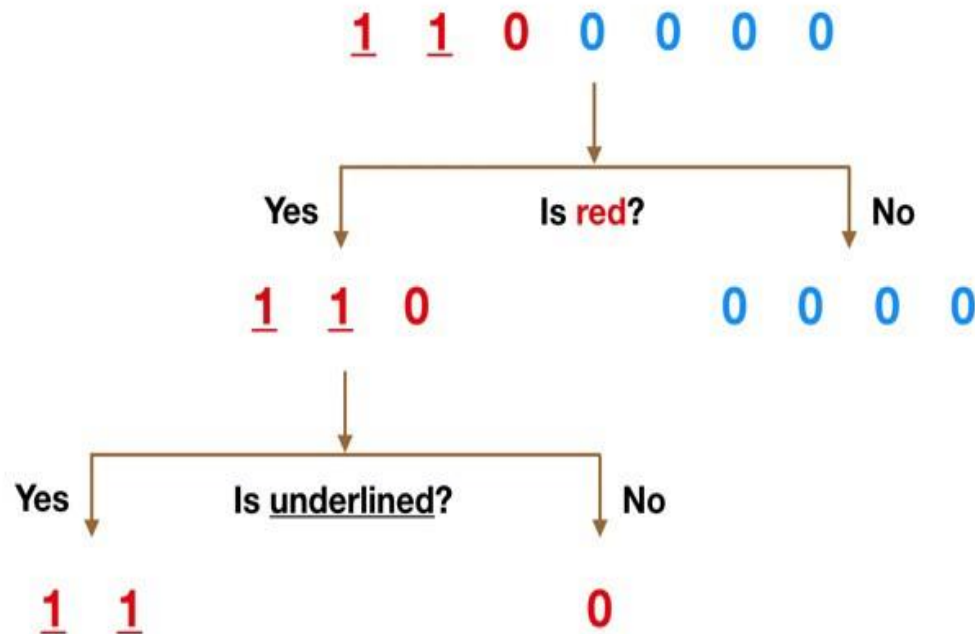
Thuật toán Random Forest là một thuật toán học máy có thể sử dụng để giải cả bài toán phân loại(classification) và hồi quy(regression). Nó làm việc bằng cách xây dựng một tập hợp các cây quyết định trong quá trình training, sau đó kết hợp kết quả trả về của mỗi cây để đưa ra quyết định dự đoán. cuối cùng. Thuật toán Random Forest có một số lợi thế như khả năng miễn nhiễm với những dữ liệu vô nghĩa, các đặc trưng không quan trọng và nhầm lẫn trong dữ liệu đầu vào.[7]

Đầu tiên, thuật toán này xây dựng một cây quyết định từ những dữ liệu này để có một mô hình. Tùy thuộc vào công thức thông tin entropy, thuật toán sẽ xây dựng quy tắc quyết định để có kết quả mong muốn. Tuy nhiên, nếu chúng ta dựa trên một cây quyết định, nó có thể gây ra rất nhiều lỗi. Do đó, chúng ta nên sử dụng nhiều cây quyết định với điểm khởi tạo ngẫu nhiên, sau đó tổng hợp chúng lại với nhau, điều này cho chúng ta kết quả an toàn hơn. Điều này tạo thành Rừng ngẫu nhiên, trong đó từ Rừng có nghĩa là quyết định hình thành từ một số cây.

Random Forest có điểm mạnh gì ?[8]

- Random Forest algorithm có thể sử dụng cho cả bài toán Classification và Regression.
- Random Forest làm việc được với dữ liệu thiếu giá trị.

- Khi Forest có nhiều cây hơn, chúng ta có thể tránh được việc Overfitting với tập dữ liệu.
- Có thể tạo mô hình cho các giá trị phân loại.



Hình 2.4: Ví dụ về cây quyết định [9].

Ví dụ cây quyết định đơn giản [Hình 2.8]. Nó có lẽ dễ hiểu hơn nhiều về cách cây quyết định hoạt động thông qua một ví dụ. Hãy tưởng tượng rằng tập dữ liệu của chúng ta bao gồm các số ở đầu hình bên trái. Chúng ta có hai 1 và năm 0 (1 và 0 là các lớp của chúng tôi) và mong muốn tách các lớp bằng các tính năng của chúng. Các tính năng là màu sắc (đỏ so với màu xanh) và quan sát có được gạch chân hay không. Màu sắc có vẻ như là một tính năng khá rõ ràng để phân chia vì tất cả nhưng một trong số 0 là màu xanh lam. Vì vậy, chúng ta có thể sử dụng câu hỏi, Có phải nó màu đỏ không? Để phân chia nút đầu tiên của chúng ta. Bạn có thể nghĩ về một nút trong cây là điểm mà đường dẫn chia thành hai - các quan sát đáp ứng các tiêu chí đi xuống nhánh Có và các nút không đi xuống nhánh Không.

Chi nhánh No (blues) hiện tại đều là 0 nên chúng tôi đã hoàn thành ở đó, nhưng chi nhánh Yes của chúng tôi vẫn có thể được phân chia thêm. Bây giờ chúng ta có thể sử dụng tính năng thứ hai và hỏi, Có phải nó được gạch chân không? Để tạo ra sự phân chia thứ hai..

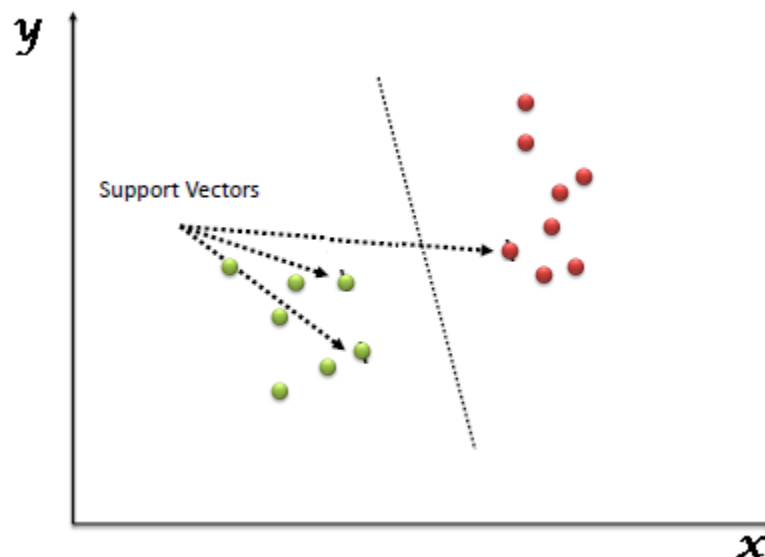
Hai số 1 được gạch chân đi xuống phân nhóm Có và số 0 không được gạch chân sẽ đi xuống phân nhóm bên phải và tất cả chúng ta đã hoàn tất. Cây quyết định của chúng ta đã có thể sử dụng hai tính năng để phân tách dữ liệu một cách hoàn hảo.[9]

Rõ ràng trong cuộc sống thực, dữ liệu của chúng ta sẽ không rõ ràng nhưng logic mà cây quyết định sử dụng vẫn giữ nguyên. Tại mỗi nút, nó sẽ hỏi -

2.2.4. Thuật toán SVM (Support Vector Machine).

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Thuật toán SVM sử dụng nguyên tắc lề tối đa để áp dụng một siêu phẳng tối ưu giữa các điểm dữ liệu. Các vector hỗ trợ rất quan trọng vì chúng xác định trọng lượng và cũng là sự hội tụ của các phân loại. Một hàm kernel được sử dụng để xây dựng mô hình lề tối đa và tìm siêu phẳng tối ưu.

Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu [Hình 2.9] là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" phân chia các lớp. Đường bay - nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.[10]



Hình 2.5: Thuật toán SVM [10].

Support Vectors hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát, Support Vector Machine là một biên giới để chia hai lớp tốt nhất.

2.3. Kết chương.

Chương này giúp chúng ta cái nhìn tổng quan về các loại học máy đồng thời đi sâu tìm hiểu về học có giám sát và các thuật toán nhằm chọn ra phương pháp phù hợp với bài toán phân loại trái cây.

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ.

3.1. Giao diện máy khách.

3.1.1. Thư viện PyQt

PyQt là Python interface của Qt, kết hợp của ngôn ngữ lập trình Python và thư viện Qt, là một thư viện bao gồm các thành phần giao diện điều khiển (widgets graphical control elements). PyQt được phát triển bởi Riverbank Computing Limited.

PyQt API bao gồm các module bao gồm số lượng lớn với các classes và functions hỗ trợ cho việc thiết kế ra các giao diện giao tiếp với người dùng của các phần mềm chức năng. [11]

Các class của PyQt5 được chia thành các module, bao gồm :

- + QtCore : là module bao gồm phần lõi không thuộc chức năng GUI, ví dụ dùng để làm việc với thời gian, file và thư mục, các loại dữ liệu, streams, URLs, mime type, threads hoặc processes.
- + QtGui : bao gồm các class dùng cho việc lập trình giao diện (windowing system integration), event handling, 2D graphics, basic imaging, fonts và text.
- + QtWidgets : bao gồm các class cho widget, ví dụ : button, hộp thoại, ... được sử dụng để tạo nên giao diện người dùng cơ bản nhất
- + QtMultimedia : thư viện cho việc sử dụng âm thanh, hình ảnh, camera,...
- + QtBluetooth : bao gồm các class giúp tìm kiếm và kết nối với các thiết bị có giao tiếp với phần mềm.
- + QtNetwork : bao gồm các class dùng cho việc lập trình mạng, hỗ trợ lập trình TCP/IP và UDP client , server hỗ trợ việc lập trình mạng.
- + QtPositioning : bao gồm các class giúp việc hỗ trợ xác định vị.
- + Enginio : module giúp các client truy cập các Cloud Services của Qt.

- + QtWebSockets : cung cấp các công cụ cho WebSocket protocol.
- + QtWebKit : cung cấp các class dùng cho làm việc với các trình duyệt Web dựa trên thư viện WebKit2.
- + QtWebKitWidgets : các widget cho WebKit.
- + QtXml : các class dùng cho làm việc với XML file.
- + QtSvg : dùng cho hiển thị các thành phần của SVG file.
- + QSql : cung cấp các class dùng cho việc làm việc với dữ liệu.
- + QTest : cung cấp các công cụ cho phép test các đơn vị của ứng dụng.

```
2
3  from PyQt5 import QtCore, QtWidgets
4  from PyQt5.QtWidgets import QFileDialog
5  import os
6  import requests
7  import csv
8  import json
9  import matplotlib.pyplot as plt
10
```

Hình 3.1: Import PyQt5 và các thư viện cần thiết.

Chúng ta có thể dễ dàng import thư viện PyQt5 [Hình 3.1].

3.1.2. *Thư viện Matplotlib.*

Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của bạn và Matplotlib là một trong những giải pháp như vậy cho người dùng Python. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotlib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở. [12]

Một Matplotlib figure có thể được phân loại thành nhiều phần như dưới đây:

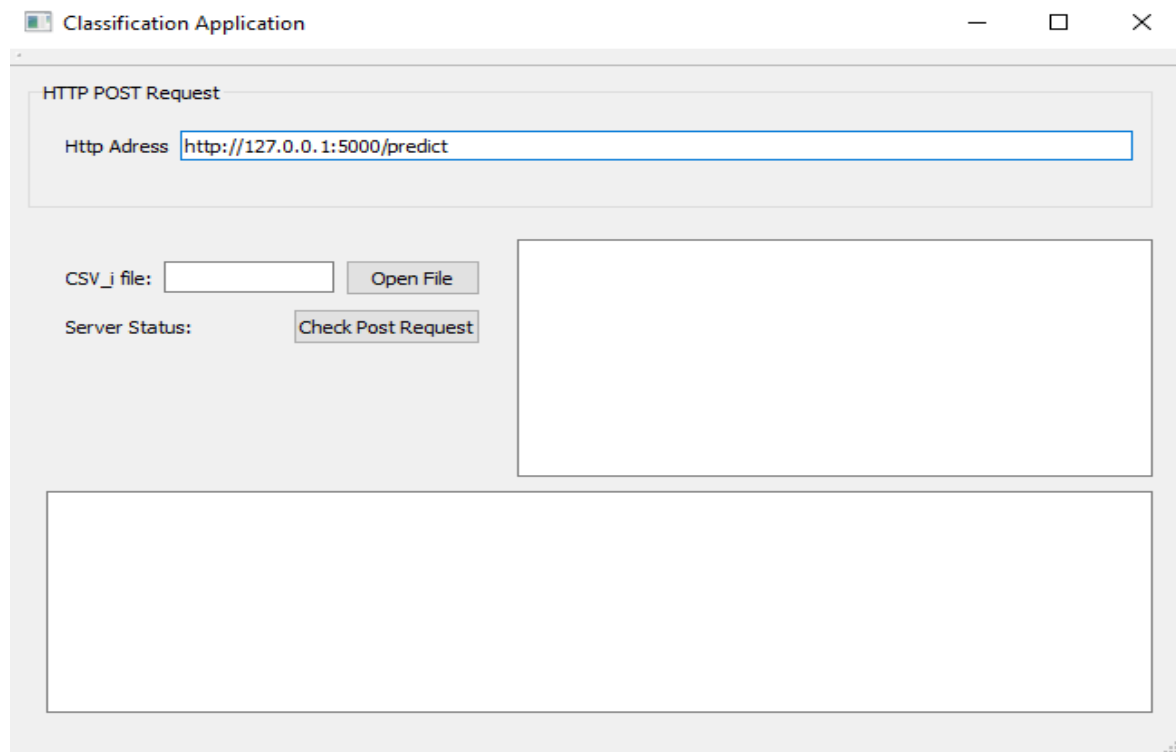
- Figure: Như một cái cửa sổ chứa tất cả những gì bạn sẽ vẽ trên đó.

- Axes: Thành phần chính của một figure là các axes (những khung nhỏ hơn để vẽ hình lên đó). Một figure có thể chứa một hoặc nhiều axes. Nói cách khác, figure chỉ là khung chứa, chính các axes mới thật sự là nơi các hình vẽ được vẽ lên.
- Axis: Chúng là dòng số giống như các đối tượng và đảm nhiệm việc tạo các giới hạn biểu đồ.
- Artist: Mọi thứ mà bạn có thể nhìn thấy trên figure là một artist như Text objects, Line2D objects, collection objects. Hầu hết các Artists được gắn với Axes.

Pyplot là một module của Matplotlib cung cấp các hàm đơn giản để thêm các thành phần plot như lines, images, text, v.v. vào các axes trong figure.

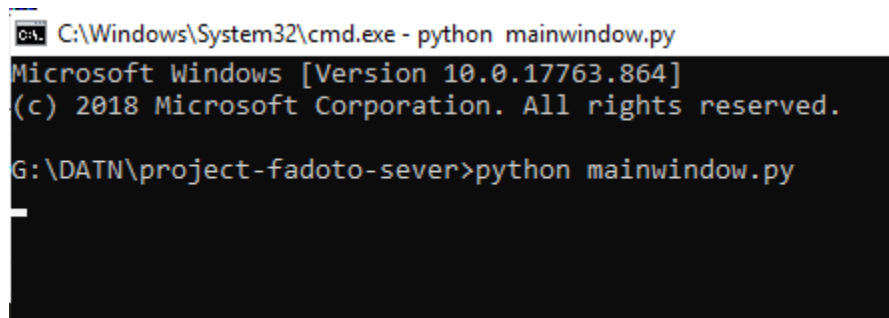
3.1.3. Giao diện.

Giao diện và một số chức năng.



Hình 3.2: Giao diện máy khách.

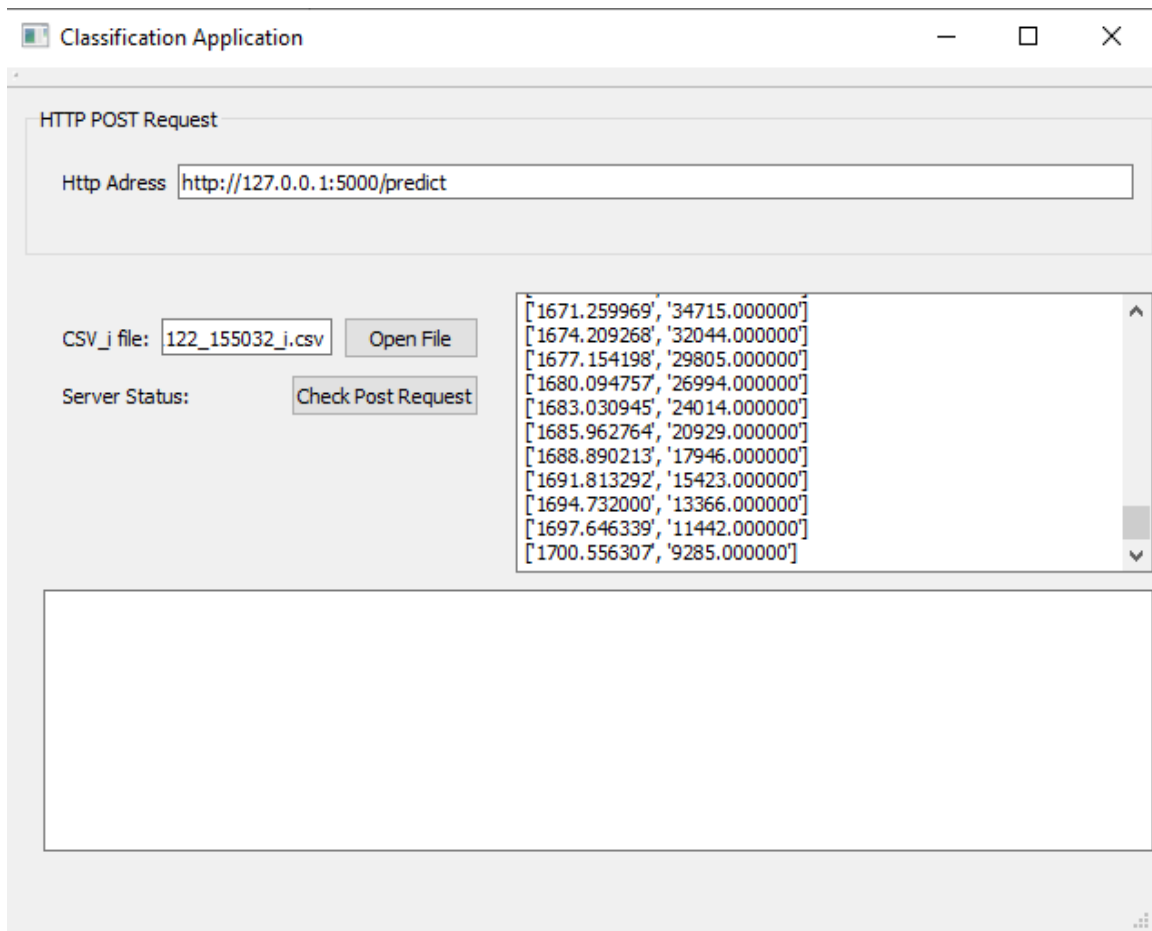
Để mở giao diện, tại command line gõ câu lệnh python mainwindow.py [Hình 3.6]



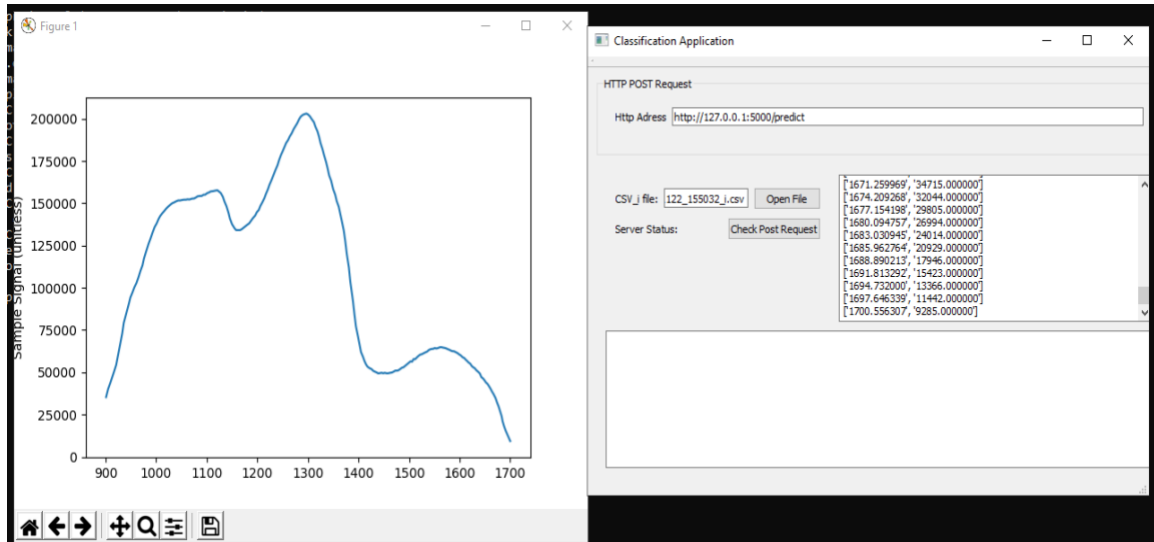
```
C:\Windows\System32\cmd.exe - python mainwindow.py
Microsoft Windows [Version 10.0.17763.864]
(c) 2018 Microsoft Corporation. All rights reserved.

G:\DATN\project-fadoto-sever>python mainwindow.py
```

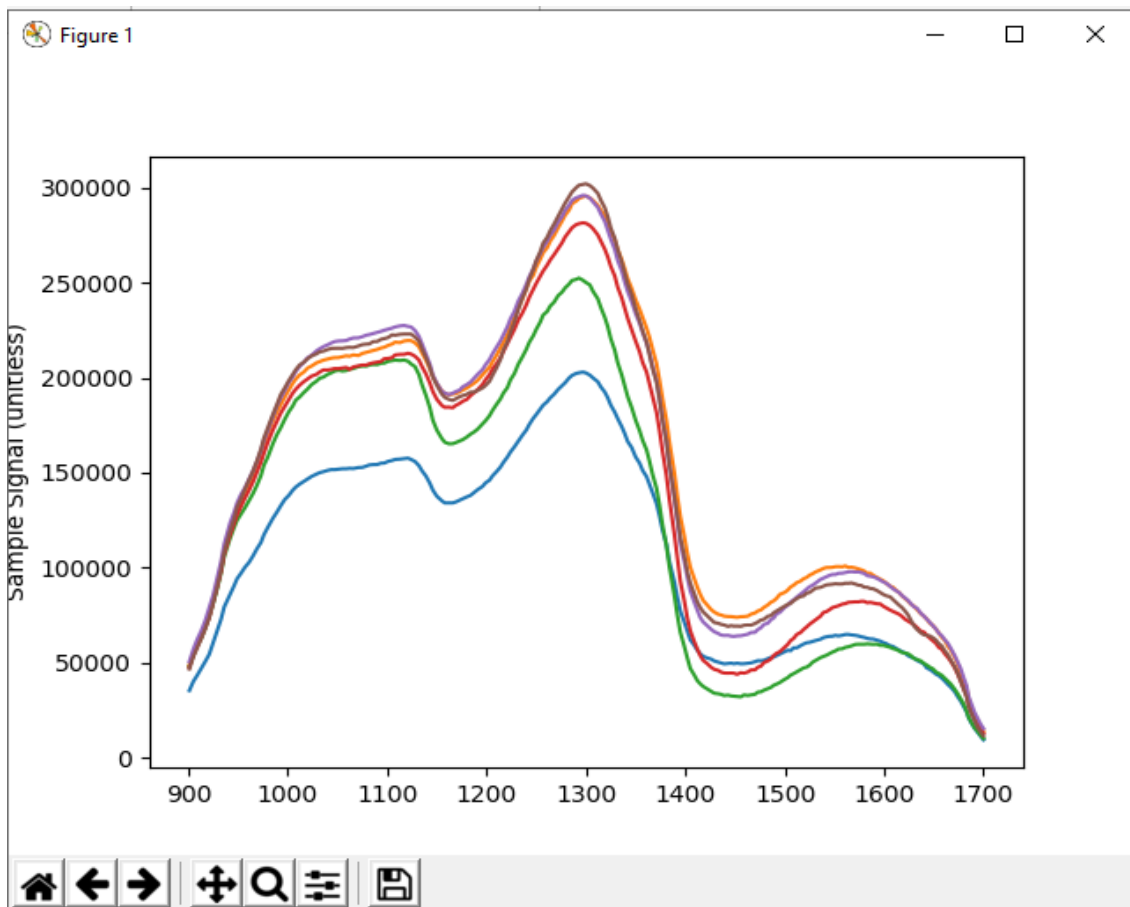
Hình 3.3: Mở giao diện



Hình 3.4: Dữ liệu được đọc và ghi ra màn hình.

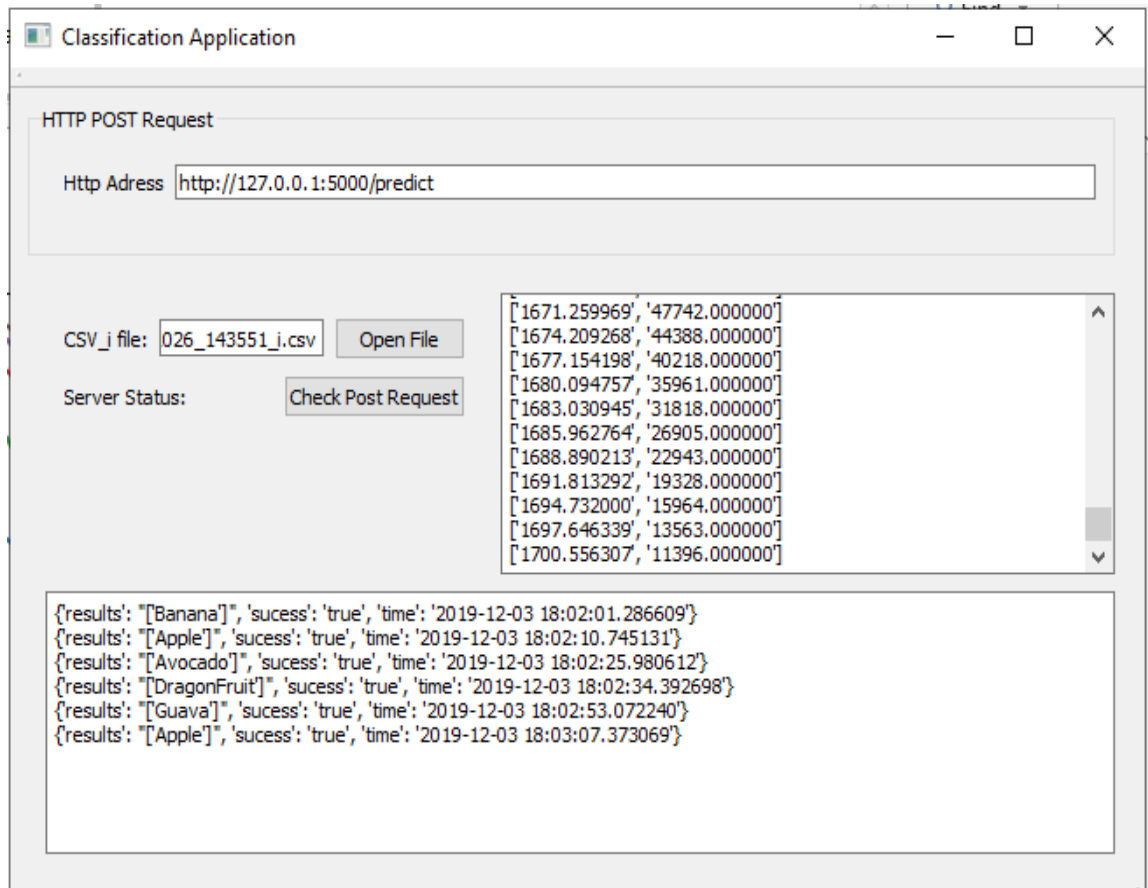


Hình 3.5: Dữ liệu được vẽ dưới dạng biểu đồ.



Hình 3.6: So sánh sự biến thiên của phổ cường độ giữa nhiều file dữ liệu của các loại quả khác nhau.

Tại giao diện máy khách [Hình 3.2] có các chức năng như mở tệp, đọc tệp và ghi ra màn hình [Hình 3.4]. Ngoài ra, có thêm chức năng gửi dữ liệu cho máy chủ và nhận lại kết quả ghi lại trên màn hình kết quả [Hình 3.7]. Chương trình cũng cho phép vẽ sự biến thiên của phổ cường độ của các loại quả để quan sát [Hình 3.6].



Hình 3.7: Kết quả trả về từ máy chủ.

3.2. Xây dựng máy chủ.

3.2.1. Thư viện Numpy.

Numpy là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó. Numpy đặc biệt hữu ích khi thực hiện các hàm liên quan tới Đại Số Tuyến Tính. [13].

Để cài đặt numpy nếu bạn có Anaconda chỉ cần gõ **conda install numpy** hoặc sử dụng tools pip **pip install numpy**.

3.2.2. Thư viện Pandas.

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi. [14].

Tại sao sử dụng thư viện pandas?

- DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
- Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
- Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
- Dễ dàng thay đổi bố cục của dữ liệu;
- Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
- Có thể thêm, xóa các cột dữ liệu;
- Tập hợp hoặc thay đổi dữ liệu với group by cho phép bạn thực hiện các toán tử trên tập dữ liệu;
- Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;
- Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;

- Tối ưu về hiệu năng;
- Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo,...

Cài đặt Pandas

Sử dụng pip: `pip install pandas`

Sử dụng conda: `conda install pandas`

3.2.3. Thư viện requests.

Requests module là một thư viện hỗ trợ chúng ta có thể gửi bất kỳ một loại request HTTP nào một cách đơn giản nhất. Ta có thể thực hiện các tác vụ như gửi request tới server cũng như xử lý response một cách dễ dàng.

Cài thư viện Requests.

Có thể sử dụng command line để cài thư viện request bằng cách sử dụng:

Sử dụng pip: `pip install requests`

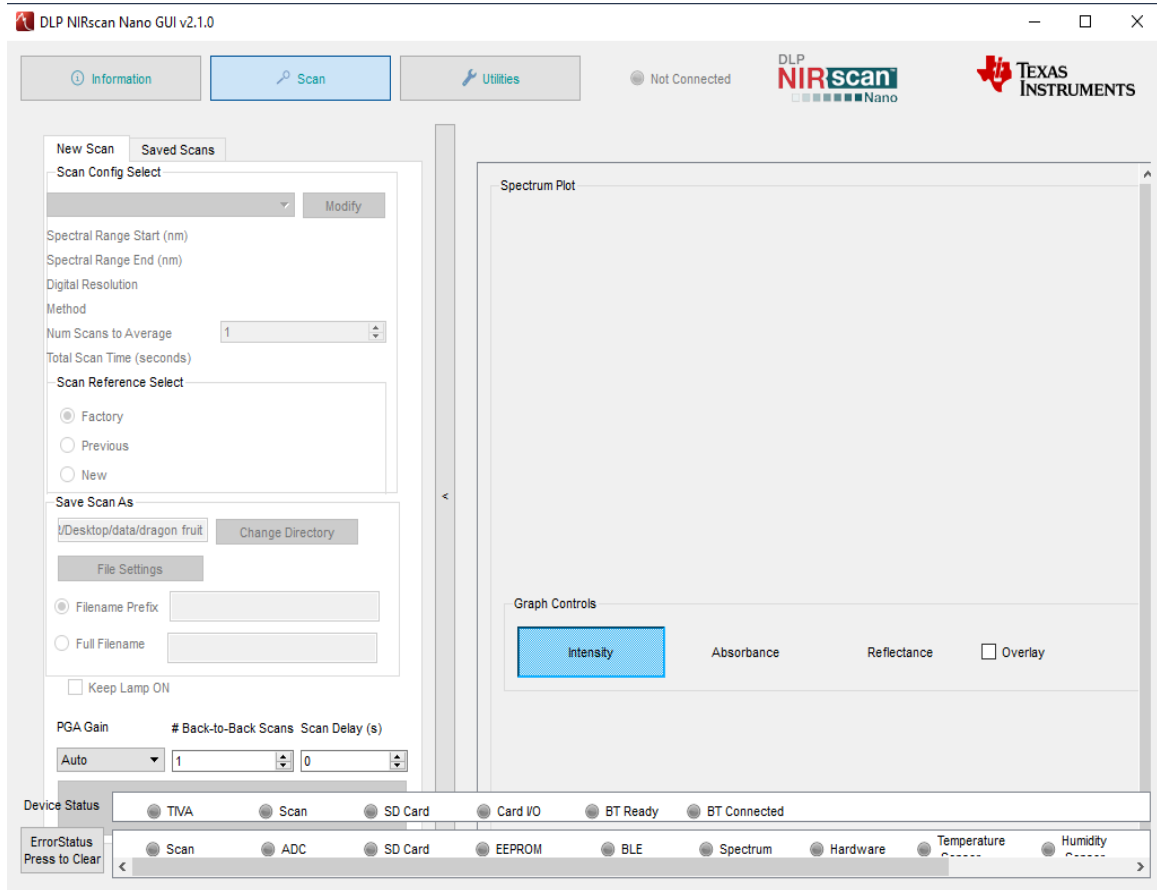
3.3. Huấn luyện mô hình.

3.3.1. Thu thập dữ liệu.

Khi thu thập dữ liệu, tôi sử dụng thiết bị DLP NIRscan Nano và phần mềm DLP NIRscan Nano GUI v2.1.0 [hình 3.8] kèm theo của hãng Texas Instrument. Phần mềm trích xuất dữ liệu khá đầy đủ các số liệu như bước sóng, phổ cường độ hay độ hấp thụ. Trong đề tài này tôi chỉ sử dụng phổ cường độ để huấn luyện mô hình. Phần mềm này có các tính năng sau:

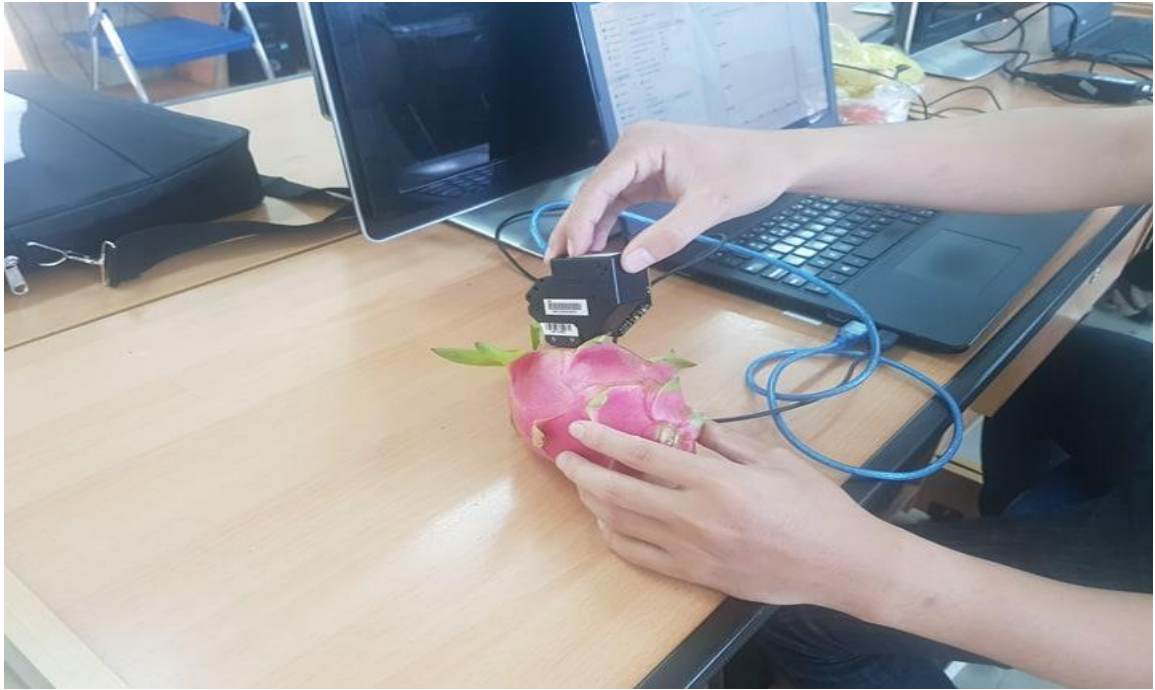
- Phần mềm cho phép thiết bị kết nối thông qua cổng USB hoặc Bluetooth của máy tính
- Phần mềm cho phép quét và trích xuất ra file có đuôi .csv có các chỉ số khác nhau như cường độ, độ hấp thụ và độ phản xạ.
- Bước sóng sử dụng trong khi quét được để ở giá trị mặc định là từ 900nm đến 1700nm.

- Kết quả có thể sai lệch giữa các lần đo vì nó tùy thuộc vào điều kiện nhiệt độ và ánh sáng tại nơi đo. Điều kiện thích hợp để sử dụng máy là nhiệt độ phòng khoảng 25 °C.

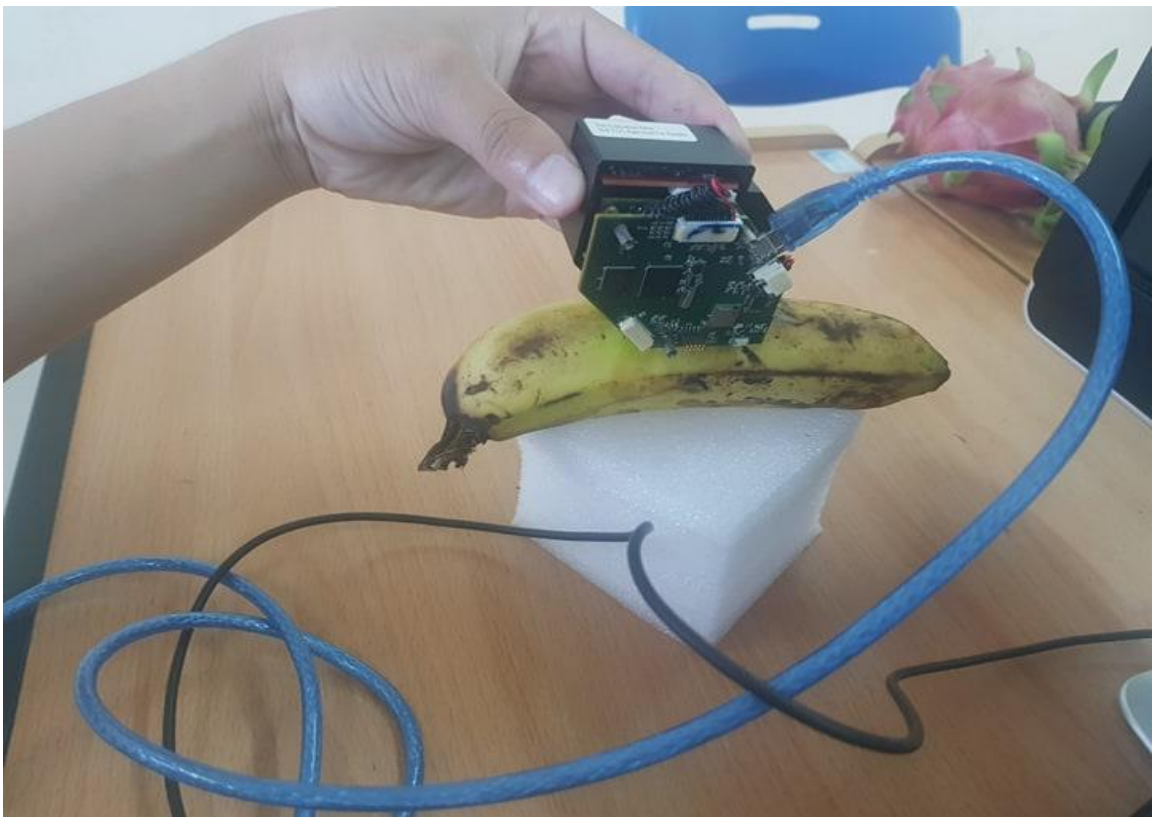


Hình 3.8: Phần mềm DLP NIRscan Nano GUI v2.1.0.

Tôi thu thập dữ liệu từ một số loại trái cây: Bơ, Thanh Long, Xoài, Chuối, Khoai Tây, Cà Chua, Ổi, Vải, Táo. Tôi quét 10 lần mỗi quả khoảng gần 100 mẫu dữ liệu. Tổng cộng có khoảng gần 900 mẫu dữ liệu để tạo thành tập dữ liệu sử dụng cho học máy.



Hình 3.9: Thu thập dữ liệu từ quả Thanh Long.



Hình 3.10: Thu thập dữ liệu từ quả Chuối.

Cách thu thập dữ liệu.

Đầu tiên, để thu được dữ liệu tốt, chúng ta cần đặt máy đo ở phòng có nhiệt độ và ánh sáng thích hợp. Tiếp theo, bật phần mềm thu thập trước khi kết nối với cổng USB. Khi thiết bị bắt đầu hoạt động, đèn led màu xanh lá cây sẽ sáng lên. Chúng ta có thể bắt đầu thu thập. Để thu được dữ liệu chuẩn, cần đặt ống kính lên mặt phẳng của trái cây [Hình 3.9] để đảm bảo ánh sáng chiếu vào một góc vuông. Sau đó, nhấn nút Scan trên phần mềm để quét và lưu dữ liệu vào máy tính. Hình trên là hai ví dụ về cách để thu thập dữ liệu phổ từ trái cây sử dụng máy đo phổ cận hồng ngoại.[Hình 3.9] [Hình 3.10]

3.3.2. Mô tả dữ liệu thô.

Máy đo NIR có thể xuất dữ liệu theo 3 định dạng: cường độ, độ hấp thụ và độ phản xạ [Hình 3.11]. Có giá trị cường độ là giá trị thực sự được đo bằng cảm biến bên trong của máy quang phổ, 2 biến khác được tính từ công thức như sau:

Độ hấp thụ = $-\log_{10}$ (Tham chiếu cường độ / cường độ)

Độ phản xạ = cường độ / cường độ tham chiếu

Với tham chiếu cường độ là cường độ ánh sáng phản xạ trong cảm biến khi chúng ta quét một vật thể có độ phản xạ 100%.

	A	B	C	D
22	Wavelength (nm)	Absorbance (AU)	Reference Signal (unitless)	Sample Signal (unitless)
23	901.800507	0.30506	89642	44407
24	905.72725	0.305507	100046	49510
25	909.649623	0.302843	108729	54138
26	913.567626	0.305989	117933	58297
27	917.481259	0.307392	128117	63127
28	921.390522	0.311952	140731	68618
29	925.295415	0.315619	155925	75387
30	929.195937	0.323116	173200	82306
31	933.09209	0.325804	190865	90141
32	936.983873	0.32694	207703	97837
33	942.166118	0.330704	228011	106476
34	946.047703	0.333018	241582	112214
35	949.924919	0.338802	253596	116236
36	953.797764	0.345974	265238	119581
37	957.66624	0.353672	277281	122814
38	961.530345	0.358662	288983	126535
39	965.39008	0.363555	300757	130215
40	969.245445	0.366068	312492	134515
41	973.09644	0.36624	324122	139466
42	976.943065	0.367004	335288	144017
43	980.78532	0.367767	345344	148076
44	985.901528	0.364783	357815	154481

Hình 3.11: Một file dữ liệu có chứa đầy đủ cả ba định dạng.

Nhưng ở bài toán này, tôi chỉ sử dụng file dữ liệu phổ cường độ để làm dữ liệu đầu vào cho học máy [Hình 3.12]. Khi sử dụng phần mềm của nhà sản xuất, chúng ta có thể dễ dàng tùy chỉnh để trích xuất file dữ liệu mong muốn.

	A	B	C	D
1	Wavelength	Sample Signal (unitless)		
2	901.8005	44407		
3	905.7273	49510		
4	909.6496	54138		
5	913.5676	58297		
6	917.4813	63127		
7	921.3905	68618		
8	925.2954	75387		
9	929.1959	82306		
10	933.0921	90141		
11	936.9839	97837		
12	942.1661	106476		
13	946.0477	112214		
14	949.9249	116236		
15	953.7978	119581		
16	957.6662	122814		
17	961.5303	126535		
18	965.3901	130215		
19	969.2454	134515		
20	973.0964	139466		
21	976.9431	144017		
22	980.7853	148076		
23	985.9015	154481		

Hình 3.12: Một file dữ liệu chỉ có chỉ số bước sóng và phổ cường độ.

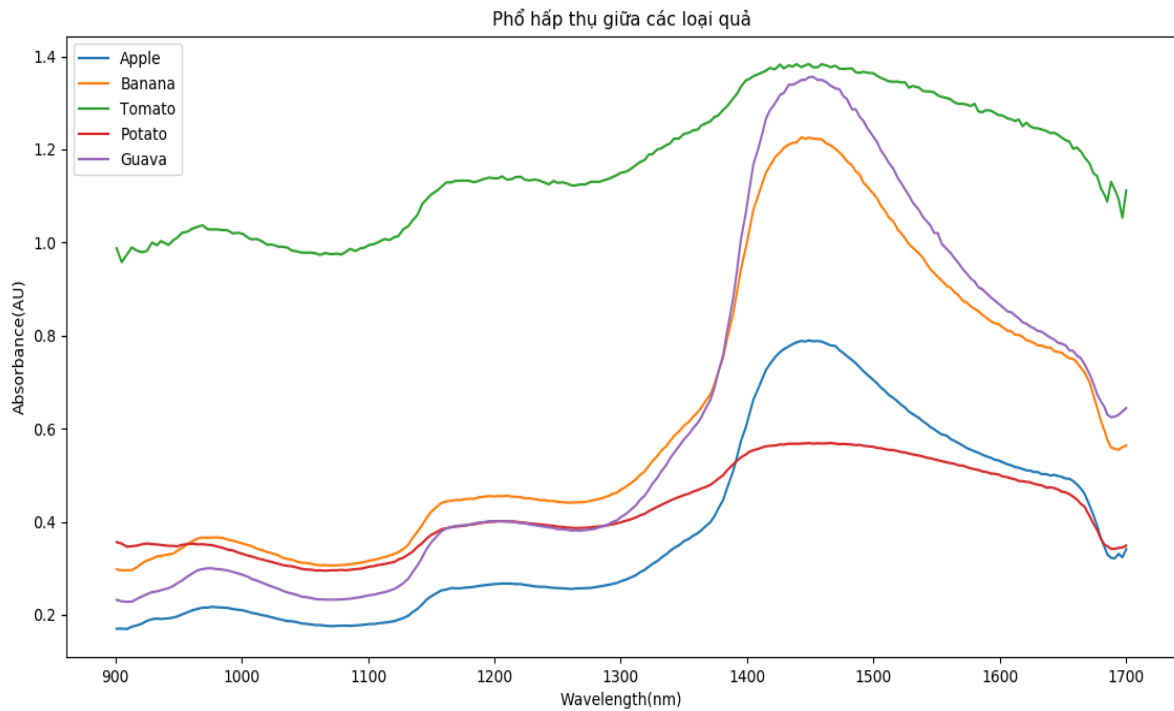
3.3.3. Thống kê dữ liệu.

Kết quả tôi thu được dữ liệu từ các loại trái cây. Số lượng dữ liệu thu được để sử dụng trong huấn luyện mô hình và kiểm tra kết quả.

STT	Tên quả	Số lượng dữ liệu dùng trong huấn luyện	Số lượng dữ liệu dùng trong kiểm tra
1	Chuối	120	30
2	Táo	80	20
3	Ổi	80	20
4	Xoài	80	20
5	Bơ	120	30
6	Vải	80	20
7	Thanh Long	80	20
8	Cà Chua	100	30
9	Khoai Tây	80	20

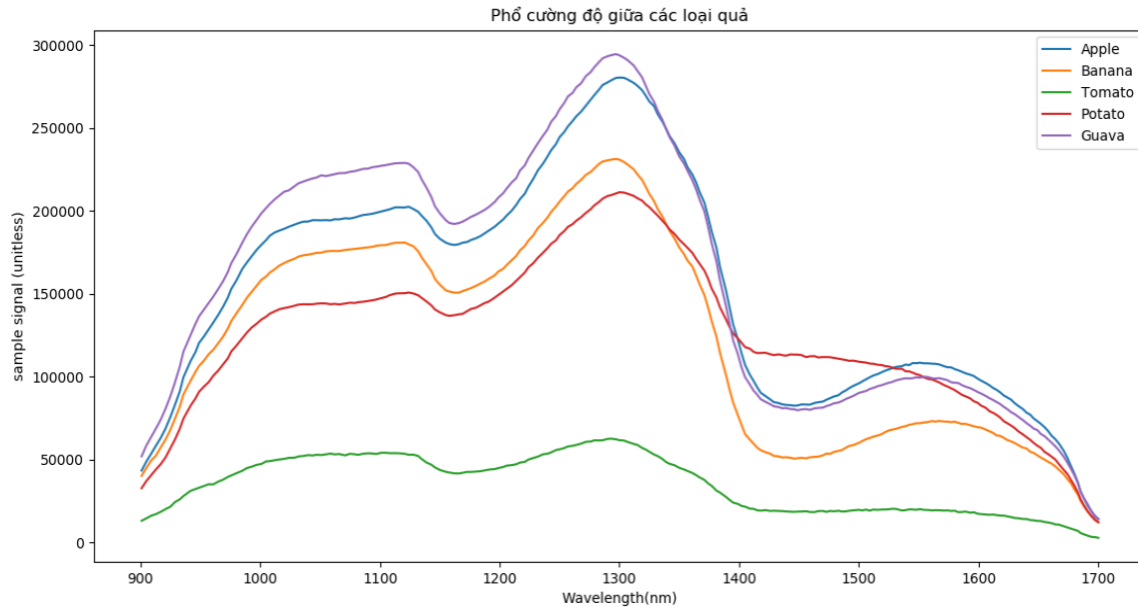
3.3.4. Tiền xử lý dữ liệu và trích chọn đặc trưng.

Đây là bước cực kỳ quan trọng, vì nó ảnh hưởng tới kết quả cũng như độ chính xác sau này. Đầu tiên, để xem quang phổ của các loại trái cây khác nhau tôi vẽ chúng lên biểu đồ để quan sát.



Hình 3.13: Quang phổ của độ hấp thụ.

Quan sát hình 3.13. Chúng ta có thể dễ dàng phân biệt bằng mắt thường phổ hấp thụ của quả Cà Chua so với các loại quả khác. Tuy nhiên, khi so sánh giữa các loại quả khác với nhau, thật khó để phân biệt giữa chúng kể cả với mắt thường. Tôi chuyển qua vẽ và quan sát phổ cường độ.

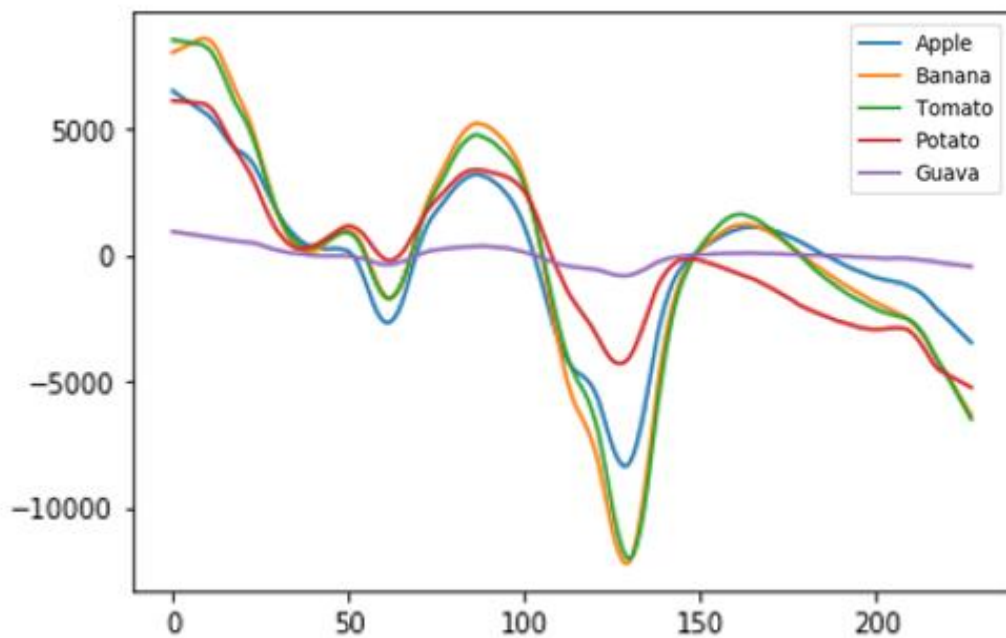


Hình 3.14: Phổ cường độ.

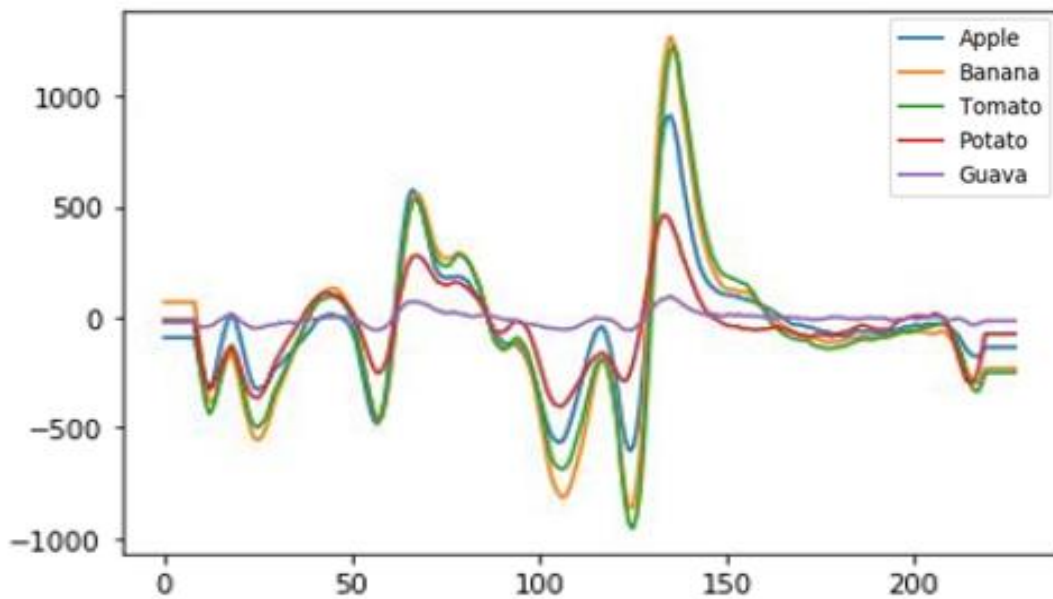
Chúng ta quan sát [Hình 3.14] rằng phổ cận hồng ngoại (near-infrared hay NIR) của các loại trái cây khác nhau là khá tương tự nhau, đặc biệt đối với Táo và Ổi rất khó quan sát sự khác biệt ngay cả với mắt thường. Vì vậy, tôi chuyển sang bước tiếp theo là trích chọn các đặc trưng có tính phân biệt giữa các loại quả phù hợp với việc nhận dạng. Phổ cường độ là dữ liệu thô thu được từ cảm biến NIR, vì vậy trong trường hợp này tôi sẽ tìm đặc trưng dựa trên phổ cường độ của các loại quả.

Trích chọn đặc trưng.

Để trích chọn đặc trưng, tôi dựa vào kết quả trong bài báo [15]. Chúng tôi coi mỗi giá trị cường độ ở mỗi bước sóng của phổ hồng ngoại là một biến dự báo. Vì mỗi dữ liệu phổ NIR có 228 giá trị bước sóng khác nhau, tôi sẽ dùng tất cả giá trị đó như dữ liệu đầu vào của mô hình học máy. Thử nghiệm cho thấy hiệu suất nhận dạng rất thấp do mô hình học máy khó “học” được các sự khác biệt nhỏ trong phổ của các loại quả. Đó là lý do tại sao tôi dùng phép lấy đạo hàm để tìm ra các sự khác biệt rõ ràng hơn giữa phổ của các loại quả.[Hình 3.15]



Hình 3.15. Đạo hàm bậc 1 của phổ NIR của các loại quả.



Hình 3.16: Đạo hàm bậc 2 của phổ NIR của các loại quả.

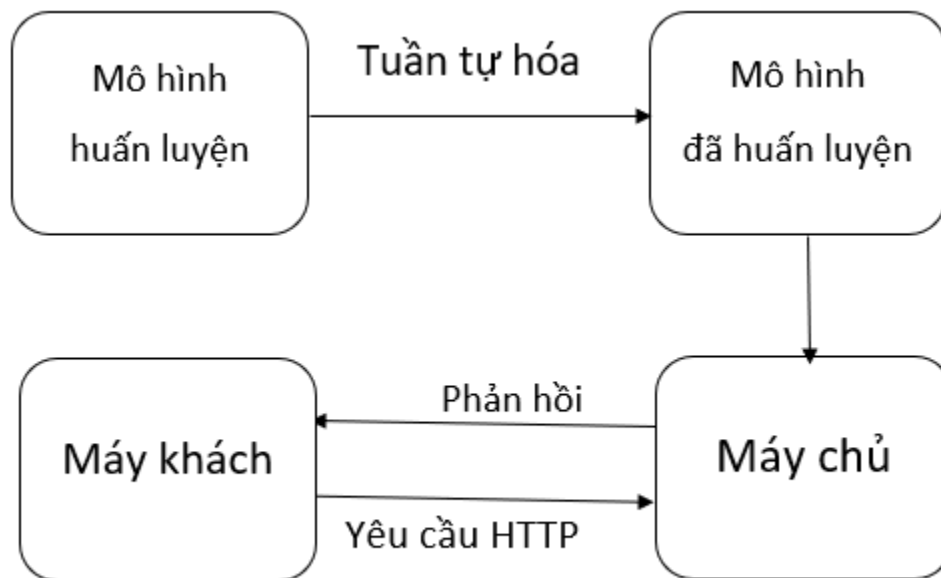
Trong trường hợp đạo hàm bậc hai [Hình 3.16], chúng ta quan sát thấy các biến thể sắc nét hơn giúp máy dễ dàng hơn trong việc trích chọn đặc trưng để đưa ra dự đoán.

Với phương pháp này, chúng tôi có được độ chính xác lên tới hơn 90% với thuật toán KNN.

3.3.5. Triển khai học máy trên máy chủ.

Sau khi huấn luyện mô hình, chúng ta chuyển sang bước tiếp theo là lưu mô hình. Thật vậy, python cung cấp thư viện pickle, cho phép tuần tự hóa và giải tuần tự hóa các đối tượng python trong tệp nhị phân. Vì vậy, nhờ đó chúng ta có thể tuần tự hóa đối tượng trong kernel python một cách dễ dàng.

Trong bước thứ hai, chúng ta giải tuần tự hóa các tệp nhị phân mà tôi đã chuẩn bị, sau đó tải tệp đó vào kernel Python. Sau đó tôi sử dụng thư viện flask, thư viện này sẽ lấy dữ liệu đến theo yêu cầu của HTTP Post, sau đó gọi mẫu để trả về kết quả cho máy khách.[Hình 3.17]



Hình 3.17: Quá trình gửi yêu cầu từ máy khách và trả lại kết quả từ máy chủ.

3.4. Đánh giá kết quả.

3.4.1. Đánh giá thuật toán KNN.

Bảng 3.1 cho thấy hiệu suất hoạt động của thuật toán phân loại bằng KNN. Nó thể hiện độ chính xác khi thay đổi K và số các đặc trưng được trích xuất. Hệ thống có độ chính xác tốt nhất khi $K = 7$ với số chiều vec tơ đặc trưng là 288 với tỷ lệ là 86,49%. Sử dụng phương pháp đánh giá confusion matrix để đánh giá và đo độ chính xác tôi thu được bảng sau đây với thuật toán KNN.

Bảng 3.1: Độ chính xác với thuật toán KNN khi K thay đổi.

K	1	2	3	4	5	6	7	8	9	10
Độ chính xác(%)	83.41	82.98	82.98	83.46	83.95	84.98	86.49	93.51	83.13	80.00

3.4.2. So sánh với các thuật toán phân loại khác.

Khi so sánh độ chính xác với các thuật toán khác, tôi thu được kết quả như dưới đây.

- Với thuật toán SVM khi thay đổi các hàm nhân (kernal) ta có kết quả như bảng sau.

Bảng 3.2: Độ chính xác với thuật toán SVM.

Hàm nhân(kernal)	Độ chính xác (%)
Linear	85.24
Polynomial	83.90
Sigmoid	35.12
RBF	79.02

- Với thuật toán Naive Bayes, khi thay đổi các hàm tham số mô hình ta có kết quả như bảng sau.

Bảng 3.3: Độ chính xác với thuật toán Naive Bayes.

Mô hình	Độ chính xác (%)
Gaussian	81.46
Multinomial	80.24
Bernoulli	76.05

- Với thuật toán Random Forest, khi thay đổi các tham số trạng thái ngẫu nhiên với hàm RandomForestClassifier, ta có kết quả như bảng sau.

Bảng 3.4: Độ chính xác với thuật toán Random Forest.

Random_state	1	2	3	4	5	6	7	8	9	10
Độ chính xác(%)	81.95	80	83.41	80	80.49	80.49	81.46	82.93	80.49	82.4

3.4.3. Thảo luận.

Tất cả các phương pháp đều đạt độ chính xác trên 80% với 9 loại trái cây, các phương pháp như KNN giữ hiệu suất cao nhất. Điều này là do số lượng mẫu không đồng đều trên mỗi quả. Tuy nhiên, thuật toán KNN giữ hiệu suất cao nhất trong trường hợp quan trọng, với $K=7$ có độ chính xác là 86,49% Vì vậy, tôi chọn thuật toán KNN làm phân loại chính cho ứng dụng.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đồ án đã đạt được những kết quả sau:

Về mặt lý thuyết:

- Ứng dụng các kiến thức tìm hiểu được để thiết kế được giao diện cho máy khách.
- Sử dụng thư viện Numpy, Pandas để xây dựng Server.
- Thu thập được gần 900 mẫu dữ liệu từ hoa qua sử dụng máy đo quang phổ cận hồng ngoại.
- Sử dụng thuật toán KNN cho học máy.

Về mặt thực tiễn ứng dụng:

- Đã chạy thử và kết quả có độ chính xác hơn 85% cho mỗi loại trái cây.

Tuy nhiên, đồ án còn tồn tại các vấn đề như sau:

- Cần chạy được nhiều máy cùng một lúc.
- Chưa kiểm tra cùng lúc nhiều file của nhiều loại trái cây khác nhau.

2. Hướng phát triển thêm.

Bên cạnh kết quả chúng tôi thu được, vẫn còn những điều mà tôi không thể làm được. Điều đặc biệt là trả lời câu hỏi về ngưỡng nồng độ chất độc trong sản phẩm mà máy quang phổ có thể làm khác nhau.

Một điều quan trọng khác là việc sử dụng phổ cường độ, không phù hợp lắm. Thật vậy, chúng tôi đọc các giá trị cường độ đến trực tiếp từ cảm biến. Tuy nhiên, nếu chúng ta đọc trực tiếp từ cảm biến, giá trị này có thể thay đổi theo quy trình sản xuất, nhiệt độ, ánh sáng và tất cả các yếu tố khác. Do đó, chúng ta nên hiệu chỉnh cường độ tham chiếu và sử dụng phổ hấp thụ để có độ đồng nhất trong các phép đo.

TÀI LIỆU THAM KHẢO.

- [1] Texas Instruments Incorporate, “ DLP® NIRscan™ Nano EVM User's Guide”.
<http://www.ti.com/lit/ug/dlpu030g/dlpu030g.pdf>, 6/2015–Revised 8/2017.
- [2] wikipedia.org “Học có giám sát”
https://vi.wikipedia.org/wiki/Học_có_giám_sát/
- [3] machinelearningcoban.com “ Phân nhóm các thuật toán Machine Learning”,
<https://machinelearningcoban.com/2016/12/27/categories/> , 2016
- [4] “Bắt đầu với Scikit-learn” <http://ml4vn.blogspot.com/2017/08/bat-au-voi-scikit-learn-cac-khai-niem.html>, 2017
- [5] “K-nearest neighbors” <https://machinelearningcoban.com/2017/01/08/knn/>, 2017
- [6] Nguyễn Duy Sim “Phân loại bằng Naive Bayes” <https://viblo.asia/p/phan-loai-bang-naive-bayes-phan-1-naQZRJAjZvx>, 2018
- [7] Nguyễn Hồng Việt “Cài đặt bộ lọc Random Forest để giải bài toán”
<https://techblog.vn/cai-dat-bo-loc-random-forest-de-giai-bai-toan-ocr-trong-moi-truong->
- [8] Couhpcode.wordpress.com “random forest, thế nào là một rừng ngẫu nhiên”
<https://couhpcode.wordpress.com/2018/01/24/random-forest-the-nao-la-mot-rung-ngau-nhien/>, 2018
- [9] Tony Yiu “Understanding Random Forest”
<https://towardsdatascience.com/understanding-random-forest>. 6/2019
- [10] Nguyễn Long Trần “Một chút về thuật toán SVM” <https://viblo.asia/p/mot-chut-ve-thuat-toan-svm-support-vector-machine-algorithm->, 2017

- [11] Mlab, “Lập trình giao diện với PyQt5 cho RaspberryPi” <http://mlab.vn/17161-bai-7-lap-trinh-giao-dien-voi-pyqt5-cho-raspberrypi-phan-1.html>, 2017
- [12] Nguyễn Văn Hoàng, “Giới thiệu về Matplotlib (một thư viện rất hữu ích của Python dùng để vẽ đồ thị)” <https://viblo.asia/p/gioi-thieu-ve-matplotlib-mot-thu-vien-rat-huu-ich-cua-python-dung-de-ve-do-thi-yMnKMN6gZ7P>, 7/1019
- [13] Nguyễn Văn Hoàng “Giới thiệu về Numpy (một thư viện chủ yếu phục vụ cho khoa học máy tính của Python)” <https://viblo.asia/p/gioi-thieu-ve-numpy-mot-thu-vien-chu-yeu-phuc-vu-cho-khoa-hoc-may-tinh-cua-python-maGK7kz9Kj2>. 4/2019
- [14] Nguyễn Văn Hiếu “Pandas Python Tutorial” <https://viblo.asia/p/pandas-python-tutorial.2018>.
- [15] F. Kosmowski, T. Worku, “Evaluation of a miniaturized NIR spectrometer for cultivar identification: The case of barley, chickpea and sorghum in Ethiopia”, PLoS ONE 13(3): e0193620. <https://doi.org/10.1371/journal.pone.0193620>, 2018.