

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

NIÊN LUẬN CƠ SỞ
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

NHẬN DẠNG BỆNH HẠI TRÊN
CÂY KHOAI TÂY

Sinh viên: Phan Bảo Luân
Mã số: B2111851
Khóa: K47

Cần Thơ, 03/2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

NIÊN LUẬN CƠ SỞ
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

NHẬN DẠNG BỆNH HẠI TRÊN
CÂY KHOAI TÂY

Người hướng dẫn
TS Phạm Thế Phi

Sinh viên thực hiện
Phan Bảo Luân
Mã số: B2111851
Khóa: K47

Cần Thơ, 03/2024

Lời cảm ơn

Để hoàn thành Niên luận cơ sở ngành Công nghệ thông tin, đầu tiên em gửi lời cảm ơn chân thành nhất đến tất cả giảng viên của Trường Công nghệ Thông tin và Truyền thông đã tận tình giảng dạy, chỉ bảo những kiến thức cơ bản đến chuyên sâu về ngành để hôm nay em có thể hoàn thành được niên luận cơ sở.

Đặc biệt, em gửi đến thầy Ths. Phạm Thế Phi đã giúp đỡ em trong quá trình làm đề tài, đã cung cấp rất nhiều kiến thức, ý tưởng để em có thể hoàn thành tốt bài niên luận.

Dù hoàn thành bài niên luận nhưng sẽ khó tránh khỏi sai sót. Mong thầy cô tận tình chỉ điểm, góp ý chân thành.

Em chân thành cảm ơn

Cần Thơ, ngày 30 tháng 12 năm 2023

Sinh viên thực hiện

Phan Bảo Luân

Mục lục

| | |
|--|----|
| Lời cảm ơn..... | 1 |
| Mục lục | 2 |
| Biểu đồ hình ảnh | 5 |
| Danh mục bảng | 6 |
| Tóm lược..... | 7 |
| Phần giới thiệu | 8 |
| 1. Đặt vấn đề | 8 |
| 2. Mục tiêu đề tài..... | 8 |
| 3. Đối tượng và phạm vi nghiên cứu..... | 8 |
| 4. Phương pháp nghiên cứu..... | 8 |
| 5. Nội dung nghiên cứu | 9 |
| 6. Bố cục của niên luận | 9 |
| Phần nội dung | 10 |
| Chương I: Đặc tả yêu cầu | 10 |
| 1. Bài toán phát hiện bệnh trên cây khoai tây | 10 |
| 1.1. Giới thiệu về cây khoai tây | 10 |
| 1.2. Bệnh early blight(úa sớm, đốm vòng)..... | 10 |
| 1.3. Bệnh late blight(mốc sương, héo muộn, sương mai) | 11 |
| 2. Cách tiếp cận và giải quyết vấn đề..... | 12 |
| Chương II: Cơ sở lý thuyết | 13 |
| 2. Ảnh trong máy tính | 13 |
| 2.1. Hệ màu RGB | 13 |
| 2.2. Ảnh màu | 13 |
| 2.3. Ảnh xám | 15 |
| 3. Tăng cường dữ liệu ảnh(Image Data Augmentation) | 15 |
| 3.1. Thao tác vị trí (Position manipulation) | 16 |
| 3.1.1. Thay đổi kích thước(Resizing) | 16 |
| 3.1.2. Xoay(Rotating) | 16 |
| 3.1.3. Lật(Flipping) | 16 |
| 3.2. Thao tác về màu(Color manipulation) | 17 |
| 3.2.1. Độ sáng(Brightness) | 17 |

| | | |
|-------------|---|----|
| 3.2.2. | Độ tương phản(Contrast)..... | 17 |
| 3.2.3. | Độ bão hòa(Saturation) | 18 |
| 3.2.4. | Sắc thái(Hue)..... | 18 |
| 3.3. | Thao tác về hình ảnh(Image manipulation) | 18 |
| 3.3.1. | Làm mờ(blur) | 18 |
| 3.3.2. | Làm sắc nét(Sharpening)..... | 19 |
| 3.3.3. | Cắt ngẫu nhiên(Random Cropping) | 19 |
| 4. | Tiền xử lý ảnh(Image Preprocessing) | 19 |
| 5. | Mạng nơ-ron(Neural Network) | 20 |
| 5.1. | Mạng nơ-ron..... | 20 |
| 5.2. | Mô hình mạng nơ-ron | 20 |
| 5.3. | Hàm kích hoạt | 21 |
| 6. | Phép tích chập(Convolution) | 22 |
| 6.1. | Convolution..... | 22 |
| 6.2. | Padding..... | 23 |
| 6.3. | Stride | 24 |
| 6.4. | Ý nghĩa của phép tích chập | 24 |
| 7. | Mạng nơ-ron tích chập(Convolutional Neural Network) | 25 |
| 7.1. | Lớp đầu vào(input layer)..... | 26 |
| 7.2. | Lớp tích chập (convolutional layer)..... | 26 |
| 7.3. | Lớp gộp(pooling layer) | 27 |
| 7.4. | Lớp trải phẳng(flatten layer) | 28 |
| 7.5. | Lớp kết nối đầy đủ(fully-connected layer)..... | 28 |
| 7.6. | Lớp đầu ra(output layer) | 29 |
| Chương III: | Kết quả thực hiện..... | 30 |
| 1. | Công nghệ sử dụng..... | 30 |
| 1.1. | Google Colab | 30 |
| 1.2. | Keras | 30 |
| 2. | Xây dựng mô hình..... | 31 |
| 2.1. | Khai báo những thư viện cần thiết | 31 |
| 2.2. | Chuẩn bị dữ liệu | 31 |
| 2.2.1. | Chuẩn bị dữ liệu | 31 |
| 2.2.2. | Chia tập dữ liệu | 32 |

| | |
|------------------------------------|----|
| 2.2.3. Tăng cường dữ liệu..... | 33 |
| 2.2.4. Tiền xử lý dữ liệu | 33 |
| 2.3. Xây dựng mô hình..... | 33 |
| Chương IV: Đánh giá kiểm thử | 36 |
| 1. Các tiêu chí đánh giá | 36 |
| 1.1. Độ chính xác(accuracy)..... | 36 |
| 1.2. Precision | 37 |
| 1.3. Recall..... | 37 |
| 2. Kết quả đánh giá..... | 37 |
| Phần kết luận..... | 40 |
| 1. Kết quả đạt được | 40 |
| 2. Hướng phát triển | 40 |
| Tài liệu tham khảo | 41 |
| Phụ lục | 42 |

Biểu đồ hình ảnh

| | |
|--|----|
| Hình 1 Lá cây khoai tây bị bệnh đốm vòng..... | 11 |
| Hình 2 Bệnh mốc sương trên lá cây khoai tây..... | 12 |
| Hình 3 Màu RGB(đỏ, xanh dương, xanh lá) | 13 |
| Hình 4 Ảnh màu cầu bàn tây ở đồi Bà Nà..... | 14 |
| Hình 5 Ma trận biểu diễn ảnh màu | 14 |
| Hình 6 Hình ảnh ma trận điểm ảnh màu..... | 14 |
| Hình 7 Ảnh xám(Ảnh trắng đen) | 15 |
| Hình 8 Hình ảnh có các kích thước khác nhau | 16 |
| Hình 9 Hình ảnh được xoay ở các góc khác nhau | 16 |
| Hình 10 Hình ảnh bị lật từ trái qua phải | 17 |
| Hình 11 Hình ảnh bị thay đổi độ sáng | 17 |
| Hình 12 Hình ảnh có độ tương phản cao | 17 |
| Hình 13 Hình ảnh khi tăng độ bão hòa | 18 |
| Hình 14 Hình ảnh khi thay đổi sắc thái | 18 |
| Hình 15 Hình ảnh bị làm mờ | 18 |
| Hình 16 Hình ảnh được làm rõ nét | 19 |
| Hình 17 Hình ảnh được cắt ngẫu nhiên | 19 |
| Hình 18 Giá trị điểm ảnh trước và sau khi chuẩn hóa | 20 |
| Hình 19 Mạng nơ-ron | 21 |
| Hình 20 Hàm kích hoạt Sigmoid Hình 21 Hàm kích hoạt ReLu | 22 |
| Hình 22 Phép tích chập với kernel 3*3 | 23 |
| Hình 23 Phép tích chập với padding=1 | 23 |
| Hình 24 Phép tích chập với padding=1 stride=2 | 24 |
| Hình 25 Phép tích chập thực hiện với bộ lọc Sobel | 24 |
| Hình 26 Phép tích chập với bộ lọc Prewitt..... | 25 |
| Hình 27 Mô hình mạng nơ-ron tích chập | 26 |
| Hình 28 Phép tích chập thực hiện với một bộ lọc 2*2*3 | 27 |
| Hình 29 Lớp pooling với pooling size =2, padding 0, stride=2 | 27 |
| Hình 30 Bản đồ đặc trưng được trải phẳng | 28 |
| Hình 31 Các deep learning framework phổ biến..... | 30 |
| Hình 32 Một vài hình ảnh về tập huấn luyện | 32 |
| Hình 33 Chia tập dữ liệu huấn luyện thành 3 phần | 32 |
| Hình 34 Tăng cường dữ liệu..... | 33 |
| Hình 35 Chuẩn hóa kích thước ảnh và chuẩn hóa điểm ảnh | 33 |
| Hình 36 Mô hình mạng nơ-ron tích chập | 34 |
| Hình 37 Biên dịch mô hình mạng nơ-ron..... | 35 |
| Hình 38 Ma trận nhầm lẫn(Confusion Matrix)..... | 36 |
| Hình 39 Biểu đồ độ chính xác và loss của mô hình | 37 |
| Hình 40 Ma trận nhầm lẫn kết quả với dữ liệu kiểm thử | 38 |
| Hình 41 Kết quả các tiêu chí đánh giá..... | 38 |
| Hình 42 Kết quả dự đoán với dữ liệu kiểm thử | 39 |

Danh mục bảng

| | |
|---|----|
| Bảng 1: Bảng tham số của mô hình mạng nơ-ron tích chập..... | 35 |
|---|----|

Tóm lược

Nhận dạng hình ảnh là một trong những tác vụ cơ bản của thị giác máy tính nhờ vậy mà ta biết hình ảnh được nhận dạng là cái gì. Ứng dụng của bài toán nhận dạng hình ảnh lên lĩnh vực nông nghiệp, đặc biệt là trong công cuộc phòng chống bệnh hại trên cây trồng đã mang lại lợi ích kinh tế to lớn người nông dân. Trong bài niên luận này sẽ giới thiệu bệnh hại thường gặp trên cây khoai tây và giới thiệu về sử dụng mô hình mạng nơ-ron tích chập trong việc nhận dạng hình ảnh. Do thời gian thực hiện có hạn, dữ liệu xây dựng mô hình nhận dạng bệnh trên cây khoai tây chỉ tập trung vào 3 lớp: lớp khỏe mạnh, lớp bị bệnh đốm vòng và lớp bị bệnh mốc sương. Từ dữ liệu đã thu thập được, xây dựng một mô hình máy học để nhận dạng bệnh trên cây khoai tây. Mô hình mạng nơ-ron tích chập có các lớp như sau: dữ liệu đầu vào là hình ảnh, hình ảnh sau khi qua lớp tích chập và lớp gộp trở thành bản đồ đặc trưng, bản đồ đặc trưng qua lớp trải phẳng trở thành một vector đặc trưng và trở thành đầu vào của lớp kết nối đầy đủ, qua một số lớp kết nối đầy đủ cuối cùng tạo thành lớp đầu ra là kết quả dự đoán của mô hình. Sau khi mô hình xây dựng xong, mô hình được đánh giá qua các tiêu chí như độ chính xác, hàm lỗi và các chỉ số đánh giá khác như recall, precision, f1. Kết quả đánh giá cho thấy mô hình nhận dạng bệnh trên cây khoai tây có độ chính xác cao.

Phân giới thiệu

1. Đặt vấn đề

Đối với các bất kỳ loại cây lương thực, thực phẩm nào đều không có khả năng chống lại các sâu bệnh hại đến từ môi trường bên ngoài. Nếu như không phát hiện kịp thời, xử lý thích đáng sẽ gây thiệt hại kinh tế lớn đối với người trồng nhất là những vùng có diện tích canh tác lớn. Cây khoai tây cũng không phải ngoại lệ, một số bệnh hại đều có dấu hiệu và thời gian phát bệnh trên cây nếu phát hiện sớm sẽ giúp người trồng khoai tây ngăn chặn bệnh hại phát sinh và lan tràn đến các cây khỏe mạnh.

Vậy có cách để phát hiện sớm được bệnh trên cây khoai tây? Với diện tích trồng khoai tây nhỏ thì còn có thể kiểm tra từng cây mỗi ngày, nhưng diện tích lớn thì việc này rất tốn thời gian, công sức nhưng hiệu quả không cao. Thay vì dùng người để làm việc này, có thể áp dụng trí tuệ nhân tạo để giải quyết vấn đề đặt ra.

Với sự phát triển vượt bậc của lĩnh vực trí tuệ nhân tạo kết hợp về sự bùng nổ dữ liệu, ta đã chứng kiến nhiều sự đột phá trong lĩnh vực học máy, đặc biệt là thị giác máy tính. Bằng phương pháp học sâu mà việc nhận dạng, phân loại hình ảnh trở nên dễ dàng và độ chính xác được cải thiện rất lớn.

“Bài toán phát hiện bệnh trên cây khoai tây” có thể được giải quyết bằng cách sử dụng hình ảnh của cây khoai tây bị bệnh để huấn luyện một mô hình máy học để nhận dạng, phát hiện bệnh trên cây khoai tây.

2. Mục tiêu đề tài

Mục tiêu của đề tài là dùng phương pháp học sâu cài đặt một mô hình máy học cho phép phát hiện, phân loại một số bệnh trên lá cây khoai tây.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là bệnh trên cây khoai tây. Do thời gian hạn chế và dữ liệu thu thập về bệnh trên cây khoai tây còn thiếu nên phạm vi nghiên cứu của đề tài là phân loại một số bệnh trên cây khoai tây như late blight(mốc sương, héo muộn, sương mai) và lá bệnh early blight(đốm vòng, úa sớm) với cây khỏe mạnh. Số lớp mà mô hình có thể phân loại là 3.

4. Phương pháp nghiên cứu

Trong thị giác máy tính, bài toán nhận dạng hình ảnh là bài toán rất phổ biến. Có nhiều phương pháp có thể thực hiện:

- + Phương pháp máy học truyền thống: sử dụng các thuật toán để trích xuất đặc trưng từ ảnh và sử dụng các thuật toán máy học như K-Nearest Neighbor(KNN), Decision Tree, Bayes, Support Vector Machine(SVM)... để phân loại hình ảnh.
- + Phương pháp học sâu: dựa vào mô hình mạng nơ-ron tích chập (Convolutional Neural Network) để phân loại hình ảnh.

Nhờ vào sự phát triển phần cứng máy tính, các bài toán nhận dạng hình ảnh đều áp dụng phương pháp học sâu để giải quyết nên đề tài này sẽ sử dụng phương pháp học sâu để giải quyết bài toán phát hiện bệnh trên cây khoai tây.

5. Nội dung nghiên cứu

Nội dung nghiên cứu chủ yếu là xây dựng một mô hình phân loại ảnh bệnh trên cây khoai tây và dự đoán cây khoai tây hiện trạng như thế nào. Các công việc cần thực hiện như sau:

- + Chuẩn bị tập dữ liệu ảnh đã phân loại thành 3 lớp khỏe mạnh, bệnh early blight và bệnh late blight.
- + Chia tập dữ liệu thành 3 phần: 1 phần để huấn luyện (training set), một phần để kiểm tra (validation set), một phần để kiểm thử (testing set).
- + Tiền xử lý dữ liệu: điều chỉnh tất cả các ảnh về cùng kích thước và chuẩn hóa giá trị điểm ảnh.
- + Tăng cường dữ liệu: làm giàu dữ liệu để mô hình huấn luyện tăng độ chính xác.
- + Xây dựng mô hình: Dùng phương pháp học sâu xây dựng một mô hình nhận dạng bệnh trên lá cây khoai tây.
- + Kiểm thử mô hình: dùng tập kiểm thử (testing set) để kiểm tra độ chính xác của mô hình đã huấn luyện.

6. Bố cục của niên luận

Dựa trên mục tiêu đã được đề ra, nội dung của bài niên luận được trình bày thành những nội dung cụ thể sau:

- + **Chương I: Đặc tả yêu cầu:**
- + **Chương II: Cơ sở lý thuyết:** Chương này sẽ giới thiệu những khái niệm cơ bản liên quan việc nhận dạng hình ảnh. Tìm hiểu khái niệm ảnh, một số thao tác tăng cường ảnh, giới thiệu về phép tích chập và thành phần cấu tạo của một mạng nơ-ron tích chập là mô hình để nhận dạng hình ảnh
- + **Chương III: Kết quả thực hiện:** Chương này giới thiệu về các công nghệ sử dụng để xây dựng mô hình, nêu lên từng bước thực hiện xây dựng mô hình nhận dạng cụ thể.
- + **Chương IV: Đánh giá kiểm thử:** Chương này giới thiệu một số chỉ tiêu đánh giá mô hình sau khi huấn luyện và trình bày kết quả đánh giá mô hình.

Phân nội dung

Chương I: Đặc tả yêu cầu

1. Bài toán phát hiện bệnh trên cây khoai tây

1.1. Giới thiệu về cây khoai tây

Khoai tây là loài cây nông nghiệp ngắn ngày, trồng lấy củ chứa tinh bột. Là loại củ mọc trên rễ, có nguồn gốc ở Nam Mỹ và được đưa đến châu Âu trong thế kỷ 16 sau đó đã được trồng rộng rãi trên thế giới. Là loại cây trồng phổ biến thứ tư về mặt sản lượng tươi - xếp sau lúa, lúa mì và ngô.

Khoai tây ở Việt Nam chủ yếu trồng vào vụ đông, một số nơi có khí hậu lạnh, như Lâm Đồng, Lào Cai được trồng vụ thu, hoặc vụ xuân hè.

Tổng diện tích trồng khoai tây hàng năm của cả nước khoảng từ 25.000 - 30.000 ha, tập trung ở các tỉnh đồng bằng và một số tỉnh miền núi phía Bắc.

Sản lượng khoai tây ở Việt Nam bao gồm hơn 10 giống. Giống Thường Tín vẫn còn được trồng khoảng 8,5% diện tích cả nước, nhất là các tỉnh Thái Bình, Hải Dương, Nam Định và Ninh Bình. Các giống nhập từ châu Âu như Diamant, Mariella và Nicola đang được trồng chủ yếu ở đồng bằng sông Hồng với 15,19% tổng diện tích. Đặc biệt giống VT2 của Trung Quốc chiếm tới 66% tổng diện tích khoai tây vì nó cho năng suất khá cao (khoảng 16-20 tấn/ha).

1.2. Bệnh early blight(úa sớm, đốm vòng)

Bệnh đốm vòng hay còn gọi là bệnh úa sớm, xảy ra chủ yếu ở cà chua, khoai tây. Bệnh phát triển mạnh trong điều kiện nhiệt độ và ẩm độ cao. Mầm bệnh tồn tại trên tàn dư cây trồng ít nhất là một năm. Nguyên nhân của bệnh đốm vòng do nấm *Alternaria solani* gây ra.

Mặc dù có tên là "sớm" nhưng các triệu chứng ở lá thường xảy ra trên các lá già. Nếu không được kiểm soát, bệnh đốm vòng có thể làm giảm năng suất đáng kể.

Ở khoai tây, bệnh này khiến khoai tây bị rụng lá sớm, gây thiệt hại đến 50% năng suất hàng năm. Sự lây nhiễm ban đầu xảy ra trên các lá già, với các đốm nâu đen phát triển chủ yếu ở tâm lá, sau đó lá chuyển sang màu vàng và khô hoặc rụng. Trên thân cây, các vết đốm không có đường viền rõ ràng (so với các vết đốm trên lá).

Môi trường ẩm, ẩm (24-29°C) là điều kiện thích hợp cho bệnh đốm vòng. Nấm *Alternaria solani* lây bệnh và sinh sản bằng bào tử. Bào tử xâm nhập trực tiếp vào biểu bì lá hoặc xâm nhập qua khí khổng. Sự lây lan mầm bệnh là do các bào tử phân tán nhờ gió, nước mưa hoặc nước tưới văng lên lá. Bệnh đốm vòng có nhiều chu kỳ lây nhiễm trong một mùa.

Dưới đây là hình ảnh của lá cây khoai tây bị bệnh cây khoai tây bị nhiễm bệnh đốm vòng trên mặt lá có nhiều vết đốm do nấm gây ra.



Hình 1 Lá cây khoai tây bị bệnh đốm vòng

1.3. Bệnh late blight(mốc sương, héo muộn, sương mai)

Bệnh sương mai khoai tây (hay còn gọi bệnh nấm mốc, mốc sương) do nấm *Phytophthora infestans* gây ra. Ký chủ chính là khoai tây, nhưng *P. infestans* cũng có thể lây nhiễm sang các cây trồng khác, bao gồm cà chua, cà tím,... Các nguồn này có thể đóng vai trò là nguồn truyền bệnh cho khoai tây.

Bệnh thường xuất hiện đầu tiên ở mép lá chóp lá, tạo vết xám xanh nhạt sau đó lan rộng vào phiến lá. Phần giữa vết bệnh chuyển sang màu nâu đen và xung quanh vết bệnh thường có lớp cảnh bào tử màu trắng xốp bao phủ như một lớp mốc trắng như sương muối làm cho lá chết lụi nhanh chóng.

Bệnh phát triển thuận lợi trong điều kiện thời tiết ẩm độ cao và nhiệt độ xuống thấp từ 18-22°C, độ ẩm không khí cao bệnh phát triển mạnh.

Dưới đây là hình ảnh cây khoai tây bị bệnh mốc sương, trên lá có nhiều bào tử màu trắng xốp bao phủ.



Hình 2 Bệnh mốc sương trên lá cây khoai tây

2. Cách tiếp cận và giải quyết vấn đề

Nhận dạng hình ảnh là một nhánh của trí tuệ nhân tạo cho phép xác định và nhận dạng các ảnh và đối tượng trong hình ảnh kỹ thuật số. Về cơ bản, một thuật toán nhận dạng hình ảnh sử dụng các thuật toán máy học hoặc học sâu phân tích từng điểm ảnh để nhận dạng hình ảnh.

Các mô hình huấn luyện theo phương pháp học sâu có cách thức hoạt động như sau: dữ liệu đầu vào sẽ đi qua nhiều lớp khác nhau, mỗi lớp sẽ giữ lại những đặc trưng dữ liệu đầu vào và sẽ làm đầu vào của lớp tiếp theo. Cứ thế cho đến lớp cuối cùng chính là kết quả đầu ra.

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) thể hiện rõ ưu thế của mình trong việc nhận dạng hình ảnh. Áp dụng mô hình mạng này không cần bước trích xuất đặc trưng ảnh như các mô hình máy học khác vì bản thân trong mô hình mạng nơ-ron tích chập đã có lớp để trích xuất đặc trưng ảnh. Hơn nữa độ chính xác của mô hình cao hơn với các thuật toán máy học khác.

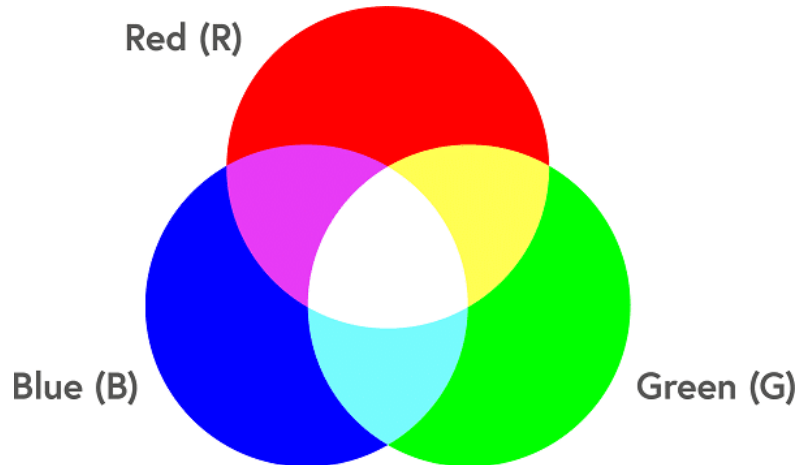
Trong đề giải quyết bài toán “**phát hiện bệnh trên cây khoai tây**” sử dụng mô hình mạng nơ-ron tích chập để phân loại các loại bệnh trên lá cây khoai tây.

Chương II: Cơ sở lý thuyết

2. Ảnh trong máy tính

2.1. Hệ màu RGB

RGB là tên viết tắt tiếng Anh của hệ màu gồm 3 màu Red(đỏ), Green(xanh lá) và Blue(xanh dương) là 3 màu chính của ánh sáng khi tách ra từ lăng kính. Khi trộn 3 màu trên với tỉ lệ nhất định có thể tạo ra nhiều màu khác nhau.



Hình 3 Màu RGB(đỏ, xanh dương, xanh lá)

Mỗi màu có miền giá trị từ 0 đến 255 với 0 là không có màu đến 255 là màu sáng nhất được biểu diễn thành dạng (r, g, b) ứng với một màu nhất định. Số màu mà hệ RGB tối đa có thể biểu diễn là $255 \times 255 \times 255 = 16581375$ triệu màu.

Từ Hình 3, khi trộn 3 màu trong hệ màu RGB sẽ tạo thành những màu khác nhau tùy theo tỉ lệ của 3 màu trên.

2.2. Ảnh màu

Điểm ảnh(pixel) là đơn vị đo lường khi nói đến chất lượng hay kích thước của hình ảnh, điểm ảnh là đơn vị nhỏ nhất cấu tạo nên ảnh kỹ thuật số.

Hình ảnh là sự kết hợp của nhiều điểm ảnh lại với nhau, được biểu diễn thành một ma trận điểm ảnh

Hình 4 dưới đây là ví dụ minh họa cho ảnh màu, đối với máy tính, ảnh màu được biểu diễn thành một ma trận điểm ảnh, trong đó mỗi điểm ảnh có 3 giá trị nằm trong khoảng giá trị $[0, 255]$ để biểu diễn một màu trong hệ màu, 3 giá trị này được sắp xếp theo thứ tự hệ màu biểu diễn mà được gọi tên ví dụ ảnh biểu diễn dưới hệ màu RGB.



Hình 4 Ảnh màu cầu bàn tay ở đồi Bà Nà

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,800} \\ w_{2,1} & w_{2,2} & \dots & w_{2,800} \\ \dots & \dots & \dots & \dots \\ w_{600,1} & w_{600,2} & \dots & w_{600,800} \end{bmatrix}$$

Hình 5 Ma trận biểu diễn ảnh màu

Chất lượng của hình ảnh chính là phụ thuộc vào số lượng điểm ảnh. Số lượng điểm ảnh càng nhiều hình ảnh càng sắc nét và chi tiết. Với mỗi một màu trong điểm ảnh sẽ được biểu diễn bằng hệ RGB là bộ (r, g, b) nên ảnh màu sẽ được biểu diễn thành:

$$\begin{bmatrix} (r_{1,1}, g_{1,1}, b_{1,1}) & (r_{1,2}, g_{1,2}, b_{1,2}) & \dots & (r_{1,800}, g_{1,800}, b_{1,800}) \\ (r_{2,1}, g_{2,1}, b_{2,1}) & (r_{2,2}, g_{2,2}, b_{2,2}) & \dots & (r_{2,800}, g_{2,800}, b_{2,800}) \\ \dots & \dots & \dots & \dots \\ (r_{600,1}, g_{600,1}, b_{600,1}) & (r_{600,2}, g_{600,2}, b_{600,2}) & \dots & (r_{600,800}, g_{600,800}, b_{600,800}) \end{bmatrix}$$

Hình 6 Hình ảnh ma trận điểm ảnh màu

Để tiện lưu trữ, tách ma trận trên thành 3 ma trận. Mỗi ma trận đại diện cho từng giá trị màu red, green, blue. Mỗi ma trận được tách ra còn gọi là channel nên ảnh màu có 3 channel: channel red, channel green, channel blue.

Khi dữ liệu có một chiều được gọi là vector, khi dữ liệu có 2 chiều được gọi là ma trận, khi dữ liệu nhiều hơn 2 chiều được gọi là tensor.

Ảnh màu trên máy tính được biểu diễn dạng tensor 3 chiều do 3 ma trận(channel) red, green, blue xếp chồng lên nhau.

2.3. Ảnh xám



Hình 7 Ảnh xám(Ảnh trắng đen)

Tương tự ảnh màu, ảnh xám cũng biểu diễn bằng ma trận điểm ảnh tuy nhiên chỉ cần một ma trận là đủ để biểu diễn ảnh màu. Mỗi điểm ảnh có giá trị trong khoảng $[0, 255]$. Giá trị 0 là màu đen, giá trị 255 là màu trắng. Càng gần 0 thì càng tối, càng gần 255 thì càng sáng.

3. Tăng cường dữ liệu ảnh(Image Data Augmentation)

Phần lớn các mô hình máy học thường gặp khó khăn với môi trường dữ liệu thực tế. Khi đối mặt với dữ liệu thực thì hiệu suất không cao mặc dù khi huấn luyện thì hiệu suất rất tốt.

Vấn đề phát sinh là dữ liệu không đủ khiến cho mô hình máy học không học được đầy đủ thông tin. Phần lớn các dữ liệu mã nguồn mở có kích thước nhỏ vì việc thu thập dữ liệu khó khăn.

Để giải quyết vấn đề khan hiếm dữ liệu, các kỹ thuật tăng cường dữ liệu ra đời. Tăng cường dữ liệu cho phép các kỹ sư tạo ra mẫu dữ liệu mới từ dữ liệu gốc để sử dụng huấn luyện mô hình. Tăng cường dữ liệu dường như đã trở thành một quá trình không thể thiếu trong việc huấn luyện mô hình máy học.

Tăng cường dữ liệu hình ảnh là quá trình tạo ra các phiên bản hình ảnh mới từ tập ảnh huấn luyện để tăng tính đa dạng cho tập huấn luyện.

Đối với máy tính, hình ảnh bản chất là một mảng 2 số chiều. Những con số này đại diện cho giá trị điểm ảnh. Khi điều chỉnh những con số này có thể tạo ra những hình ảnh mới, ảnh tăng cường. Những hình ảnh tăng cường này giống với ảnh gốc nhưng chứa những thông tin khái quát giúp mô hình máy học huấn luyện tốt hơn. Hình ảnh là một trong những dữ liệu dễ tăng cường nhất do có sẵn nhiều kỹ thuật liên quan.

3.1. Thao tác vị trí (Position manipulation)

3.1.1. Thay đổi kích thước(Resizing)

Tăng hoặc giảm kích thước hình ảnh so với ảnh gốc sẽ tạo ra các ảnh có kích thước khác nhau. Hình 8 tuy cùng một tấm ảnh nhưng khi thay đổi kích thước, từ gốc độ máy tính đây là những hình hoàn toàn khác nhau nhưng nội dung hình ảnh là một.



Hình 8 Hình ảnh có các kích thước khác nhau

3.1.2. Xoay(Rotating)

Xoay hình ảnh một góc từ -360 đến 360 độ để tạo thành một ảnh mới. Hình 9 biểu diễn hình ảnh được xoay ở các 45 độ và -45 độ, khi hình ảnh được xoay có thể dẫn đến kích thước hình ảnh bị thay đổi.



Hình 9 Hình ảnh được xoay ở các góc khác nhau

3.1.3. Lật(Flipping)

Tương tự với xoay, lật hình ảnh là từ ảnh ban đầu lật ảnh từ trái qua phải, phải qua trái, trên xuống dưới, dưới lên trên để tạo ra

ảnh mới. Hình 10 biểu diễn hình ảnh bị lật từ trái qua phải, sau khi lật kích thước hình ảnh không bị thay đổi.



Hình 10 Hình ảnh bị lật từ trái qua phải

3.2. Thao tác về màu(Color manipulation)

3.2.1. Độ sáng(Brightness)

Thay đổi độ sáng của tất cả các màu trong hình ảnh để tạo thành ảnh mới. Hình 11 có hình bên trái độ sáng bị giảm đi, còn hình bên phải thì độ sáng được tăng lên.



Hình 11 Hình ảnh bị thay đổi độ sáng

3.2.2. Độ tương phản(Contrast)

Thay vì thay đổi độ sáng toàn bộ hình ảnh, độ tương phản chỉ thay đổi độ sáng một phần màu nào đó trong ảnh. Hình 12 biểu diễn hình ảnh có độ tương phản cao, nhìn vào ảnh có cảm giác như có ánh sáng chiếu vào ảnh.



Original image

Contrast image

Hình 12 Hình ảnh có độ tương phản cao

3.2.3. Độ bão hòa(Saturation)

Mức độ màu sắc trong một tấm ảnh, tăng độ bão hòa làm ảnh nổi bật hơn, giảm độ bão hòa làm cho ảnh nhạt đi. Hình 13 khi thấy hình ảnh bên phải tăng độ bão hòa, cảm giác hình ảnh trở nên sống động hơn.



Hình 13 Hình ảnh khi tăng độ bão hòa

3.2.4. Sắc thái(Hue)

Sắc thái chính là màu chủ đạo trong hình ảnh, với các màu chủ đạo khác nhau trong nhóm các màu sơ cấp(đỏ, cam, vàng, lục, lam, chàm, tím) và nhóm màu thứ cấp(sự kết hợp của những màu sơ cấp). Hình 14 là hình ảnh khi thay màu chủ đạo thành màu tím.



Hình 14 Hình ảnh khi thay đổi sắc thái

3.3. Thao tác về hình ảnh(Image manipulation)

3.3.1. Làm mờ(blur)

Không phải lúc nào hình ảnh cũng sắc nét, thao tác làm mờ ảnh từ ảnh gốc ban đầu thành ảnh bị mờ đi tạo nên ảnh mới. Hình 15 biểu diễn hình ảnh bị làm mờ.



Hình 15 Hình ảnh bị làm mờ

3.3.2. Làm sắc nét(Sharpening)

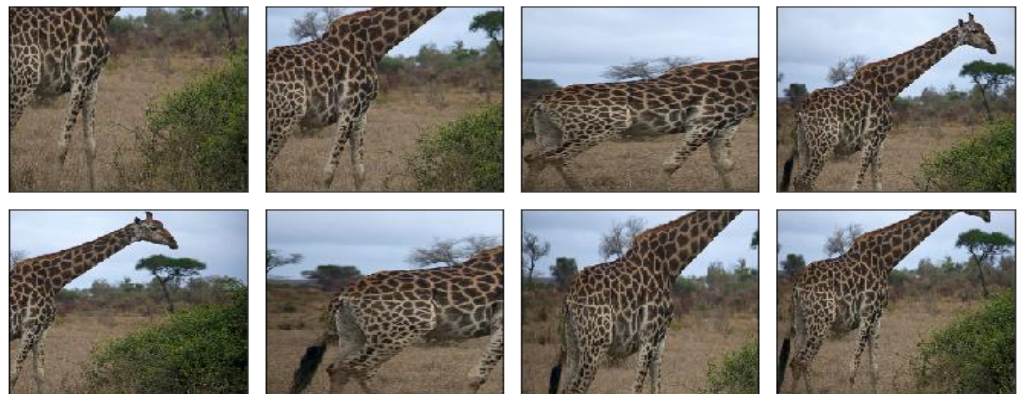
Ngược lại với làm mờ, làm sắc nét ảnh giúp tăng chất lượng hình ảnh, chi tiết hình ảnh sẽ rõ ràng hơn. Hình 16 biểu diễn ảnh được làm sắc nét.



Hình 16 Hình ảnh được làm rõ nét

3.3.3. Cắt ngẫu nhiên(Random Cropping)

Cắt ngẫu nhiên tạo ra các hình ảnh không trọn vẹn như góc nhìn bị che khuất, góc chụp không tốt phù hợp với hình ảnh thực tế. Hình 17 cùng biểu diễn hình ảnh con hươu cao cổ những ở các góc cắt khác nhau



Hình 17 Hình ảnh được cắt ngẫu nhiên

Tất cả các kỹ thuật được đề cập ở trên có thể giúp tăng cường kích thước và độ đa dạng của tập dữ liệu. Giúp tiết kiệm thời gian chi phí thay vì thu thập thủ công, tăng độ chính xác cho mô hình với hình ảnh thực tế.

4. Tiền xử lý ảnh(Image Preprocessing)

4.1. Thay đổi kích thước(Resizing)

Resizing là quá trình tăng hoặc giảm kích thước của ảnh, cụ thể hơn là về mặt hình ảnh 2 chiều, tăng hoặc giảm chiều rộng hoặc chiều cao bằng cách tính toán lại các giá trị điểm ảnh cho ảnh mới(ảnh đã thay đổi).

Trong mạng nơ-ron tích chập, kỹ thuật resizing được sử dụng với nhiều mục đích khác nhau:

- + Chuẩn hóa kích thước ảnh: Ảnh đầu vào có nhiều kích thước khác nhau cần được chuẩn hóa về cùng kích thước cho phù hợp với mạng nơ-ron
- + Giảm kích thước ảnh: Ảnh kích thước lớn làm tăng chi phí tính toán và thời gian huấn luyện mô hình
- + Tăng kích thước ảnh: Một số trường hợp cần tăng kích thước để cung cấp thêm thông tin cho mô hình.

4.2. Chuẩn hóa điểm ảnh(Rescaling)

Rescaling là quá trình thay đổi miền giá trị của các điểm ảnh. Mặc định các giá trị điểm ảnh nằm trong kích thước là $[0, 255]$. Trong các mô hình mạng nơ-ron tích chập, kỹ thuật rescaling được sử dụng để:

- + Chuẩn hóa giá trị điểm ảnh: Dùng phép chia $1/255$ để chuyển đổi giá trị điểm ảnh từ $[0,255]$ thành $[0,1]$ giúp dữ liệu ảnh đồng nhất và dễ xử lý hơn
- + Tăng tốc độ huấn luyện mô hình: Phạm vi tính toán nhỏ lại

| | |
|------------------|------------------|
| [| [|
| [100, 150, 200], | [0.4, 0.6, 0.8], |
| [50, 100, 150], | [0.2, 0.4, 0.6], |
| [255, 255, 255] | [1.0, 1.0, 1.0] |
|] |] |

Hình 18 Giá trị điểm ảnh trước và sau khi chuẩn hóa

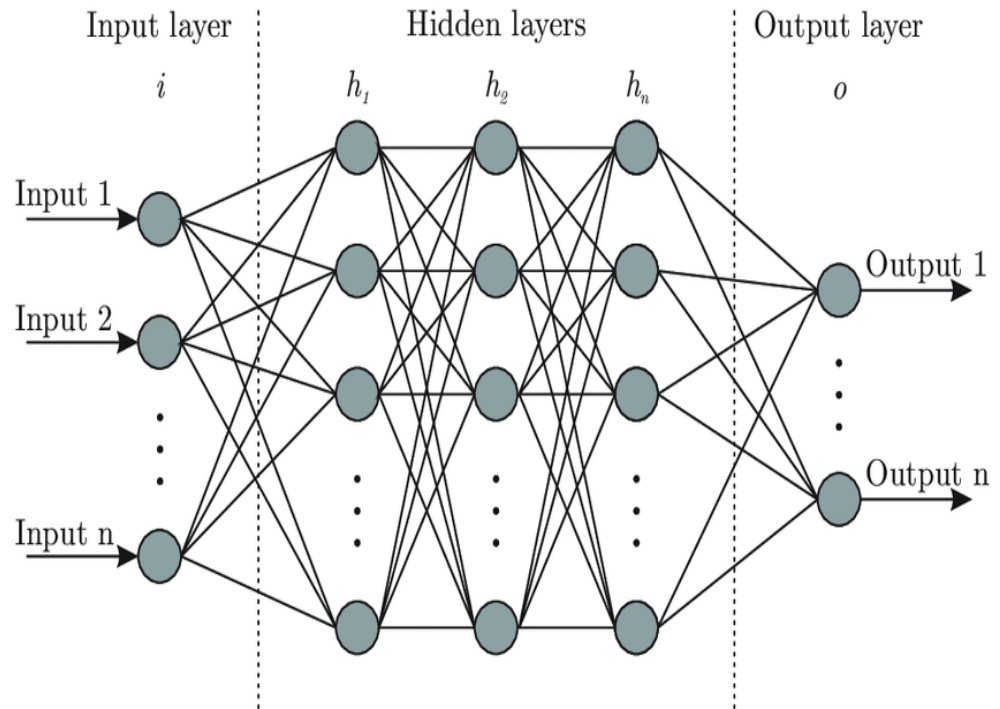
5. Mạng nơ-ron(Neural Network)

5.1. Mạng nơ-ron

Mạng nơ-ron là một phương thức trong lĩnh vực trí tuệ nhân tạo được lấy cảm hứng từ sự hoạt động của các nơ-ron thần kinh trong bộ não con người. Đây là một loại quy trình trong máy học, được gọi là học sâu(Deep Learning).

5.2. Mô hình mạng nơ-ron

Bộ não người chính là nguồn cảm hứng cho kiến trúc mạng nơ-ron. Các tế bào nơ-ron tạo thành một mạng lưới phức tạp có tính liên kết cao, gửi tín hiệu qua lại lẫn nhau giúp con người xử lý thông tin. Tương tự như vậy, mô phỏng kiến trúc mạng của các nơ-ron trong bộ não người để tạo ra mạng nơ-ron nhân tạo với mỗi nơ-ron nhân tạo chính là nút(node) trong mạng, các nút này liên kết với nhau tạo thành mạng giúp truyền thông tin qua giữa các nút. Hình bên dưới mô tả cấu trúc của một mạng nơ-ron nhân tạo gồm 3 loại lớp đầu vào, lớp ẩn và lớp kết quả.



Hình 19 Mạng nơ-ron

Nhìn vào hình 19 thấy được một mô hình mạng nơ-ron đơn giản được cấu tạo như sau:

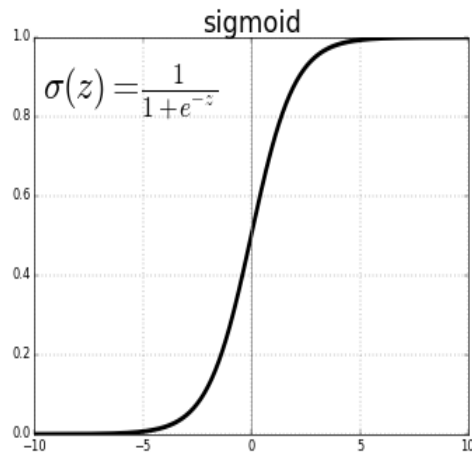
- + Lớp đầu vào(input layer): Lớp này bao gồm các nơ-ron nhận giá trị đầu vào và chuyển chúng sang lớp tiếp theo. Số lượng nơ-ron trong lớp đầu vào được xác định bởi kích thước của dữ liệu đầu vào.
- + Lớp ẩn(hidden layers): Các lớp này không tiếp xúc với đầu vào hoặc đầu ra và có thể được coi là công cụ tính toán của mạng nơ-ron. Các nơ-ron của mỗi lớp ẩn lấy tổng trọng số của các đầu ra từ lớp trước, áp dụng hàm kích hoạt và chuyển kết quả cho lớp tiếp theo. Mạng có thể có 0 hoặc nhiều lớp ẩn.
- + Lớp đầu ra(output layer): Lớp cuối cùng tạo ra đầu ra cho các đầu vào nhất định. Số lượng nơ-ron trong lớp đầu ra phụ thuộc vào số lượng đầu ra có thể có mà mạng được thiết kế để tạo ra.

Mỗi nơ-ron trong một lớp được kết nối với mọi nơ-ron ở lớp tiếp theo, tạo nên một mạng được kết nối đầy đủ. Sức mạnh của sự kết nối giữa các nơ-ron được biểu thị bằng các trọng số và việc học trong mạng nơ-ron bao gồm việc cập nhật các trọng số này dựa trên lỗi của đầu ra.

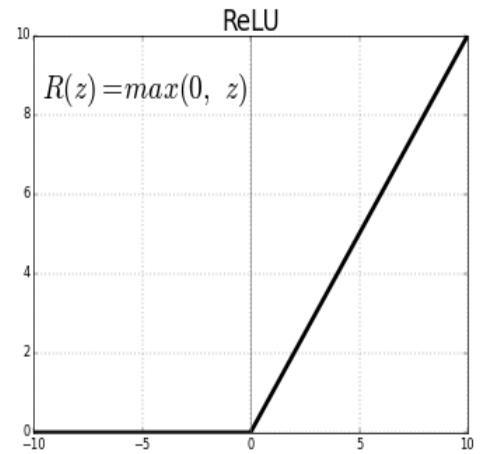
5.3. Hàm kích hoạt

Hàm kích hoạt (activation function) đóng vai trò quan trọng trong mạng nơ-ron, đưa các thuộc tính phi tuyến qua hàm kích hoạt cho phép mô hình tìm hiểu các mẫu phức tạp. Các hàm kích hoạt phổ biến như hàm sigmoid, ReLu(Rectified Linear Unit).

Hình 20 và Hình 21 là đồ thị của hai hàm kích hoạt phi tuyến phổ biến nhất trong mạng nơ-ron.



Hình 20 Hàm kích hoạt Sigmoid



Hình 21 Hàm kích hoạt ReLu

Nếu hàm kích hoạt là tuyến tính, mạng nơ-ron chỉ tìm hiểu được mối quan hệ tuyến tính giữa đầu vào và đầu ra. Điều này có nghĩa là mạng sẽ không thể tính toán gần đúng các phương trình phức tạp hay các phương trình phi tuyến tính.

Hàm kích hoạt phi tuyến làm tăng độ phức tạp của mạng nơ-ron bằng cách đưa vào tính phi tuyến tính để cho mạng nơ-ron có thể biểu diễn dữ liệu phức tạp hơn.

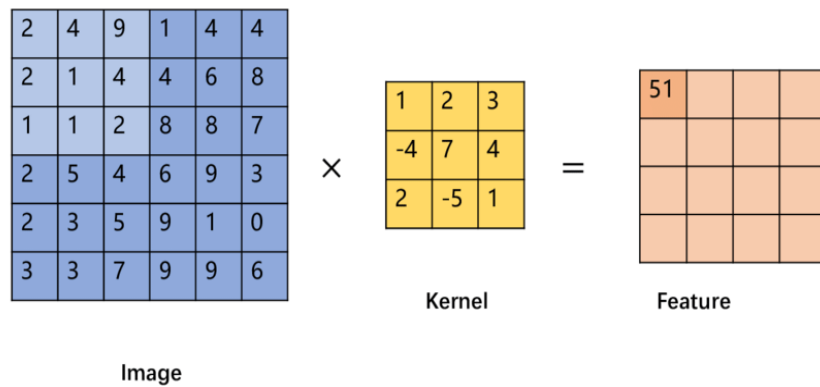
6. Phép tích chập(Convolution)

6.1. Convolution

Phép tích chập (Convolution) trong xử lý ảnh là một phép toán toán học được sử dụng để trích xuất các đặc điểm quan trọng từ ảnh. Nó hoạt động bằng cách di chuyển một ma trận nhỏ, được gọi là bộ lọc (kernel), qua ảnh và tính toán tích vô hướng giữa các giá trị pixel trong ảnh và các giá trị tương ứng trong bộ lọc.

Ví dụ phép tích chập, một ảnh xám được biểu diễn bằng ma trận A có kích thước $m \times n$. Ta định nghĩa một bộ lọc(kernel) có kích thước $k \times k$ với k là số lẻ 1, 3, 5...

Hình 22 thực hiện phép tích chập với bộ lọc 3×3 để tạo thành bản đồ đặc trưng(những đặc trưng được trích xuất của ảnh).



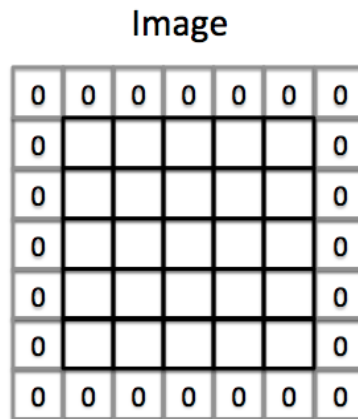
Hình 22 Phép tích chập với kernel 3*3

Trong ma trận A(Image) lấy một ma trận X có kích thước bằng với kernel sau đó nhân ma trận X với kernel ta thu được giá trị điền vào ma trận Y(Feature). Lần lượt di chuyển kernel từ trái qua phải từ trên xuống dưới từng bước sẽ thu được ma trận Y(Feature) trong đó kết quả $51 = 2*1 + 4*2 + 9*3 + 2*(-4) + 1*7 + 4*4 + 2*1 + (-5)*1 + 1*2$

Ma trận đặc trưng Y có kích thước nhỏ hơn X. Kích thước của ma trận Y là $(m - k + 1) * (n - k + 1)$.

6.2. Padding

Khi thực hiện phép tích chập như trên sẽ làm cho ma trận kết quả nhỏ hơn ma trận gốc, hơn nữa các điểm ảnh ở viền, góc có số lần tích chập ít hơn các điểm ảnh ở vị trí khác. Như vậy để giải quyết vấn đề nêu trên ta thêm giá trị 0 ở phần viền ngoài ma trận X.



Hình 23 Phép tích chập với padding=1

Sau phép tích chập thu được ma trận kết quả Y bằng với ma trận điểm ảnh gốc X. Phép tính này được gọi là phép tích chập với padding=1. Padding=k nghĩa là thêm k vector 0 ở mỗi phía(trên dưới, trái, phải) ma trận điểm ảnh. Hình 23 sử dụng thực hiện phép tích chập với padding=1.

Với phép padding có thể bảo toàn được kích thước của ảnh gốc, giảm thiểu mất mát thông tin với các điểm ảnh ở rìa.

6.3. Stride

Khi phép tích chập với kernel di chuyển tuần tự từ trái qua phải từ trên xuống dưới từng bước là phép tích chập thực hiện với bước nhảy stride=1. Khi stride= k với $k > 1$, ma trận kết quả thu được sẽ có kích thước nhỏ hơn rất nhiều so với ma trận kết quả gốc.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

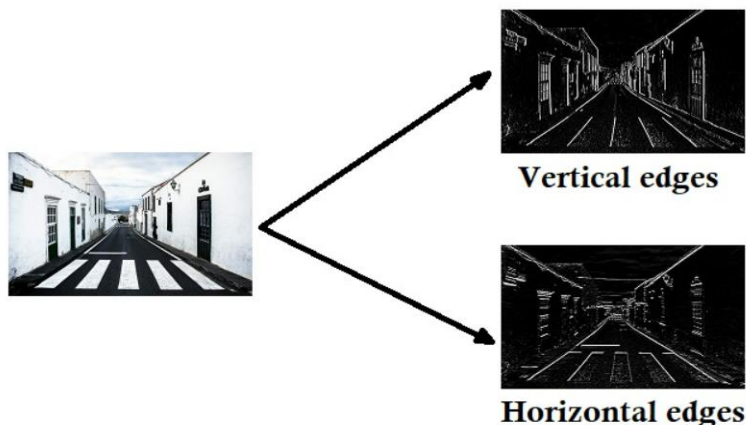
Hình 24 Phép tích chập với padding=1 stride=2

Với phép stride > 1 giúp kiểm soát độ phân giải, độ chi tiết, giảm kích thước của ảnh. Hình 24 sử dụng phép stride=2 kết hợp với padding=1, sau khi thực hiện phép tích chập, kết quả thực hiện chỉ còn là phần màu cam trên hình ảnh.

6.4. Ý nghĩa của phép tích chập

Phép tích chập đóng vai trò quan trọng trong xử lý ảnh cũng như mạng nơ-ron tích chập. Phép tích chập trích xuất đặc trưng quan trọng từ ảnh ví dụ như cạnh, đường viền, kết cấu, v.v.

+ Phát hiện cạnh



Hình 25 Phép tích chập thực hiện với bộ lọc Sobel

Bộ lọc Sobel có độ nhạy cảm cao với các đường ngang và dọc được sử dụng để phát hiện các cạnh rõ ràng và sắc nét. Hình 25 sau khi sử dụng bộ lọc Sobel để lập bản đồ đặc trưng, các cạnh đường thẳng nằm dọc và ngang hiện rõ hơn.

+ **Xác định các đường viền**



Hình 26 Phép tích chập với bộ lọc Prewitt

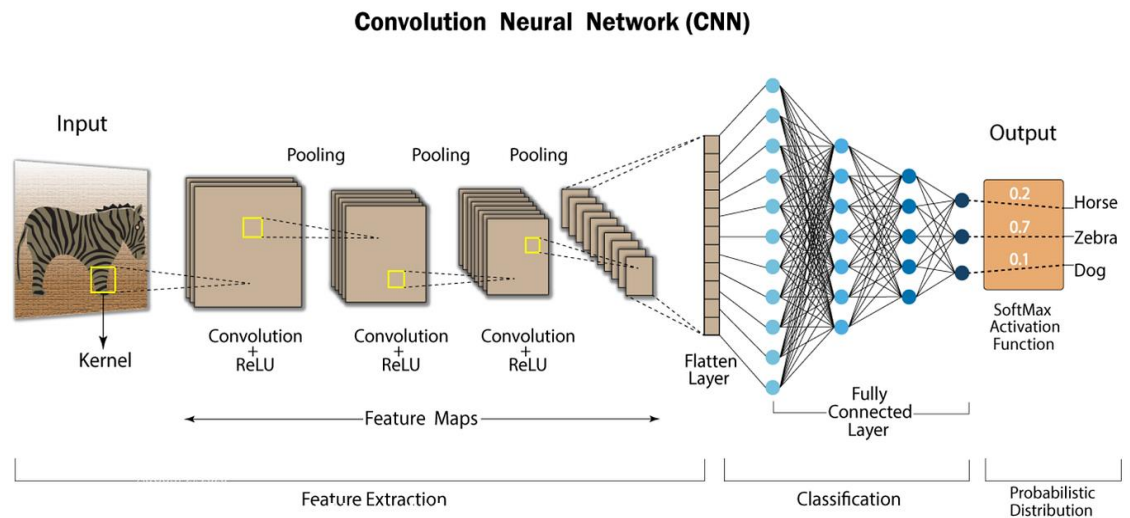
Bộ lọc Prewitt nhạy cảm với các đường cong giúp phát hiện các đường viền. Hình 26 sử dụng bộ lọc Prewitt để lập bản đồ đặc trưng, ngược lại với bộ lọc Sobel, bộ lọc Prewitt dễ dàng phát hiện những đường cong trong ảnh.

7. **Mạng nơ-ron tích chập(Convolutional Neural Network)**

Yann LeCun, giám đốc Nhóm nghiên cứu AI của Facebook, là người tiên phong về mạng nơ-ron tích chập. Ông đã xây dựng mạng nơ-ron tích chập đầu tiên có tên LeNet vào năm 1988. LeNet được sử dụng cho các nhiệm vụ nhận dạng ký tự như đọc mã zip và chữ số.

Một số ứng dụng nổi tiếng như là nhận dạng gương mặt, phát hiện đối tượng của xe tự lái, chẩn đoán bệnh dựa trên hình ảnh y khoa đều thực hiện nhờ vào mạng nơ-ron tích chập, đã cải tiến độ chính xác hơn là các thuật toán trước đó. Dưới đây là ví dụ minh họa về mạng nơ-ron tích chập và cách nó hoạt động:

Giả sử có một tập hình ảnh các con vật như ngựa, lừa, chó, v.v. Mục tiêu của bài toán là cho một tấm ảnh con vật bất kỳ, nhận dạng nó là con gì. Đầu tiên là cung cấp một tấm ảnh con vật làm đầu vào của mạng nơ-ron. Các lớp ẩn thực hiện việc trích xuất đặc trưng bằng các phép tính và các thao tác khác nhau. Các lớp ẩn bao gồm lớp tích chập(convolution layer), lớp gộp(pooling layer) thực hiện trích xuất các đặc trưng từ ảnh. Cuối cùng có lớp kết nối đầy đủ(fully connected layer) xác định con vật trong ảnh. Hình 27 mô tả đầy đủ các lớp đã thực hiện ở trên.



Hình 27 Mô hình mạng nơ-ron tích chập

7.1. Lớp đầu vào(input layer)

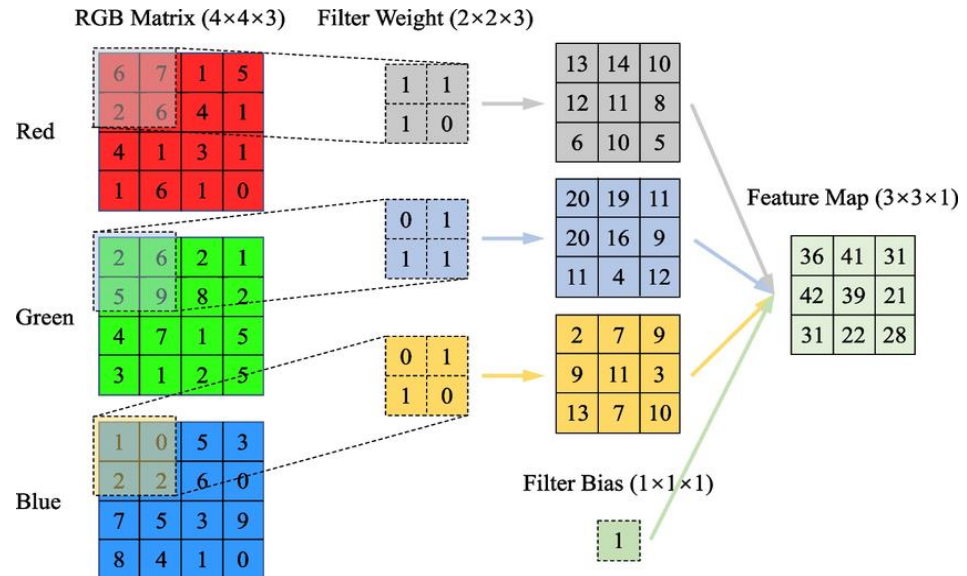
Đây là lớp nhận dữ liệu đầu vào cho mô hình. Trong mạng nơ-ron tích chập, dữ liệu đầu vào thường là một danh sách hình ảnh. Mỗi hình ảnh được biểu diễn dưới dạng tensor 3 chiều $m \times n \times k$.

7.2. Lớp tích chập (convolutional layer)

Đây là lớp để trích xuất đặc trưng từ tập dữ liệu đầu vào. Đối với phép tích chập được giới thiệu ở trên áp dụng cho cho ma trận hình ảnh được sử dụng bộ lọc là một ma trận kích thước $k \times k$. Tuy nhiên ảnh màu thì có đến 3 channel red, green, blue nên khi muốn trích xuất đặc trưng cho hình ảnh này cần định nghĩa một bộ lọc là một tensor 3 chiều có kích thước $k \times k \times 3$.

Các quy tắc padding, và stride tương tự với ma trận ảnh xám. Mỗi bộ lọc khác nhau sẽ trích xuất được những đặc trưng khác nhau từ hình ảnh, nên mỗi lớp tích chập sẽ sử dụng nhiều bộ lọc để trích xuất được nhiều đặc trưng từ hình ảnh. Mỗi bộ lọc sẽ cho đầu ra một bản đồ đặc trưng(feature map). Giả sử sử dụng k bộ lọc, kết quả đầu ra của lớp tích chập sẽ là bản đồ đặc trưng với độ sâu là k .

Hình 28 bên dưới thực hiện phép tích chập với ảnh màu, từ 3 ma trận điểm ảnh sau đó thực hiện phép tích chập với bộ lọc có kích thước là 2×2 và cuối cùng gộp lại thành một bản đồ đặc trưng.



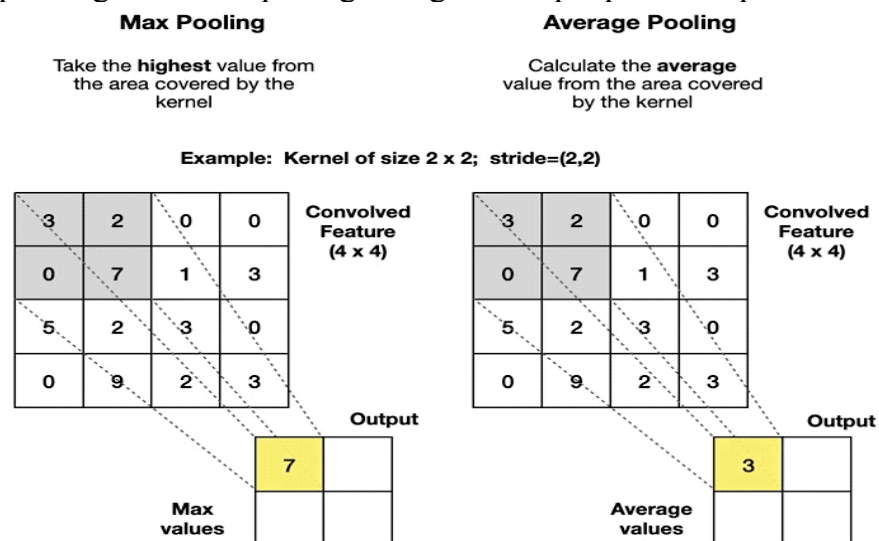
Hình 28 Phép tích chập thực hiện với một bộ lọc $2 \times 2 \times 3$

Sau đó, kết quả đầu ra của lớp tích chập sẽ sử dụng một hàm kích hoạt phi tuyến để thêm tính phi tuyến vào cho mạng. Một số hàm phi tuyến phổ biến như ReLU, sigmoid, v.v. Sau đó sẽ đưa kết quả làm đầu vào của lớp tiếp theo.

7.3. Lớp gộp(pooling layer)

Lớp gộp thường nằm xen giữa các lớp tích chập, để giảm kích thước dữ liệu nhưng vẫn giữ được những thuộc tính quan trọng. Việc giảm kích thước dữ liệu giúp giảm chi phí tính toán trong mô hình.

Gọi pooling size có kích thước $k \times k$. Đầu vào của lớp gộp là một bản đồ đặc trưng có kích thước $H \times W \times D$. Tách bản đồ đặc trưng thành D ma trận có kích thước $H \times W$. Với mỗi ma trận, trên vùng kích thước $k \times k$ của trên các ma trận, tìm giá trị lớn nhất hoặc trung bình của vùng kích thước rồi điền vào ma trận kết quả. Quy tắc padding và stride áp dụng tương tự như phép tích chập.



Hình 29 Lớp pooling với pooling size = 2, padding 0, stride=2

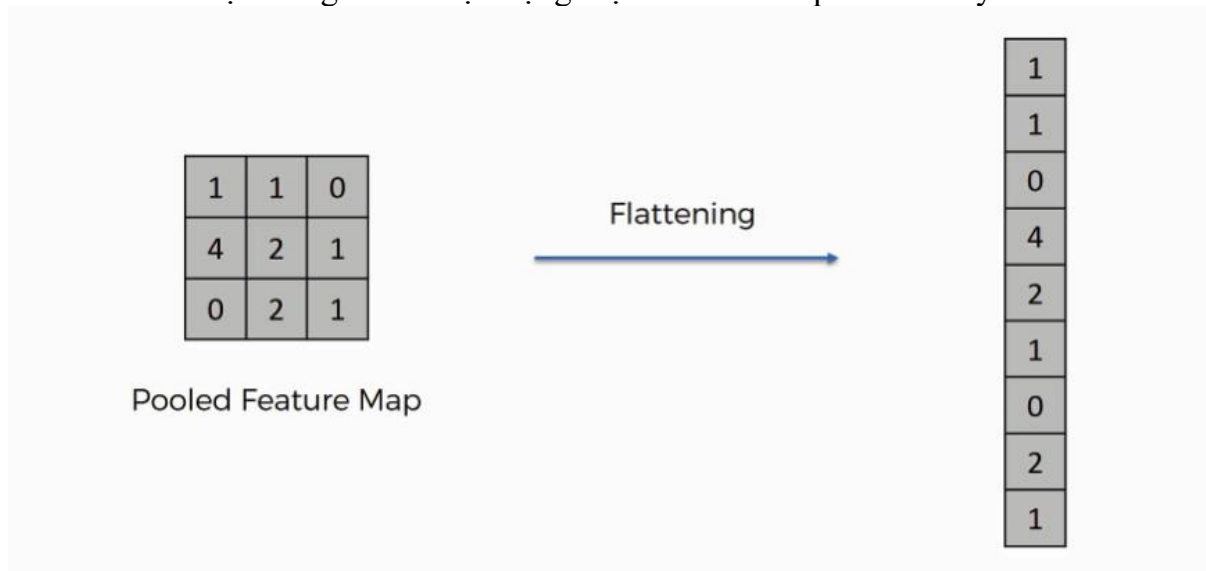
Nhưng hầu hết trường hợp, lớp gộp thường dùng $\text{size}=(2,2)$ và $\text{stride}=2$, $\text{padding}=0$. Khi đó kết quả chiều cao và chiều rộng dữ liệu sẽ giảm đi một nửa, độ sâu không thay đổi.

Có hai loại pooling phổ biến: Max Pooling, Average Pooling. Hình 29 mô tả về cách hoạt động của hai loại gộp phổ biến là lấy giá trị lớn nhất và lấy giá trị trung bình trong vùng gộp.

7.4. Lớp trải phẳng(flatten layer)

Lớp trải phẳng được sử dụng để chuyển đổi bản đồ đặc trưng nhận được sau khi thực hiện các thao tác ở lớp tích chập và lớp gộp thành định dạng mà lớp kết nối đầy đủ mong muốn làm đầu vào.

Bản đồ đặc trưng cơ bản là một mảng nhiều chiều chứa các giá trị điểm ảnh, lớp kết nối đầy đủ lại yêu cầu mảng một chiều làm đầu vào để xử lý. Vì vậy mà lớp trải phẳng được sử dụng để làm phẳng bản đồ đặc trưng thành một mảng một chiều cho lớp kết nối đầy đủ.



Hình 30 Bản đồ đặc trưng được trải phẳng

7.5. Lớp kết nối đầy đủ(fully-connected layer)

Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.

7.6. Lớp đầu ra(output layer)

Nhận đầu vào từ lớp kết nối đầy đủ trước đó và thực hiện hàm kích hoạt có tác vụ phân loại như là sigmoid với 2 lớp, nếu nhiều hơn 2 lớp thì dùng hàm softmax để chuyển đổi đầu ra của mỗi lớp thành xác suất của mỗi lớp.

Chương III: Kết quả thực hiện

1. Công nghệ sử dụng

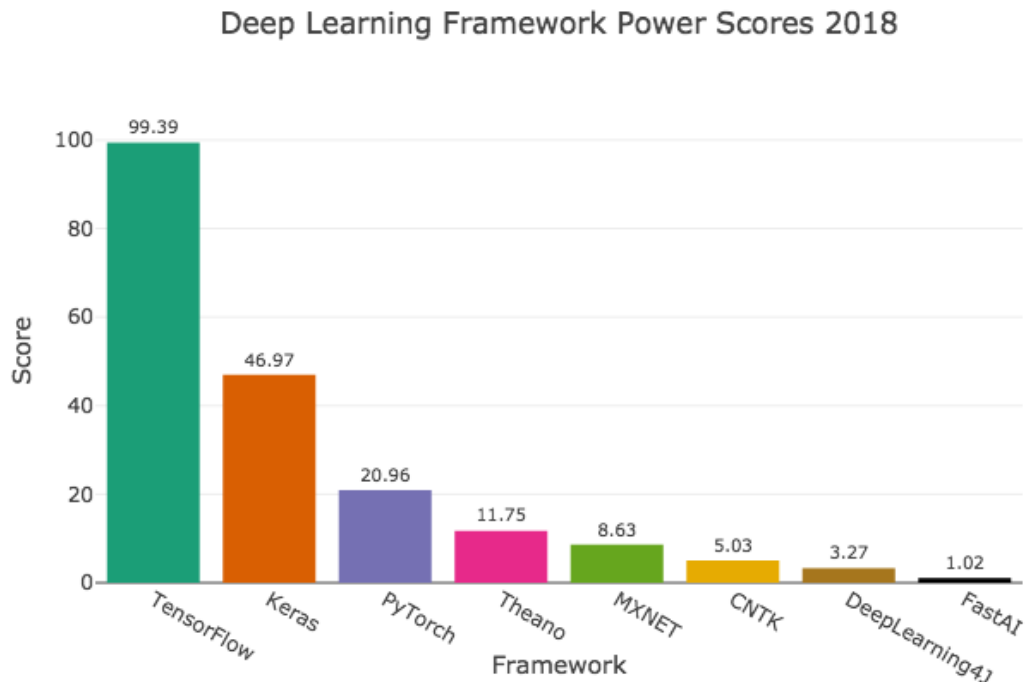
1.1. Google Colab

Google Colab, hay còn gọi là Google Colaboratory, là một môi trường Jupyter Notebook miễn phí được lưu trữ trên nền tảng đám mây Google. Colab cho phép bạn viết mã Python, thực thi các tác vụ học máy và trực quan hóa dữ liệu một cách trực tuyến, mà không cần cài đặt phần mềm hay cấu hình máy tính.

1.2. Keras

Bằng python thuần có thể xây dựng được một mạng nơ-ron tích chập để giải quyết bài toán được đặc ra nhưng làm như vậy sẽ rất mất thời gian, công sức mà kết quả lại không đạt được như mong muốn vì vậy mà các deep learning framework ra đời hỗ trợ việc xây dựng các mô hình học sâu dễ dàng hơn, hiệu suất cao, tiết kiệm chi phí.

Deep learning framework là tập hợp các thư viện, các nền tảng phần mềm chuyên dụng cung cấp các công cụ để xây dựng, huấn luyện, triển khai mô hình học sâu hiệu quả. Deep learning rất hữu ích với các kiến trúc mạng nơ-ron phức tạp chẳng hạn như mạng nơ-ron tích chập đã cho ra hiệu suất vượt trội so với không sử dụng framework. Các framework cung cấp một loạt các chức năng bao gồm vi phân tự động(automatic differentiation), tối ưu thuật toán, tăng tốc GPU và hỗ trợ đa dạng các mạng nơ-ron khác nhau.



Hình 31 Các deep learning framework phổ biến

Có thể thấy tensorflow là framework phổ biến nhất tuy nhiên khá khó sử dụng cho người mới bắt đầu cho nên trong niên luận này sẽ sử dụng keras, dễ sử dụng, thân thiện hơn với người dùng nhưng có đầy đủ tính năng mà một deep learning framework có.

Keras là một API mạng nơ-ron mã nguồn mở viết bằng python được thiết kế cho phép trải nghiệm nhanh với các mạng nơ-ron sâu(deep neural network). Nó được phát triển với trọng tâm là tính thân thiện với người dùng, tính module, khả năng mở rộng cho phép người dùng xây dựng mô hình học sâu một cách dễ dàng.

2. Xây dựng mô hình

2.1. Khai báo những thư viện cần thiết

Keras là thư viện để xây dựng mô hình mạng nơ-ron tích chập. Thư viện cung cấp module khởi tạo một mô hình và có các lớp được xây dựng sẵn như lớp tích chập, lớp gộp, lớp trải phẳng, lớp kết nối đầy đủ để xây dựng mạng nơ-ron tích chập. Mô hình được khởi tạo có thể được huấn luyện với dữ liệu huấn luyện và được kiểm tra với dữ liệu kiểm thử.

Matplotlib là thư viện dùng để trực quan hóa dữ liệu hình ảnh cây khoai tây và tạo biểu kết quả đánh giá mô hình huấn luyện.

Google Colab cung cấp API kết nối với google drive nơi mà dự án đặt dữ liệu để huấn luyện mô hình.

NumPy là thư viện được sử dụng để thao tác với mảng và ma trận một cách hiệu quả.

Module os trong Python là một thư viện cung cấp các hàm để tương tác với hệ điều hành, cho phép bạn thực hiện các thao tác như quản lý tập tin, quản lý thư mục, lấy thông tin hệ thống.

2.2. Chuẩn bị dữ liệu

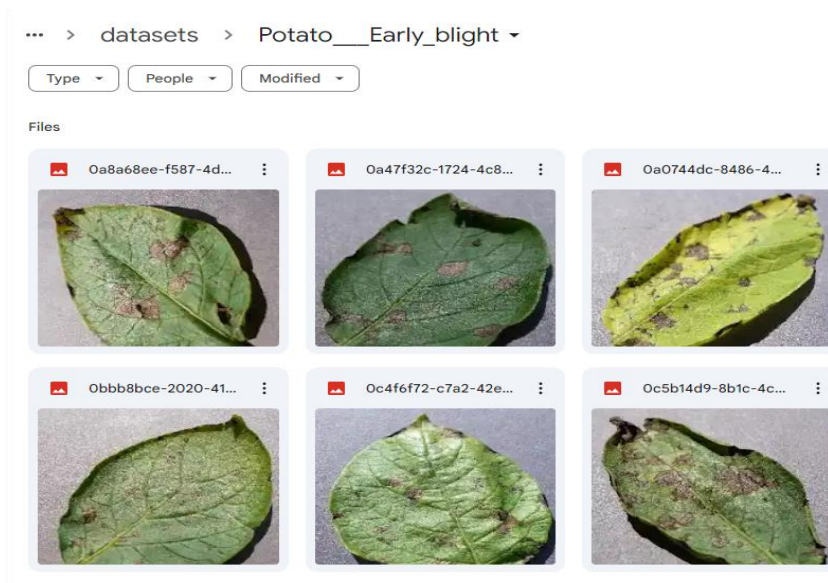
2.2.1. Chuẩn bị dữ liệu

Keras cung cấp API cho phép tải dữ liệu hình ảnh từ thư mục, cụ thể ở đây là thư mục trên google drive nơi chứa datasets huấn luyện.

Sử dụng hàm của thư viện keras để tải dữ liệu hình ảnh: *image_dataset_from_directory* vào biến. Với bộ tham số như hình trên thì dữ liệu được tải vào như sau: dữ liệu đã được xáo trộn lên, kích thước của mỗi hình ảnh là 256x256 và được chia thành các batch khác nhau, mỗi batch chứa 32 hình ảnh.

Batch size là số lượng mẫu dữ liệu trong một lần huấn luyện. Ví dụ, trong bài toán phân loại chó mèo, chọn batch size =32, nghĩa là 1 lần lặp ta sẽ cho ngẫu nhiên 32 bức nhìn chó hoặc mèo chạy lan truyền tiến trong mạng neural network.

Kết quả sau khi tải dữ liệu là có 2152 tấm ảnh được chia thành 3 lớp như sau: lớp khỏe mạnh, lớp ứa sớm, lớp héo muộn.



Hình 32 Một vài hình ảnh về tập huấn luyện

2.2.2. Chia tập dữ liệu

Chia tập dữ liệu thành 3 phần tập huấn luyện (training set), tập kiểm tra(validation set) và tập kiểm thử(test set) với tỉ lệ 8:1:1.

Mục đích cuối cùng của bài toán không phải là dự đoán tốt với tập dữ liệu huấn luyện đang có mà là dữ liệu thực tế vì vậy mà để mô hình học được tổng quan hóa dữ liệu thay vì chỉ biết nhớ dữ liệu trong tập huấn luyện. Để đánh giá xem mô hình có hiệu năng như thế nào ngoài thực tế, người ta chia tập dữ liệu thành 3 tập training set, validation set, test set.

Sử dụng tập train set để huấn luyện mô hình và đánh giá mô hình trên tập validation set. Nếu có nhiều mô hình hơn thì mô hình nào có hiệu năng tốt hơn trong tập validation set thì mô hình đó được chọn. Và cuối cùng mô hình tốt nhất sẽ được đánh giá dựa trên tập test set làm hiệu suất của mô hình khi dùng thực tế.

```
Dataset batch size 68
Train batch size= 54, val batch size= 6 test batch size =8
```

Hình 33 Chia tập dữ liệu huấn luyện thành 3 phần

Hình 33 là kết quả sau khi chia tập dữ liệu thành 3 phần tổng cộng có 68 batch, phần huấn luyện chứa 54 batch, phần kiểm tra chứa 6 batch và phần kiểm thử chứa 8 batch, mỗi batch có 32 tấm ảnh ngẫu nhiên về lá cây khoai tây.

2.2.3. Tăng cường dữ liệu

Tạo thêm dữ liệu huấn luyện bằng data augmentation giúp đa dạng tập huấn luyện. Sử dụng cắt ngẫu nhiên và xoay ngẫu nhiên để tăng tính đa dạng dữ liệu.

```
data_augmentation= tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    layers.experimental.preprocessing.RandomRotation(0.2),
])
```

Hình 34 Tăng cường dữ liệu

Hình 34 trong mô hình sử dụng 2 cách để làm giàu dữ liệu hình ảnh: đó là cắt ngẫu nhiên ngang và dọc hình ảnh kết hợp với xoay ngẫu nhiên.

2.2.4. Tiền xử lý dữ liệu

Chuẩn hóa kích thước ảnh đầu vào và chuẩn hóa phạm vi điểm ảnh giúp mô hình tính toán tốt hơn. Hình 35 cho thấy mô hình sử dụng ảnh có kích thước 256* 256 và sau khi chuẩn hóa phạm vi điểm ảnh giao động [0, 1].

```
resize_and_rescale= tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1.0/255),
])
```

Hình 35 Chuẩn hóa kích thước ảnh và chuẩn hóa điểm ảnh

2.3. Xây dựng mô hình

Mô hình chung của bài toán nhận dạng hình ảnh bằng mạng nơ-ron tích chập là Hình ảnh đầu vào → Lớp tích chập → Lớp Pooling → Lớp kết nối đầy đủ → Lớp kết quả.

Trong lớp tích chập sử dụng bộ lọc có kích thước (3,3) và hàm kích hoạt là relu. Trong lớp gộp, sử dụng kích thước max pooling với kích thước là (2*2). Dưới đây là hình ảnh mô hình mạng nơ-ron tích chập xây dựng để nhận dạng hình ảnh.

```

model= tf.keras.Sequential(
    [
        resize_and_rescale,
        data_augmentation,
        layers.Conv2D(32, (3, 3),activation='relu', input_shape=input_shape ),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3),activation='relu', ),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3),activation='relu', ),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3),activation='relu', ),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3),activation='relu', ),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3),activation='relu', ),
        layers.MaxPooling2D((2, 2)),
        layers.Flatten(),
        layers.Dense(64, activation='relu'),
        layers.Dense(n_classes, activation='softmax')
    ]
)

```

Hình 36 Mô hình mạng nơ-ron tích chập

| STT | Tên lớp | Tham số | Hàm kích hoạt |
|-----|----------------|-------------------------------------|---------------|
| 1 | Lớp tích chập | kernel number=32. kernel size=(3,3) | ReLU |
| 2 | Lớp gộp | pooling size=(2,2) | |
| 3 | Lớp tích chập | kernel number=64, kernel size=(3,3) | ReLU |
| 4 | Lớp gộp | pooling size=(2,2) | |
| 5 | Lớp tích chập | kernel number=64, kernel size=(3,3) | ReLU |
| 6 | Lớp gộp | pooling size=(2,2) | |
| 7 | Lớp tích chập | kernel number=64 ,kernel size=(3,3) | ReLU |
| 8 | Lớp gộp | pooling size=(2,2) | |
| 9 | Lớp tích chập | kernel number=64, kernel size=(3,3) | ReLU |
| 10 | Lớp gộp | pooling size=(2,2) | |
| 11 | Lớp tích chập | kernel number=64, kernel size=(3,3) | ReLU |
| 12 | Lớp gộp | pooling size=(2,2) | |
| 13 | Lớp trải phẳng | | |

| STT | Tên lớp | Tham số | Hàm kích hoạt |
|-----|--------------------|----------------|---------------|
| 14 | Lớp kết nối đầy đủ | node number=64 | ReLU |
| 15 | Lớp kết quả | node number=3 | softmax |

Bảng 1: Bảng tham số của mô hình mạng nơ-ron tích chập

Adam là thuật toán tối ưu hóa được sử dụng rộng rãi trong học máy, đặc biệt là cho các mô hình mạng nơ-ron. Adam là một thuật toán thích ứng, nghĩa là nó tự điều chỉnh tốc độ học cho mỗi tham số của mô hình trong quá trình huấn luyện.

Categorical Cross Entropy (CCE) là một hàm mất mát (loss function) thường được sử dụng trong các bài toán phân loại đa lớp (multi-class classification) với mạng nơ-ron. Nó đo lường sự khác biệt giữa phân bố xác suất dự đoán của mô hình (tức là kết quả đầu ra của mạng) và phân bố xác suất thực tế (ground truth) của các lớp. Nói cách khác, CCE cho biết mô hình dự đoán các lớp chính xác đến mức nào.

```
[ ] model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
    metrics=['accuracy']
)
```

Hình 37 Biên dịch mô hình mạng nơ-ron

Huấn luyện mô hình với dữ liệu hình ảnh với epochs là 20 tương ứng với 20 lần huấn luyện.

Chương IV: Đánh giá kiểm thử

1. Các tiêu chí đánh giá

Để đánh giá một mô hình máy học hoạt động có tốt hay không ta đánh giá hiệu năng trên tập dữ liệu mới cụ thể là tập test set đã được chia ra trước khi huấn luyện mô hình. Việc đánh giá mô hình giúp xác định xem mô hình có thành công hay chưa, mức độ thành công là bao nhiêu, có nên dừng lại huấn luyện, có nên cải tiến cái gì. Bài toán nhận dạng bệnh trên lá cây khoai tây là bài toán phân lớp nên có một số tiêu chí đánh giá sau: độ chính xác (accuracy), precision, recall.

Khi thực hiện bài toán phân loại sẽ có 4 trường hợp dự đoán xảy ra:

- True Positive(TP): đối tượng thuộc lớp Positive mô hình dự đoán lớp Positive(dự đoán đúng).
- True Negative(TN): đối tượng thuộc lớp Negative mô hình dự đoán Negative(dự đoán đúng).
- False Negative(FN): đối tượng thuộc lớp Positive mô hình dự đoán Negative(dự đoán sai).
- False Positive(FP): đối tượng thuộc lớp Negative mô hình dự đoán Positive(dự đoán sai).

Hình 38 biểu diễn ma trận nhầm lẫn.

| | | True Class | |
|-----------------|----------|------------|----------|
| | | Positive | Negative |
| Predicted Class | Positive | TP | FP |
| | Negative | FN | TN |

Hình 38 Ma trận nhầm lẫn(Confusion Matrix)

1.1. Độ chính xác(accuracy)

Accuracy được định nghĩa là tỷ lệ phần trăm dự đoán đúng cho dữ liệu thử nghiệm. Nó có thể được tính toán dễ dàng bằng cách chia số lần dự đoán đúng cho tổng số lần dự đoán.

$$accuracy = \frac{\text{correct predictions}}{\text{all predictions}}$$

1.2. Precision

Precision được định nghĩa là phần nhỏ của các ví dụ có liên quan (tích cực thực sự) trong số tất cả các ví dụ được dự đoán là thuộc một lớp nhất định.

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

1.3. Recall

Recall được định nghĩa là phần nhỏ của các ví dụ được dự đoán thuộc về một lớp so với tất cả các ví dụ thực sự thuộc về lớp đó.

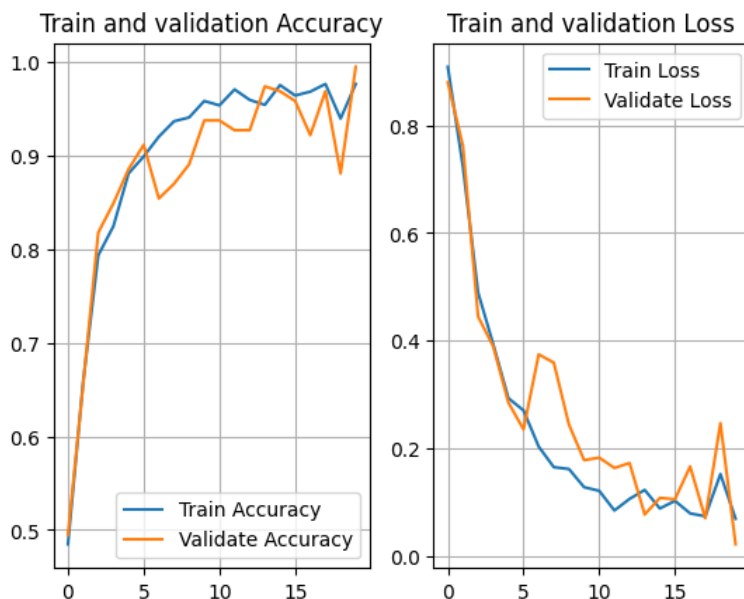
$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Precision và recall sẽ rất hữu ích với trường hợp dữ liệu phân bố không đồng đều. Ví dụ như phát triển một thuật toán phân loại dự đoán xem ai đó có mắc bệnh hay không. Giả sử, tỷ lệ dân số mắc bệnh là 1%, công cụ dự đoán đưa ra kết quả là tất cả mọi người đều không mắc bệnh. Vậy công cụ vẫn đạt tỷ lệ chính xác là 99%. Tuy nhiên, chúng ta sẽ không tìm được ai là người mắc bệnh. Vậy độ hữu ích của công cụ là 0%.

Vì vậy, điều quan trọng là phải đánh giá cả Precision và Recall của một mô hình để xác định ra được người bị bệnh hay không mà không có nhầm lẫn.

2. Kết quả đánh giá

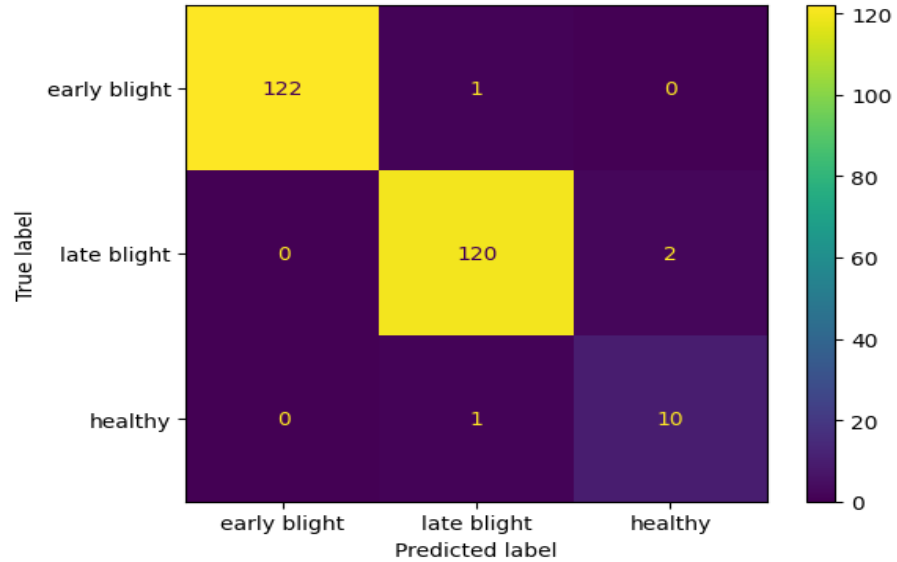
- Biểu đồ độ chính xác và loss:



Hình 39 Biểu đồ độ chính xác và loss của mô hình

Nhìn vào hình 39 thấy rằng độ chính xác được cải thiện dần và sai số giảm đi qua 20 lần học

- Ma trận nhầm lẫn(confusion matrix):



Hình 40 Ma trận nhầm lẫn kết quả với dữ liệu kiểm thử

Từ ma trận này cho thấy có sự chênh lệch giữa lá khỏe mạnh và lá bệnh dẫn đến có thể dự đoán sai, cần thêm tiêu chí precision và recall để đánh giá mô hình/

- Các tiêu chí đánh giá như độ chính xác, precision, recall:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0.0 | 1.00 | 1.00 | 1.00 | 138 |
| 1.0 | 1.00 | 0.97 | 0.98 | 101 |
| 2.0 | 0.85 | 1.00 | 0.92 | 17 |
| accuracy | | | 0.99 | 256 |
| macro avg | 0.95 | 0.99 | 0.97 | 256 |
| weighted avg | 0.99 | 0.99 | 0.99 | 256 |

Hình 41 Kết quả các tiêu chí đánh giá

Mô hình dự đoán khá tốt với các lá bị bệnh tuy nhiên với các lá khỏe mạnh còn hạn chế vì số lượng chênh lệch dữ liệu kiểm thử giữa lá khỏe mạnh và lá bệnh.

- Kiểm tra với hình ảnh cụ thể:

Actual: Potato__Late_blight
 Predicted: Potato__Late_blight
 Confidence: 99.24 %



Actual: Potato__healthy
 Predicted: Potato__healthy
 Confidence: 98.8 %



Actual: Potato__Early_blight
 Predicted: Potato__Early_blight
 Confidence: 99.99 %



Actual: Potato__Early_blight
 Predicted: Potato__Early_blight
 Confidence: 99.9 %



Actual: Potato__Early_blight
 Predicted: Potato__Early_blight
 Confidence: 99.43 %



Actual: Potato__Early_blight
 Predicted: Potato__Early_blight
 Confidence: 99.66 %



Hình 42 Kết quả dự đoán với dữ liệu kiểm thử

Phần kết luận

1. Kết quả đạt được

- Huấn luyện thành công mô hình phát hiện bệnh trên lá cây khoai tây gồm có 3 lớp: khỏe mạnh, mốc sương, đốm nâu.
- Phân biệt được các loại bệnh với độ chính xác cao, độ chính xác tổng thể lên đến hơn 90%.
- Khi kiểm thử cho thấy khả năng phân biệt các loại bệnh với hình ảnh thực tế với độ tin cậy cao, giúp phát hiện bệnh trên lá cây khoai tây hiệu quả.

2. Hướng phát triển

- Tiếp tục nghiên cứu nâng cao chất lượng, cải thiện độ chính xác của mô hình.
- Mở rộng mô hình phát hiện thêm nhiều loại bệnh liên quan đến lá cây khoai tây.
- Phát triển mô hình thành một ứng dụng web hoặc di động để dễ dàng cho người sử dụng.

Tài liệu tham khảo

Các website tham khảo

- [1] *stanford* Available: <https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-convolutional-neural-networks>
- [2] *elcom* Available: <https://www.elcom.com.vn/deep-learning-la-gi-ung-dung-cua-hoc-sau-trong-thuc-tien-1666582364>
- [3] *deepai* Available: <https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network>
- [4] *ibm* Available: <https://www.ibm.com/topics/convolutional-neural-networks>
- [5] *datahacker* Available: <https://datahacker.rs/edge-detection/>
- [6] *sisu.ut.ee* Available: <https://sisu.ut.ee/imageprocessing/book/3>
- [7] *raghul-719.medium* Available <https://raghul-719.medium.com/basics-of-computer-vision-1-image-resizing-97fca504cd63>
- [8] *picsellia* Available <https://www.picsellia.com/post/image-data-augmentation>
- [9] *simplilearn* Available: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network>
- [10] *machinelearningmastery* Available <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>
- [11] *funo* Available: <https://funo.vn/kien-thuc-nong-nghiep/phan-bon-dam/su-dung-dam-hop-ly-giup-han-che-benh-dom-vong-o-khoai-tay>
- [12] *camnangcaytrong* Available <http://camnangcaytrong.com/benh-moc-suong-heo-muon-bd60.html>

Sách tham khảo

Sách deep learning cơ bản - Nguyễn Thanh Tuấn
<https://nttuan8.com/sach-deep-learning-co-ban/>

Phụ lục

1. Dataset <https://www.kaggle.com/datasets/arjuntejaswi/plant-village>
2. Github https://github.com/LuanPhan2911/potato_disease_classification