

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



HCMUTE

Báo Cáo Cuối Kỳ : NHẬN DIỆN BIẾN BÁO GIAO THÔNG VIỆT NAM

Nhóm: 05CLC

Giảng viên tổ chức : PGS TS. Nguyễn Trường Thịnh

Môn học : Trí Tuệ Nhân Tạo

Sinh viên thực hiện: Trịnh Tuấn Vũ

MSSV : 19146014

TP.HCM, 6/2022

Mục lục

MỤC LỤC HÌNH ẢNH	3
LỜI NÓI ĐẦU	4
CHƯƠNG 1 : TỔNG QUAN VỀ TÀI NHẬN DIỆN BIỂN BÁO GIAO THÔNG VIỆT NAM	6
1.1 Đặt vấn đề	6
1.2 Tình hình nghiên cứu trong và ngoài nước	6
1.3 Giới hạn đề tài	7
1.4 Bố cục đề tài	7
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	8
2.1 Các khái niệm cơ bản	8
2.1.1 Trí tuệ nhân tạo	8
2.1.2 Học máy	9
2.1.3 Deep learning	10
2.1 Neural Network	11
2.2 Convolutional Neural Network	13
2.3.1 Định nghĩa mạng nơron tích chập	13
2.3.2 Mô hình mạng nơron tích chập	14
2.3.3 tính chất của mạng nơron tích chập	17
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	20
3.1 Sơ đồ phân rã chức năng	20
3.2 Sơ đồ sử dụng	20
3.3 Nhận dạng biển báo giao thông	21
3.3.1 Khám phá tập dữ liệu	21
3.3.2 Xây dựng mô hình	22
3.3.3 Tranning	23
CHƯƠNG 4 KẾT QUẢ THỰC NGHIỆM	25
4.1 Đánh giá mô hình	25
4.2 Kết quả nhận diện hình ảnh sử dụng real-time	25
CHƯƠNG 5 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN	27
5.1 Kết quả đạt được	27
5.2 Hướng phát triển	27
Tài Liệu Tham Khảo	28

MỤC LỤC HÌNH ẢNH

Hình 1 giao thông ở Việt Nam	6
Hình 2 Cấu trúc trí tuệ nhân tạo	8
Hình 3. Quan hệ giữa dữ liệu và khả năng dự đoán của thuật toán. Nguồn: Coursera	10
Hình 4 mối quan hệ giữa ngõ vào và ngõ ra của một node.	11
Hình 5 hình đồ thị hàm sigmoid và hàm tanh	12
Hình 6 đồ thị hàm ReLu	13
Hình 7 Minh họa tích chập	14
Hình 8 hình mô hình mạng nowrron tích chập.....	14
Hình 9 Tích chập trên bản đồ kích hoạt	17
Hình 10 Hình kết nối cục bộ.....	18
Hình 11 Quá trình tổng hợp.....	19
Hình 12 Một trình tự CNN để phân loại các chữ số viết tay	19
Hình 13 Sơ đồ chức năng	20
Hình 14 Sơ đồ sử dụng	21
Hình 15 Hình ảnh về hàm mất mát và độ chính xác của model sau khi train	25
Hình 16 Hình ảnh nhận dạng 4 loại biển báo giao thông	26

LỜI NÓI ĐẦU

Chắc hẳn bạn đã từng nghe về xe ô tô tự lái mà hành khách hoàn toàn có thể tin tưởng vào chiếc xe để di chuyển. Nhưng để đạt được mức độ chính xác và an toàn thì các phương tiện cần phải hiểu và tuân theo các quy tắc giao thông. Trong thế giới của trí tuệ nhân tạo và sự tiến bộ trong công nghệ, nhiều nhà nghiên cứu và các công ty lớn như Tesla, Uber, Google, Mercedes-Benz, Toyota, Ford, Audi, v.v. đang nghiên cứu về xe tự hành và xe tự lái. Để đạt được độ chính xác trong công nghệ này, các phương tiện cần phải giải thích được các biển báo giao thông và đưa ra quyết định sao cho phù hợp. Chính vì thế em đã có ý tưởng tạo nên một chương trình nhận diện biển báo giao thông.

Mục tiêu mà em muốn hướng đến để đưa lên sử dụng trên xe oto để và phát triển đề tài của mình trong tương lai, nhưng hiện tại bây giờ trong trường học là sinh viên năm ba em chỉ phát triển chương trình với chức năng nhận diện được các biển báo giao thông rồi hiển thị ra đó là biển báo gì ở trên chương trình.

Nhận xét của giảng viên

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Chữ ký của giảng viên

CHƯƠNG 1 : TỔNG QUAN ĐỀ TÀI NHẬN DIỆN BIỂN BÁO GIAO THÔNG VIỆT NAM

Chương này trình bày tình hình nghiên cứu trong và ngoài nước về đề tài nhận diện biển báo giao thông . Từ đó đưa ra lí do chọn đề tài và tóm tắt quá trình thực hiện .

1.1 Đặt vấn đề

Chắc hẳn bạn đã từng nghe về xe ô tô tự lái mà hành khách hoàn toàn có thể tin tưởng vào chiếc xe để di chuyển . Nhưng để đạt được mức độ chính xác và an toàn thì các phương tiện cần phải hiểu và tuân theo các quy tắc giao thông . Trong thế giới của trí tuệ nhân tạo và sự tiến bộ trong công nghệ , nhiều nhà nghiên cứu và các công ty lớn như Tesla , Uber , Google , Mercedes - Benz , Toyota , Ford , Audi , v.v. đang nghiên cứu về xe tự hành và xe tự lái . Để đạt được độ chính xác trong công nghệ này , các phương tiện cần phải giải thích được các biển báo giao thông và đưa ra quyết định sao cho phù hợp



Hình 1 giao thông ở Việt Nam

1.2 Tình hình nghiên cứu trong và ngoài nước

Đối với bài toán phân loại biển báo giao thông , tức là nhận diện bức ảnh này thuộc loại biển báo gì . Đối với mỗi ngõ vào , ta phân loại vào 1 class nhất định. Đây là bài toán phân lớp (Classification) . Có nhiều thuật iotoán để phân lớp dữ liệu , từ cổ điển đến hiện đại . Nhưng mạng Nơ- ron tích chập được đánh giá là thuật toán hiệu quả , tránh được tác động từ môi trường , nhiễu , sự thay đổi khoảng cách ảnh đến camera . Trong đề tài này , mục tiêu chúng tôi đặt ra là xây dựng mô hình nhận diện biển báo giao thông trong ảnh và video . Từ đó so sánh hiệu quả giữa các mô hình khác nhau . Với bài toán nhận diện biển báo , gồm hai bài toán lớn là phát hiện và phân loại vật thể . Có rất nhiều cách để phân loại vật thể bao gồm R - CNN , Fast R - CNN , Faster R - CNN và một số thuật toán khác.

1.3 Giới hạn đề tài

Không thể nhận diện cùng lúc được nhiều biển báo, khoảng cách nhận diện gần, không nhận diện được tất cả biển báo giao thông ở việt nam.

Không có web, app.

1.4 Bố cục đề tài

Nội dung đề tài được trình bày trong 5 chương sau:

Chương 1: Tổng quan về đề tài: Chương này nêu ra được tình hình nghiên cứu hiện nay, lý do, giới hạn của đề tài.

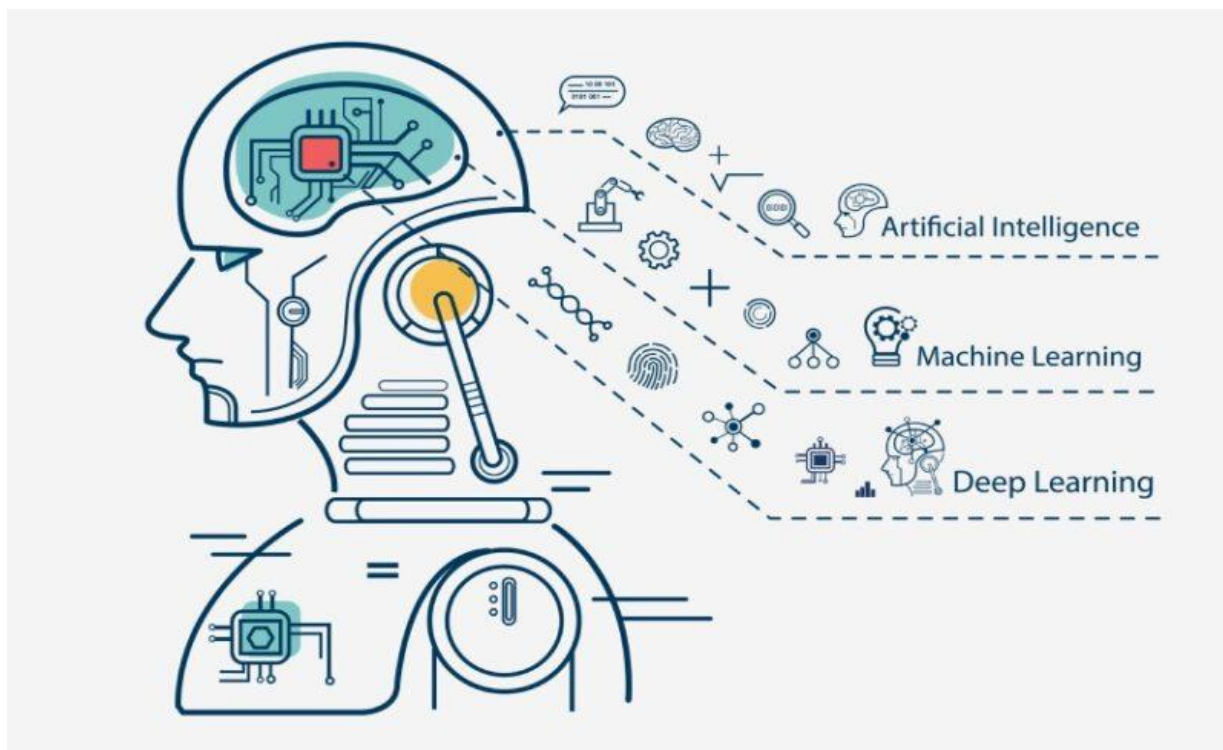
Chương 2: Cơ sở lý thuyết. Chương này đưa các lý thuyết chính liên quan đến: Ảnh số, quá trình xử lý ảnh, mạng nơ-ron nhân tạo, mạng nơ-ron tích chập (CNN), các phương pháp nhận diện biển báo giao thông.

Chương 3: Thiết kế hệ thống nhận diện biển báo giao thông Việt Nam dùng mạng nơ-ron tích chập. Chương này nêu lên được yêu cầu chi tiết của hệ thống, sơ đồ khối của hệ thống và hoạt động của từng khối, thiết kế từng khối, lưu đồ chính, lưu đồ chi tiết của chương trình mô phỏng trên colab

Chương 4: Kết quả thực nghiệm. Trình bày về kết quả mô phỏng

Chương 5: Kết luận và hướng phát triển

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2 Cấu trúc trí tuệ nhân tạo

2.1 Các khái niệm cơ bản

2.1.1 Trí tuệ nhân tạo

Trong khoa học máy tính, Trí tuệ nhân tạo hay Trí tuệ nhân tạo (tiếng Anh: Artificial Intelligence), đôi khi được gọi là trí thông minh nhân tạo, là trí thông minh được thể hiện bằng máy móc, trái ngược với trí thông minh tự nhiên được con người thể hiện. Thông thường, thuật ngữ "Trí tuệ nhân tạo" thường được sử dụng để mô tả các máy móc (hoặc máy tính) bắt chước các chức năng "nhận thức" mà con người liên kết với tâm trí con người, như "học tập" và "giải quyết vấn đề"

Trí tuệ nhân tạo có thể được phân thành ba loại hệ thống khác nhau: Trí tuệ nhân tạo phân tích, lấy cảm hứng từ con người và nhân tạo. Trí tuệ nhân tạo phân tích chỉ có các đặc điểm phù hợp với Trí tuệ nhận thức; tạo ra một đại diện nhận thức về thế giới và sử dụng học tập dựa trên kinh nghiệm trong quá khứ để thông báo các quyết định trong tương lai. Trí tuệ nhân tạo lấy cảm hứng từ con người có các yếu tố từ trí tuệ nhận thức và cảm xúc; hiểu cảm xúc của con người, ngoài các

yếu tố nhận thức và xem xét chúng trong việc ra quyết định. Trí tuệ nhân tạo nhân cách hóa cho thấy các đặc điểm của tất cả các loại năng lực (nghĩa là trí tuệ nhận thức, cảm xúc và xã hội), có khả năng tự ý thức và tự nhận thức được trong các tương tác.

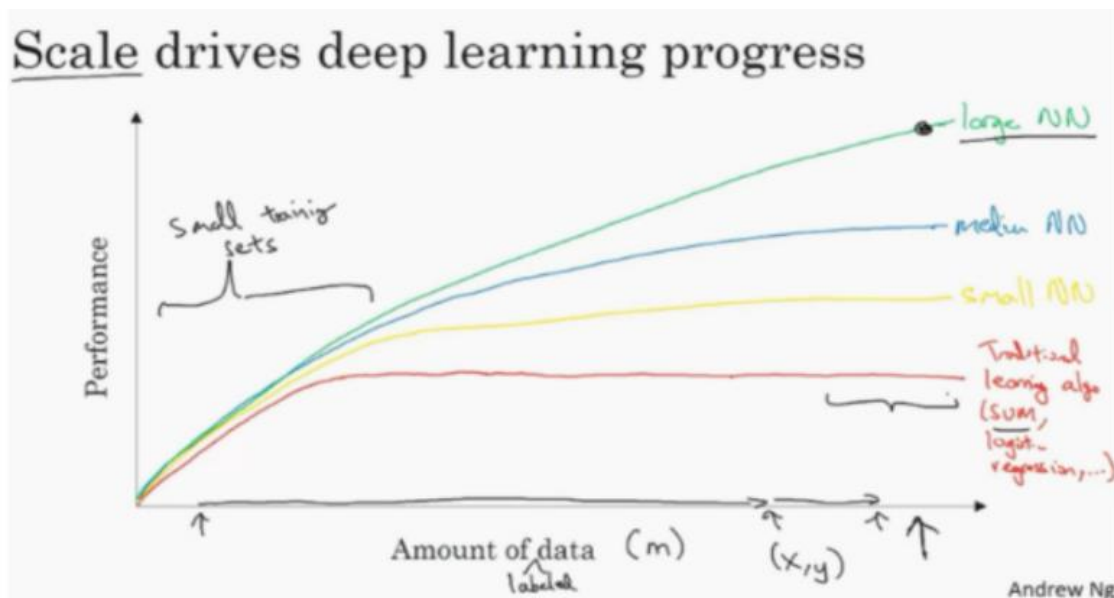
2.1.2 Học máy

Máy học (Machine Learning) hiểu đơn giản là tập hợp các phương pháp, các giải thuật khiến cho máy tính có thể thích ứng một cách độc lập và đưa ra các quyết định giống như con người do đó người ta xếp “Máy học” là một lĩnh vực của trí tuệ nhân tạo liên quan đến nghiên cứu và xây dựng các kỹ thuật cho phép các hệ thống “học”, “suy nghĩ” tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Máy học có liên quan lớn đến suy diễn thống kê (statistical inference), vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, Máy học tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán khó, một phần của Máy học là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được. Vì thế Máy học cho phép máy tính tìm thấy những thông tin giá trị ẩn sâu mà không được lập trình một cách rõ ràng nơi để tìm. Phương pháp phân tích quy nạp: thông qua việc ước lượng quan hệ giá trị giữa các tham số từ tập dữ liệu chuẩn đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có. Phương pháp phân tích suy diễn: Máy học/phân biệt các khái niệm dựa vào các luật. Phương pháp này cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ máy tính. Phân loại giải thuật trong Máy học: Học có giám sát (Supervised Learning): cung cấp cho máy tính một tập dữ liệu nguồn và đích mẫu tương ứng (Training dataset) để làm cơ sở xây dựng ra model mong muốn. Máy tính được “huấn luyện” thông qua các mô hình được quy định. Các tham số sinh ra được từ tập dữ liệu huấn luyện sẽ được sử dụng để thẩm định lại tính thích hợp của mô hình trên tập dữ liệu kiểm tra (Testing dataset). Trong các bài toán thực, dữ liệu ngõ vào mới lại được huấn luyện nhằm cải tiến model hiện tại. Điều này chính là mô tả cho khái niệm học có giám sát. Học không giám sát: chỉ cung cấp tập nguồn cho máy tính mà không có tập

đầu ra, sau đó máy tính phải tự thực hiện phân nhóm hay làm giảm độ phức tạp của dữ liệu và các mẫu mới. Điều này đặc biệt có ích trong việc lưu trữ hoặc xử lý dữ liệu trong tương lai (Đưa về bài toán học có giám sát khi đã có dữ liệu thực tế) Các bài toán học không giám sát cũng được chia làm hai loại dựa vào tập nguồn của chúng

2.1.3 Deep learning

Deep Learning (Học sâu) là một nhánh nhỏ của Machine Learning (Máy học). bắt nguồn từ thuật toán Neural Network . Tuy nhiên , gần đây Deep Learning mang lại nhiều thành tựu kỹ thuật nhờ vào sự phát triển của Big Data (Dữ liệu lớn) , với tập dữ liệu càng lớn giúp phát triển các mạng Neural Network sâu hơn với khả năng dự đoán chính xác cao hơn . Bắt nguồn từ các loại hình kinh doanh internet truyền thống như tìm kiếm web và quảng cáo . Deep Learning ngày nay đã tạo ra nhiều lợi ích hơn trong cuộc sống hằng ngày. Ví dụ như trong lĩnh vực y tế , việc phân tích các hình ảnh X - quang được ứng dụng vào Deep Learning để nghiên cứu và dự đoán các khả năng mang bệnh của người khám . Và đặc biệt Deep Learning còn được ứng dụng trong các xe tự hành để dự đoán các tình huống giao thông



Hình 3. Quan hệ giữa dữ liệu và khả năng dự đoán của thuật toán. Nguồn: Coursera

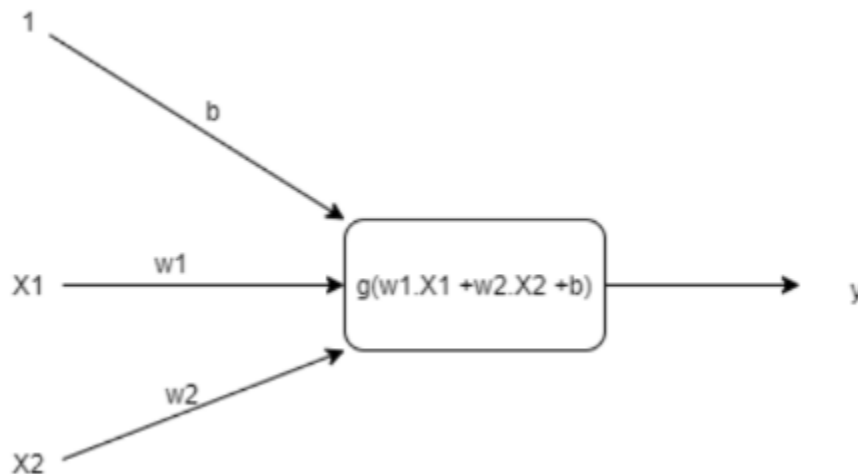
Các ứng dụng khác nhau cũng ta có thể áp dụng các Neuron Network khác:

Các ứng dụng	Neural Network có thể áp dụng
Đoán giá nhà	Standard Neural Network
Xe tự hành	Custom, Hybrid Neural Network
Nhận diện giọng nói, máy dịch	Current Neural Network
Nhận diện vật thể	Convolutional Neural Network

2.1 Neural Network

Neural Network là một mạng lưới gồm nhiều lớp được lấy cảm hứng từ người. Ở đó, lớp đầu tiên để đưa ra các đặc tính của vật của dự đoán vào được gọi là Input Layer. Và layer cuối cùng mang kết quả dự đoán được gọi là Output Layer. Một mạng neuron có thể có hoặc không có các lớp ở giữa Input Layer và Output Layer gọi là Hidden Layer, các Hidden Layer này giúp cho tỉ lệ dự đoán chính xác cao hơn tuy nhiên việc huấn luyện cũng tốn nhiều thời gian và dung lượng hơn. Mỗi Layer là tập hợp nhiều node, các node của lớp sau kết nối toàn bộ các node của lớp trước.

Mỗi node trong hidden layer và output layer thực hiện các công việc sau: Liên kết với tất cả các node ở layer trước đó với các hệ số w riêng. Mỗi node có 1 hệ số bias b riêng. Từ đó w, b biểu thị mối quan hệ giữa node sau.



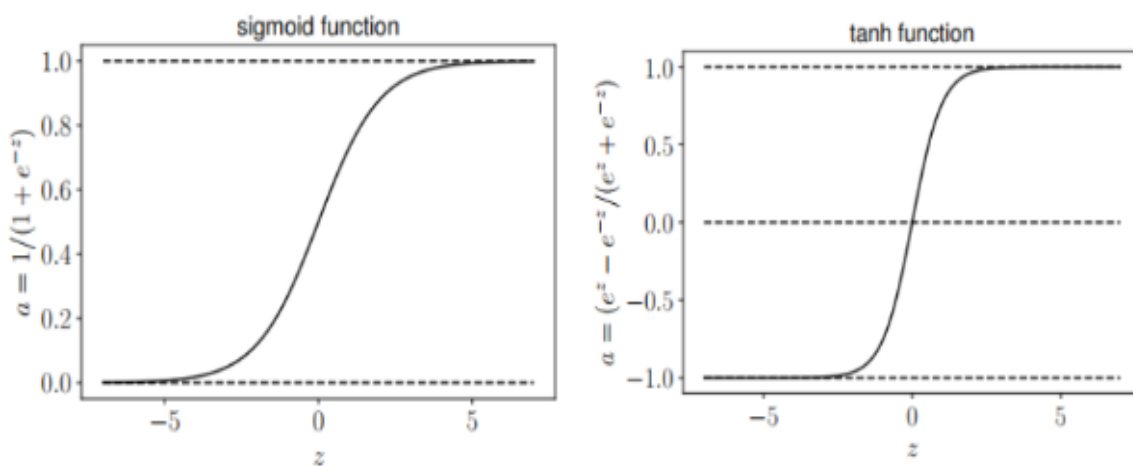
Hình 4 mối quan hệ giữa ngõ vào và ngõ ra của một node.

Node phía trên đầu vào là X_1, X_2 , trọng số w_1, w_2 . Ngõ ra là y kết quả của một hàm phi tuyến và một hàm tuyến tính. Hàm phi tuyến g còn được gọi là một hàm kích hoạt (activation) giúp tăng khả năng học của mạng neural.

Các hàm phi kích hoạt thường được sử dụng là:

- Sigmoid: Một hàm sigmoid là một hàm bị chặn, hàm số khả vi, hàm thực mà được định nghĩa cho tất cả giá trị thực và chứa một đạo hàm không âm ở mỗi điểm và có chính xác một điểm uốn cong (điểm quan trọng làm cho hàm có hình chữ S). Một hàm sigmoid và một cường cong sigmoid đều nói về cùng một đối tượng giống nhau.
- Tanh : Giá trị ngõ ra được chuyển về trong khoảng $[-1, 1]$ khiến nó có tính chất tâm không (zero - centered) , theo công thức :

$$a = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$
 Khi đó , ngõ ra bằng 1 khi ngõ vào lớn và ngõ ra bằng -1 khi ngõ vào nhỏ . Hàm Tanh với tính chất tâm không giúp các dữ liệu được phân bố quanh điểm 0 , tanh khi lấy đạo hàm có cả phần dương và phần âm giúp việc hội tụ trở nên tốt hơn . Tuy nhiên hàm tanh không giải quyết được vấn đề bão hòa khi giá trị ngõ vào quá lớn hoặc quá nhỏ của sigmoid.

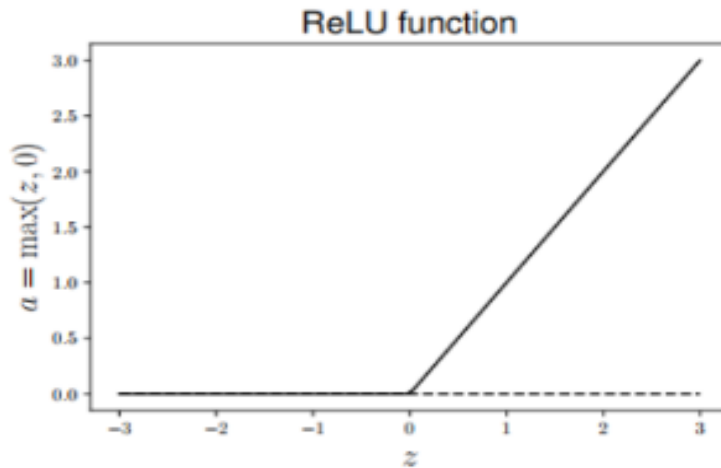


Hình 5 hình đồ thị hàm sigmoid và hàm tanh

- ReLU và leaky ReLU : Lấy ngưỡng giá trị ở 0 (Thay thế các giá trị âm bằng 0) :

$$g(x) = \max(0, x)$$
 Hàm ReLU loại bỏ các giá trị âm , việc hội tụ cũng diễn ra nhanh hơn khi không còn bị bão hòa ở hai đầu như hàm Sigmoid và hàm Tanh . Tuy nhiên, với các giá trị âm nhỏ gần 0 , việc giữ lại đạo hàm vẫn có giá trị khi lấy Gradient,

nhưng lại bị triệt tiêu gây ra hiện tượng “ Dying ReLU ” . Để khắc phục việc đó, hàm Leaky ReLU ra đời và là biến thể của ReLU : $g(x) = \max(x, 0)$, các giá trị âm lớn gần như xấp xỉ bằng 0 trong khi các giá trị âm nhỏ vẫn mang giá trị khi đạo hàm. Nhưng cả hai hàm RELU và Leaky ReLU lại không có đạo hàm tại 0.

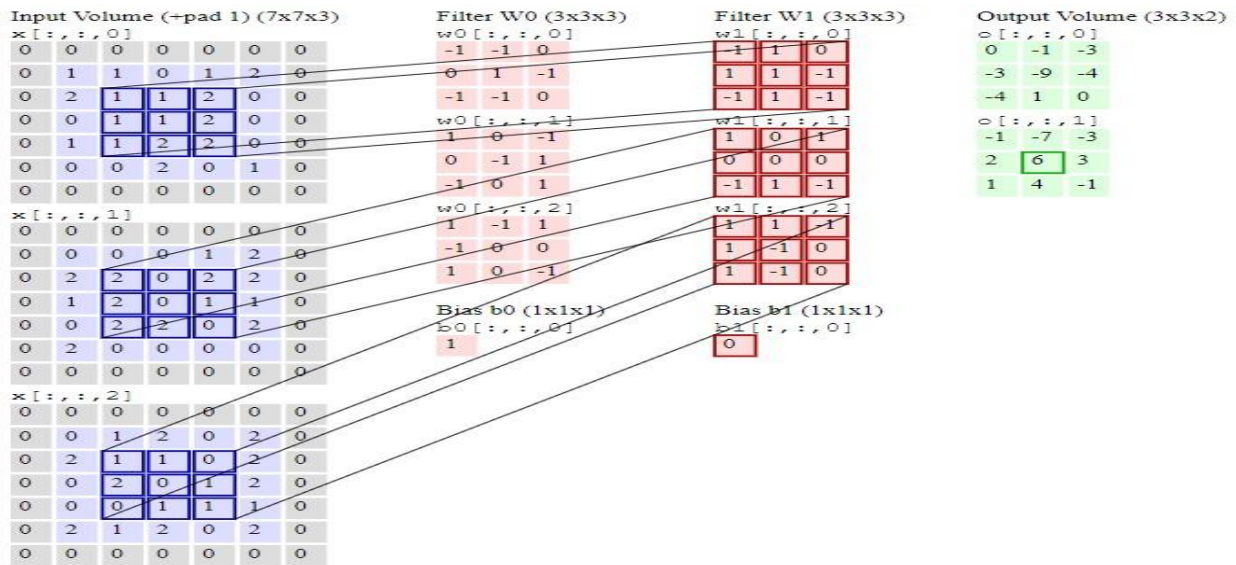


Hình 6 đồ thị hàm ReLu

2.2 Convolutional Neural Network

2.3.1 Định nghĩa mạng nơ-ron tích chập

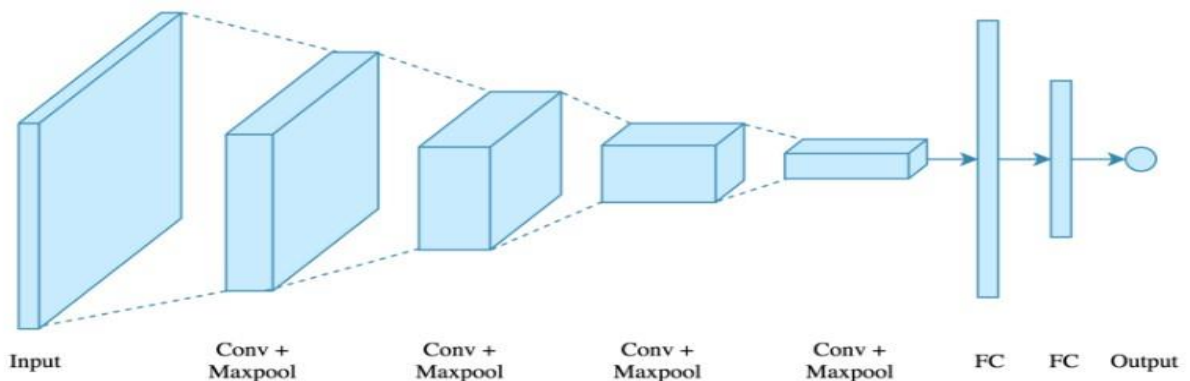
Tích chập được ứng dụng phổ biến trong lĩnh vực thị giác máy tính. Thông qua các phép tích chập, các đặc trưng chính từ ảnh được chiết xuất và truyền vào các lớp tích chập (layer convolution). Mỗi một lớp tích chập sẽ bao gồm nhiều đơn vị mà kết quả ở mỗi đơn vị là một phép biến đổi tích chập từ layer trước đó thông qua phép nhân tích chập với bộ lọc.



Hình 7 Minh họa tích chập

2.3.2 Mô hình mạng nơron tích chập

đầu vào được cho qua một bộ lọc chạy dọc bức ảnh. Bộ lọc có kích thước là (3x3 hoặc 5x5) và áp dụng phép tích vô hướng để tính toán, cho ra một giá trị duy nhất. Đầu ra của phép tích chập là một tập các giá trị ảnh được gọi là mạng đặc trưng (features map).



Hình 8 hình mô hình mạng nơron tích chập

CNNs có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling lớp sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Về cơ bản thiết kế của một mạng

neuron tích chập 2 chiều có dạng như sau: **INPUT** \rightarrow **[[CONV \rightarrow RELU]** $N \rightarrow$ **POOL?** $M \rightarrow$ **[FC \rightarrow RELU]*K \rightarrow FC**

Trong đó:

- **Input:** Lớp đầu vào
- **Conv Layer (Lớp tích chập):** Mục tiêu của các lớp tích chập là trích chọn các đặc trưng của ảnh đầu vào.
- **ReLU Layer (Lớp biến đổi):** ReLU layer áp dụng các kích hoạt (activation function) $\max(0, x)$ lên đầu ra của Conv Layer, có tác dụng đưa các giá trị âm về thành 0. Layer này không thay đổi kích thước của ảnh và không có thêm bất kì tham số nào. Mục đích của lớp ReLU là đưa ảnh một mức ngưỡng, ở đây là 0. Để loại bỏ các giá trị âm không cần thiết mà có thể sẽ ảnh hưởng cho việc tính toán ở các layer sau đó.
- **Pool layer (Lớp tổng hợp):** Pool Layer thực hiện chức năng làm giảm chiều không gian của đầu vào và giảm độ phức tạp tính toán của model ngoài ra Pool Layer còn giúp kiểm soát hiện tượng overfitting. Thông thường, Pool layer có nhiều hình thức khác nhau phù hợp cho nhiều bài toán, tuy nhiên Max Pooling là được sử dụng nhiều vào phổ biến hơn cả với ý tưởng cũng rất sát với thực tế con người đó là: Giữ lại chi tiết quan trọng hay hiểu ở trong bài toán này chính giữ lại pixel có giá trị lớn nhất.
- **Fully_Connected (Lớp kết nối hoàn toàn):** Tại lớp mạng này, mỗi một neuron của layer này sẽ liên kết tới mọi neuron của lớp khác. Để đưa ảnh từ các layer trước vào mạng này, buộc phải dãn phẳng bức ảnh ra thành 1 vector thay vì là mảng nhiều chiều như trước. Tại layer cuối cùng sẽ sử dụng 1 hàm kinh điển trong học máy mà bất kì ai cũng từng sử dụng đó là softmax để phân loại đối tượng dựa vào vector đặc trưng đã được tính toán của các lớp trước đó.

Các kí hiệu $[N]$, $[M]$ hoặc $[*K]$ ám chỉ cấu trúc bên trong $[]$ có thể lặp lại nhiều lần liên tiếp nhau. M , K là số lần lặp lại. Kí hiệu \rightarrow đại diện cho các lớp liên kế nhau mà lớp đứng trước \rightarrow sẽ làm đầu vào cho lớp đứng sau.

Như vậy ta có thể thấy một mạng nơron tích chập về cơ bản có 3 quá trình khác nhau:

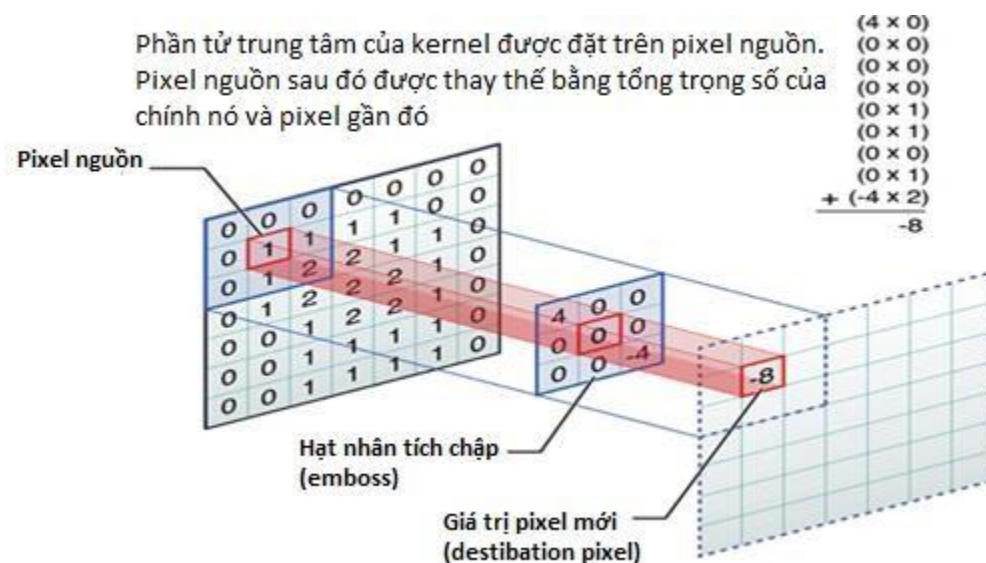
Quá trình chiết xuất đặc trưng: Thông qua các tích chập giữa ma trận đầu vào với bộ lọc để tạo thành các đơn vị trong một lớp mới. Quá trình này có thể diễn ra liên tục ở phần đầu của mạng và thường sử dụng hàm kích hoạt relu.

Quá trình tổng hợp: Các lớp ở về sau quá trình chiết xuất đặc trưng sẽ có kích thước lớn do số đơn vị ở các lớp sau thường tăng tiến theo cấp số nhân. Điều đó làm tăng số lượng hệ số và khối lượng tính toán trong mạng nơron. Do đó để giảm tải tính toán chúng ta sẽ cần giảm chiều của ma trận hoặc giảm số đơn vị của lớp. Vì mỗi một đơn vị sẽ là kết quả đại diện của việc áp dụng 1 bộ lọc để tìm ra một đặc trưng cụ thể nên việc giảm số đơn vị sẽ không khả thi. Giảm kích thước ma trận thông qua việc tìm ra 1 giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh. Do đó quá trình giảm chiều ma trận được áp dụng. Quá trình này gọi là tổng hợp.

Quá trình kết nối hoàn toàn: Sau khi đã giảm số lượng tham số đến một mức độ hợp lý, ma trận cần được làm dẹt (flatten) thành một vector và sử dụng các kết nối hoàn toàn giữa các lớp. Quá trình này sẽ diễn ra cuối mạng tích chập và sử dụng hàm kích hoạt là relu. Kết nối cuối cùng sẽ dẫn tới các đơn vị là đại diện cho mỗi lớp với hàm kích hoạt là softmax nhằm mục đích tính xác suất.

2.3.3 tính chất của mạng nowrron tích chập

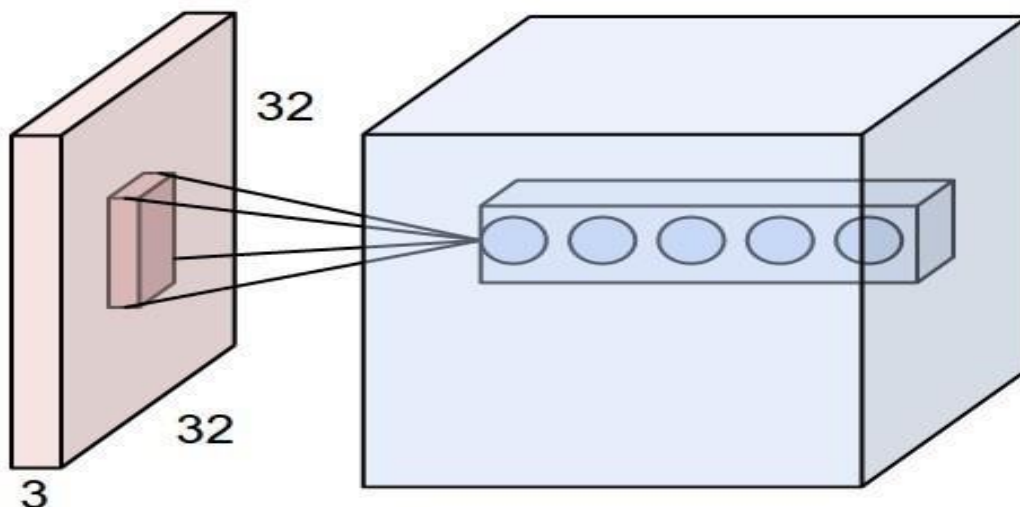
Tính kết nối trượt: Khác với các mạng nơ ron thông thường, mạng nơ ron tích chập không kết nối tới toàn bộ hình ảnh mà chỉ kết nối tới từng *vùng địa phương* (local region) có kích thước bằng kích thước bộ lọc của hình ảnh đó. Các bộ lọc sẽ trượt theo chiều của ảnh từ trái qua phải và từ trên xuống dưới đồng thời tính toán các giá trị tích chập và điền vào *bản đồ kích hoạt* (activation map).



Hình 9 Tích chập trên bản đồ kích hoạt

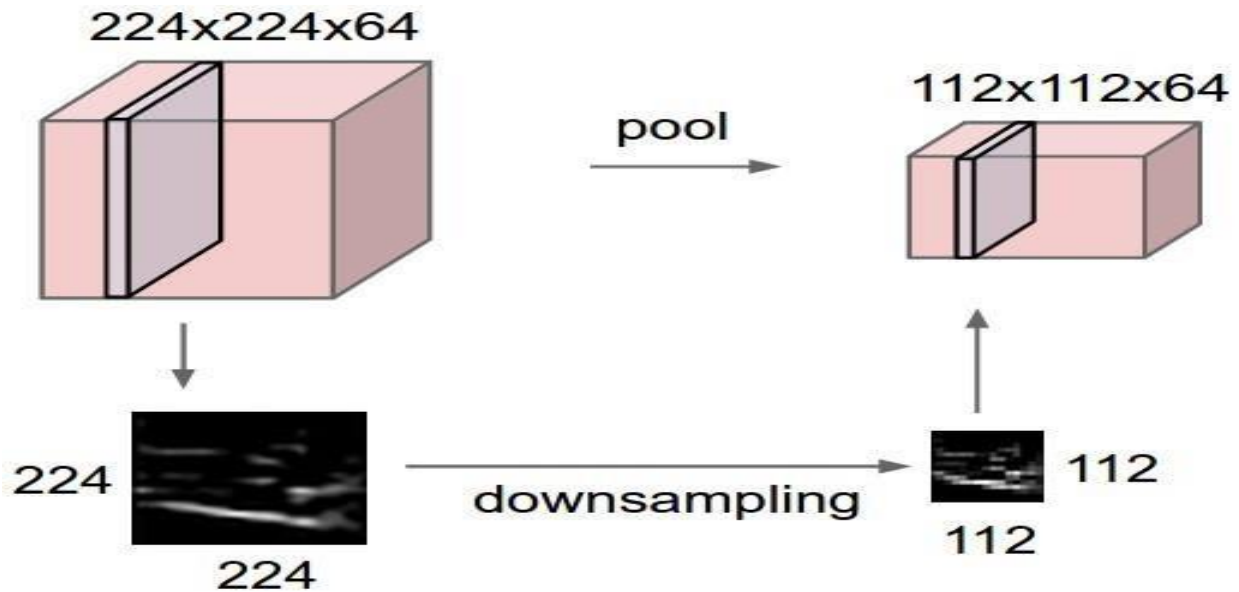
Tính chia sẻ kết nối và kết nối cục bộ: Chúng ta đã biết quá trình biến đổi trong mạng tích chập sẽ kết nối các khối nơ ron 3D. Tuy nhiên các đơn vị sẽ không kết nối tới toàn bộ khối 3D trước đó theo chiều width và height mà chúng sẽ chọn ra các *vùng địa phương* có kích thước bằng với bộ lọc giống như quá trình tính tích chập. Các vùng địa phương sẽ được chia sẻ chung một bộ siêu tham số gọi là trường tiếp nhận (receptive field) của bộ lọc. Tuy nhiên các kết nối cục bộ chỉ diễn ra theo chiều width và height. Kết nối sẽ mở rộng hoàn toàn theo chiều depth. Như vậy số tham số trong một lớp sẽ là $F \times F \times DF \times F \times D$ (F, DF, D lần lượt là kích thước bộ lọc và chiều depth).

Mỗi bộ lọc đại diện cho một khả năng chiết xuất một đặc trưng nào đó. Do đó khi đi qua toàn bộ các vùng địa phương của khối nơ ron 3D, các đặc trưng được chiết xuất sẽ hiển thị trên lớp mới.



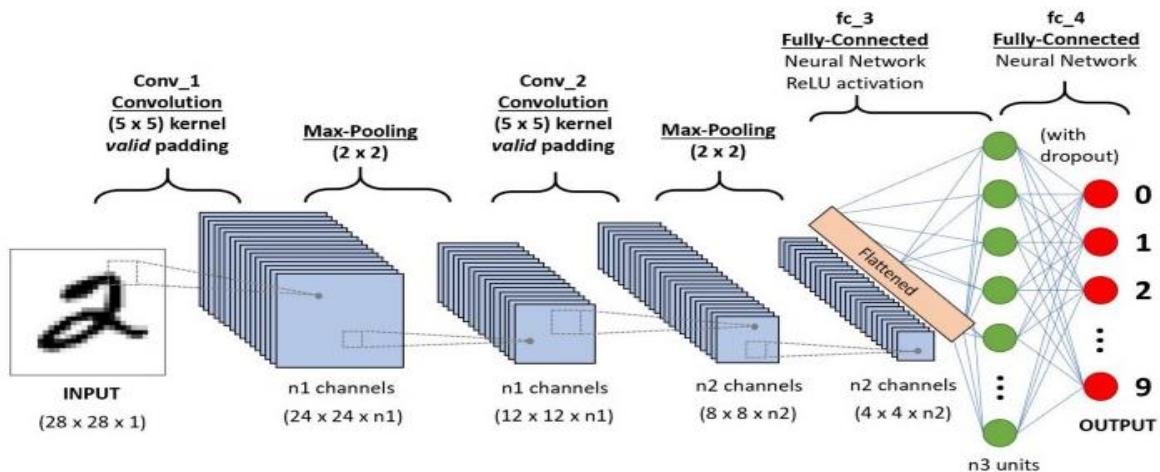
Hình 10 Hình kết nối cục bộ

Tính tổng hợp: Chúng ta tưởng tượng rằng ở các lớp tích chập gần cuối số tham số sẽ cực kì lớn do sự gia tăng của chiều depth và thông thường sẽ theo cấp số nhân. Như vậy nếu không có một cơ chế kiểm soát sự gia tăng tham số, chi phí tính toán sẽ cực kì lớn và vượt quá khả năng của một số máy tính cấu hình yếu (*Như máy của mình chẳng hạn, hơi đáng buồn*). Một cách tự nhiên là chúng ta sẽ giảm kích thước các chiều width và height (down sampling) mà vẫn giữ nguyên được các đặc trưng của khối. Các thực hiện tương tự như tính tích chập nhưng thay vì tính tích hadamard giữa ma trận bộ lọc và vùng địa phương ta sẽ tính trung bình (average pooling) hoặc giá trị lớn nhất (max pooling) của các phần tử trong vùng địa phương. Trước đây các tính trung bình được áp dụng nhiều nhưng các mô hình hiện đại đã thay thế bằng giá trị lớn nhất do tốc độ tính max nhanh hơn so với trung bình.



Hình 11 Quá trình tổng hợp

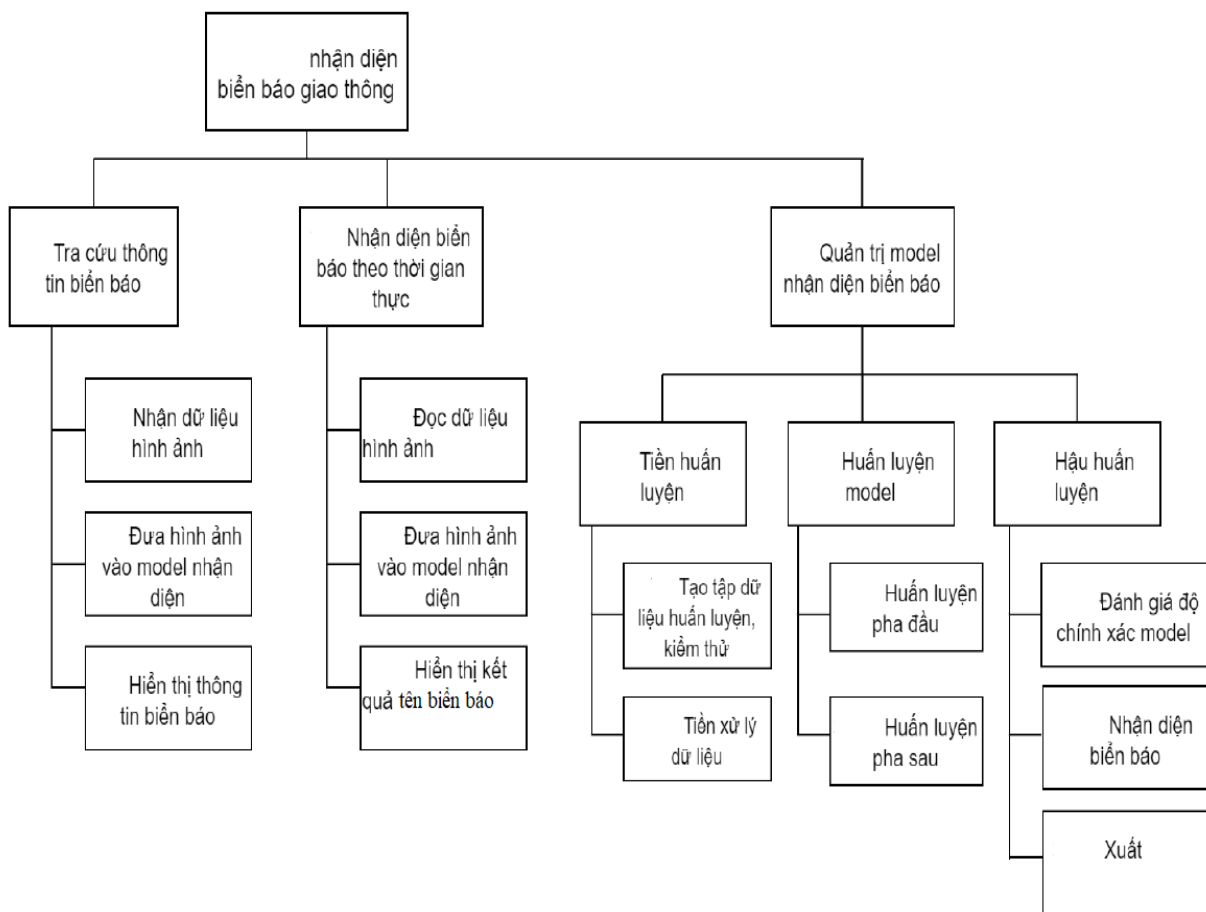
Độ phức tạp phát hiện hình ảnh tăng dần: Ở lớp đầu tiên hình ảnh mà chúng ta có chỉ là những giá trị pixels. Sau khi đi qua lớp thứ 2 máy tính sẽ nhận diện được các hình dạng cạnh, rìa và các đường nét đơn giản. Càng ở những lớp tích chập về sau càng có khả năng phát hiện các đường nét phức tạp hoặc vật thể. Đầu ra ở lớp cuối cùng là xác suất thuộc về mỗi lớp.



Hình 12 Một trình tự CNN để phân loại các chữ số viết tay

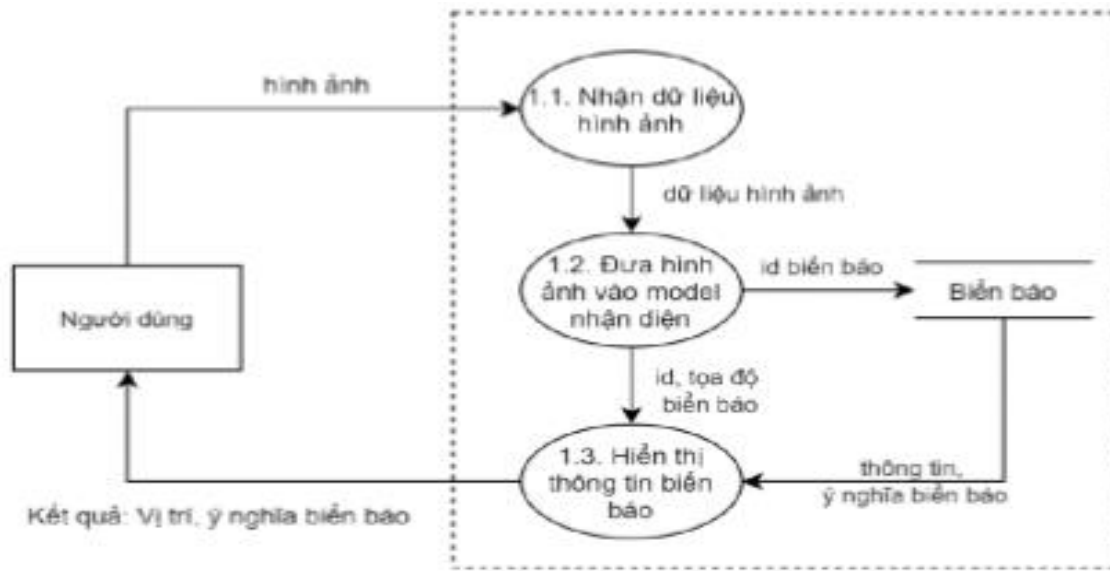
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Sơ đồ phân rã chức năng



Hình 13 Sơ đồ chức năng

3.2 Sơ đồ sử dụng



Hình 14 Sơ đồ sử dụng

3.3 Nhận dạng biển báo giao thông

- Gồm 4 bước:
 - Khám phá tập dữ liệu.
 - Xây dựng mô hình CNN.
 - Training và xác thực.
 - Kiểm tra mô hình.

3.3.1 Khám phá tập dữ liệu

- Thư mục train có chứa 43 thư mục con, mỗi thư mục đại diện cho một lớp khác nhau được đánh số thứ tự từ 0 đến 42.

- Thư viện PIL được sử dụng để chuyển nội dung hình ảnh thành một mảng.

```
# Import Ảnh
count = 0
images = []
classNo = []
myList = os.listdir(path)
print("Tổng số Classes: ", len(myList))
noOfClasses = len(myList)
print("Importing Classes....")
for x in range(0, len(myList)):
    myPicList = os.listdir(path + "/" + str(count))
    for y in myPicList:
        curImg = cv2.imread(path + "/" + str(count) + "/" + y)
        curImg = cv2.resize(curImg, (32,32))
        images.append(curImg)
        classNo.append(count)
    print(count, end=" ")
    count += 1
print(" ")
# chuyển đổi danh sách thành mảng
images = np.array(images)
classNo = np.array(classNo)

# Split Data
X_train, X_test, y_train, y_test = train_test_split(images, classNo, test_size=testRatio)
X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train, test_size=validationRatio)
```

- Sau đó ta sẽ lưu tất cả các hình ảnh và nhãn của chúng vào lists images và classNo.
- Chúng ta cần chuyển các lists thành các numpy array để làm đầu vào cho model.
- Kích thước của dữ liệu là (39209, 30, 30, 3), có nghĩa là có 39209 hình ảnh, mỗi ảnh có kích thước 30×30 pixel, số 3 ở cuối thể hiện dữ liệu là hình ảnh màu (giá trị RGB).
- Sử dụng phương thức train_test_split () của sklearn để phân chia bộ dữ liệu thành train set và test set
- Sử dụng phương thức to_categorical của keras.utils để xử lý các nhãn của y_train và y_test bằng one-hot encoding.

3.3.2 Xây dựng mô hình

- Để phân loại các hình ảnh thành các danh mục tương ứng, chúng ta sẽ xây dựng mô hình CNN.
- CNN là mô hình tốt nhất cho mục đích phân loại hình ảnh.
- Kiến trúc mô hình CNN của chúng ta như sau:

```
# CNN Model dựa theo LeNet
def myModel():
    num_filters = 60
    size_of_filter = (5, 5) # KERNEL size.
    size_of_filter2 = (3, 3)
    size_of_pool = (2, 2) # Pooling size
    model = Sequential()
    model.add(Conv2D(num_filters, size_of_filter, input_shape=((32, 32, 3)[0], (32, 32, 3)[1], 1), activation='relu'))
    model.add(Conv2D(num_filters, size_of_filter, activation='relu'))
    model.add(MaxPooling2D(pool_size=size_of_pool))

    model.add(Conv2D(num_filters // 2, size_of_filter2, activation='relu'))
    model.add(Conv2D(num_filters // 2, size_of_filter2, activation='relu'))
    model.add(MaxPooling2D(pool_size=size_of_pool))
    model.add(Dropout(0.5)) # => Giảm Overfitting

    model.add(Flatten())
    model.add(Dense(500, activation='relu')) # HIDDEN LAYER
    model.add(Dropout(0.5))
    model.add(Dense(num_classes, activation='softmax')) # OUTPUT LAYER
    # COMPILER MODEL
    model.compile(Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

3.3.3 Tranning

Sau khi xây dựng kiến trúc mô hình, chúng ta sẽ huấn luyện mô hình bằng cách sử dụng `model.fit()`. Và tiến hành lưu model.

```

model = myModel()
print(model.summary())
history = model.fit(dataGen.flow(X_train, y_train, batch_size=50),
                    epochs=35, validation_data=(X_validation, y_validation), shuffle=1)

# PLOT
plt.figure(1)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['training', 'validation'])
plt.title('loss')
plt.xlabel('epoch')
plt.figure(2)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['training', 'validation'])
plt.title('Accuracy')
plt.xlabel('epoch')
plt.show()
score = model.evaluate(X_test, y_test, verbose=0)
print('Test Score:', score[0])
print('Test Accuracy:', score[1])

# Lưu model vào file .h5 để sử dụng cho real-time
model.save('/content/drive/MyDrive/AI/data/TrafficSign-Classification-CNN-main/modelfix.h5')
cv2.waitKey(0)

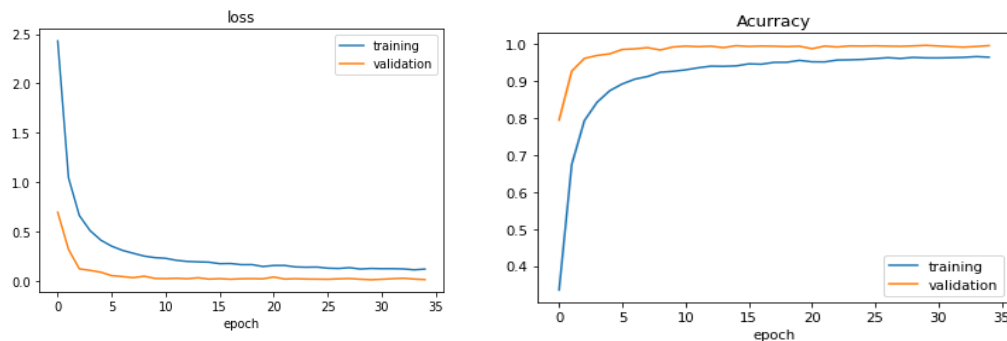
```


CHƯƠNG 4 KẾT QUẢ THỰC NGHIỆM

4.1 Đánh giá mô hình

Dữ liệu thu thập cho đề án này bao gồm 43 thư mục hình ảnh với hơn 40000 ảnh dữ liệu về biển báo giao thông. Những hình ảnh được lưu trong các thư mục được đánh số từ 0 đến 42.

Kết quả sau khi train:



Hình 15 Hình ảnh về hàm mất mát và độ chính xác của model sau khi train

Sau khi training kết quả đạt được là khoảng 99,69%, như vậy mô hình của chúng ta khá chính xác.

4.2 Kết quả nhận diện hình ảnh sử dụng real-time

Phần mềm được viết bằng python và chạy trên VScode

Sử dụng thư viện cv2 để đọc và xử lý ảnh từ webcam và tiến hành dự đoán kết quả và trả kết quả qua khung hiện trên màn hình máy tính.

Kết quả đạt được:



Hình 16 Hình ảnh nhận dạng 4 loại biển báo giao thông

Nhận xét: kết quả trả về với độ chính xác cao tuy nhiên, chỉ nhận diện được với những ảnh ở gần và trong nền không phức tạp.

CHƯƠNG 5 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được

Sau thời gian học tập, nghiên cứu và tìm hiểu đề em đã hoàn thành được sản phẩm đúng với mục tiêu đã đề ra.

Tỉ lệ nhận diện đối với tập dữ liệu có sẵn cũng tương đối cao, đạt 99,69%. Vì đây là tập dữ liệu tốt, các hình ảnh chụp rõ nét, thay đổi về hướng mặt, độ sáng không quá nhiều.

5.2 Hướng phát triển

Trong tương lai để chương trình được hoàn thiện hơn nữa thì chúng em cần sự giúp đỡ rất nhiều từ phía thầy cô và nhà trường để đề tài có thể ứng dụng vào những chiếc xe oto trong thực tế.

Tài Liệu Tham Khảo

- [1] Vũ Hữu Tiệp nghiên cứu sinh ngành Học Máy và Thị Giác Máy Tính. “Machine learning cơ bản”.
- [2] Y. LeCun and Y. Bengio. “Convolutional networks for images, speech, and time-series.” In M. A. Arbib, editor, *The Handbook of Brain Theory and Noron Networks*. MIT Press, 1995
- [3] Nguyễn Thanh Hải. (2014). “Giáo trình xử lý ảnh (Ngành Điện – Điện tử)”. Nhà xuất bản đại học quốc gia, Trường Đại học Sư Phạm Kỹ Thuật TPHCM.

Các website tham khảo:

- [4] <http://noronnetworksanddeeplearning.com/chap6.html>
- [5] <http://noronnetworksanddeeplearning.com/chap2.html>
- [6] www.emgucv.com
- [7] www.opencv.com