**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN TỐT NGHIỆP

**XÂY DỰNG HỆ THỐNG GÁN NHÃN VÀ NHẬN DIỆN THUỐC LÁ ĐIỆN TỬ**

Sinh viên thực hiện: Đào Duy Đán

Giảng viên hướng dẫn: TS. Nguyễn Thọ Thông

HÀ NỘI, NĂM 2023

# MỤC LỤC

[MỤC LỤC 2](#_Toc153417559)

[DANH MỤC HÌNH ẢNH 3](#_Toc153417560)

[DANH MỤC BẢNG BIỂU 3](#_Toc153417561)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ 3](#_Toc153417562)

[**I.** **TỔNG QUAN ĐỒ ÁN TỐT NGHIỆP.** 4](#_Toc153417563)

[**1.1.** **Giới thiệu đề tài.** 4](#_Toc153417564)

[**1.2.** **Cơ sở lý thuyết.** 5](#_Toc153417565)

[1.3. Các công nghệ sử dụng. 12](#_Toc153417566)

[**II.** **XÂY DỰNG MÔ HÌNH HỌC SÂU.** 12](#_Toc153417567)

[2.1. Tổng quan về bài toán nhận diện thuốc lá điện tử. 12](#_Toc153417568)

[2.2. Thu thập dữ liệu. 12](#_Toc153417569)

[2.3. Chuẩn bị dữ liệu. 12](#_Toc153417570)

[2.4. Cấu trúc và kiến trúc mô hình Faster R-CNN. 12](#_Toc153417571)

[2.5. Lựa chọn thư viện và công cụ. 12](#_Toc153417572)

[2.6. Các tham số và thành phần chính của mô hình. 12](#_Toc153417573)

[2.7. Huấn luyện mô hình. 12](#_Toc153417574)

[**III.** **PHÂN TÍCH THIẾT KẾ HỆ THỐNG.** 12](#_Toc153417575)

[3.1. Phân tích yêu cầu hệ thống. 12](#_Toc153417576)

[3.2. Đặc tả yêu cầu hệ thống. 12](#_Toc153417577)

[3.3. Thiết kế hệ thống gán nhãn. 12](#_Toc153417578)

[**IV.** **THỰC NGHIỆM, SO SÁNH VÀ ĐÁNH GIÁ.** 12](#_Toc153417579)

[4.1. Môi trường thực nghiệm – cấu hình. 12](#_Toc153417580)

[4.2. Kết quả thực nghiệm. 12](#_Toc153417581)

[4.3. So sánh với các phương pháp khác. 12](#_Toc153417582)

[4.3.1. Thuật toán Fast R-CNN. 13](#_Toc153417583)

[4.4. Đánh giá hiệu suất mô hình và phân tích kết quả. 13](#_Toc153417584)

[4.5. Kết luận. 13](#_Toc153417585)

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

1. **TỔNG QUAN ĐỒ ÁN TỐT NGHIỆP.**
   1. **Giới thiệu đề tài.**

Với sự phát triển không ngừng của lĩnh vực thị giác máy tính trong thập kỷ gần đây đã mở ra những cánh cửa mới cho ứng dụng thực tế như phân loại, trích xuất đặc trưng và phát hiện đối tượng trong hình ảnh. Cùng với đó là sự phát triển nhanh chóng của học sâu đã giúp chúng ta tiến xa hơn với việc tiếp cận vấn đề trên. Học sâu cho phép mô hình hóa tự động các đặc trưng phức tạp từ dữ liệu giúp chúng ta giải quyết các thách thức đặt ra trong việc nhận diện đối tượng trong các hình ảnh đa dạng.

Trong thời đại công nghệ hiện đại ngày nay, người tiêu dùng thường xuyên tiếp cận với những loại quảng cáo trực tuyến trên website, mạng xã hội và ứng dụng di động. Một trong những loại quảng cáo rất phổ biến và đã trở thành một thách thức lớn với sức khỏe người tiêu dùng, nhất là với những người trẻ tuổi đó là thuốc lá điện tử (e-cigarettle). Tại Việt Nam hiện nay xuất hiện tràn lan các sản phẩn thuốc lá điện tử không rõ nguồn gốc có giá thành rất rẻ, chủ yếu là hàng kém chất lượng, rất độc hại và đã gây ra rất nhiều ca ngộ độc. Vì vậy hiện nay nước ta và rất nhiều quốc gia trên thế giới đã thực hiện các biện pháp nhằm kiểm soát, hạn chế quảng cáo và buôn bán loại thuốc lá này.

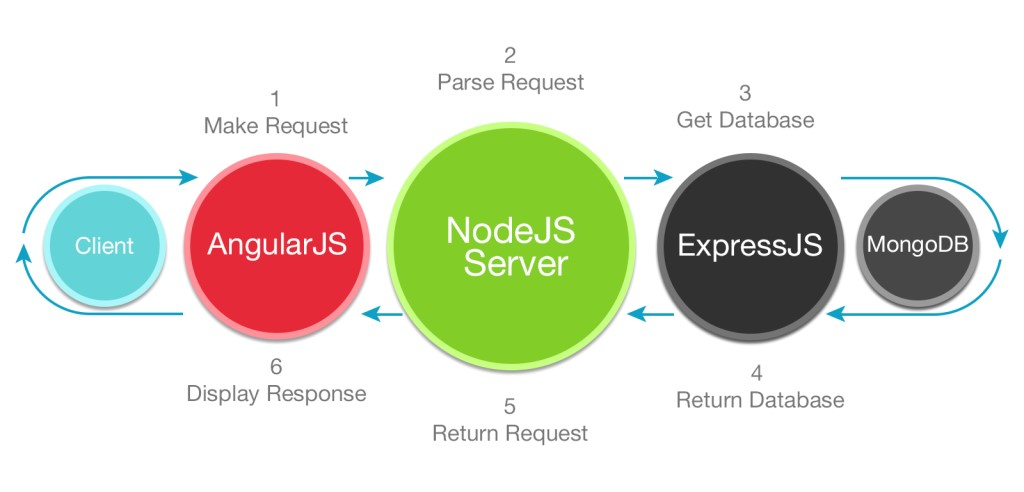
Từ sự phát triển của công nghệ và tình hình thực tiễn đã nêu ở trên, đồ án này sẽ tập trung vào việc xây dựng một hệ thống gán nhãn và dùng thuật toán Faster R-CNN để xây dựng mô hình nhận diện đối tượng thuốc lá điện tử. Hệ thống gán nhãn dùng để gán nhãn những đối tượng thuốc lá điện tử thu thập được từ thực tế và từ đó trích xuất ra dữ liệu đúng định dạng dùng cho việc huấn luyện. Thuật toán Faster R-CNN là một trong những thuật toán tiên tiến nhất trong lĩnh vực nhận diện đối tượng, với khả năng tự động tạo ra các vùng đề xuất (Region Proposals) thông qua RPN và sau đó phân loại đối tượng trong các vùng này.

Các nội dung chủ yếu sẽ hướng tới việc phân tích thiết kế, xây dựng hệ thống gán nhãn và nêu các ý tưởng, phạm vi và cài đặt thuật toán Faster R-CNN trong việc nhận diện thuốc lá điện tử. Mục tiêu của đề tài không chỉ cung cấp giải pháp hiệu quả mà còn mang lại những hiệu quả tích cực cho xã hội đặc biệt trong lĩnh vực nghiên cứu y tế trong việc có thể kiểm soát và hạn chế việc buôn bán thuốc lá điện tử.

* 1. **Cơ sở lý thuyết.**
     1. **MEAN.js stack.**

Thuật ngữ MEAN.js là một giải pháp mã nguồn mở JavaScript đầy đủ được sử dụng để xây dựng các trang web và ứng dụng web động. MEAN là từ viết tắt của MongoDB, ExpressJS, NodeJS và AngularJS là các thành phần chính của MEAN.js stack.

Về cơ bản nó được phát triển để giải quyết các vấn đề phổ biến khi kết nối các khung đó (MongoDB, ExpressJS, NodeJS, AngularJS), xây dựng được một bộ khung mạnh mẽ để hỗ trợ nhu cầu phát triển hàng ngày và giúp các lập trình viên sử dụng các phương pháp thực hành tốt hơn khi làm việc với các thành phần JavaScript phổ biến. Stack nghĩa là sử dụng cơ sở dữ liệu, máy chủ web ở back end, ở giữa sẽ có logic và khả năng kiểm soát ứng dụng cũng như sự tương tác với người dùng ở front end.



*Hình 1: Cấu trúc MEAN.js stack và các luồng hoạt động.*

Lựa chọn MEAN.js cho hệ thống gán nhãn được thực hiện với những lý do cụ thể, đặc biệt tập trung vào sự linh hoạt và hiệu suất trong xử lý dữ liệu đa dạng để phát triển hệ thống gán nhãn. Điều đó được thể hệ qua các thành phần bên trong nó:

1. **MongoDB.**

MongoDB là một database hướng tài liệu, một dạng NoSQL database. Vì vậy, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. Các dữ liệu được lưu trữ trong document kiểu JSON nên việc truy vấn rất nhanh.

Dữ liệu trong MongoDB được lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao. Điều này tạo điều kiện lưu trữ các thông tin về ảnh cũng như thông tin các đối tượng được gán nhãn trong ảnh vào JSON file một cách nhanh chóng và khi truy vấn có thêm thông tin về gán ảnh vào đối tượng một cách dễ dàng với định dạng JSON.

Ngoài ra với đặc tính hiệu suất cao, dễ dàng mở rộng lưu trữ và dữ liệu được ghi đệm lên RAM giúp hạn chế việc truy cập vào ổ cứng của MongoDB có thể cho phép hệ thống lưu trữ rất nhiều ảnh mà vẫn đảm bảo được tốc độ truy vấn dữ liệu nhanh.

1. **ExpressJS.**

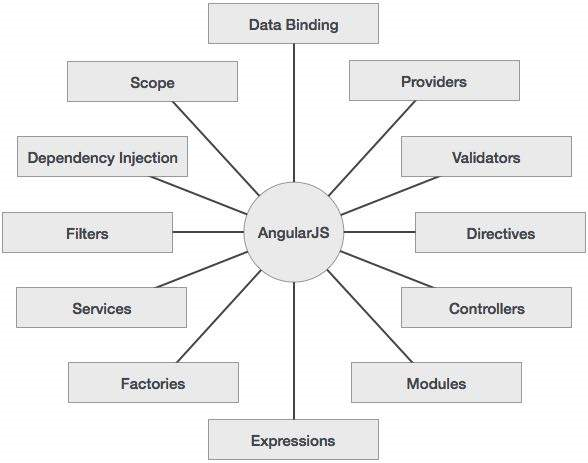
ExpressJS là một khung ứng dụng web NodeJS được tối giản và linh hoạt, cung cấp một bộ tính năng mạnh mẽ để phát triển các ứng dụng web và di động. Nó tạo điều kiện cho sự phát triển nhanh chóng của các ứng dụng Web dựa trên NodeJS. Rất dễ dàng để cấu hình và tùy chỉnh, cho phép xây dựng các ứng dụng, module một cách dễ dàng nhanh chóng. Nó chỉ định các routes của một ứng dụng tùy thuộc và HTTP method và URLs. Và nó cũng cho phép thiết lập middlewares để phản hồi HTTP requests.

1. **AngularJS.**

AngularJS là một framework phát triển web mã nguồn mở được phát triển bởi Google, được sử dụng để phát triển ứng dụng web đơn trang (SPAs). Được viết bằng Javascript, Angular cung cấp một cơ sở hạ tầng mạnh mẽ để phát triển ứng dụng web phức tạp và linh hoạt. Angular được tổ chức theo mô hình MVC giúp tách rõ ràng các thành phần của ứng dụng, tạo điều kiện thuận lợi cho quản lý mã nguồn và bảo trì.

Các tính năng cốt lõi của AngularJS có thể kể đến như:

* Data binding – Sự đồng bộ hóa dữ liệu tự động giữa các thành phần model và views.
* Scope – Đây là những đối tượng tham chiếu tới model. Nó hoạt động như một sự liên kết giữa controller và view.
* Controller – Đây là các hàm JavaScript được liên kết với một phạm vi cụ thể.
* Services – AngularJS đi kèm với một số dịch vụ tích hợp sẵn như $http để tạo XMLHttpRequests. Đây là những đối tượng đơn lẻ chỉ được khởi tạo một lần trong ứng dụng.
* Directives – là các thành phần mở rộng cho các thẻ html dùng để bổ trợ các thuộc tính nâng cao cho các thẻ html và cập nhật các thành phần giao diện web. Một số thuộc tính có thể kể đến như ngBind, ngModel,…
* Templates – Đây là chế độ xem được hiển thị với thông tin từ controller và model.
* Routing – dùng để chuyển đổi giữa các views.
* Dependency Injection – AngularJS có tích hợp dependency injection subsystem giúp nhà pháp triển tạo, hiểu và kiểm tra các ứng dụng một cách dễ dàng.

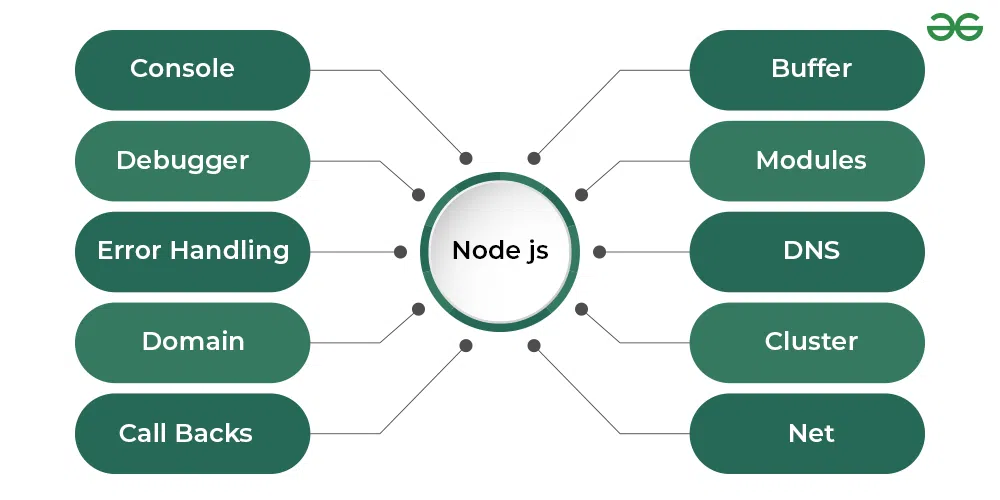


*Hình 2: Các thành phần cốt lõi của AngularJS.*

1. **NodeJS.**

NodeJS là một nền tảng phía máy chủ được sử dụng để phát triển các ứng dụng web như các trang video clip, forum và đặc biệt là các trang mạng xã hội phạm vi hẹp. Nó cung cấp một thư viện phong phú gồm nhiều JavaScript module khác nhau giúp đơn giản hóa việc phát triển ứng dụng web bằng NodeJS ở mức độ lớn. Nó được xây dựng trên công cụ JavaScript V8 của Google Chrome vì vậy nó thực thi mã rất nhanh.

NodeJS = Runtime Environment + JavaScript Library.



*Hình 3: Một số thành phần quan trọng trong NodeJS.*

* + 1. **Tổng quan về học sâu.**
       1. **Lý thuyết về bài toán nhận diện đối tượng trong thị giác máy tính.**

Thị giác máy tính (Computer Vision) là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh, phát hiện các đối tượng, tạo ảnh, siêu phân giải hình ảnh và còn nhiều hơn nữa. Object Detection – Nhận diện đối tượng có lẽ là khía cạnh sâu sắc nhất của thị giác máy tính do số lần sử dụng trong thực tế.

Object Detection là một nhiệm vụ trong thị giác máy tính, trong đó mục tiêu là phát hiện và định vị các đối tượng được quan tâm trong hình ảnh hoặc video. Nhiệm vụ này liên quan đến việc xác định vị trí và ranh giới của các đối tượng trong ảnh và phân loại các đối tượng thành các loại khác nhau. Nó tạo thành một phần quan trọng trong nhận dạng thị giác, bên cạnh việc phân loại và truy xuất hình ảnh.

Các phương pháp nhận diện đối tượng được bắt đầu sử dụng và triển khai sớm liên quan đến việc sử dụng các thuật toán cổ điển. Tuy nhiên, các thuật toán cổ điển này không thể đạt hiệu suất đủ để làm việc trong các điều kiện khác nhau.

Việc áp dụng đột phát và nhanh chóng của Deep Learning – Học sâu vào năm 2012 đã đưa vào sự tồn tại các thuật toán và phương pháp phát hiện đối tượng hiện đại và chính xác cao như R-CNN, Fast R-CNN, Faster R-CNN, RetinaNet,… và nhanh hơn nhưng rất chính xác như SSD và YOLO.

* + - 1. **Cơ sở lý thuyết về học sâu.**

Học sâu (Deep Learning) là một phần trong một nhánh rộng hơn các phương pháp học máy dựa trên mạng nơ ron nhân tạo kết hợp với việc học biểu diễn đặc trưng (representation learning). Học sâu được sử dụng để dạy máy tính xử lý dữ liệu theo cách được lấy cảm hứng từ bộ não con người. Mô hình học sâu có thể nhận diện nhiều hình mẫu phức tạp trong hình ảnh, văn bản, âm thanh và các dữ liệu khác để tạo ra thông tin chuyên sâu và dự đoán chính xác.

Các thuật toán học sâu là các mạng nơ ron được lập mô hình theo bộ não con người. Một bộ não con người chứa hàng triệu nơ ron được kết nối với nhau, làm việc cùng nhau để tìm hiểu và xử lý thông tin. Tương tự, các mạng nơ ron học sâu, hay mạng nơ ron nhân tạo, được tạo thành từ nhiều lớp nơ ron nhân tạo hoạt động cùng nhau trong máy tính. Các nơ ron nhân tạo là những module phần mềm được gọi là nút (node), sử dụng các phép toán để xử lý dữ liệu. Các mạng nơ ron nhân tạo là những thuật toán học sâu sử dụng các nút này để giải quyết các vấn đề phức tạp.



*Hình 4: Cấu trúc cơ bản của một mô hình học sâu.*

Một mạng nơ ron chuyên sâu sẽ có các thành phần sau.

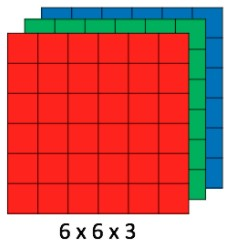
* Lớp đầu vào (Input Layer): Một mạng nơ ron nhân tạo sẽ có một số nút để nhập dữ liệu đầu vào.
* Lớp ẩn (Hidden Layer): Lớp đầu vào xử lý và chuyển dữ liệu đến các lớp sâu hơn trong mạng nơ ron. Các lớp ẩn này xử lý thông tin ở các cấp độ khác nhau, thích ứng với hành vi của mình khi nhận được thông tin mới. Các mạng học sâu có hàng trăm lớp ẩn để có thể phân tích một vấn đề từ nhiều góc độ khác nhau.
* Lớp đầu ra (Output Layer): Lớp đầu ra bao gồm các nút xuất dữ liệu. Các mô hình học sâu xuất ra đáp án “có” hoặc “không” nếu chỉ có hai nút trong lớp đầu ra. Mặt khác, các mô hình xuất ra nhiều đáp án hơn sẽ có nhiều nút hơn.

Học sâu thường được thực hiện thông qua học có giám sát, trong đó mô hình cố gắng dự đoán đầu ra dựa trên đầu vào và điều chỉnh trọng số sao cho sự chênh lệch giữa dự đoán và thực tế là nhỏ nhất.

Tuy nhiên, học sâu không chỉ giới hạn trong phạm vi học có giám sát. Có cả phương pháp học không giám sát và bán giám sát, nơi mô hình tự học cấu trúc của dữ liệu mà không cần nhãn.

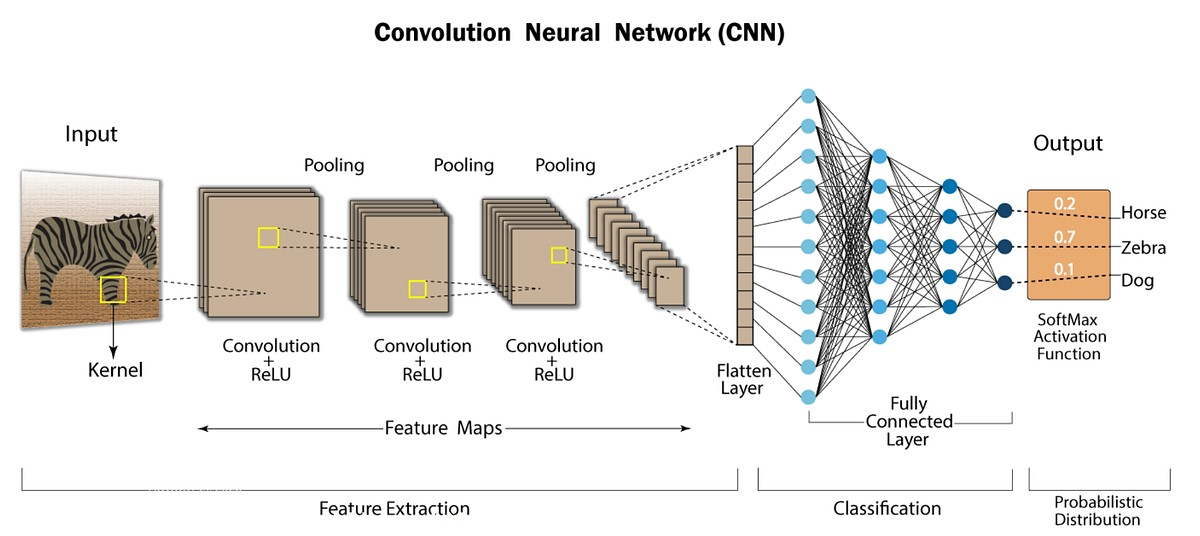
* + - 1. **Mạng nơ ron tích chập – CNN (Convolutional Neural Networks).**

CNN là một loại mạng nơ ron thị giác máy tính được thiết kế đặc biệt cho việc xử lý và nhận diện dữ liệu không gian như ảnh. CNN phân loại hình ảnh bằng cách lấy một hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định. Máy tính coi hình ảnh đầu vào là một mảng pixel và nó phụ thuộc và độ phân giải của hình ảnh. Dựa trên độ phân giải của hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao, W: Chiều rộng, D: Độ dày).



*Hình 5: Hình ảnh là mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).*

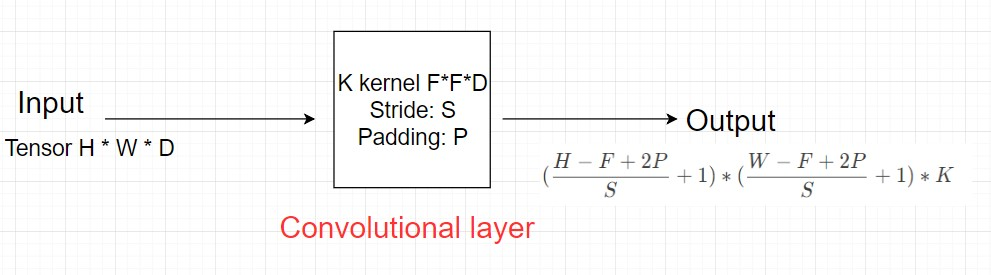
Về kỹ thuật, mô hình CNN để huấn luyện và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua một loạt các lớp tích chập (Convolutional) và các bộ lọc (Kernels) được áp dụng để trích xuất đặc trưng từ hình ảnh đầu vào, với hàm kích hoạt giữ lại thông tin quan trọng. Tiếp theo, lớp pooling giảm kích thước không gian của biểu diễn đặc trưng bằng cách chọn giá trị lớn nhất hoặc giá trị trung bình từ các vùng đặc trưng. Các lớp Fully Connected sau đó tạo ra biểu diễn phẳng để áp dụng softmax activation, chuyển đổi đầu ra thành xác suất dự đoán cho từng lớp. Mô hình này được huấn luyện thông qua việc điều chỉnh trọng số dựa trên hàm mất mát và kiểm tra dữ liệu mới để đánh giá hiệu suất.



*Hình 6: Cấu trúc mạng nơ ron tích chập – CNN.*

1. **Lớp tích chập – Convolutional Layer**

Là lớp đầu tiên để trích xuất các tính năng từ hình ảnh đầu vào. Tích chập duy trì mối quan hệ giữa các pixel bằng cách tìm hiểu các tính năng hình ảnh bằng cách sử dụng các ô vuông nhỏ của dữ liệu đầu vào.



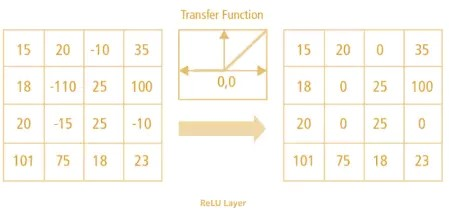
Một số khái niệm chính trong lớp kích hoạt là: Filter Map, Stride, Padding, Feature Map.

* Filter Map: CNN sử dụng Filter để áp dụng vào các vùng của hình ảnh. Những Filter này có thể xem là một ma trận 3 chiều gồm các con số là tham số đầu vào.
* Stride: Stride được hiểu là khi chúng ta dịch chuyển Fiter Map theo Pixel và dựa vào giá trị từ trái sang phải. Stride đơn giản biểu thị cho sự dịch chuyển này.
* Padding: Chính là những giá trị 0 được thêm vào lớp Input.
* Feature Map: Đây là kết quả được hiển thị sau mỗi lần Filter Map quét qua Input. Cứ mỗi lần quét qua sẽ thấy được sự xuất hiện của quá trình tính toán.

1. **Hàm phi tuyến – ReLU.**

ReLu là một hàm phi tuyến với đầu ra là

ReLu quan trọng bởi vì dữ liệu trong các vấn đề mà chúng ta tìm hiểu là các giá trị tuyến tính không âm.

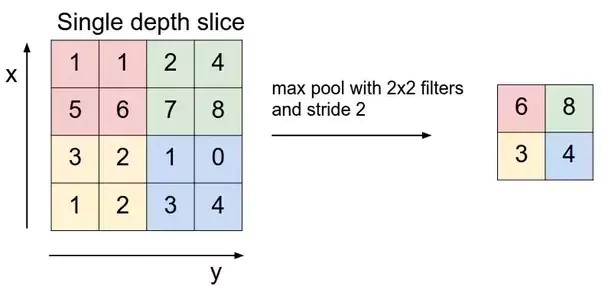


Có một số hàm phi tuyến khác như tanh, sigmoid cũng có thể thay thế cho ReLu. Nhưng hầu hết người ta dùng ReLu bởi vì nó có hiệu suất tốt.

1. **Lớp gộp – Pooling Layer.**

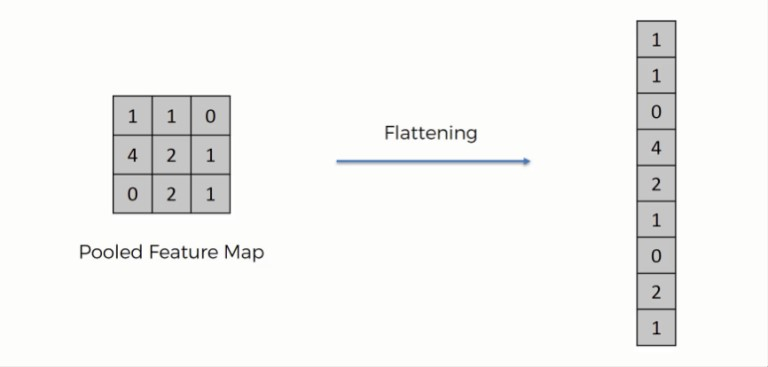
Lớp pooling sẽ giảm bớt số lượng tham số khi hình ảnh quá lớn. Không gian pooling còn được gọi là lấy mẫu con hoặc lấy mẫu suống làm giảm kích thước của mỗi map nhưng vẫn giữ lại thông tin quan trọng. Một số loại pooling:

* Max Pooling: Lấy lấy tử lớn nhất từ ma trận đối tượng.
* Average Pooling: Lấy tổng trung bình từ ma trận đối tượng.
* Sum Pooling: Tổng tất cả phần tử trong ma trận đối tượng.



1. **Fully connected layer.**

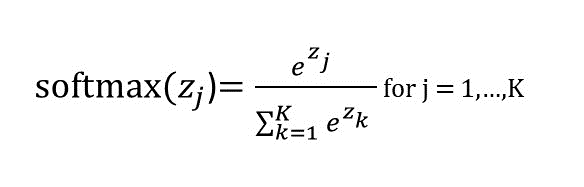
Sau khi ảnh được truyền qua nhiều Convolutional Layer và Pooling Layer thì model đã học được tương đối các đặc điểm của ảnh (ví dụ: đầu tank thuốc lá điện tử, khung thuốc lá điện tử,…) thì tensor của output của layer cuối cùng, kích thước H\*W\*D sẽ được chuyển về một vector kích thước HxWxD



Sau đó vector này được kết nối với một số lớp được kết nối đầy đủ giống như Mạng nơ ron nhân tạo và thực hiện các phép toán tương tự.

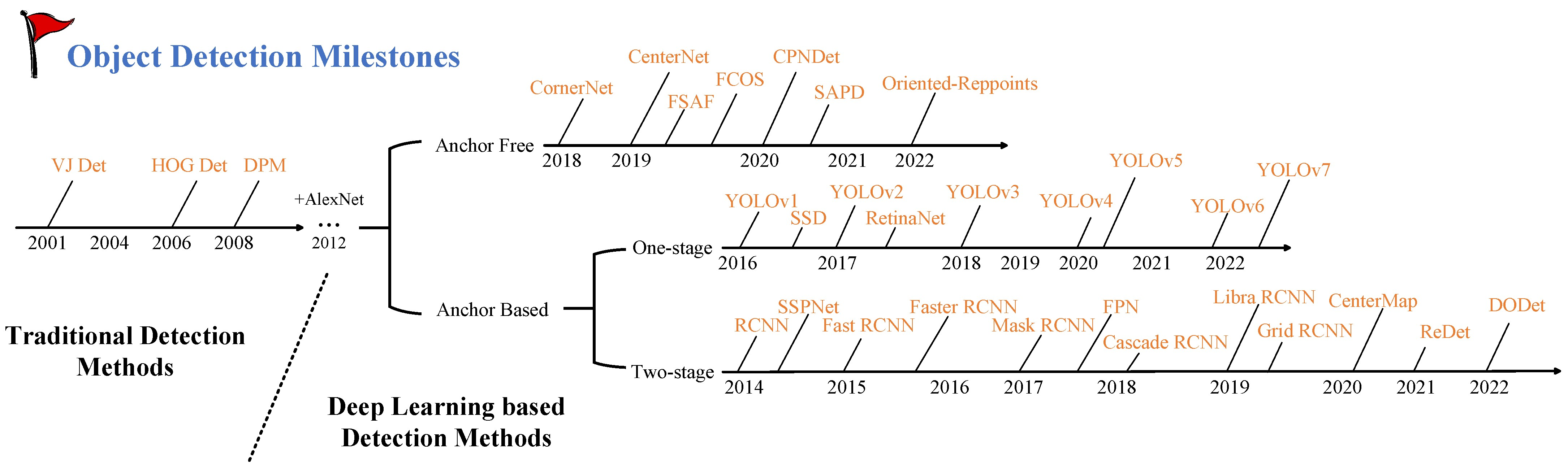
1. **Hàm kích hoạt softmax – softmax activation function.**

Sau khi đi qua các lớp được kết nối đầy đủ, lớp cuối cùng sử dụng chức năng kích hoạt softmax (thay vì ReLu) để nhận xác suất của đầu vào thuộc một lớp cụ thể (phân loại). Và cuối cùng, chúng ta có xác suất của đối tượng thuộc các lớp khác nhau.



* + - 1. **Mạng tương tác đặc trưng – Feature Pyramid Network (FPN).**

* + - 1. **Các cột mốc phát triển của lĩnh vực phát hiện đối tượng.**



* + - 1. Một số thuật toán trong lĩnh vực nhận phát hiện đối tượng trong thị giác máy tính.
  1. Các công nghệ sử dụng.

1. **XÂY DỰNG MÔ HÌNH HỌC SÂU.**
   1. Tổng quan về bài toán nhận diện thuốc lá điện tử.
   2. Thu thập dữ liệu.
   3. Chuẩn bị dữ liệu.
   4. Cấu trúc và kiến trúc mô hình Faster R-CNN.
   5. Lựa chọn thư viện và công cụ.
   6. Các tham số và thành phần chính của mô hình.
   7. Huấn luyện mô hình.
2. **PHÂN TÍCH THIẾT KẾ HỆ THỐNG.**
   1. Phân tích yêu cầu hệ thống.
   2. Đặc tả yêu cầu hệ thống.
   3. Thiết kế hệ thống gán nhãn.
3. **THỰC NGHIỆM, SO SÁNH VÀ ĐÁNH GIÁ.**
   1. Môi trường thực nghiệm – cấu hình.
   2. Kết quả thực nghiệm.
   3. So sánh với các phương pháp khác.
      1. Thuật toán Fast R-CNN.
   4. Đánh giá hiệu suất mô hình và phân tích kết quả.
   5. Kết luận.