NHẬN DẠNG LÁ CÂY THUỐC BẰNG THUẬT TOÁN CNN

MỤC LỤC

DANH MỤC HÌNH

MỞ ĐẦU

1. Lí do chọn đề tài

Trong giai đoạn gần đây, cụm từ cách mạng công nghiệp lần thứ tư được nhắc đến rất nhiều trên truyền thông. Khái niệm cách mạng công nghiệp 4.0 được đưa ra dựa trên xu thế bùng nổ của nhiều công nghệ mới như: công nghệ truyền dẫn tốc độ cao, CPU rất mạnh, GPU hàng ngàn nhân, công nghệ chế tạo bộ nhớ đạt nhiều đột phá. Song song với đó, công nghệ dần làm các thiết bị nhỏ lại, nhưng sức mạnh tăng lên đáng kể. Công nghệ mạch nhúng cũng bùng nổ không kém với nhiều mạch nhúng nhỏ, sử dụng lượng điện khiêm tốn, nhưng khả năng tính toán lên tới hàng ngàn tỉ phép tính trên giây, tương đương sức mạnh của 1 siêu máy tính cách đây khoảng 20 năm. Ví dụ như board mạch Nvidia's Jetson TX1 được ra mắt năm 2015, có thể nằm vừa trong lòng bàn tay và chỉ sử dụng 10 oát điện, đã có sức mạnh tính toán tới 1 ngàn tỉ phép tính trên giây, điều mà cách đó tròn 20 năm, vào năm 1996, siêu máy tính ASCI Red của Intel, phải sử dụng tới 6000 vi xử lý Pentium Pros, vận hành với 1000 kW mới có thể đạt được sức mạnh tính toán trên.

Máy tính hiện nay có sức mạnh tính toán lớn nhưng giá thành lại ở mức phổ thông, dẫn tới người làm nghiên cứu rất dễ dàng để có thể tự kiểm nghiệm được các lý thuyết về trí tuệ nhân tạo từ nhiều năm trước. Cùng với mã nguồn mở, hiện làn sóng trí tuệ nhân tạo đang bùng nổ mạnh mẽ trong thời gian gần đây, và đem lại rất nhiều ứng dụng trong đời sống.

Ngày nay, với sự phát triển của công nghệ bán dẫn, máy tính ngày càng nhỏ đi, năng lượng tiêu thụ ngày càng thấp xuống, trong khi tốc độ xử lý lại ngày càng tăng lên. Với những ưu điểm như vậy, chúng ta có thể thấy rất nhiều thiết bị thông minh đã và đang hiện diện mọi nơi trong đời sống, với camera nhiều điểm ảnh, bộ nhớ trong lớn và vi xử lý mạnh như: điện thoại thông minh, máy ảnh kỹ thuật số, camera hành trình,... Ngoài ra, với sự bùng nổ của xu hướng mạng vạn vật IOT, người ta có thể sẽ còn thấy rất nhiều thiết bị thông minh mới xuất hiện: xe ô tô tự lái, thiết bị bay không người lái tự giao hàng,... Có thể thấy, việc sử dụng trí thông minh nhân tạo để khai thác dữ liệu hình ảnh trong các thiết bị thông minh trong tương lai đã và đang trở thành xu hướng. Từ nhận định trên tôi quyết định chọn nội dung “Tìm hiểu mạng CNN và ứng dụng nhận dạng lá cây thuốc chữa bệnh trong dân gian” để làm đề tài báo cáo cuối kì trong môn trí tuệ nhân tạo.

2. Cấu trúc đề tài

Báo cáo được tổ chức gồm 5 chương gồm:

-Chương 1: Giới thiệu về bài toán phân loại lá cây thuốc, giới thiệu một số hướng tiếp cận, ưu nhược điểm của các hướng tiếp cận, khó khăn và thách thức. Cuối cùng là

hướng giải quyết.

-Chương 2: Trình bày về một số kiểu mạng nơ-ron và cơ chế lan truyền ngược. Cuối cùng là giới thiệu về cấu tạo và cách hoạt động của mạng nơ-ron tích chập. Trình bày tổng quan về bài toán nhận dạng bằng mạng nơ-ron tích chập (CNN)

-Chương 3: Thu thập và xử lý dữ liệu. Tìm hiểu các bài báo về các lá cây thuốc chữa bệnh trong dân gian. Sau khi đã có danh sách các lá cây thuốc chữa bệnh tiến hành đi thu thập hình ảnh của các lá cây thuốc nhằm phục vụ việc huấn luyện model.

-Chương 4: Xây dựng mô hình huấn để giải quyết bài toán tìm lá cây thuốc bằng hình ảnh.

-Chương 5: Cuối cùng là phần kết luận kết quả đã đạt được và nêu ra những tồn tại, dựa vào đó để đưa ra những mục tiêu và phương hướng phát triển cho hệ thống sau này.

CHƯƠNG 1
TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu

Việt Nam là một nước có rất nhiều loại lá cây thuốc quý hiếm, ngành đông y đã có một bề dày lịch sử đáng kể. Hiện nay ngành thuốc tây y đã phát triển đột phá về công nghệ nhưng không thể phủ nhận giá trị mà các loại thuốc đông y mang lại, vẫn còn đó những loại bệnh mà cần phải kết hợp giữa đông và tây y.

Trên thế giới đã có một số công trình nghiên cứu về nhận dạng lá cây thuốc. Bằng cách vận dụng kỹ thuật xử lý ảnh và nhận dạng đã đạt được một số kết quả khả quan. Các hệ thống nhận dạng chỉ được thử nghiệm trên điều kiện lý tưởng, các mẫu vật được chụp một cách rõ ràng và có thể thấy rõ chi tiết của lá, nhưng chưa áp dụng vào cuộc sống đại trà.

Với công nghệ phát triển đột phá trong thập kỉ qua, việc đưa máy học và mạng nơ-ron nhân tạo vào để phân loại đã không là một điều gì quá khó khăn. Hiện tại ở Việt Nam vẫn chưa có nhiều công trình kết hợp học sâu để phân loại lá cây thuốc. Tôi quyết định thực hiện đề tài này với mục đích nghiên cứu các công nghệ mới hiện nay để tạo tiền đề cho việc ứng dụng trí tuệ nhân tạo vào thực tiễn đời sống.

1.2. Hướng tiếp cận

- Thu thập kiến thức và dữ liệu về lá cây thuốc dân gian trong điều trị bệnh
- Tìm hiểu mạng học sâu, các mô hình mạng nơ-ron và cách hoạt động của chúng.
- Xây dựng một hệ thống phân tích và nhận dạng được những loại lá cây thuốc đã được thu thập.

1.3. Khó khăn và thách thức

- Mô hình thực hiện bài toán cần phải có bộ dữ liệu lá cây thuốc để huấn luyện Model. Đây là thách thức lớn nhất của bài toán, cần rất nhiều thời gian để thu thập và xử lý.
- Đề tài tập trung vào các lá cây thuốc dân gian để chữa bệnh vì vậy tôi cũng phải tìm kiếm những nguồn tin uy tín và đáng tin cậy trên các trang web cây thuốc lớn tại Việt Nam.
- Khó khăn tiếp theo là vấn đề tạo nên một giao diện dễ dàng sử dụng cho người dụng. Vì tôi chưa được tiếp cận nhiều với lập trình hướng đối tượng chuyên nghiệp.

1.4. Hướng giải quyết

- Để giải quyết bài toán phân loại các cây thuốc, tôi sử dụng thuật toán CNN.

- Bước đầu cần tìm kiếm thu thập và xây dựng kho dữ liệu hình ảnh lá cây thuốc được lưu truyền trong dân gian và có trong kho sách về thuốc chữa bệnh của Việt Nam. Từ đó chọn lọc và tiến hành gán nhãn để thực hiện bước huấn luyện.
- Mô hình sẽ học và trích lọc các đặc trưng của từng loại lá cây khác nhau.
- Bước cuối cùng là đưa ứng dụng kết hợp với model phân loại lên giao diện để đưa đến tay người dùng. Model được đưa vào phần tìm kiếm bằng hình ảnh, từ đó sẽ đưa ra được tên của loại lá cây mà người dùng đang tìm kiếm và các bài thuốc sử dụng loại lá cây đó.

CHƯƠNG 2
CƠ SỞ LÝ THUYẾT

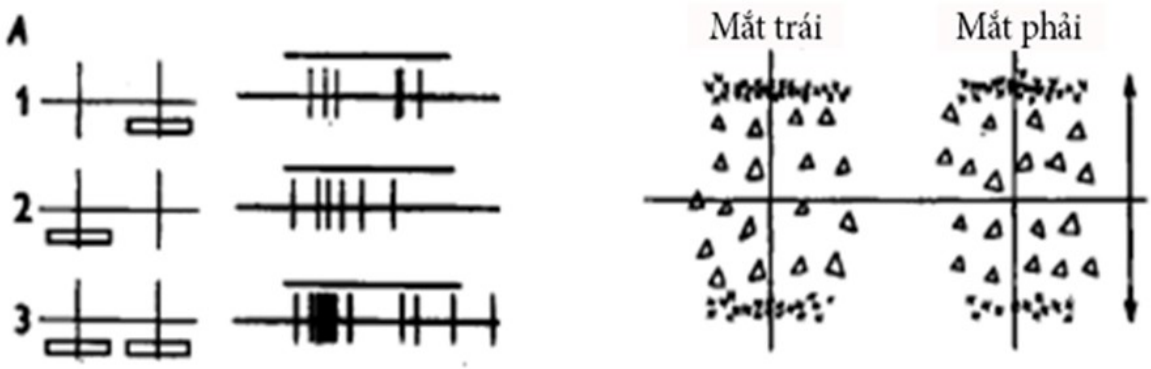
2.1. Giới thiệu về mạng nơ-ron

2.1.1 Lịch sử của nơ-ron nhân tạo

Vào năm 1943, nhà thần kinh học Warren McCulloch đã cùng nhà toán học Walter Pitts đã viết một cuốn sách về cách mạng thần kinh hoạt động. Và họ đã thực hiện mô phỏng một mạng thần kinh đơn giản trên một mạch điện.

Vào năm 1949, Donald Hebb đã viết cuốn sách Organization of Behavior. Điểm nhấn chính là mạng thần kinh nào được sử dụng nhiều sẽ được tăng cường.

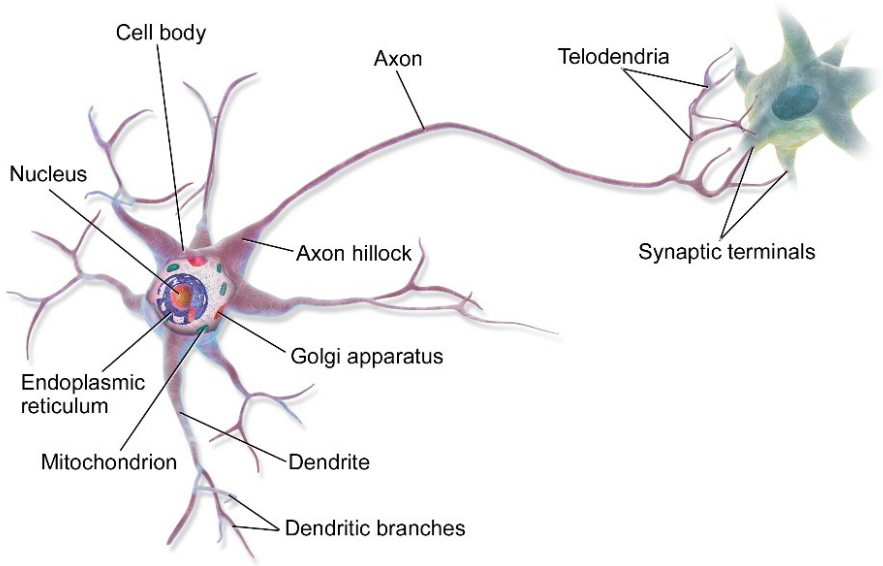
Vào năm 1959, David Hubel và Torsten Wiesel đã xuất bản cuốn sách Receptive fields of single neurons in the cat's striate cortex, miêu tả về phản ứng của các tế bào thần kinh thị giác trên loài mèo, cũng như cách loài mèo ghi nhớ và nhận diện hình dạng trên kiến trúc vỏ não của nó.



Hình 2.1 Hình ảnh thí nghiệm của David Hubel và Torsten Wiesel trên mèo

Vào năm 1989, Yann LeCun đã áp dụng thuật toán học cho mạng nơ-ron theo kiểu lan truyền ngược vào kiến trúc mạng nơ-ron tích chập của Fukushima. Sau đó vài năm, LeCun đã công bố LeNet-5. Có thể nói, LeNet-5 là một trong những mạng nơ-ron tích chập sơ khai nhất, tuy nhiên các dấu ấn của nó vẫn tồn tại tới ngày nay, có thể thấy thông qua một số thành phần thiết yếu mà các mạng nơ-ron tích chập của ngày nay vẫn đang sử dụng.

2.1.2 Cấu tạo và quá trình xử lý của một nơ-ron sinh học



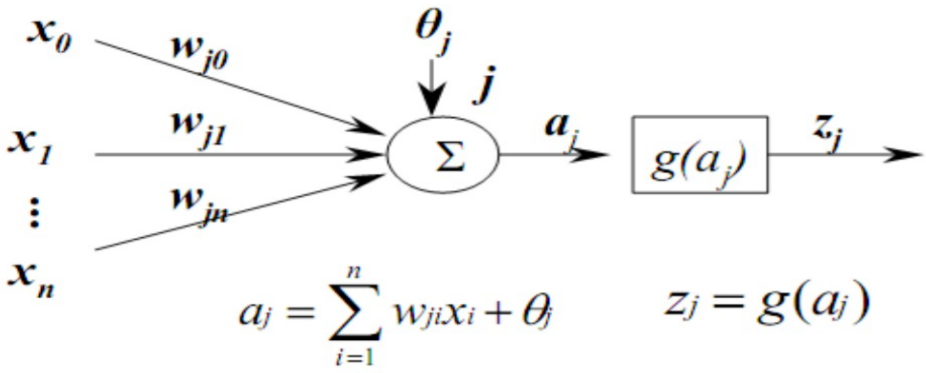
Hình 2.2 Cấu trúc một nơ-ron sinh học

Một số chức năng cơ bản của một nơ-ron sinh học:

- Cell body: là nơi xử lý tất cả tín hiệu đưa vào;
- Dendrite: là nơi nhận các xung điện vào trong nơ-ron;
- Axon: là nơi đưa tín hiệu ra ngoài sau khi được xử lý bởi nơ-ron;
- Synaptic terminals: vị trí nằm giữa Dendrite và Axon, đây là điểm liên kết đầu ra của nơ-ron này với đầu vào của nơ-ron khác.

2.1.3 Cấu tạo và quá trình xử lý của một nơ-ron nhân tạo

Dựa vào cấu tạo của một nơ-ron sinh học, các nhà khoa học nghiên cứu và lập trình đã đưa ra kiến trúc của một nơ-ron nhân tạo:



Hình 2.3 Cấu trúc xử lý của nơ-ron nhân tạo

Trong đó:

X_i : Là danh sách các đầu vào. Số lượng thuộc tính đầu vào thường nhiều hơn một, do dữ liệu thô đầu vào thường là một vector nhiều chiều, hoặc nhiều nơ-ron tầng

trước kết nối tới một nơ-ron tầng sau;

W_{ji} : Các trọng số tương ứng với các đầu vào;

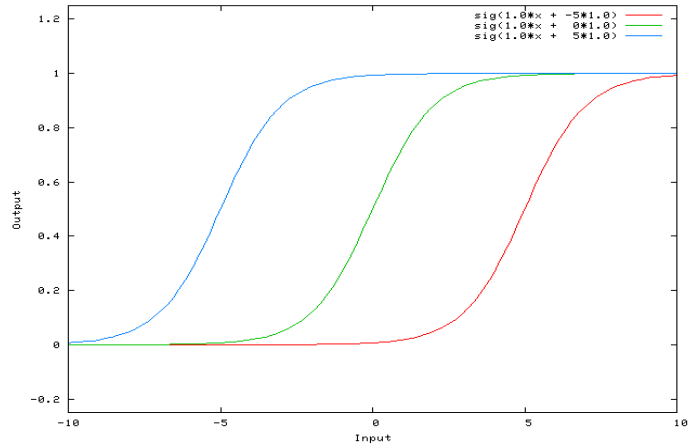
Θ_j : độ lệch (bias). Độ lệch được đưa vào sau khi khi tính toán xong hàm tổng, tạo ra giá trị cuối cùng trước khi đưa vào hàm truyền. Mục đích của việc thêm vào độ lệch nhằm dịch chuyển chức năng của hàm kích hoạt sang trái hoặc phải, giúp ích khi mạng được huấn luyện.

A_j : đầu vào mạng (net-input). Có chức năng tính tổng các tích của các đầu vào và trọng số tương ứng.

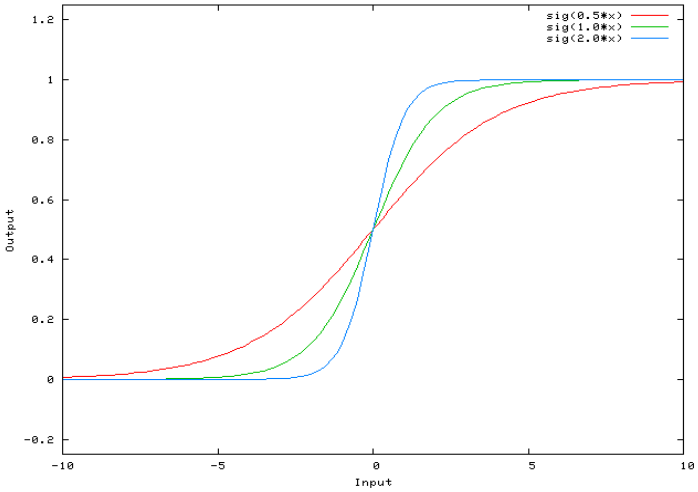
G(x): hàm chuyển (hàm kích hoạt).

Z_j : đầu ra của nơron

Một số hàm kích hoạt phổ biến hiện nay: Sigmoid, TanH, ReLU,..



Hình 2.4 Kết quả hàm sigmoid khi không sử dụng bias



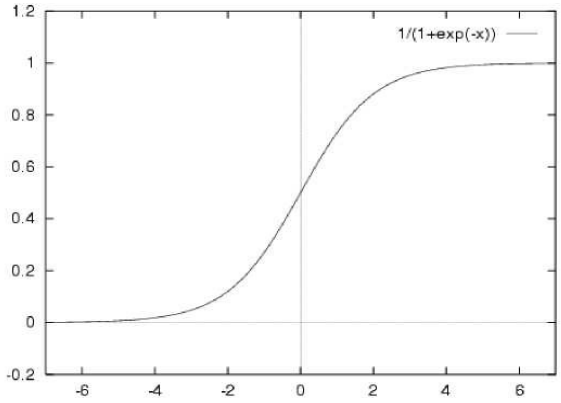
Hình 2.5 Kết quả hàm sigmoid khi sử dụng bias

2.1.4. Các mô hình hàm kích phổ hoạt phổ biến của mạng nơ rơn nhân tạo

2.1.4.1. Hàm Sigmoid

- Biểu diễn hàm:
- Đạo hàm của hàm:

Hàm sigmoid được sử dụng vì ngưỡng của nó nằm trong khoảng (0, 1). Do đó, hàm này được sử dụng nhiều cho các mô hình dự đoán xác suất đầu ra, tức kết quả chỉ tồn tại trong khoảng từ 0 đến 1: khi đầu vào là số dương lớn, đầu ra của hàm sigmoid gần bằng 1. Khi nhỏ hơn 0, đầu ra gần bằng 0. Tuy nhiên, việc tối ưu của hàm này khó khăn, nguyên nhân vì nếu giá trị đầu vào của hàm là 1 số rất lớn, thì đầu ra của hàm càng về 2 đầu xấp xỉ 1 hoặc 0, nên tốc độ hội tụ sẽ rất chậm.

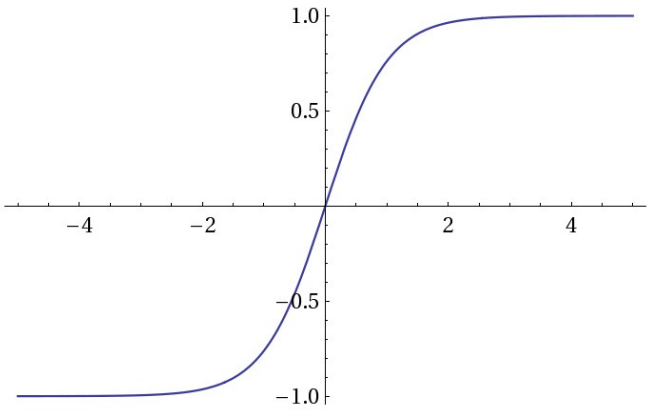


Hình 2.6 Đồ thị hàm Sigmoid

2.1.4.2. Hàm TanH

- Biểu diễn hàm:
- Đạo hàm của hàm:

Hàm TanH được sử dụng vì đầu ra của hàm nằm trong khoảng (-1,1), thích hợp với các mô hình đầu ra có ba giá trị: âm, trung tính (0) và dương. Chúng ta có thể thấy rõ hơn điều này trong hình minh họa .

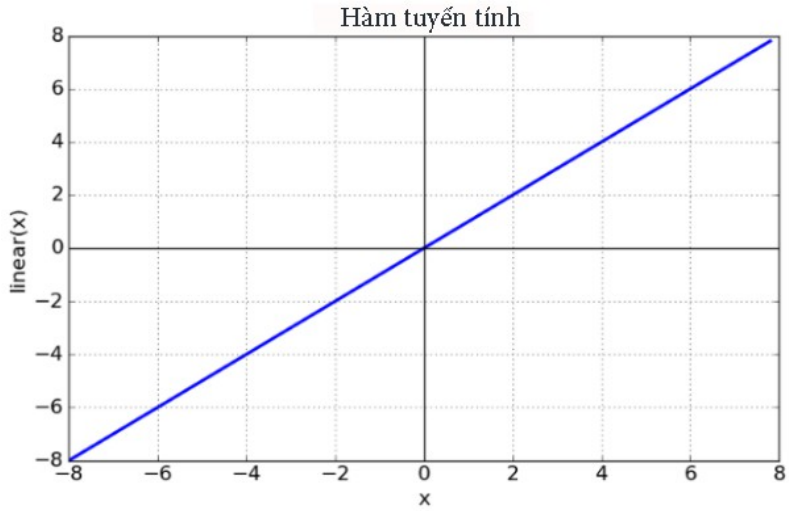


Hình 2.7 Đồ thị hàm TanH

2.1.4.3. Hàm tuyến tính

- Biểu diễn hàm: •
- Đạo hàm của hàm:

Hàm tuyến tính áp dụng thao tác nhận dạng trên dữ liệu với dữ liệu đầu ra tỷ lệ thuận với dữ liệu đầu vào.

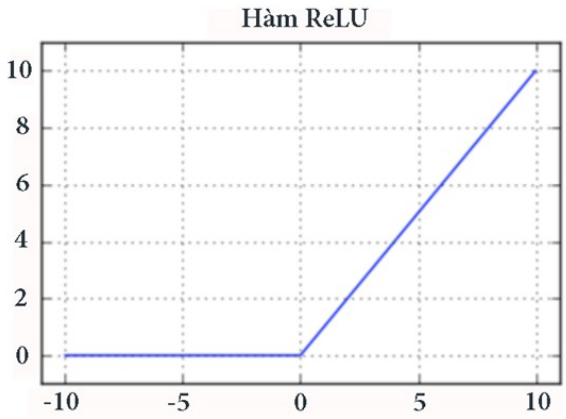


Hình 2.8 Đồ thị hàm tuyến tính

2.1.4.4. Hàm ReLU

- Biểu diễn hàm:
- Đạo hàm của hàm:

Hàm RELU áp dụng với những trường hợp cần đầu ra nằm trong khoảng (0,+∞), có tốc độ tính toán rất nhanh, gán các giá trị âm trở thành 0 ngay lập tức, phù hợp cho việc huấn luyện từ dữ liệu chuẩn. Tuy nhiên, điều này khiến hàm ReLU không ánh xạ các giá trị âm một cách thích hợp.

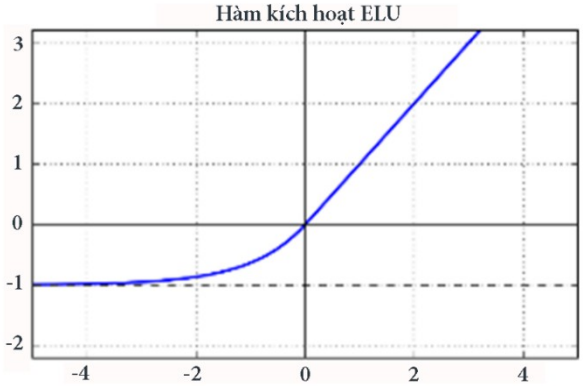


Hình 2.9 Đồ thị hàm ReLU

2.1.4.5. Hàm ELU

- Biểu diễn hàm:
- Đạo hàm của hàm:

Hàm ELU là một biến thể của hàm RELU. Hàm thường được sử dụng khi ngưỡng đầu ra của nó nằm trong khoảng (-1, +∞). Hàm ELU khắc phục hạn chế ánh xạ các giá trị âm của hàm ReLU.



Hình 2.10 Đồ thị hàm ELU

2.2. Mạng nơ-ron nhân tạo

2.2.1. Giới thiệu mạng nơ rọn nhân tạo

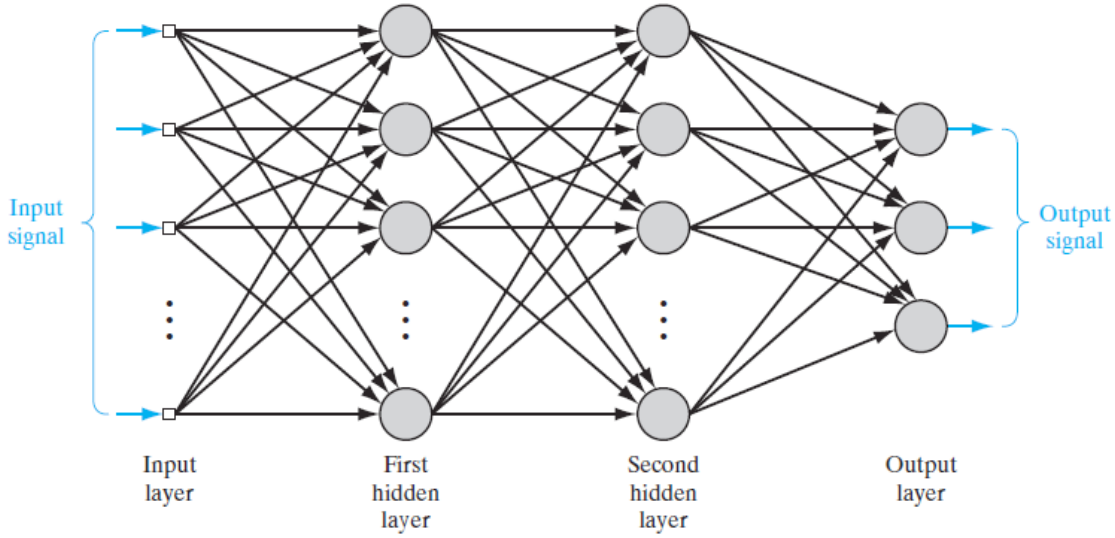
Mạng nơ-ron nhân tạo (Artificial Neural Network) là một chuỗi các giải thuật lập trình, mô phỏng dựa trên cách hoạt động của mạng lưới thần kinh trong não bộ các sinh vật sống. Mạng nơ-ron nhân tạo được sử dụng để tìm ra mối quan hệ của một tập dữ liệu thông qua một thiết kế kiến trúc chứa nhiều tầng ẩn (hidden layer), mỗi tầng lại chứa nhiều nơ-ron. Các nơ-ron được kết nối với nhau và độ mạnh yếu của các liên kết được biểu hiện qua trọng số liên kết.

Lập trình thông thường có thể làm được rất nhiều phần mềm lớn, như tính toán mô phỏng các vụ nổ hạt nhân trong siêu máy tính ở các phòng thí nghiệm, hoặc tái hiện các tế bào ở cấp độ phân tử để phân tích các thử nghiệm thuốc. Một siêu máy tính có thể tính toán được nhiều tỉ phép tính trên giây, tuy nhiên lập trình thông thường lại gặp khó khăn trong việc nhận ra các mẫu đơn giản, ví dụ như nhận diện mặt người, điều mà một bộ não sinh học xử lý nhanh và chính xác hơn nhiều.

Áp dụng với các kỹ thuật học sâu, mạng nơ-ron nhân tạo hiện nay đang được áp dụng để giải quyết những vấn đề mà lập trình theo logic thông thường khó có thể giải quyết được. Do đó, mạng nơ-ron nhân tạo đang nhanh chóng trở nên phổ biến, và là xu thế trên nhiều lĩnh vực.

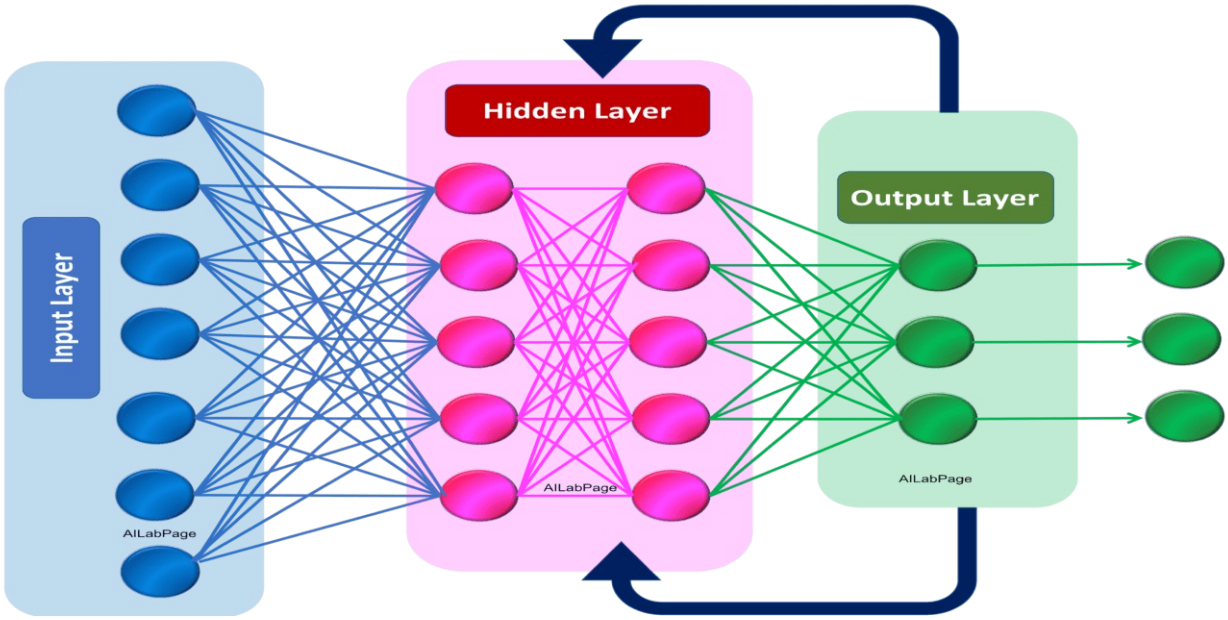
2.2.2. Một số kiểu mạng nơ-ron

Có hai kiểu mạng nơ-ron chính: mạng nơ-ron truyền thẳng (feedforward neural network) và mạng nơ-ron hồi quy (recurrent neural network).



Hình 2.11 Mạng nơ-ron truyền thẳng

Dễ thấy, ở mạng nơ-ron truyền thẳng, các nơ-ron trong tầng ẩn n + 1 đều được kết nối với các nơ-ron trong tầng n. Do có nhiều tầng ẩn nên chúng ta có thể thấy rằng mạng truyền thẳng kéo dài trong không gian, và là không có bất kỳ đường tuần hoàn (cyclic path) nào nằm trong mạng. Mạng nơ-ron truyền thẳng rất phổ biến hiện nay.



Hình 2.12 Mạng nơ-ron hồi quy (RNN)

Công dụng chính của RNN là khi sử dụng google hoặc Facebook, các giao diện này có thể dự đoán từ tiếp theo mà bạn sắp nhập. RNN có các vòng lặp để cho phép thông tin tồn tại. Điều này làm giảm độ phức tạp của các tham số, không giống như các mạng nơ-ron khác. Các mạng thần kinh này được coi là khá tốt để lập mô hình dữ liệu trình tự.

Mạng nơron hồi quy là một biến thể kiến trúc tuyến tính của mạng đệ quy. Chúng có một "bộ nhớ" do đó nó khác với các mạng nơ-ron khác. Bộ nhớ này ghi nhớ tất cả các thông tin về những gì đã được tính toán ở trạng thái trước đó. Nó sử dụng các tham số giống nhau cho mỗi đầu vào vì nó thực hiện cùng một nhiệm vụ trên tất cả các đầu vào hoặc các lớp ẩn để tạo ra đầu ra.

2.2.3. Mạng nơ-ron lan truyền ngược

2.2.3.1. Tổng quan về mạng nơ-ron lan truyền ngược

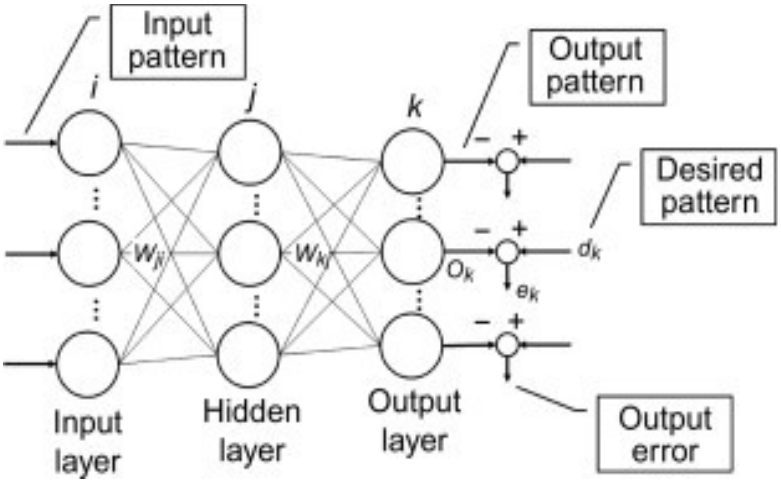
Giải thuật lan truyền ngược được mô tả ngắn gọn như sau:

Bước 1: Lan truyền. Giai đoạn lan truyền có hai bước, lan truyền tiến và lan truyền ngược. Bước vào lan truyền tiến là đưa dữ liệu huấn luyện vào các mạng nơ-ron và tính toán đầu ra. Sau đó, dựa vào kết quả đầu ra so sánh với dữ liệu huấn luyện. Chúng ta sử dụng lan truyền ngược để cập nhật ngược lại các trọng số cho các nơ-ron trong các tầng trước đó.

Bước 2: Cập nhật trọng số. Mạng cập nhật các giá trị của trọng số của nơ-ron theo hàm lỗi của kết quả đầu ra.

Bước 3: Lặp lại hai bước trên đến khi sai số tối thiểu.

2.2.3.2. Cách thức lan truyền ngược



Hình 2.13 Sơ đồ mạng hồi quy

Bước 1: Khởi tạo các giá trị ban đầu của w_{ji}, w_{kj}, θ_j, θ_k, và η (> 0).

Bước 2: Các giá trị đầu ra mong muốn d_k, k = 1, 2,…, K tương ứng với dữ liệu đầu vào x_i, i = 1, 2,...,I.

Bước 3: Tính toán đầu ra của các nơ-ron trong lớp ẩn và lớp đầu ra bằng

Bước 4: Tính toán sai số e_k và sai số tổng quát δ_k, δ_j bằng

Bước 5: Nếu e_k đủ nhỏ cho tất cả k, KẾT THÚC và các trường hợp khác

Bước 6: Quay lại bước 3.

2.3. Mạng nơ-ron tích chập Convolutional Neural Networks (CNN)

2.3.1. Giới thiệu

CNN là kiến trúc lý tưởng khi giải quyết vấn đề dữ liệu hình ảnh, một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh.

2.3.2. Cấu trúc mạng CNN

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

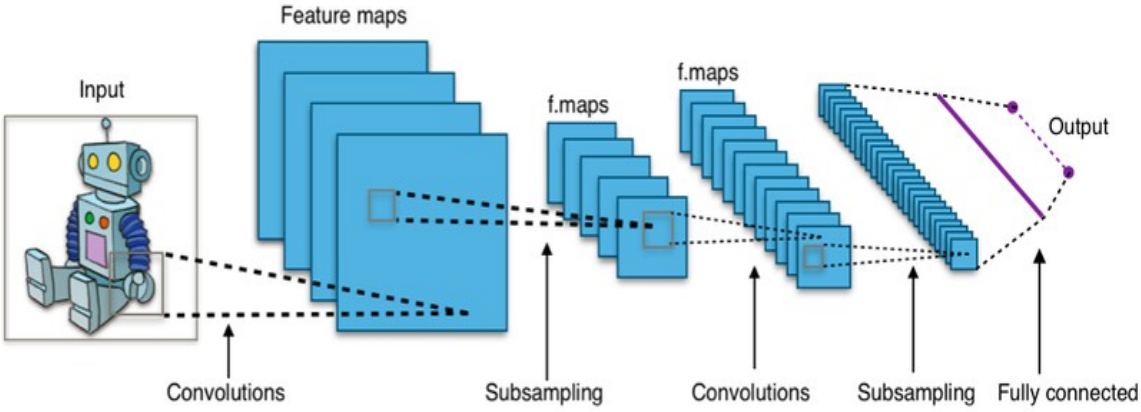
Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNN sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp

ảnh.



Trong mô hình CNN có 2 khía cạnh cần quan tâm là tính bất biến (Location Invariance) và tính kết hợp (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

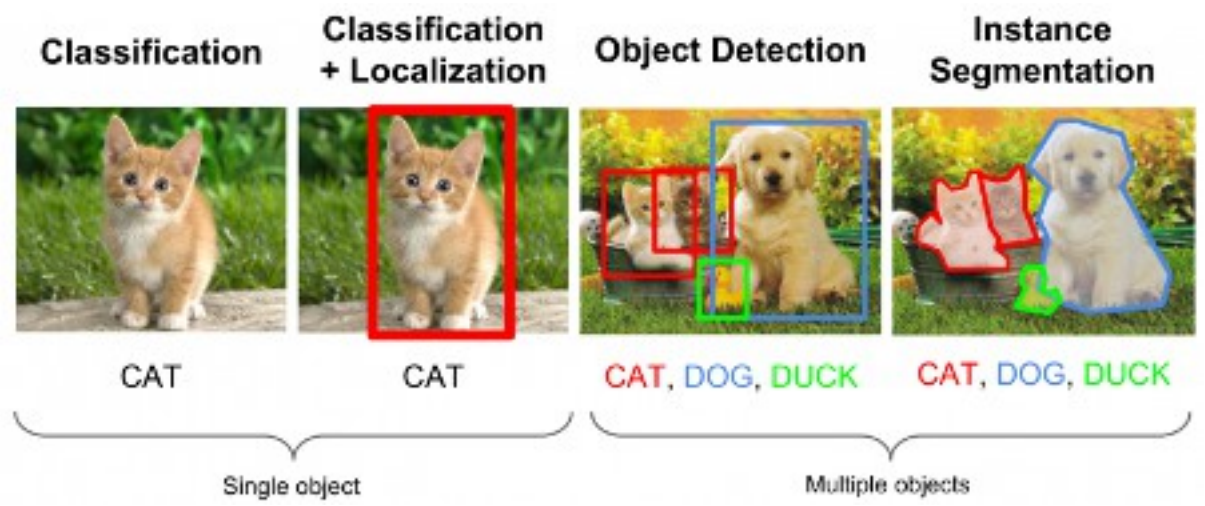
Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

Mạng CNN sử dụng 3 ý tưởng cơ bản:

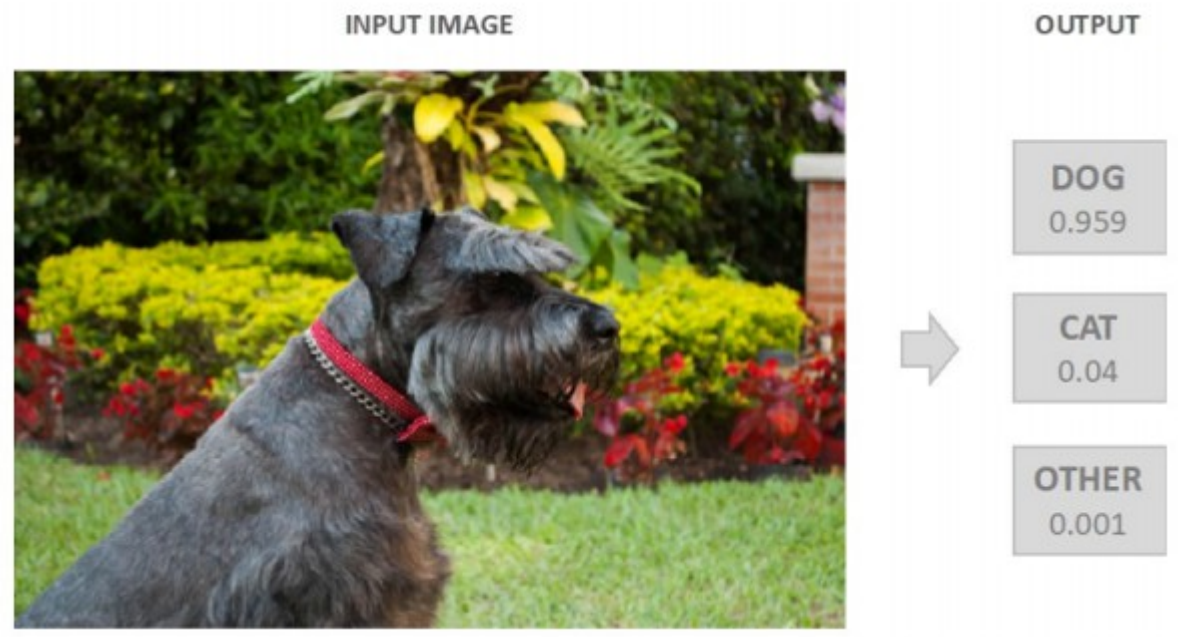
- Các trường tiếp nhận cục bộ (local receptive field)
- Trọng số chia sẻ (shared weights)
- Tổng hợp (pooling)

2.3.3. Ứng dụng của CNN

Mặc dù CNN chủ yếu được sử dụng cho các vấn đề về computer vision, nhưng điều quan trọng là đề cập đến khả năng giải quyết các vấn đề học tập khác của họ, chủ yếu liên quan đến tích chuỗi dữ liệu. Ví dụ: CNN đã được biết là hoạt động tốt trên chuỗi văn bản, âm thanh và video, đôi khi kết hợp với các mạng khác quả cầu kiến trúc hoặc bằng cách chuyển đổi các chuỗi thành hình ảnh có thể được xử lý của CNN. Một số vấn đề dữ liệu cụ thể có thể được giải quyết bằng cách sử dụng CNN với chuỗi dữ liệu là các bản dịch văn bản bằng máy, xử lý ngôn ngữ tự nhiên và gắn thẻ khung video, trong số nhiều người khác.

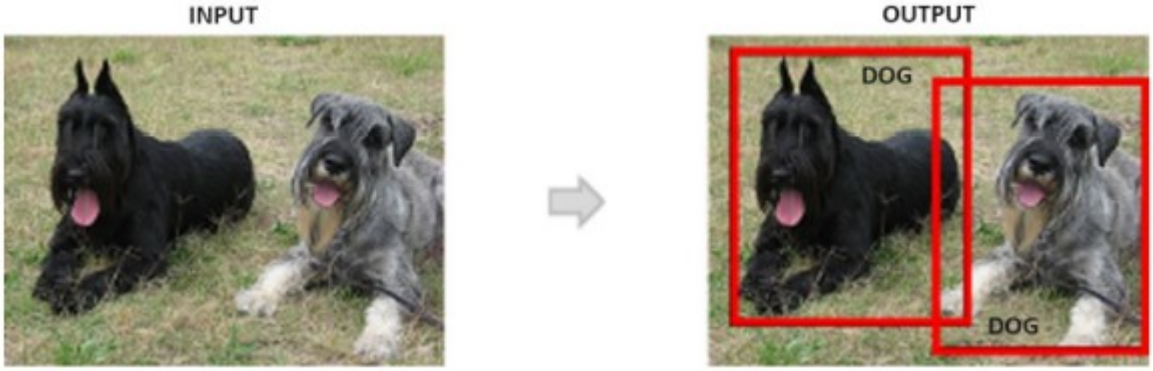


Classification: Đây là nhiệm vụ được biết đến nhiều nhất trong computer vision. Ý tưởng chính là phân loại nội dung chung của hình ảnh thành một tập hợp các danh mục, được gọi là nhãn. Ví dụ: phân loại có thể xác định xem một hình ảnh có phải là của một con chó, một con mèo hay bất kỳ động vật khác. Việc phân loại này được thực hiện bằng cách xuất ra xác suất của hình ảnh thuộc từng lớp, như được thấy trong hình ảnh sau:

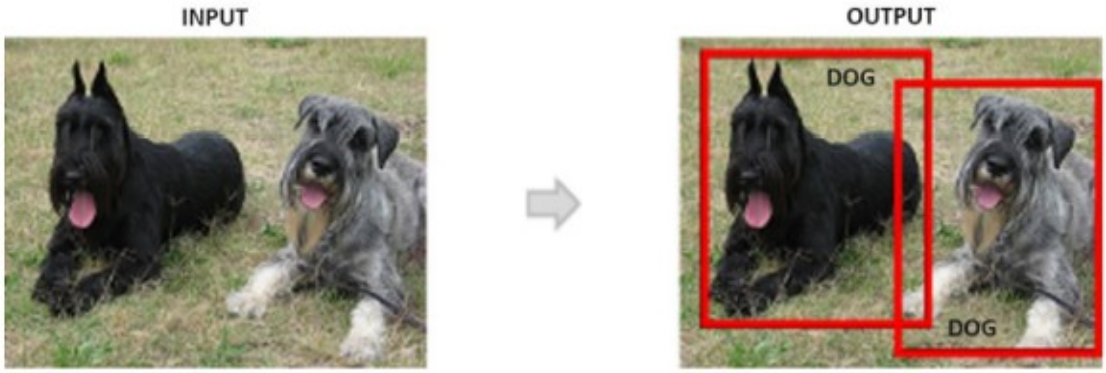


Localization: Mục đích chính của localization là tạo ra một hộp giới hạn mô tả vị trí của đối tượng trong hình ảnh. Đầu ra bao gồm một nhãn lớp và một hộp giới hạn. Tác vụ này có thể được sử dụng trong cảm biến để xác định xem một đối tượng ở bên trái hay bên phải của màn hình:

Detection: Nhiệm vụ này bao gồm thực hiện localization trên tất cả các đối tượng trong ảnh. Các đầu ra bao gồm nhiều hộp giới hạn, cũng như nhiều nhãn lớp (một cho mỗi hộp). Nhiệm vụ này được sử dụng trong việc chế tạo ô tô tự lái, với mục tiêu là có thể xác định vị trí các biển báo giao thông, đường, ô tô khác, người đi bộ và bất kỳ đối tượng nào khác có thể phù hợp để đảm bảo trải nghiệm lái xe an toàn:

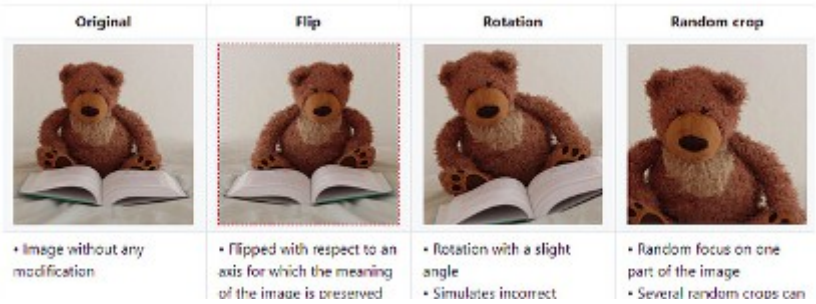


Segmentation: Nhiệm vụ ở đây là xuất ra cả nhãn lớp và đường viền của mỗi đối tượng hiện diện trong hình ảnh. Điều này chủ yếu được sử dụng để đánh dấu các đối tượng quan trọng của hình ảnh cho phân tích sâu hơn. Ví dụ: tác vụ này có thể được sử dụng để phân định rõ ràng khu vực tương ứng với khối u trong hình ảnh phổi của bệnh nhân. Hình sau mô tả cách vật thể quan tâm được phác thảo và gán nhãn:



2.3.4. Tăng cường dữ liệu (Data Augmentation)

Hiện nay trong deep learning thì vấn đề dữ liệu có vai trò rất quan trọng. Chính vì vậy có những lĩnh vực có ít dữ liệu để cho việc train model thì rất khó để tạo ra được kết quả tốt trong việc dự đoán. Do đó người ta cần đến một kỹ thuật gọi là tăng cường dữ liệu (data augmentation) để phục vụ cho việc nếu có ít dữ liệu. Phương thức data aumentation cơ bản:



Flip (Lật): lật theo chiều dọc, ngang miễn sao ý nghĩa của ảnh (label) được giữ nguyên hoặc suy ra được. Ví dụ nhận dạng quả bóng tròn, thì lật kiểu gì cũng ra quả bóng. Còn với nhận dạng chữ viết tay, lật số 8 vẫn là 8, nhưng 6 sẽ thành 9 (theo chiều ngang) và không ra số gì theo chiều dọc.

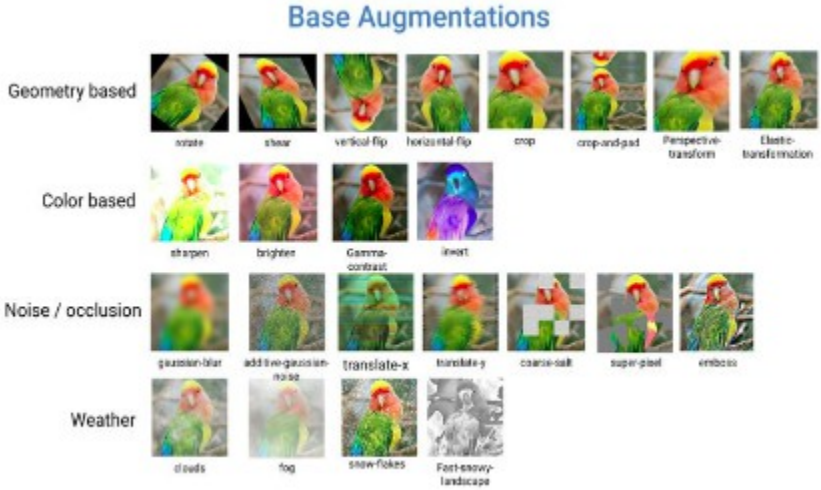
Random crop (Cắt ngẫu nhiên): cắt ngẫu nhiên một phần của bức ảnh. Lưu ý là khi cắt phải giữ thành phần chính của bức ảnh mà ta quan tâm. Như ở nhận diện vật thể, nếu ảnh được cắt không có vật thể, vậy giá trị nhãn là không chính xác.

Color shift (Chuyển đổi màu): Chuyển đổi màu của bức ảnh bằng cách thêm giá trị vào 3 kênh màu RGB. Việc này liên quan tới ảnh chụp đôi khi bị nhiễu --> màu bị ảnh hưởng.

Noise addition (Thêm nhiễu): Thêm nhiễu vào bức ảnh. Nhiễu thì có nhiều loại như nhiễu ngẫu nhiên, nhiễu có mẫu, nhiễu cộng, nhiễu nhân, nhiễu do nén ảnh, nhiễu mờ do chụp không lấy nét, nhiễu mờ do chuyển động… có thể kể hết cả ngày.

Information loss (Mất thông tin): Một phần của bức hình bị mất. Có thể minh họa trường hợp bị che khuất.

Constrast change (Thay đổi độ tương phản): thay độ tương phản của bức hình, độ bão hòa.


Base Augmentations
Geometry based
rotate
shear
vertical-flip
horizontal-flip
crop
crop-and-pad
Perspective-transform
Elastic-transformation
Color based
sharpen
brighten
Gamma-contrast
invert
Noise / occlusion
gaussian-blur
additive-gaussian-noise
translate-x
translate-y
coarse-salt
super-pixel
emboss
Weather
clouds
fog
snow-flakes
Fast-snowy-landscape

CHƯƠNG 3
THU THẬP DỮ LIỆU VÀ XỬ LÝ DỮ LIỆU

3.1. Giới thiệu

Trong chương này tôi sẽ trình bày quá trình thu thập, xử lý và gán nhãn dữ liệu nhằm xây dựng bộ dữ liệu (tập hợp các hình ảnh về các loại lá cây thuốc) phục vụ việc huấn luyện và đánh giá model.

Trải qua quá trình tìm hiểu các loại bệnh, tôi đã tìm hiểu được 40 loài cây khác nhau hỗ trợ điều trị được một số bệnh nhẹ cũng như các bệnh nặng được lưu lại trong những cuốn sách về các bài thuốc dân gian như: cảm mạo phong nhiệt, cảm cúm, ngạt mũi, nhức đầu, chứng ho mới hoặc ho lâu ngày, hen suyễn, viêm họng, viêm Amydal, viêm đường tiết niệu, viêm đại tràng (trường ung), cơ thể suy nhược, kém ăn, kém ngủ, thần kinh suy nhược, hoa mắt chóng mặt, đau dạ dày, tiêu chảy, táo bón, ung thư cổ tử cung, ung thư tuyến tiền liệt, ung thư gan, ung thư máu, ung thư phổi,…

3.2. Một số bài thuốc có sử dụng lá cây trong dân gian

3.2.1. Bạc hà

Công năng, chủ trị: Sơ phong, thanh nhiệt, thấu chẩn, sơ can, giải uất, giải độc. Chữa cảm mạo phong nhiệt, cảm cúm, ngạt mũi, nhức đầu, đau mắt đỏ, thúc đẩy sởi mọc, ngực sườn đầy tức.

Liều lượng, cách dùng: Ngày dùng 12 - 20g, hãm vào nước sôi 200 ml, cách 3 giờ uống một lần.

Hình 3.1 lá bạc hà

3.2.2. Bạch hoa xà thiệt thảo

Công năng, chủ trị: Thanh nhiệt giải độc, lợi niệu thông lâm, tiêu ung tán kết. Chữa phế nhiệt, hen suyễn, viêm họng, viêm Amydal, viêm đường tiết niệu, viêm đại tràng (trường ung). Dùng ngoài chữa vết thương, rắn cắn, côn trùng đốt.

Liều lượng, cách dùng: Ngày dùng 15 - 60g (khô) sắc uống. Dùng ngoài, giã nát đắp tại chỗ.

Hình 3.2 Lá bạch hoa xà thiệt thảo

3.2.3. Cốt khí

Công năng, chủ trị: Khu phong trừ thấp, hoạt huyết, thông kinh, chỉ khái (giảm ho), hóa đờm, chỉ thống. Chữa đau nhức gân xương, ngã sưng đau ứ huyết, bế kinh, hoàng đản, ho nhiều đờm, mụn nhọt lở loét.

Liều lượng, cách dùng: Ngày dùng 9 - 15g, sắc uống, dùng ngoài sắc lấy nước để bôi, rửa, hoặc chế thành cao, bôi.

Hình 3.3 Lá cốt khí

3.2.4 Trinh nữ hoàng cung

Công năng, chủ trị: Tiêu ung, bài nùng. Hỗ trợ chữa ung thư vú, ung thư cổ tử cung, ung thư tuyến tiền liệt.

Liều lượng, cách dùng: Ngày dùng 3 - 5g, sao vàng, sắc uống.

Hình 3.4 Lá trinh nữ hoàng cung

3.2.5. Tía tô

Công năng, chủ trị: Hành khí, khoan trung, chỉ thống, an thai. Chữa khí uất vùng ngực, ngực sườn đày tức, thượng vị đau, ợ hơi, nôn mửa. Lá và cành tía tô chữa động thai. Hạt tía tô (tô tử) giảm ho trừ đàm.

Liều lượng, cách dùng: Ngày dùng 5 - 9g, sắc uống.

Hình 3.5 Lá tía tô

3.2.6. Lá mơ

Công năng, chủ trị: Thanh nhiệt, giải độc. Chữa lỵ trực khuẩn.

Liều lượng, cách dùng: Lá tươi 30 - 50g, lau sạch, thái nhỏ trộn với trứng gà, bọc

vào lá chuối đem nướng hoặc áp chảo cho chín. Ngày ăn 2 - 3 lần, trong 5 - 8 ngày.

Hình 3.6 Lá mơ

3.2.7. Dừa cạn

Công năng, chủ trị: Ung thư máu

Liều lượng, cách dùng: Dừa cạn 15g, cây xạ đen 30g, lá bồ công anh 30g. Các vị thuốc đen rửa sạch, sắc với 1,5 lít nước, sắc cạn còn 700ml chia 3 lần uống sau bữa ăn 30 phút.

Hình 3.7 lá cây dừa cạn

3.2.8. Cây cúc tần

Công năng, chủ trị: Phát tán phong nhiệt, tiêu độc, lợi tiểu, tiêu đàm. Chữa cảm mạo phong nhiệt, sốt không ra mồ hôi, phong thấp, tê bại, đau nhức xương khớp.

Liều lượng, cách dùng: Ngày dùng 8 - 16g, sắc uống.

Hình 3.8 Lá cúc tần

3.2.9. Dâu tằm

Công năng, chủ trị: Vỏ rễ dâu có tác dụng thanh phế nhiệt bình suyễn, tiêu thũng, giảm ho, trừ đờm, hạ suyễn. Chữa phế nhiệt, ho suyễn, hen, ho ra máu, trẻ con ho gà, phù thũng, bụng trướng to, tiểu tiện không thông; Lá dâu có tác dụng tán phong thanh nhiệt, thanh can, sáng mắt. Chữa cảm mạo phong nhiệt, phế nhiệt, ho, viêm họng, nhức đầu, mắt đỏ, chảy nước mắt, đậu lào, phát ban, cao huyết áp, mất ngủ; Cành dâu có tác dụng trừ phong thấp, lợi các khớp, thông kinh hoạt lạc, tiêu viêm. Chữa phong thấp đau nhức các đầu xương, cước khí, sưng lở, chân tay co quắp.

Liều lượng, cách dùng: Vỏ rễ: ngày dùng 6 - 12g (có thể dùng tới 20 - 40g), sắc uống. Lá: ngày dùng 5 - 12g, sắc uống. Cành: ngày dùng: 9 - 15g (có thể dùng tới 40 - 60g), sắc uống.

Hình 3.9 Lá dâu tằm

CHƯƠNG 4
XÂY DỰNG BÀI TOÁN

4.1. Các bước chi tiết trong thuật toán

Bước 1: Import Modules cần thiết

```
import numpy as np
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import tensorflow as tf
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling
2D, Flatten
from tensorflow.keras.optimizers import Adam
from keras.callbacks import EarlyStopping
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Bước 2: Load data

```
from google.colab import drive
drive.mount('/content/drive')
data_train='/content/drive/MyDrive/Colab Notebooks/DATASETLACAYTHUOC/Tr
ain'
data_validation='/content/drive/MyDrive/Colab Notebooks/DATASETLACAYTHU
OC/Test'
train=ImageDataGenerator(rescale=1/255)
validation=ImageDataGenerator(rescale=1/255)
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
training_data=train_datagen.flow_from_directory('/content/drive/MyDrive
/Colab Notebooks/DATASETLACAYTHUOC/Train',
                                                target_size=(150,150),
                                                batch_size=32,
                                                class_mode ='categorical
')
validation_data=train_datagen.flow_from_directory('/content/drive/MyDri
ve/Colab Notebooks/DATASETLACAYTHUOC/Validation',
                                                target_size=(150,150),
                                                batch_size=32,
                                                class_mode ='categorical
')
kieuhinh=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range
=0.2, horizontal_flip=True)
training_data=kieuhinh.flow_from_directory('/content/drive/MyDrive/Cola
b Notebooks/DATASETLACAYTHUOC/Train',target_size=(150,150), batch_size=
32, class_mode='categorical')
```

```
validation_data=kieuhinh.flow_from_directory('/content/drive/MyDrive/Co
lab Notebooks/DATASETLACAYTHUOC/Validation',target_size=(150,150), batc
h_size=32, class_mode='categorical')
```

Bước 3: Create Model

```
model=Sequential()
model.add(Conv2D(128,
(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',
input_shape=(150,150,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,
(3,3),activation='relu',kernel_initializer='he_uniform',padding='same')
)
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(128,activation='relu',kernel_initializer = 'he_uniform'
))
#model.add(Dropout(0.2))
model.add(Dense(40,activation='softmax'))
model.summary()
```

Bước 4: Training

```
training_data.class_indices
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy',met
rics = ['accuracy'])
callbacks=[EarlyStopping(monitor='val_loss',patience=20)]
history=model.fit(training_data,
                  steps_per_epoch=len(training_data),
                  batch_size = 64,
                  epochs=200,
                  validation_data=validation_data,
                  validation_steps=len(validation_data),
                  callbacks=callbacks,
                  verbose = 1)
```

Bước 5: Độ chính xác

```
score = model.evaluate(validation_data,verbose=0)
print('Sai số kiểm tra là: ',score[0])
print('Độ chính xác kiểm tra là: ',score[1])
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['Train','Validation'])
plt.show
```

Bước 6: Predict

```
model.save("/content/drive/MyDrive/Colab Notebooks/Model_CNN_NHAN_DIEN_
LA_CAY_THUOC/Model_CNN_Nhan_Dien_La_Cay_Thuoc.h5")
from tensorflow.keras.models import load_model
model=load_model('/content/drive/MyDrive/Colab Notebooks/Model_CNN_NHAN
_DIEN_LA_CAY_THUOC/Model_CNN_Nhan_Dien_La_Cay_Thuoc.h5')
```

```
img=load_img("/content/drive/MyDrive/Colab Notebooks/DATASETLACAYTHUOC/
Test/LaBacHa/Image_ 1.jpg",target_size=(150,150))
plt.imshow(img)
img=img_to_array(img)
img=img.astype('float32')
img=img/255
img=np.expand_dims(img,axis=0)
result=model.predict(img)
if round(result[0][0])==1:
  prediction='Lá Bạc Hà'
if round(result[0][1])==1:
  prediction='Lá Bạch Đầu Ông'
if round(result[0][2])==1:
  prediction='Lá Bạch Truất'
if round(result[0][3])==1:
  prediction='Lá Bố Chính Sâm'
if round(result[0][4])==1:
  prediction='Lá Cây Bồ Công Anh'
if round(result[0][5])==1:
  prediction='Lá Cây Bồ Kết'
if round(result[0][6])==1:
  prediction='Lá Cây Bồ Quân'
if round(result[0][7])==1:
  prediction='Lá Cây Cúc Áo'
if round(result[0][8])==1:
  prediction='Lá Cây Diếp Cá'
if round(result[0][9])==1:
  prediction='Lá Cây Đu Đủ'
if round(result[0][10])==1:
  prediction='Lá Cây Dừa Cạn'
if round(result[0][11])==1:
  prediction='Lá Cây Hàm Ếch'
if round(result[0][12])==1:
  prediction='Lá Cây Bạch Hoa Xà Thiệt Thảo'
if round(result[0][13])==1:
  prediction='Lá Cây Hoàng Cầm Râu'
if round(result[0][14])==1:
  prediction='Lá Cây Kha Tử'
if round(result[0][15])==1:
  prediction='Lá Cây Kim Ngân Hoa'
if round(result[0][16])==1:
  prediction='Lá Cây Lược Vàng'
if round(result[0][17])==1:
  prediction='Lá Cây Mãng Cầu'
if round(result[0][18])==1:
  prediction='Lá Cây Nghệ Đen'
if round(result[0][19])==1:
  prediction='Lá Cây Nghệ Vàng '
if round(result[0][20])==1:
  prediction='Lá Cây Ô Liu'
if round(result[0][21])==1:
```

```
  prediction='Lá Cây Rau Má'
if round(result[0][22])==1:
  prediction='Lá Cây Rau Sam'
if round(result[0][23])==1:
  prediction='Lá Cây Sê'
if round(result[0][24])==1:
  prediction='Lá Cây Sói Rừng'
if round(result[0][25])==1:
  prediction='Lá Cây Mộc Thông'
if round(result[0][26])==1:
  prediction='Lá Cây Trinh Nữ'
if round(result[0][27])==1:
  prediction='Lá Cây Trinh Nữ Hoàng Cung'
if round(result[0][28])==1:
  prediction='Lá Cây Xạ Đen'
if round(result[0][29])==1:
  prediction='Lá Cây Ý Dĩ'
if round(result[0][30])==1:
  prediction='Lá Cốt Khí'
if round(result[0][31])==1:
  prediction='Lá Cúc Tần'
if round(result[0][32])==1:
  prediction='Lá Dâu Tằm'
if round(result[0][33])==1:
  prediction='Lá Địa Liền'
if round(result[0][34])==1:
  prediction='Lá Gai'
if round(result[0][35])==1:
  prediction='Lá Húng Chanh'
if round(result[0][36])==1:
  prediction='Lá Khổ Sâm'
if round(result[0][37])==1:
  prediction='Lá Mơ'
if round(result[0][38])==1:
  prediction='Lá Ổi'
if round(result[0][39])==1:
  prediction='Lá Tía Tô'
print(prediction)
```

4.2. Tập dữ liệu và kết quả đạt được của thuật toán

Tổng dữ liệu: 3191

Tập train:1595

Tập validation: 796

Tập test: 800

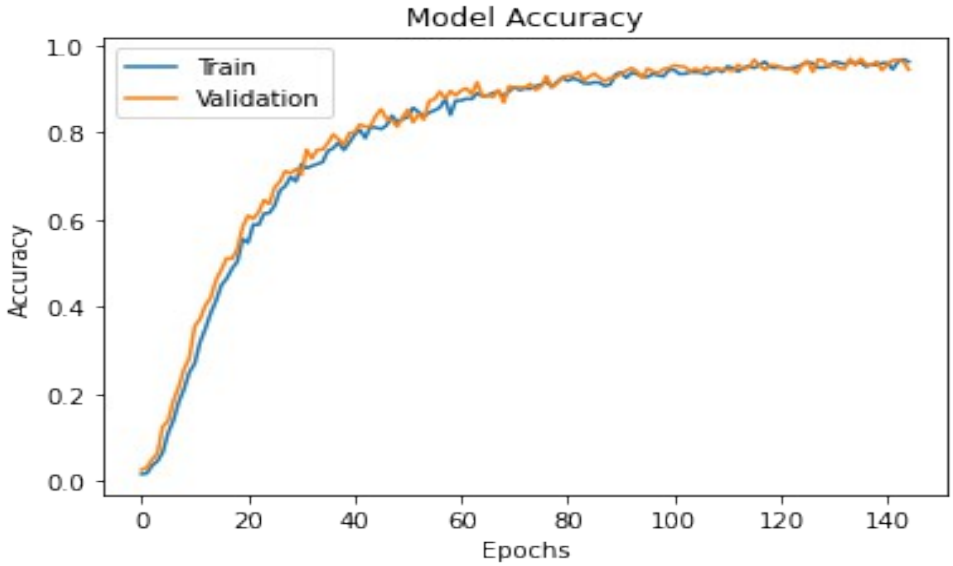
Số lớp: 40

Gán nhãn cho dữ liệu:

```
{'LaBacHa': 0,
 'LaBachDauOng': 1,
```

```
'LaBachTruat': 2,
'LaBoChinhSam': 3,
'LaCayBoCongAnh': 4,
'LaCayBoKet': 5,
'LaCayBoQuan': 6,
'LaCayCucAo': 7,
'LaCayDiepCa': 8,
'LaCayDuDu': 9,
'LaCayDuaCan': 10,
'LaCayHamEch': 11,
'LaCayHoaXaThietThao': 12,
'LaCayHoangCamRau': 13,
'LaCayKhaTu': 14,
'LaCayKimNganHoa': 15,
'LaCayLuocVang': 16,
'LaCayMangCau': 17,
'LaCayNgheDen': 18,
'LaCayNgheVang': 19,
'LaCayOliu': 20,
'LaCayRauMa': 21,
'LaCayRauSam': 22,
'LaCaySe': 23,
'LaCaySoiRung': 24,
'LaCayThongMoc': 25,
'LaCayTrinhNu': 26,
'LaCayTrinhNuHoangCung': 27,
'LaCayXaDen': 28,
'LaCayYdi': 29,
'LaCotKhi': 30,
'LaCucTan': 31,
'LaDauTam': 32,
'LaDiaLien': 33,
'LaGai': 34,
'LaHungChanh': 35,
'LaKhoSam': 36,
'LaMo': 37,
'LaOi': 38,
'LaTiaTo': 39}
```

Độ chính xác kiểm tra là: 0.9510050415992737



CHƯƠNG 5
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Đề tài nhận diện lá cây thuốc bằng thuật toán CNN đã được thực hiện với model có độ chính xác tương đối cao qua quá trình training. Qua việc thực hiện đề tài giúp tôi hiểu rõ hơn về các thuật toán CNN cũng như các thuật toán khác trong Deep Learning. Đề tài giúp tôi hiểu rõ hơn về những khó khăn gặp phải khi tìm kiếm dữ liệu, phát triển mô hình bài toán.

5.2. Hạn chế

Điểm quan trọng nhất của bài toán nhận dạng các lá cây thuốc đó là dữ liệu hình ảnh. Để tăng được độ chính xác của bài toán thì dữ liệu hình ảnh huấn luyện cần phải nhiều và đa dạng hơn nữa. Khi đó model sẽ tổng quát hơn, nhận dạng và đánh giá dữ liệu chuẩn xác hơn.

Dữ liệu hình ảnh trong bài toán mà tôi đang thực hiện đa phần được tìm kiếm trên google nên định dạng, độ nét cũng như góc chụp không được hoàn chỉnh và đa dạng. Do đó thiếu đi sự ổn định, nhiều góc chụp khác nhau của loài cây đó.

Trong quá trình thực hiện nhận diện real – time xảy ra một số lỗi, vì thời gian quá gấp nên tôi chưa kịp sửa lỗi nên phần này chưa được thực hiện.

5.3. Hướng phát triển

Để phát triển thuật toán tốt hơn, tôi cần thêm thời gian và sự tiếp cận trực tiếp với các loài cây để từ đó có sự ổn định và đa dạng trong dữ liệu.

Đề tài sẽ được tôi nghiên cứu và phát triển thêm về nhận diện bằng real – time thông qua app hoặc web để thử độ ổn định và có thể ứng dụng vào thực tiễn.

TÀI LIỆU THAM KHẢO

[1]. 70 cây thuốc nam theo quy định của bộ y tế (11/2014), https://healthvietnam.vn/thu-vien/tai-lieu-tieng-viet/y-hoc-co-truyen/70-cay-thuoc-nam-theo-quy-dinh-cua-bo-y-te-112014

[2]. Convolutional Neural Network là gì? Cách chọn tham số cho Convolutional Neural Network chuẩn chỉnh https://wiki.tino.org/convolutional-neural-network-la-gi/

[3]. Nhận diện lá cây thuốc bằng thiết bị di động https://text.123docz.net/document/4256390-nhan-dang-la-cay-thuoc-bang-thiet-bi-di-dong.htm