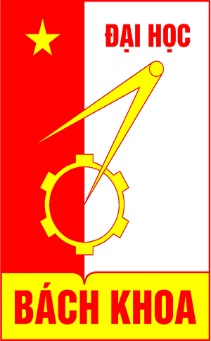
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Trường Công nghệ thông tin và Truyền thông**



**ĐỒ ÁN ĐỀ XUẤT**

**Nghiên cứu đánh giá tối ưu phương pháp nhận dạng biển số xe tự động**

**trên hệ thống nhúng**

**TRỊNH QUYẾT TIẾN**

tien.tq211262m@sis.hust.edu.vn

**Ngành Khoa học dữ liệu (Elitech)**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Phạm Ngọc Hưng |

**HÀ NỘI, 4/2023**

**MỤC LỤC**

**Danh mục hình vẽ**

**Danh mục bảng biểu**

**CHƯƠNG 1: GIỚI THIỆU**

**1.1. Giới thiệu đề tài**

Với sự phát triển nhanh chóng của kinh tế và xã hội, nước ta đang đối mặt với nhiều thách thức phức tạp, trong đó tình trạng giao thông quá tải và hỗn loạn đang là vấn đề nghiêm trọng tại các trung tâm kinh tế lớn của cả nước. Những tác động tiêu cực của vấn đề này đang gây ra những thiệt hại lớn cho nền kinh tế cũng như đời sống xã hội. Để giải quyết vấn đề này, cần có sự nâng cao ý thức và chấp hành giao thông của người dân, đồng thời cần thiết phải tập trung vào việc giám sát và quản lý giao thông.

Tuy nhiên, việc giám sát và quản lý giao thông hiện nay vẫn chủ yếu dựa trên phương pháp thủ công, trong khi số lượng phương tiện giao thông ngày càng tăng đòi hỏi cần có các giải pháp tự động hơn để giảm thiểu sức lao động của con người. Điều này đã được áp dụng từ lâu ở những nước phát triển và hiện nay, với yêu cầu đưa công nghệ vào cuộc sống và sản xuất, việc giải quyết bài toán giám sát giao thông tự động trở nên cực kỳ cấp bách.

Trong đó, bài toán nhận diện biển số xe là một trong những bài toán quan trọng để giúp quản lý phương tiện giao thông một cách hiệu quả. Tuy nhiên, để giải quyết bài toán này cần phải áp dụng các công nghệ mới như trí tuệ nhân tạo, học sâu, xử lý ảnh, và có những phương pháp đánh giá và xử lý dữ liệu để đạt được độ chính xác cao. Vì vậy, việc nghiên cứu và giải quyết bài toán này là cực kỳ cần thiết để đóng góp vào giải quyết vấn đề giao thông chung hiện nay.

Vấn đề phát hiện và nhận diện biển số xe là một trong những hướng nghiên cứu đã được quan tâm trong lĩnh vực thị giác máy tính từ lâu. Tuy nhiên, hiện nay với sự phát triển mạnh mẽ của phương pháp học sâu (deep learning), đã mở ra một hướng tiếp cận mới cho vấn đề này. Các mạng nơ-ron trong học sâu như Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) cùng với các kiến trúc mạng như LeNet[7], ImageNet [8], Fast R-CNN [9]... đang được phổ biến và được sử dụng trong nhiều ứng dụng thực tế. Đặc biệt, các kết quả đạt được thông qua các cuộc thi và nghiên cứu trong giới học thuật đều rất ấn tượng. Những phương pháp này giúp cho việc phát hiện và nhận diện biển số xe trở nên chính xác hơn, nhanh hơn và hiệu quả hơn. Việc áp dụng phương pháp học sâu này không chỉ giải quyết được bài toán giám sát giao thông tự động mà còn có thể ứng dụng trong nhiều lĩnh vực khác.

Vấn đề nhận diện biển số xe là một trong những hướng nghiên cứu quan trọng trong lĩnh vực thị giác máy tính. Tuy nhiên, nó cũng đang gặp phải một số khó khăn thách thức:

* Đầu tiên là các yếu tố do môi trường, chẳng hạn như độ phức tạp của khung cảnh chứa biển số xe, gây khó khăn cho việc phân biệt biển số xe với các đối tượng khác. Ngoài ra, độ sáng và độ nhạy sáng của các thiết bị cảm biến cũng ảnh hưởng đến kết quả phát hiện và nhận dạng.
* Các yếu tố do quá trình thu nhận hình ảnh gồm độ mờ, chất lượng ảnh thấp và độ biến dạng của ảnh. Việc căn chỉnh tiêu cự của máy quay và sự chuyển động của máy quay hoặc vật thể gây ra độ mờ. Hình ảnh hay video trải qua quá trình nén và giải nén cũng sẽ dẫn đến giảm chất lượng hình ảnh. Cuối cùng, với nhiều góc độ của máy quay, hình ảnh thu được sẽ bị biến dạng.
* Ngoài ra, quá trình huấn luyện mạng deep learning cần nhiều dữ liệu để đảm bảo tính tổng quát và độ chính xác tốt. Tuy nhiên, việc thu thập và gán nhãn số lượng lớn dữ liệu là công việc tốn nhiều thời gian và công sức.
* Việc huấn luyện mạng cũng tốn nhiều thời gian, đặc biệt là với các mạng lớn và phức tạp. Hơn nữa, để tìm được bộ siêu tham số phù hợp với kiến trúc mạng, ta phải thực hiện việc huấn luyện lại mạng với mỗi một tham số truyền vào.

Dựa vào khảo sát của các đề tài hiện nay về nhận diện biển số xe, có thể nhận thấy rằng vẫn còn một số vấn đề đáng chú ý.

* Một trong những giới hạn của các phương pháp nhận diện ký tự ứng dụng học sâu là khả năng phát hiện các ký tự có góc xoay bất kỳ. Đa phần các phương pháp hiện tại chỉ đạt được hiệu quả tối ưu với các góc xoay không quá lớn.
* Hơn nữa, đa phần các phương pháp sử dụng học sâu vẫn chưa được chứng minh là có thể áp dụng với biển số xe ở Việt Nam. Các dạng biển số xe mà các phương pháp hiện tại có thể nhận diện đều có kích thước tương đối đồng nhất và cách bố trí các ký tự không quá phức tạp, khác với biển số xe ở Việt Nam có độ phức tạp cao hơn và có nhiều dạng biển số khác nhau.
* Cuối cùng, tốc độ xử lý của các giải thuật hiện nay vẫn chưa đạt đến thời gian thực, tức là việc xử lý và nhận diện biển số xe vẫn tốn nhiều thời gian.
* Trước những khó khăn nêu trên, cần có sự cải tiến và kết hợp các phương pháp hiện tại để tạo ra một phương pháp nhận diện biển số xe hiệu quả, có thể giải quyết các vấn đề đang tồn tại. Đồng thời, đề tài cũng mong muốn có thể giải quyết được các vấn đề còn tồn tại của các đề tài đi trước đó.
* Trong bài toán nhận dạng biển số xe, thời gian phản hồi nhanh và độ chính xác cao là rất quan trọng để đảm bảo an toàn giao thông và quản lý tốt việc điều khiển phương tiện. Sử dụng hệ thống edge computing có thể cải thiện các yếu tố này bởi vì nó cho phép xử lý dữ liệu cục bộ tại nguồn phát sinh mà không phải truyền dữ liệu tới một trung tâm xử lý tập trung (centralized processing center) ở xa. Kết quả là giảm thiểu độ trễ và tăng tốc độ xử lý, đồng thời giảm tải băng thông mạng và chi phí lưu trữ dữ liệu. Hơn nữa, hệ thống edge computing có thể hoạt động độc lập và đảm bảo tính sẵn sàng cao, ngay cả khi kết nối mạng bị gián đoạn hoặc mất mát. Từ đó, việc áp dụng hệ thống edge computing vào bài toán nhận dạng biển số xe sẽ giúp cải thiện hiệu suất và độ tin cậy của hệ thống.

**1.2. Mục tiêu của đề tài**

Đề tài này tập trung vào việc đánh giá và cải thiện phương pháp nhận dạng biển số xe tự động trên hệ thống nhúng. Bằng việc kết hợp các kỹ thuật xử lý ảnh, mục tiêu là tạo ra một hệ thống nhận dạng biển số xe tự động có độ chính xác cao và tốc độ xử lý nhanh hơn so với các phương pháp khác, đạt độ chính xác trên 90% cho việc phát hiện biển số và trên 80% cho việc nhận diện chuỗi biển số, đồng thời cải thiện thời gian xử lý của mô hình để đảm bảo thực hiện ở thời gian thực. Đề tài này sẽ đưa ra các kết quả thực nghiệm và đánh giá hiệu quả của phương pháp được đề xuất.

**1.3. Ý nghĩa của đề tài**

***1.3.1. Ý nghĩa thực tiễn***

Phương pháp được đề xuất trong đề tài có thể áp dụng trong nhiều ứng dụng, như giám sát giao thông tự động, bãi giữ xe thông minh, trạm thu phí tự động và nhiều ứng dụng khác. Sử dụng phương pháp này có thể giúp giảm thời gian và công sức của con người, đồng thời giảm tình trạng kẹt xe và tăng sự tiện lợi cho người tham gia giao thông. Một trong những yếu tố quan trọng để đạt được hiệu quả cao của phương pháp là xây dựng được tập dữ liệu chính xác, đa dạng và phù hợp với các điều kiện thực tế tại Việt Nam. Các ứng dụng được phát triển từ phương pháp này sẽ mang lại lợi ích cho cả xã hội và góp phần nâng cao chất lượng cuộc sống của người dân**.**

***1.3.2. Ý nghĩa khoa học***

Đề tài đóng góp một phương pháp mới trong việc phát hiện và nhận diện biển số xe hiệu quả, với mục tiêu tăng độ chính xác và giải quyết các vấn đề còn tồn tại của các phương pháp đi trước. Nghiên cứu này có thể cung cấp cơ sở cho các nghiên cứu trong nước về sau, đặc biệt là trong lĩnh vực ứng dụng các hệ thống nhúng. Tập dữ liệu chính xác, đa dạng và sát với điều kiện thực tế của nước ta cũng được xem là một đóng góp đáng kể của nghiên cứu này.

***1.4. Phạm vi của đề tài***

Trong đề tài, tập dữ liệu được sử dụng để huấn luyện và kiểm tra mô hình chỉ bao gồm các hình ảnh biển số xe được chụp theo phương ngang, không bị che khuất, không bị hư hỏng, tróc sơn, rỉ sét và có độ mờ và độ biến dạng thấp. Vị trí của camera để thu thập hình ảnh là cố định hoặc di động với góc lệch không quá lớn. Hình ảnh có thể bị nhiễu từ thời tiết như mưa, sương mù, vv., nhưng vẫn có thể nhận dạng được đầy đủ các ký tự bằng mắt thường. Trong đề tài, chỉ sử dụng các biển số xe máy và ô tô phổ biến tại Việt Nam, tuân thủ theo Thông tư 58/2020/TT-BCA về kích thước, ký hiệu và bố trí của biển số xe. Các biển số xe của các nước khác sẽ không được xem xét trong đề tài.

***1.5 Bố cục luận văn***

Báo cáo này gồm 4 chương, trong đó Chương 2 tập trung vào giới thiệu các nghiên cứu liên quan đến bài toán phát hiện và nhận dạng biển số xe, cùng với các phương pháp giải quyết được áp dụng trong lĩnh vực này, đồng thời cung cấp những kiến thức cơ bản để đặt nền móng cho các mô hình được đề xuất trong Chương 3 bao gồm thông tin về các thí nghiệm thực hiện, các chỉ tiêu đánh giá và kết quả đánh giá của các phương pháp, đồng thời cũng đưa ra một số nhận xét dựa trên kết quả đạt được. Chương 4 đưa ra kết luận chung của báo cáo và trình bày những hướng phát triển tiềm năng trong tương lai, kèm theo danh sách các tài liệu tham khảo được sử dụng trong đề tài.

**CHƯƠNG II: CÁC NGHIÊN CỨU LIÊN QUAN VÀ CƠ SỞ LÝ THUYẾT**

Bài toán nhận dạng biển số xe là một trong những bài toán quan trọng trong lĩnh vực nhận diện văn bản ngoại cảnh. Vì vậy, các nghiên cứu liên quan đến hai bài toán này thường có sự tương quan mật thiết. Trong những năm gần đây, bài toán nhận diện biển số xe đã được quan tâm nhiều và có sự phát triển đáng kể. Thông thường, bài toán nhận dạng biển số xe bao gồm một số bước cơ bản như sau:

Untitled

Các bước cơ bản của một hệ thống nhận diện biển số xe

Từ những bước trên, có ba bài toán chính cần được giải quyết trong việc nhận dạng biển số xe: phát hiện vị trí của biển số trên hình, phân đoạn các ký tự trong biển số và nhận diện chuỗi ký tự trên biển số. Dựa trên những nghiên cứu mới đây, đề tài nhận thấy rằng hướng tiếp cận nhận dạng chuỗi ký tự trên biển số mà không phải qua bước phân đoạn ký tự rất tiềm năng, vì có nhiều ưu điểm hơn. Do đó, đề tài tập trung vào khảo sát các công trình liên quan đến phát hiện vị trí của biển số và nhận diện ký tự trên biển số. Ngoài ra, trong số các nghiên cứu gần đây còn đề cập đến việc kết hợp xử lý cả hai quá trình phát hiện và nhận dạng một cách đồng thời, tạo ra một phương pháp tổng thể hiệu quả hơn để giải quyết bài toán này.

**2.1 Phát hiện biển số xe**

**2.1.1 Các phương pháp trích xuất đặc trưng thủ công**

Trong quá khứ, các phương pháp trích đặc trưng thủ công được sử dụng phổ biến trong việc phân tích ảnh. Những đặc trưng thường được sử dụng bao gồm các đặc trưng cơ bản như góc, cạnh, vân ảnh, màu sắc và mức sáng. Các phương pháp phổ biến để trích xuất các đặc trưng này bao gồm việc nhị phân hóa ảnh, sử dụng thuật toán SIFT, HOG, phân tích thành phần liên thông và hình thái học. Sau đó, các đặc trưng này được đưa qua các bộ phân loại như AdaBoost, SVM để phân loại.

Các phương pháp trích xuất đặc trưng cơ bản thường đơn giản, dễ hiện thực và nhanh chóng, đặc biệt là khi kết hợp nhiều đặc trưng với nhau, kết quả có thể đạt được độ chính xác tương đối cao. Tuy nhiên, các phương pháp này có nhược điểm là khá nhạy cảm với sự thay đổi mức sáng, nhiễu và mờ. Hơn nữa, việc thiết kế bộ rút trích đặc trưng còn phụ thuộc khá nhiều vào tập dữ liệu, do đó cần phải đảm bảo tập dữ liệu đủ đa dạng để đảm bảo tính chính xác của phương pháp trích xuất.  
  
Dưới đây là một số bài báo sử dụng phương pháp trích xuất đặc trưng thủ công trong lĩnh vực xử lý hình ảnh:  
  
- "Vehicle License Plate Recognition Using Edge-Based Connected Component Analysis":

* Tác giả: V. R. Uthariaraj, S. Baskar, and P. G. Raj.
* Năm: 2013.
* Tạp chí: International Journal of Computer Applications
* Tóm tắt: Phương pháp trong paper này tập trung vào việc sử dụng phân tích thành phần liên thông dựa trên cạnh để nhận dạng biển số xe. Cụ thể, nó sử dụng các kỹ thuật xử lý ảnh để phát hiện và xác định vị trí của biển số xe trong hình ảnh. Sau khi xác định vùng chứa biển số, các đặc trưng thủ công như cạnh và hình dạng được sử dụng để nhận dạng ký tự trên biển số. Phương pháp này đơn giản và dễ triển khai, tuy nhiên, nó có thể không đủ robust đối với biến đổi mức sáng và nhiễu mạnh và không hiệu quả trong các tình huống phức tạp hơn.
* Ưu điểm:
  + Sử dụng phân tích thành phần liên thông dựa trên cạnh để nhận dạng biển số xe, giúp xác định vùng chứa biển số.
  + Đơn giản và dễ triển khai.
  + Hiệu suất tương đối tốt trong các tình huống ổn định về ánh sáng và nhiễu.
* Nhược điểm:
  + HOG có thể nhạy cảm với thay đổi ánh sáng và góc nhìn trong hình ảnh.
  + Yêu cầu một lượng lớn dữ liệu huấn luyện để đảm bảo tính chính xác.
  + Không thể tự động học các đặc trưng mà phải được thiết lập thủ công.

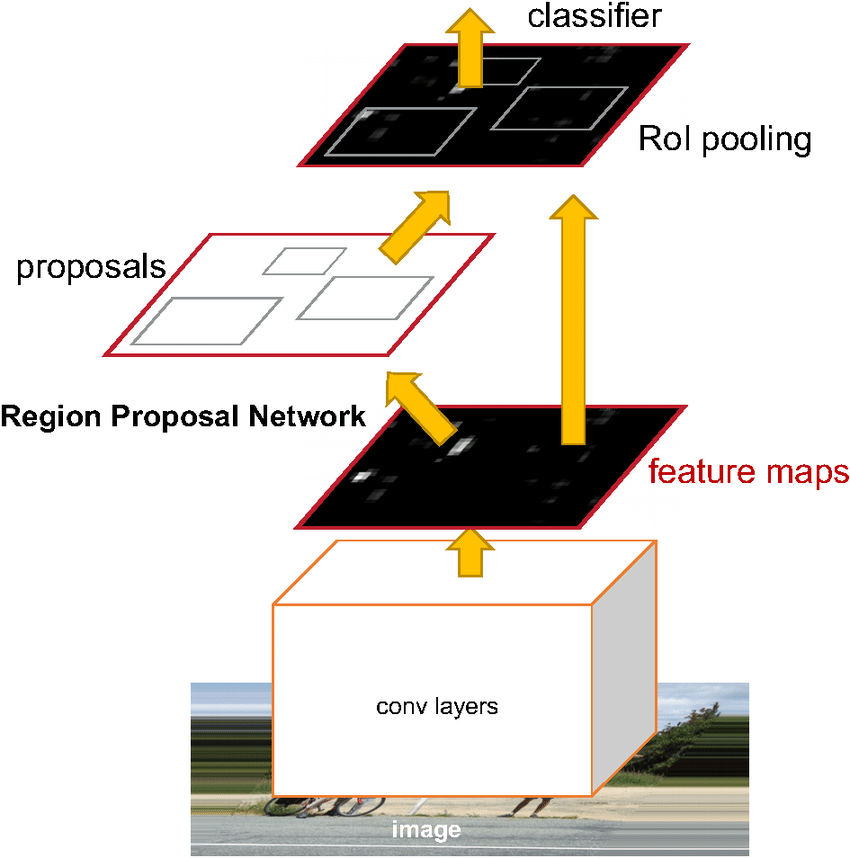
**2.1.2 Các phương pháp học sâu**

Hiện nay, phương pháp học sâu được sử dụng để phát hiện đối tượng rất đa dạng và hiệu quả. Có nhiều kiến trúc mạng được áp dụng, ví dụ như Faster RCNN, SSD, YOLO và YOLO2. Các kiến trúc này tỏ ra hiệu quả trong việc phát hiện nhiều đối tượng trong cùng một ảnh.

**Mạng Faster RCNN**

Kiến trúc mạng Faster RCNN được xây dựng bao gồm 3 phần chính. Phần đầu tiên là một mạng rút trích đặc trưng cơ bản, phần thứ hai là lớp mạng đề xuất vùng dự tuyển Region Proposal Network (RPN) và phần cuối cùng là lớp mạng dự đoán vị trí và phân loại đối tượng. Khi ảnh đầu vào được đưa qua phần mạng rút trích đặc trưng, nó sẽ cho ra một bản đồ đặc trưng (feature map). Sau đó, bản đồ đặc trưng này sẽ được đưa qua lớp mạng RPN để tạo ra các vùng dự tuyển. Các vùng dự tuyển sẽ được đưa về cùng kích thước thông qua lớp Roi Pooling. Sau đó, việc phân loại đối tượng cũng như dự đoán vị trí sẽ được thực hiện trên các vùng dự tuyển này.

Cải tiến lớn nhất của Faster RCNN so với mạng Fast RCNN đến từ việc sử dụng lớp mạng RPN để thay thế cho giải thuật tìm kiếm vét cạn. Điều này đã cải thiện đáng kể tốc độ chạy của mô hình.



Kiến trúc mạng Faster RCNN

Mạng Faster RCNN có nhiều ưu điểm, trong đó đáng chú ý nhất là tốc độ và độ chính xác được cải thiện đáng kể so với phiên bản trước đó. Tuy nhiên, một nhược điểm của kiến trúc này là việc sử dụng một lớp bản đồ đặc trưng có thể dẫn đến việc bỏ sót các đối tượng nhỏ và do đó giảm độ chính xác so với các phương pháp học sâu khác.

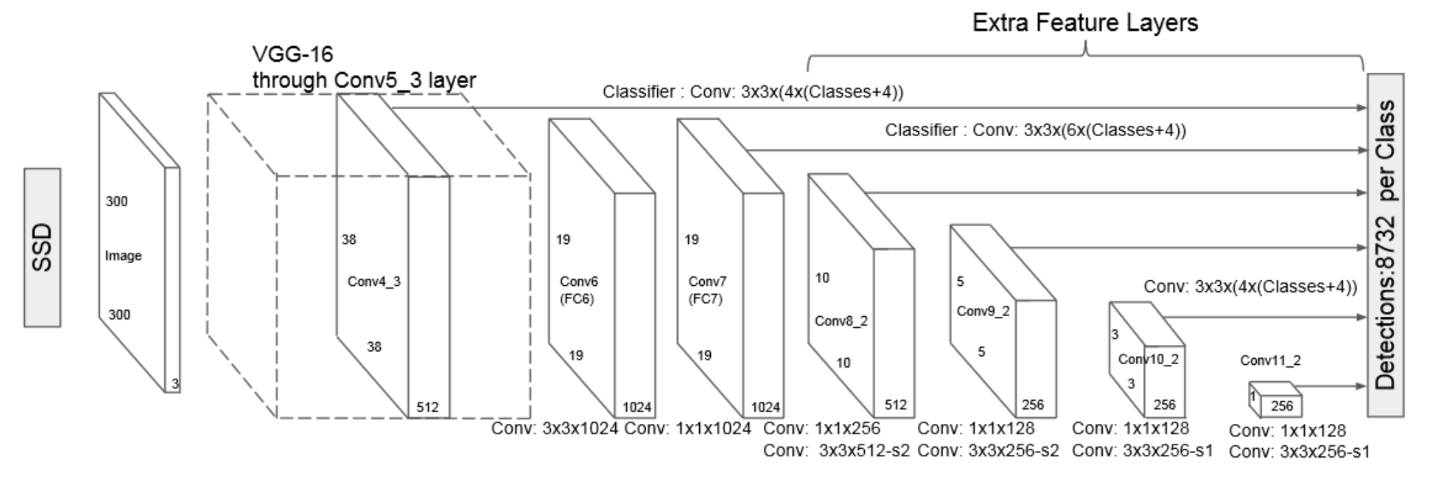
**Mạng Single Shot Multibox Detector (SSD)**

Mạng Single Shot Multibox Detector (SSD) là một mạng học sâu được thiết kế để phát hiện và phân loại đối tượng trong hình ảnh. Khác với phương pháp truyền thống sử dụng cửa sổ trượt với kích thước cố định, SSD tạo ra một tập hợp các ô chuẩn (default box) để xác định vị trí và lớp của các đối tượng trong quá trình huấn luyện. Việc sử dụng các ô chuẩn giúp cho mạng có khả năng dự đoán kích thước của các đường bao chữ nhật quanh vị trí của đối tượng một cách chính xác.

SSD áp dụng các ô chuẩn trên nhiều lớp bản đồ đặc trưng với các kích thước khác nhau để phát hiện được những đối tượng có kích thước lớn hoặc nhỏ khác nhau. Điều này giúp cho SSD có khả năng phát hiện đối tượng một cách hiệu quả và đồng thời giảm thiểu số lượng các phép tính cần thiết để xác định vị trí và lớp của đối tượng.

Mạng SSD được thiết kế với kiến trúc đầy đủ từ lớp mạng nền cho tới các lớp dự đoán vị trí và phân loại. Điều này giúp cho mạng có khả năng tích hợp việc phân loại và phát hiện vị trí trong cùng một mạng, tăng độ chính xác và giảm thời gian xử lý so với phương pháp truyền thống.

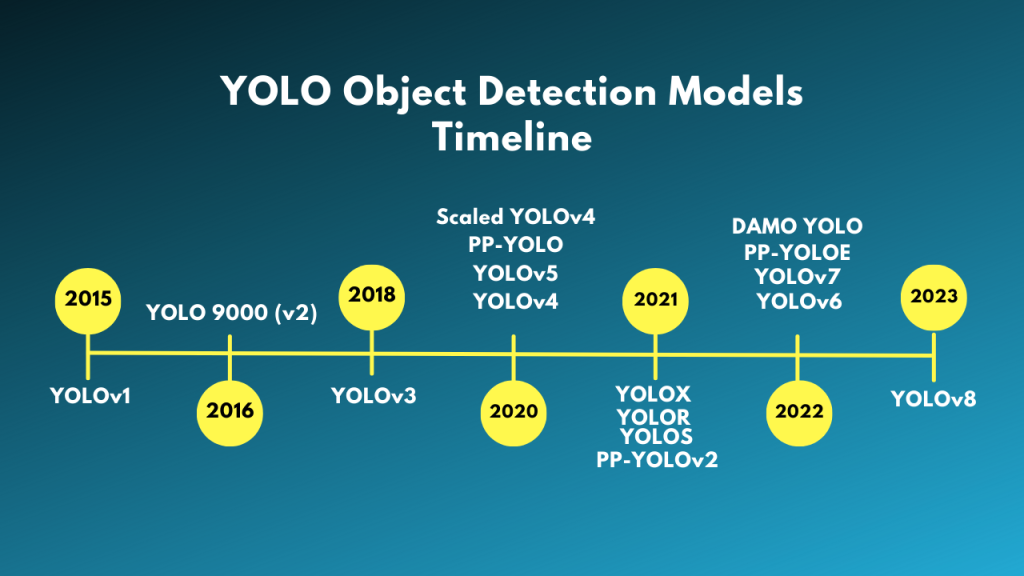
Tuy nhiên, một điểm hạn chế của SSD là số lượng các ô chuẩn lớn có thể gây khó khăn trong quá trình huấn luyện do đòi hỏi nhiều bộ nhớ hơn.



Kiến trúc mạng SSD

**Mô hình YOLO (You Only Look Once)**

Mô hình YOLO (You Only Look Once) là một mô hình học sâu được sử dụng cho việc nhận diện đối tượng trong ảnh và video. Nó được thiết kế để nhận diện và phân loại các đối tượng đồng thời trong một khung hình. Mô hình YOLO sử dụng một mạng nơ-ron tích chập để xác định vị trí, kích thước và loại của các đối tượng trong ảnh và đưa ra các dự đoán trong thời gian thực. Với độ chính xác cao và khả năng xử lý nhanh, YOLO đã trở thành một trong những mô hình phổ biến nhất trong lĩnh vực nhận diện đối tượng. YOLO xuất hiện từ năm 2015, đã phát triển qua nhất nhiều phiên bản, chúng ta có thể nhìn thấy các phiên bản như trên hình.



Sự phát triển của YOLO

* YOLOv1: Phiên bản đầu tiên của mô hình YOLO, đó là YOLOv1 đã được xuất bản bởi Joseph Redmon et al. vào năm 2015. Đây là mô hình phát hiện đối tượng (SSD - Single Stage object Detection) single stage đầu tiên đã tạo ra các mô hình SSD và tất cả các mô hình YOLO tiếp theo.
* YOLOv2: còn được gọi là YOLO 9000 được xuất bản bởi tác giả gốc của YOLOv1, Joseph Redmon. Nó đã cải thiện YOLOv1 bằng cách giới thiệu khái niệm anchor boxes và backbone tốt hơn, đó là Darknet-19.
* YOLOv3: Năm 2018, Joseph Redmon và Ali Farhadi đã xuất bản YOLOv3. YOLOv3 sử dụng backbone Darknet-53, loại bỏ các kết nối dư thừa, pretrain tốt hơn và các kỹ thuật tăng cường hình ảnh để đem lại những cải tiến.
* Ultralytics YOLO Object Detection Models: Tất cả các mô hình phát hiện đối tượng YOLO cho đến YOLOv3 đều được viết bằng ngôn ngữ lập trình C và sử dụng framework Darknet. Những người mới học sẽ gặp khó khăn khi đọc code và tinh chỉnh các mô hình. Cùng khoảng thời gian với YOLOv3, Ultralytics đã phát hành mô hình YOLO (YOLOv3) đầu tiên được triển khai bằng framework PyTorch. Nó cũng dễ tiếp cận và dễ sử dụng hơn cho việc transfer learning. Ngay sau khi xuất bản YOLOv3, Joseph Redmon đã rời khỏi cộng đồng nghiên cứu Thị giác máy tính. YOLOv4 (của Alexey và cộng sự) là mô hình YOLO cuối cùng được viết trên Darknet. Sau đó đã có rất nhiều mô hình YOLO khác nhau. Trong đó YOLOv4, YOLOX, PP-YOLO, YOLOv6 và YOLOv7 là một số mô hình nổi bật. Sau YOLOv3, Ultralytics cũng phát hành YOLOv5 thậm chí còn tốt hơn, nhanh hơn và dễ sử dụng hơn tất cả các mô hình YOLO khác. Tính đến thời điểm hiện tại, Ultralytics đã xuất bản YOLOv8, đây có lẽ là mô hình YOLO tốt nhất cho đến nay.

**YOLOv8**

* Sử dụng kiến trúc mạng nơ-ron tích chập (CNN) mới nhất, bao gồm các khối ResNet, CSP và SPP, để tăng độ chính xác và giảm thời gian huấn luyện.
* Áp dụng kỹ thuật học chuyển tiếp (transfer learning) để tận dụng các trọng số được huấn luyện trước trên các bộ dữ liệu lớn như COCO và ImageNet, giúp tiết kiệm thời gian và tài nguyên.
* Hỗ trợ nhiều định dạng đầu vào và đầu ra khác nhau, bao gồm ảnh, video, webcam, RTSP và ONNX. Ngoài ra, YOLOv8 cũng có thể xuất ra các kết quả dưới dạng JSON, CSV hoặc PASCAL VOC.
* Cung cấp nhiều tùy chọn để tùy biến mô hình theo nhu cầu của người dùng, bao gồm việc thay đổi kích thước ảnh, số lượng lớp, ngưỡng tin cậy và tốc độ khung hình.
* Có thể phát hiện và phân đoạn các đối tượng trong ảnh một cách tự động, sử dụng các phương pháp như phát hiện biên giới (boundary detection), phân vùng ngữ nghĩa (semantic segmentation) và phân vùng thể hiện (instance segmentation).

Ưu điểm:

* Có thể phát hiện đối tượng nhanh chóng và chính xác, với tốc độ lên đến 140 khung hình/giây trên GPU và 3 khung hình/giây trên CPU.

Nhược điểm:

* Cần nhiều tài nguyên để huấn luyện và chạy mô hình, do đó có thể không phù hợp cho các thiết bị nhúng hoặc có dung lượng thấp.
* Có thể gặp khó khăn khi phát hiện các đối tượng nhỏ, che khuất hoặc có hình dạng không đều.
* Có thể bị ảnh hưởng bởi các yếu tố ngoại cảnh như ánh sáng, nhiễu hoặc chuyển động.

**2.2. Nhận diện biển số xe**

**2.2.1. Phương pháp so trùng mẫu (Template matching)**

Phương pháp so trùng mẫu được sử dụng để phân loại các đối tượng dựa trên sự tương tự giữa chúng và một mẫu được cho trước. Trong bài toán nhận diện biển số xe, mẫu được sử dụng là một tập hợp các ký tự. Phương pháp này thường được áp dụng trên ảnh xám hoặc ảnh nhị phân.

Các độ đo khoảng cách như khoảng cách Mahalanobis, khoảng cách Jaccard và khoảng cách Hamming được sử dụng để so sánh các đối tượng với mẫu. Độ chính xác của phương pháp này phụ thuộc vào độ tương đồng giữa đối tượng và mẫu.

Mặc dù phương pháp so trùng mẫu đơn giản và dễ thực hiện, nhưng nó cũng có những hạn chế. Trong thực tế, kích thước của các ký tự phải cố định để áp dụng phương pháp này. Ngoài ra, phương pháp này cũng rất nhạy cảm với nhiễu, sự thay đổi mức sáng và góc quay của đối tượng. Vì vậy, việc áp dụng phương pháp này trong các bài toán thực tế có thể gặp khó khăn.

**2.2.2 Phương pháp học sâu**

Các phương pháp nhận diện biển số xe thường hoạt động trên tập dữ liệu là hình ảnh biển số xe đã được cắt sẵn. Tập dữ liệu này chỉ chứa thông tin về chuỗi ký tự trên biển số xe mà không có thông tin về vị trí của từng ký tự trong hình. Ngoài ra, các chuỗi ký tự này thường chỉ nằm trên một hàng.

***Mạng ConvNet-RNN [4]***

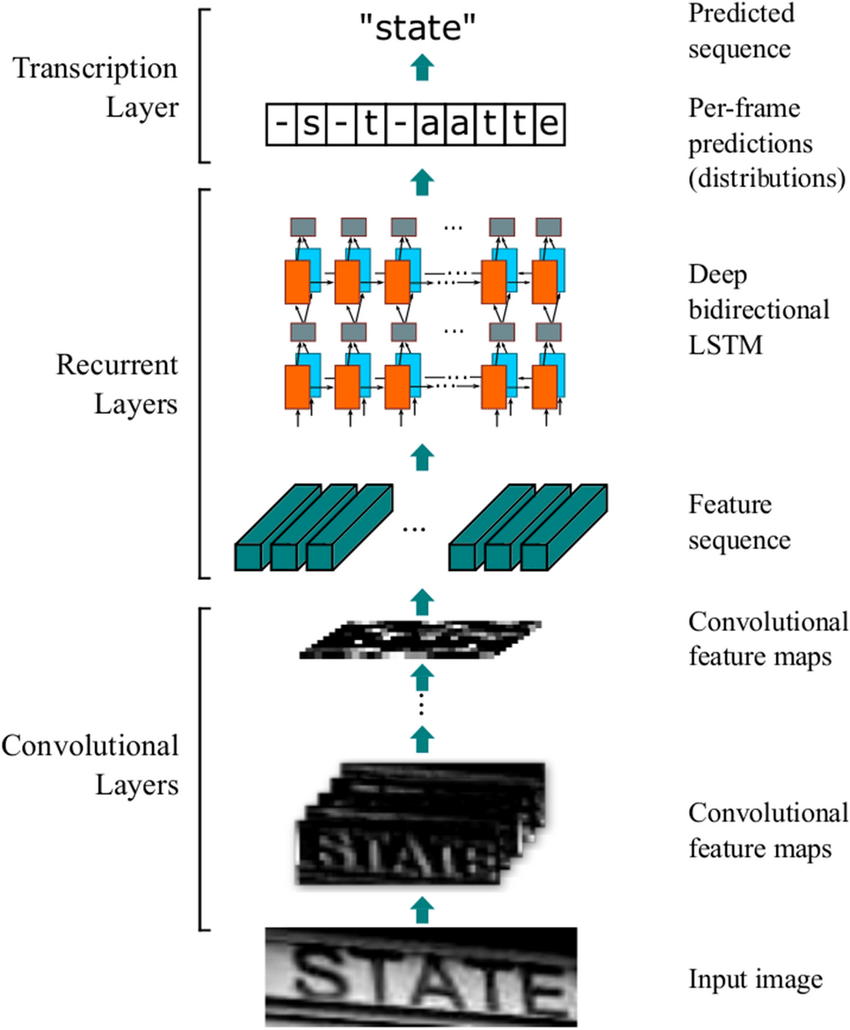
Phương pháp nhận diện biển số xe sử dụng kiến trúc mạng gồm lớp mạng VGG [14] và lớp mạng RNN. Đầu tiên, ảnh đầu vào được đưa qua lớp mạng VGG [14] để trích xuất đặc trưng. Sau đó, các đặc trưng được chuyển thành các vector và đưa vào lớp RNN để nhận dạng các ký tự tương ứng.

Phương pháp này có hiệu quả tốt hơn so với phương pháp cửa sổ trượt và không yêu cầu phân đoạn dữ liệu huấn luyện trước. Tuy nhiên, phương pháp này có hạn chế là tỷ lệ nhận dạng sai khá cao cho các cặp ký tự M và N, D và Q, T và Y, C và G.

***Mạng CRNN [1]***

Phương pháp này sử dụng hai lớp mạng chính: một lớp mạng neural network tích chập (CNN) để trích xuất đặc trưng và một lớp mạng Long-Short Term Memory (LSTM) cho việc nhận dạng các ký tự. Lớp mạng LSTM được thiết kế để chạy độc lập theo hai chiều (gọi là Bidirectional LSTM hoặc BiLSTM). Đầu vào cho lớp BiLSTM là bản đồ đặc trưng được biến đổi từ đầu ra của lớp CNN. Cuối cùng, lớp BiLSTM được kết nối với hàm lỗi Connectionist Temporal Classification (CTC), cho phép chuyển đổi các vector đặc trưng thành chuỗi các xác suất và tìm ra được chuỗi có tổng xác suất là lớn nhất. Hàm lỗi CTC được sử dụng thay cho giải thuật Hidden Markov Model (HMM), và được cho là hiệu quả hơn HMM hoặc HMM kết hợp học sâu.

Phương pháp này có tính tổng quát tương đối cao và có thể áp dụng cho các văn bản ngoại cảnh, bản nhạc và các tập dữ liệu khác. Nó chỉ sử dụng tập dữ liệu sinh tự động để huấn luyện nhưng vẫn đạt được kết quả tốt trên các tập dữ liệu thực khác. Tuy nhiên, nhược điểm của phương pháp này là chỉ có thể nhận dạng được chuỗi dữ liệu trên cùng một hàng.



Mạng CRNN

**2.3 Mô hình kết hợp phát hiện và nhận diện biển số xe**

**2.3.1. Hướng tuần tự**

Hướng tuần tự là một phương pháp trong việc xử lý ảnh mà có sự tách biệt giữa quá trình phát hiện và quá trình nhận diện. Trong đó quá trình phát hiện và quá trình nhận diện được thực hiện riêng biệt, nhưng có thể có sự tương tác giữa chúng. Phương pháp của Masood và đồng sự [17]: Phương pháp này sử dụng 3 mạng neural convolutional (CNN) để thực hiện quá trình phát hiện và nhận dạng biển số xe. Mạng đầu tiên được sử dụng để phát hiện và phân loại biển số xe, mạng thứ hai được sử dụng để phát hiện các ký tự trên biển số, và mạng cuối cùng được sử dụng để nhận dạng các ký tự đó. Phương pháp này đã đạt được độ chính xác khá cao trên các tập dữ liệu biển số xe của Mỹ và châu Âu, lần lượt là 93,44% và 94,55%. Tuy nhiên, mô hình mà phương pháp này đề xuất khá lớn với 3 mạng neural tách biệt, dẫn đến thời gian huấn luyện sẽ tốn nhiều thời gian.

Phương pháp của Li Hui và đồng sự [18]: Phương pháp này áp dụng một mạng phân loại CNN để tạo ra bản đồ xác suất của các pixel có khả năng là ký tự trong biển số xe. Bản đồ xác suất này sau đó được gom nhóm lại bằng giải thuật Non-Maxima Suppression (NMS) và Run-Length Smoothing Algorithm (RLSA) để tạo ra các nhóm pixel liên quan đến các ký tự trong biển số xe. Từ các nhóm pixel này, đường bao của biển số xe được tạo ra thông qua giải thuật phân tích các thành phần liên kết (CCA). Sau đó, vùng biển số được trích xuất từ các nhóm pixel này và đưa qua mô hình tương tự như phương pháp [1] để nhận diện chuỗi biển số xe. Ưu điểm của phương pháp này là đạt được độ chính xác khá cao so với các phương pháp trước đó với tỷ lệ 97.56%. Tuy nhiên, bước trích xuất biển số xe vẫn phải được thực hiện bằng cách thủ công, do đó có thể dẫn đến sai sót và ảnh hưởng đến độ chính xác của bước nhận dạng phía sau.

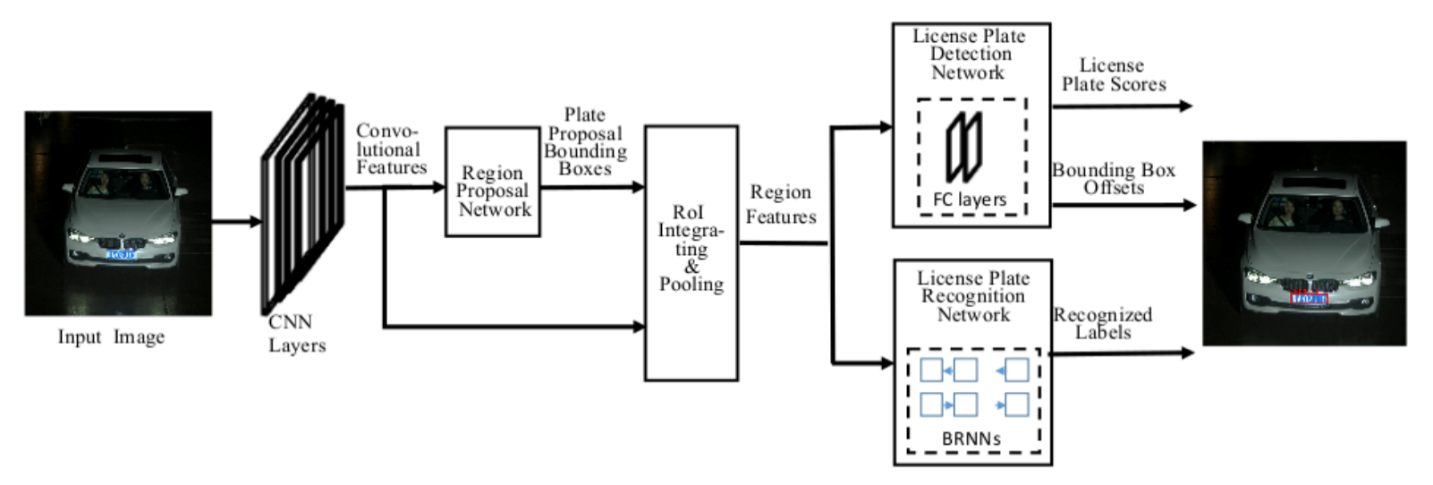
**2.3.2 Hướng tích hợp**

Hướng tích hợp là hướng tiếp cận trong xử lý ảnh mà việc phát hiện và nhận diện đối tượng được kết hợp lại để cùng hoạt động trên cùng một mạng. Trong hướng này, thông tin giữa quá trình phát hiện và nhận diện được chia sẻ thông qua bộ phân loại trung gian, giúp cải thiện độ chính xác và tốc độ xử lý.

Một số phương pháp tích hợp sử dụng bước tiền xử lý để xác định vùng quan tâm (ROI - Region of Interest) trước khi thực hiện phát hiện và nhận diện. Quá trình này giúp tập trung phân tích vào vùng quan tâm để tăng tốc độ xử lý và giảm độ phức tạp của mô hình. Tuy nhiên, việc xác định ROI đòi hỏi kiến thức chuyên môn và kinh nghiệm để đảm bảo độ chính xác và tránh sai sót trong quá trình xử lý.

Phương pháp của Li Hui và đồng sự [5]: Phương pháp này sử dụng kiến trúc mạng duy nhất bao gồm các phần: lớp mạng trích xuất đặc trưng cấp thấp, tạo vùng chứa biển số, xử lý vùng chứa biển số, phát hiện và nhận dạng biển số xe. Phương pháp này lấy ý tưởng kết hợp các mạng có sẵn thành một mạng duy nhất để giảm thiểu các bước trung gian, đặc biệt là bước phân đoạn ký tự. Nó hoạt động tốt trong các điều kiện tự nhiên như ban ngày, ban đêm, mưa và nắng.

Tuy nhiên, quá trình nhận dạng ký tự chỉ hoạt động với chuỗi ký tự trên một hàng. Chiến thuật huấn luyện tương đối phức tạp và yêu cầu một lượng lớn dữ liệu. Do đó, phương pháp này đòi hỏi sự đầu tư nghiêm túc và chú ý đến các thách thức về tính toàn vẹn và độ chính xác của dữ liệu.



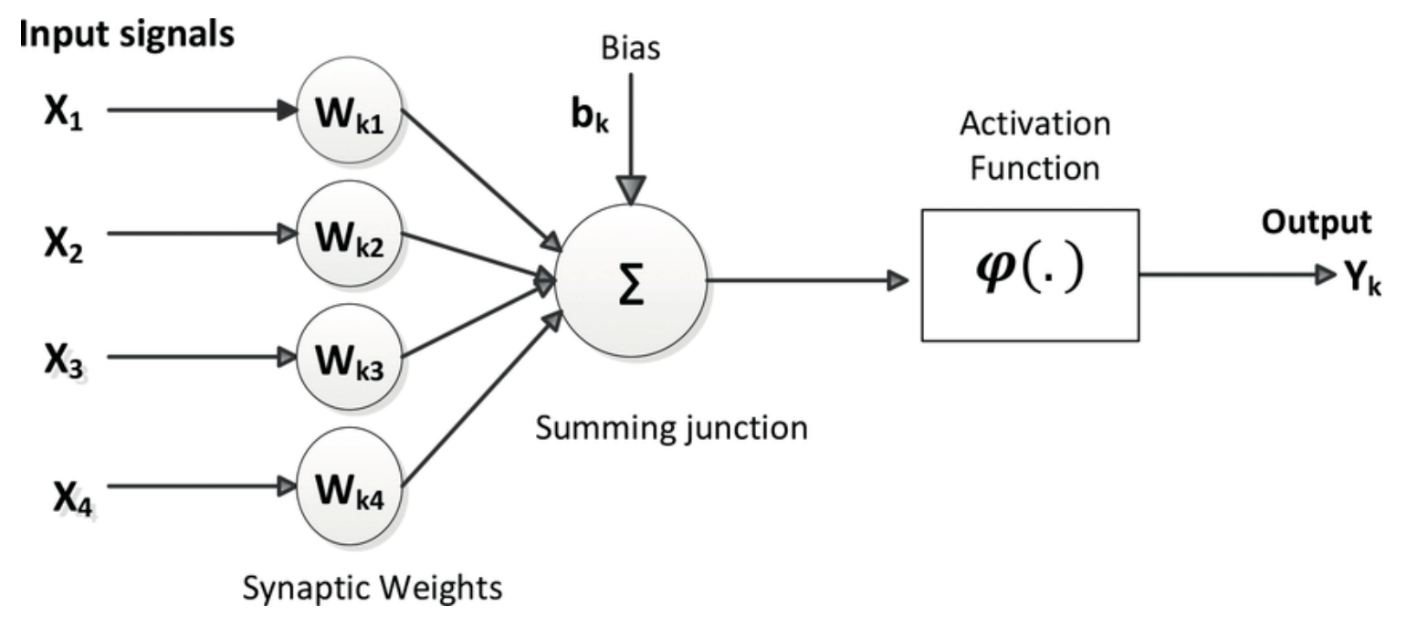
Kiến trúc tổng quan của mô hình

**2.4. Cơ sở lý thuyết**

Phần này giải thích các kiến thức nền tảng liên quan đến phương pháp học sâu, đây là cơ sở để xây dựng các phương pháp mới được đề xuất.

**2.4.1. Mạng nơ-ron truyền thẳng**

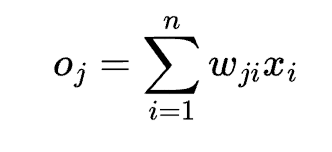
Các kiến trúc mạng học sâu hiện nay đều được bắt nguồn từ mạng nơ-ron truyền thẳng. Mạng nơ-ron truyền thẳng có thể được coi là một hàm số phi tuyến tính , trong đó dữ liệu đầu vào là một vector , và nhãn tương ứng của dữ liệu đó là . Cấu trúc của mạng nơ-ron truyền thẳng bao gồm nhiều đơn vị nơ-ron kết nối với nhau theo thứ tự.



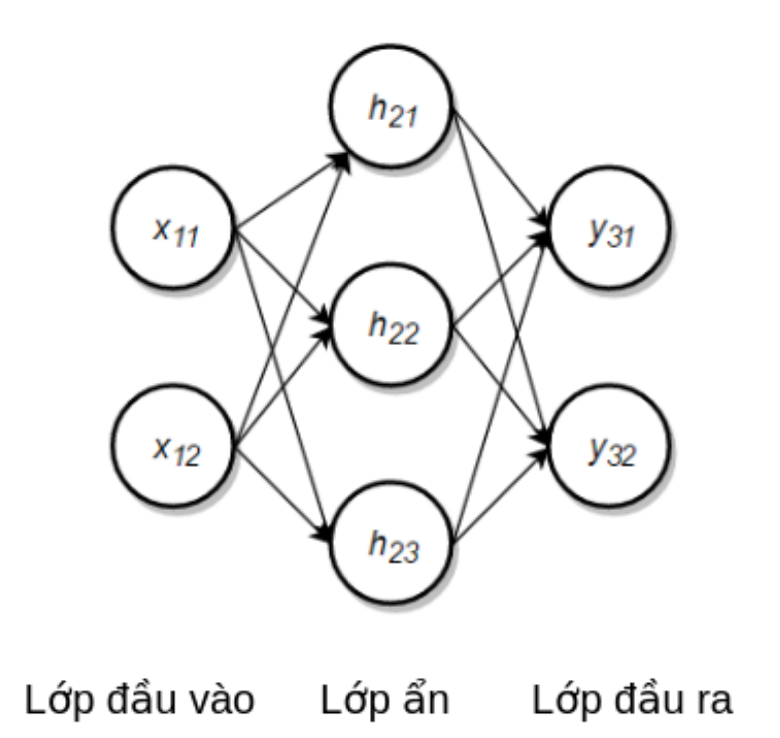
*Cấu trúc một nơ-ron*

**3.1.1 Hàm kích hoạt**

Một đơn vị tính toán bao gồm các thành phần sau: vector đầu vào , ma trận trọng số (weights) , hệ số bias , hàm tính tổng trọng số và hàm kích hoạt. Hàm tính tổng trọng số được xác định bằng công thức sau đây:



Sau khi nhận được các giá trị đầu vào từ lớp trước, mỗi nơ-ron trong lớp hiện tại sẽ tính toán tổng trọng số của các giá trị đầu vào nhân với các trọng số tương ứng. Tổng trọng số này sẽ được cộng thêm một hệ số điều chỉnh (bias) để có được giá trị tổng hợp (net input). Giá trị tổng hợp này sẽ được đưa qua một hàm kích hoạt phi tuyến để biến đổi nó thành một giá trị đầu ra trong một khoảng nhất định. Giá trị đầu ra này sẽ được gửi tới các nơ-ron trong lớp tiếp theo nếu như vẫn còn những lớp mạng phía sau. Một mạng nơ-ron truyền thẳng thông thường có cấu trúc như hình bao gồm: một lớp đầu vào, một hay nhiều lớp ẩn (hidden), và một lớp đầu ra. Lớp đầu vào là lớp nhận các dữ liệu từ bên ngoài, lớp ẩn là lớp thực hiện các tính toán và biến đổi dữ liệu, và lớp đầu ra là lớp xuất ra kết quả cuối cùng của mạng.

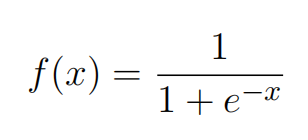


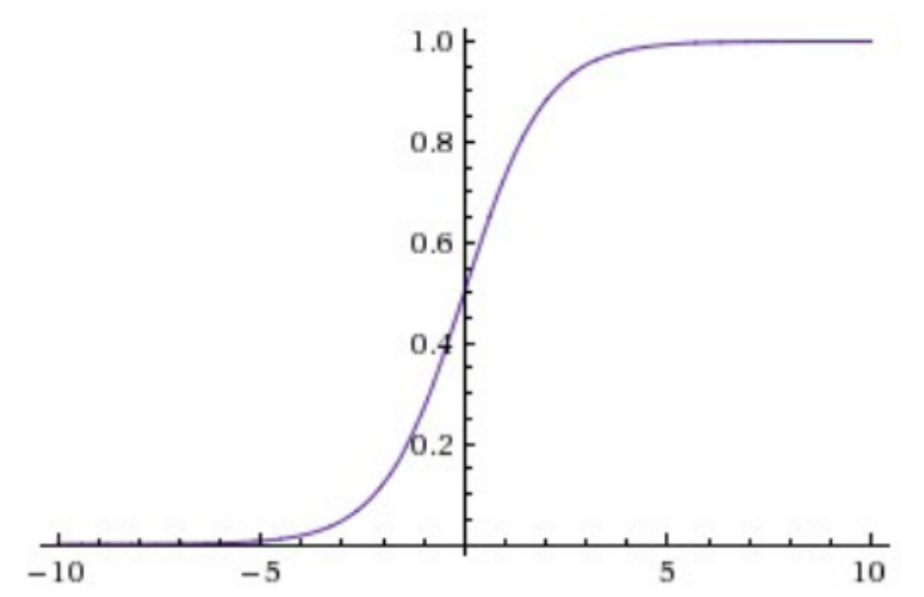
Cấu trúc mạng truyền thẳng

**3.1.1 Hàm kích hoạt**

Như đã nói ở trên, một yếu tố quan trọng trong mạng nơ-ron đó là hàm kích hoạt (activation function). Hàm kích hoạt có chức năng biến đổi giá trị tổng hợp của mỗi nơ-ron thành một giá trị đầu ra trong một khoảng nhất định. Hàm kích hoạt cũng giúp tạo ra các đặc trưng trừu tượng cho các lớp phía sau, giúp mạng nơ-ron có thể học được các mối quan hệ phi tuyến giữa dữ liệu đầu vào và đầu ra. Có nhiều loại hàm kích hoạt khác nhau được sử dụng trong mạng nơ-ron, tùy thuộc vào mục đích và bài toán cần giải quyết. Một số hàm kích hoạt phổ biến như:

**Sigmoid**

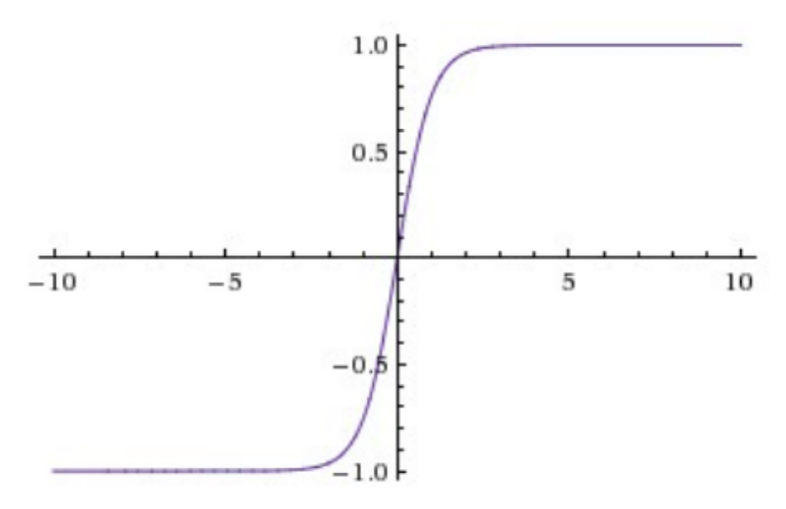




*Hàm Sigmoid*

**Tanh**

*f*(*x*) = *tanh*(*x*)



*Hàm tanh*

**ReLU**

*f*(*x*) = *max*(0*, x*)

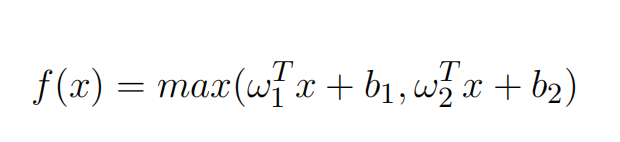


*Hàm ReLU*

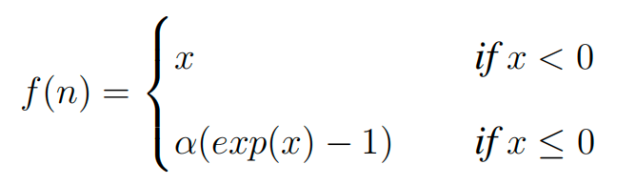
**Leaky ReLU**

*f*(*x*) = *max*(0*.*1 *∗ x, x*)

**Maxout**



**ELU**



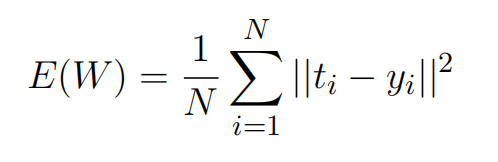
Trong thực tế, hàm kích hoạt ReLU và Leaky Relu là hai hàm được ưa chuộng nhất. Hàm kích hoạt ReLU có ưu điểm là đơn giản, nhanh và hiệu quả trong việc giảm thiểu vấn đề biến mất gradient (vanishing gradient). Hàm kích hoạt Leaky Relu là một biến thể của ReLU, có thể khắc phục được vấn đề nơ-ron chết (dying neuron) khi giá trị đầu vào âm. Hàm Tanh cũng có thể được sử dụng trong một số trường hợp, nhưng nó vẫn gặp phải vấn đề biến mất gradient khi giá trị đầu vào quá lớn hoặc quá nhỏ. Hàm Sigmoid gần như ít được sử dụng bởi vì nó có nhiều nhược điểm, như biến mất gradient, không tâm đối xứng và dễ bão hòa.

**3.1.2 Hàm lỗi**

Trong quá trình huấn luyện mạng nơ-ron, mục tiêu là giảm thiểu sai số giữa dự đoán và nhãn. Từ đó việc xây dựng hàm lỗi đóng một vai trò rất quan trọng. Hàm lỗi là một hàm số đo lường mức độ khác biệt giữa dự đoán và nhãn. Hàm lỗi có hai tính chất quan trọng là trả về giá trị không âm và phải có đạo hàm liên tục. Điều này giúp cho việc tối ưu hàm lỗi bằng các phương pháp gradient dễ dàng hơn. Theo cách hiểu đơn giản hàm lỗi sẽ phạt mô hình mỗi khi nó dự đoán sai và tỷ lệ thuận với mức độ sai của dự đoán. Càng nhiều sai số thì giá trị hàm lỗi càng cao và ngược lại. Có rất nhiều hàm lỗi được sử dụng cho những bài toán khác nhau, tuy nhiên đề tài chỉ tập trung một số hàm lỗi thông thường được sử dụng trong các bài toán phát hiện và nhận dạng như hàm Mean Square Error (MSE), Cross Entropy, Smooth-L1 và CTC. Hàm MSE là hàm tính trung bình bình phương sai số giữa dự đoán và nhãn, thường được sử dụng cho các bài toán hồi quy. Hàm Cross Entropy là hàm tính tổng âm của tích giữa nhãn và logarit của dự đoán, thường được sử dụng cho các bài toán phân loại. Hàm Smooth-L1 là hàm kết hợp giữa MSE và L1 (trung bình giá trị tuyệt đối sai số), thường được sử dụng cho các bài toán phát hiện vật thể. Hàm CTC (Connectionist Temporal Classification) là hàm tính xác suất của một chuỗi ký tự từ một chuỗi dự đoán, thường được sử dụng cho các bài toán nhận dạng ký tự.

**Hàm lỗi Mean Square Error (MSE):**

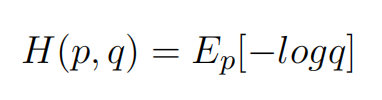
Đây là hàm lỗi đơn giản nhất mà vẫn thỏa mãn được các tính chất của một hàm lỗi tốt và cũng dễ sử dụng nhất. Hàm này định nghĩa bởi công thức:



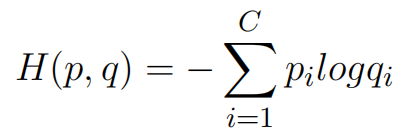
Hàm này thường được ứng dụng trong các bài toán đơn giản như hồi quy tuyến tính hay phân loại tập số liệu. Đây là những bài toán mà dữ liệu đầu vào và đầu ra có mối quan hệ tuyến tính hoặc gần tuyến tính. Tuy nhiên hàm số này khó có thể áp dụng cho các bài toán phức tạp hơn. Đó là những bài toán mà dữ liệu đầu vào và đầu ra có mối quan hệ phi tuyến, nhiễu hoặc không rõ ràng. Hàm lỗi này có thể gây ra vấn đề biến mất gradient (vanishing gradient) khi giá trị sai số quá nhỏ hoặc quá lớn, làm cho việc cập nhật trọng số của mạng nơ-ron trở nên khó khăn. Received message.

**Hàm lỗi Cross Entropy:**

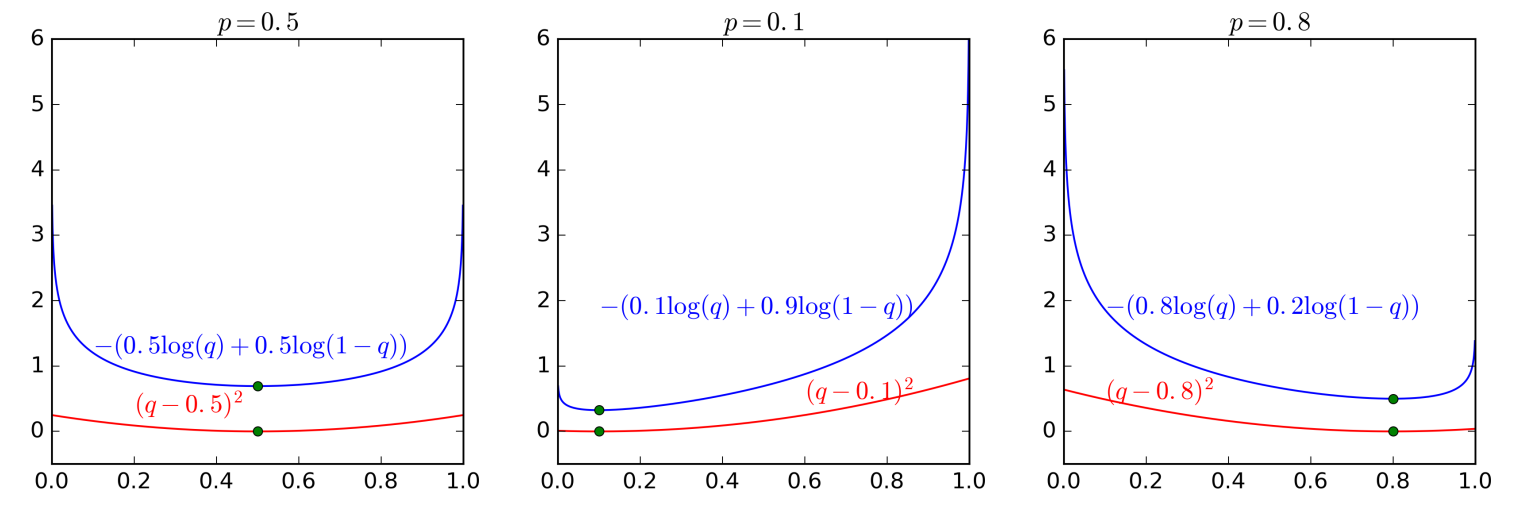
Hàm này hay được sử dụng trong các bài toán phân loại. Cross Entropy là một hàm lỗi đo lường mức độ khác biệt giữa hai phân phối xác suất *p* và *q*. Trong bài toán phân loại, *p* là phân phối xác suất của nhãn thực tế, còn *q* là phân phối xác suất của dự đoán. Cross Entropy giữa hai phân phối *p* và *q* được định nghĩa là:



Với *p* và *q* rời rạc công thức trên được viết lại:



Hàm Cross Entropy có nhiều ưu điểm so với hàm bình phương khoảng cách thông thường. Hàm bình phương khoảng cách thông thường là hàm tính trung bình bình phương sai số giữa dự đoán và nhãn. Hàm Cross Entropy có nhiều ưu điểm như: phù hợp với các bài toán phân loại nhiều lớp, không gây ra vấn đề biến mất gradient, và có thể tăng tốc độ hội tụ của mạng nơ-ron. Hình sau minh họa cho hàm lỗi trong trường hợp *C* = 2 và *p*1 lần lượt có giá trị là 0*.*5*,* 0*.*1 và 0*.*8.



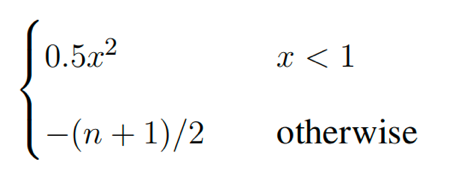
*So sánh giữa hàm Cross Entropy và hàm bình phương khoảng cách.*

*Các điểm màu* *xanh lục thể hiện các giá trị nhỏ nhất của mỗi hàm.*

Như hình ta nhận thấy hàm Cross Entropy nhận giá trị rất cao khi *p* ở xa *q*. Trong khi đó với hàm bình phương khoảng cách sữ chênh lệch là không đáng kể. Điều này có nghĩa là hàm Cross Entropy sẽ phản ánh rõ hơn mức độ sai lệch giữa dự đoán và nhãn. Do đó mô hình được huấn luyện với hàm Cross Entropy sẽ cho kết quả tốt hơn vì với mỗi dự đoán sai sẽ bị trừng phạt nặng hơn. Điều này sẽ thúc đẩy mô hình cải thiện dự đoán của mình để giảm thiểu giá trị hàm lỗi.

**Hàm lỗi Smooth-L1**

Hàm lỗi này được dùng trong các mạng phát hiện vật thể với đường bao. Hàm này giúp hồi quy các đường bao dự đoán về các đường bao đúng. Đường bao là một hình chữ nhật bao quanh vật thể trong ảnh. Đường bao có thể được biểu diễn bằng tọa độ của hai điểm đối diện hoặc bằng tọa độ của điểm trung tâm và chiều rộng và chiều cao của hình chữ nhật. Hàm lỗi này có công thức như sau:



Hình minh họa về hàm Smooth L1:



*Hàm Smooth-L1*

**Hàm lỗi CTC**

Hàm lỗi Connectionist Temporal Classification [16] (CTC) được sử dụng trong các bài toán nhận dạng chuỗi ký tự. Đây là hàm lỗi được sử dụng trong kiến trúc mạng CRNN [1] cho việc nhận dạng văn bản. Ưu điểm của hàm lỗi này là có thể nhận diện chuỗi ký tự có chiều dài bất kỳ. Hàm lỗi này tính xác suất của một chuỗi ký tự từ một chuỗi dự đoán có chiều dài lớn hơn hoặc bằng. Một vài khái niệm quan trọng trong hàm lỗi này như:

Với mỗi chuỗi đầu vào, sẽ có nhiều chuỗi dự đoán có chiều dài khác nhau tương ứng với chuỗi ban đầu. Tuy nhiên những chuỗi dự đoán này đều có thể được rút gọn về chuỗi ban đầu. Chuỗi ban đầu là chuỗi ký tự mà ta muốn nhận dạng. Những ký tự trong chuỗi ban đầu có thể được lặp lại nhiều lần trong chuỗi dự đoán. Ở giữa các ký tự hợp lệ là các ký tự trống . Các ký tự trống này là các ký tự không thuộc bộ ký tự mà ta muốn nhận dạng. Các ký tự trống này cùng với các ký tự lặp lại sẽ bị loại bỏ trong quá trình rút gọn. Quá trình rút gọn là quá trình xóa bỏ các ký tự trống và các ký tự lặp lại liên tiếp để thu được chuỗi ban đầu.



*Quá trình thu giảm chuỗi dự đoán*

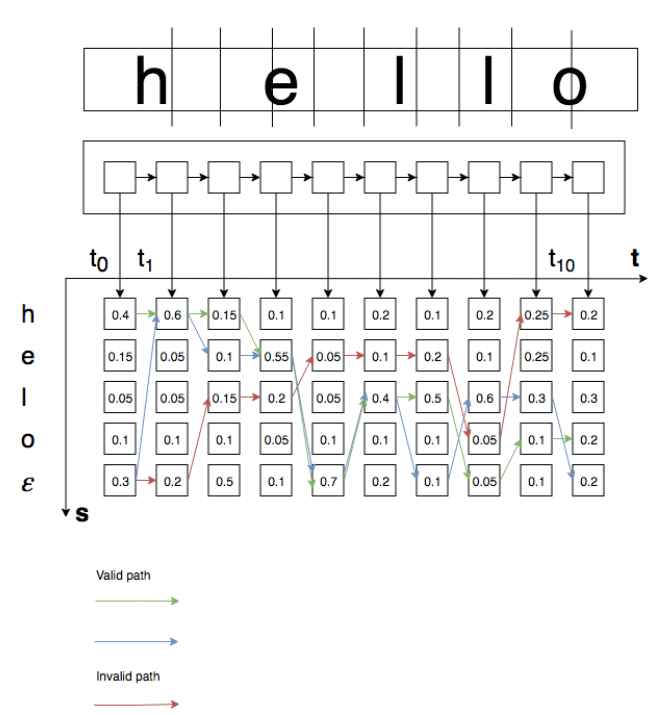
Mục tiêu của hàm lỗi CTC là tìm ra được các chuỗi mà có xác suất cao nhất trong số các chuỗi hợp lệ. Các chuỗi hợp lệ là các chuỗi mà có thể được rút gọn về chuỗi ban đầu. Điều này được thể hiện qua phương trình. Trong đó *X* là dữ liệu đầu vào, *Y* là chuỗi ban đầu. *A* là chuỗi hợp lệ trong tập các chuỗi có thể có *AX,Y*. Tập các chuỗi có thể có AX,Y là tập các chuỗi dự đoán có chiều dài bằng hoặc lớn hơn chiều dài của X và có thể được rút gọn về Y.



Để tính giá trị của hàm lỗi Connectionist Temporal Classification (CTC), ta cần thực hiện các bước sau:

* Trích xuất đặc trưng từ dữ liệu ảnh đầu vào bằng cách sử dụng các lớp mạng tích chập (CNN). Các đặc trưng này được sử dụng để đưa ra dự đoán cho mỗi cột trên đặc trưng.
* Đưa các đặc trưng đã được trích xuất qua các lớp mạng hồi quy (RNN) để tạo ra một chuỗi dự đoán. Các giá trị dự đoán này ứng với mỗi cột trên đặc trưng.
* Chuyển đổi các giá trị dự đoán thành chuỗi xác suất bằng cách sử dụng hàm Softmax. Các giá trị xác suất này đại diện cho khả năng của các ký tự tại vị trí tương ứng trên đặc trưng.
* Loại bỏ các ký tự trùng lặp hoặc rỗng bằng cách ghép nối các ký tự với nhau theo thứ tự trên đặc trưng để tạo ra một chuỗi dự đoán đầy đủ.
* Tính giá trị của hàm lỗi CTC bằng cách so sánh chuỗi dự đoán với chuỗi nhãn thật sự bằng cách sử dụng thuật toán Beam search. Thuật toán này tìm kiếm tất cả các chuỗi dự đoán có thể từ chuỗi xác suất, sau đó so sánh với chuỗi nhãn thật sự để tính toán giá trị hàm lỗi.

Điều quan trọng cần lưu ý là chiều rộng của đặc trưng không nhất thiết phải bằng số lượng ký tự. Vì vậy, trong một số trường hợp, một số cột liền nhau trên đặc trưng có thể đại diện cho một ký tự hoặc ký tự trống. Quá trình xử lý này được thực hiện để tạo ra chuỗi dự đoán đầy đủ và đúng chuẩn.



*Minh họa cách tính giá trị hàm lỗi CTC*

**3.1.3 Quá trình tối ưu hóa**

Trong quá trình huấn luyện mạng học, mục tiêu của chúng ta là tìm ra giá trị nhỏ nhất của hàm lỗi, còn gọi là hàm chi phí (cost function). Để đạt được điều này, chúng ta cần tối ưu hóa các tham số của mạng học, ví dụ như trọng số (weights) và bias, thông qua việc điều chỉnh chúng dựa trên đạo hàm của hàm chi phí.

Trong quá trình tối ưu hóa, phương pháp phổ biến nhất là Gradient Descent, trong đó chúng ta tính toán đạo hàm của hàm chi phí theo từng tham số (trọng số và bias), và sử dụng giá trị đạo hàm này để điều chỉnh các tham số. Cụ thể, mỗi lần lặp qua tập dữ liệu huấn luyện (epoch), các tham số sẽ được cập nhật theo công thức:

Wt+1 = Wt-- α \* ∇J(Wt)

Trong đó:

* Wt là trọng số (hoặc bias) hiện tại của mạng học
* Wt+1 là trọng số mới được cập nhật sau khi điều chỉnh
* α (alpha) là learning rate, đại diện cho kích thước bước nhảy khi cập nhật trọng số. Nó được sử dụng để kiểm soát tốc độ học của mạng học và tránh việc overshooting (vượt qua giá trị tối ưu).
* ∇J(Wt) là gradient của hàm chi phí theo trọng số Wt. Gradient là một vector có chiều bằng với số lượng trọng số trong mạng học, mỗi phần tử của nó là đạo hàm riêng của hàm chi phí theo từng trọng số. Gradient này sẽ cho chúng ta biết hướng tăng/giảm của hàm chi phí, và chúng ta sẽ cập nhật trọng số theo hướng ngược lại với gradient này để giảm thiểu hàm chi phí.

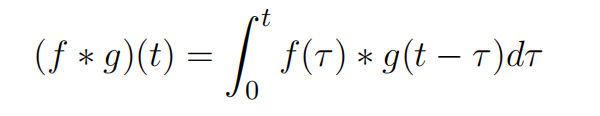
Quá trình tối ưu hóa này sẽ được lặp lại nhiều lần cho đến khi đạt được giá trị nhỏ nhất của hàm chi phí hoặc khi không có sự cải thiện đáng kể nữa. Quá trình này còn được gọi là backpropagation (lan truyền ngược), vì chúng ta tính toán đạo hàm từ phía đầu ra của mạng học và lan truyền ngược lại để cập nhật các trọng số.

**3.2. Mạng nơ-ron tích chập**

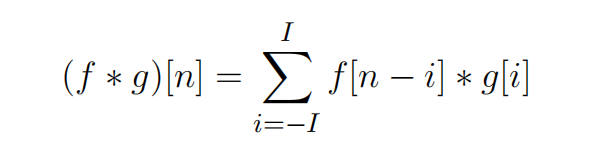
Trong thời gian gần đây, mạng nơ-ron tích chập (Convolutional Neural Network - CNN) đã thu hút sự quan tâm đặc biệt từ cộng đồng nghiên cứu về thị giác máy tính. CNN được đưa ra lần đầu bởi Yann LeCun và đồng nghiệp vào năm 1998 [19]. Mạng này đã được công nhận với ưu điểm lớn nhất là khả năng xuất sắc trong việc giải quyết các vấn đề liên quan đến thị giác máy tính và có khả năng dễ dàng mở rộng sang các loại bài toán khác.

**3.2.1 Tính chất mạng nơ-ron tích chập**

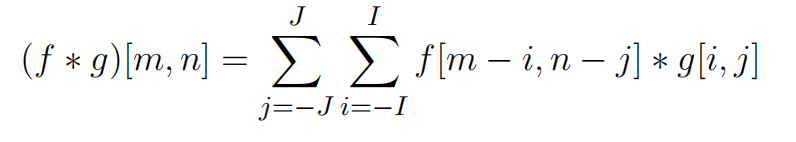
Mạng nơ-ron tích chập có cách tính toán tương tự như các mặt nạ tích chập thông thường. Phép toán tính chập được định nghĩa bẳng công thức sau:

****

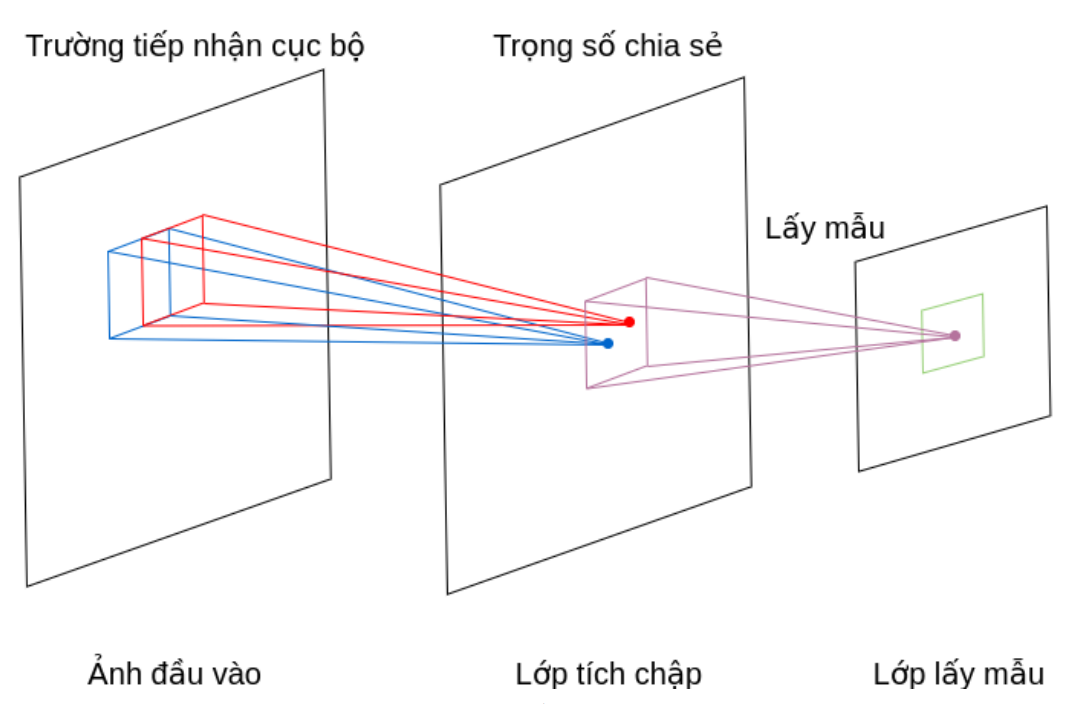
Trong miền rời rạc công thức 3.2.1 được viết lại như sau:

****

Với ảnh số công thức 3.2.1 được mở rộng như sau:

****

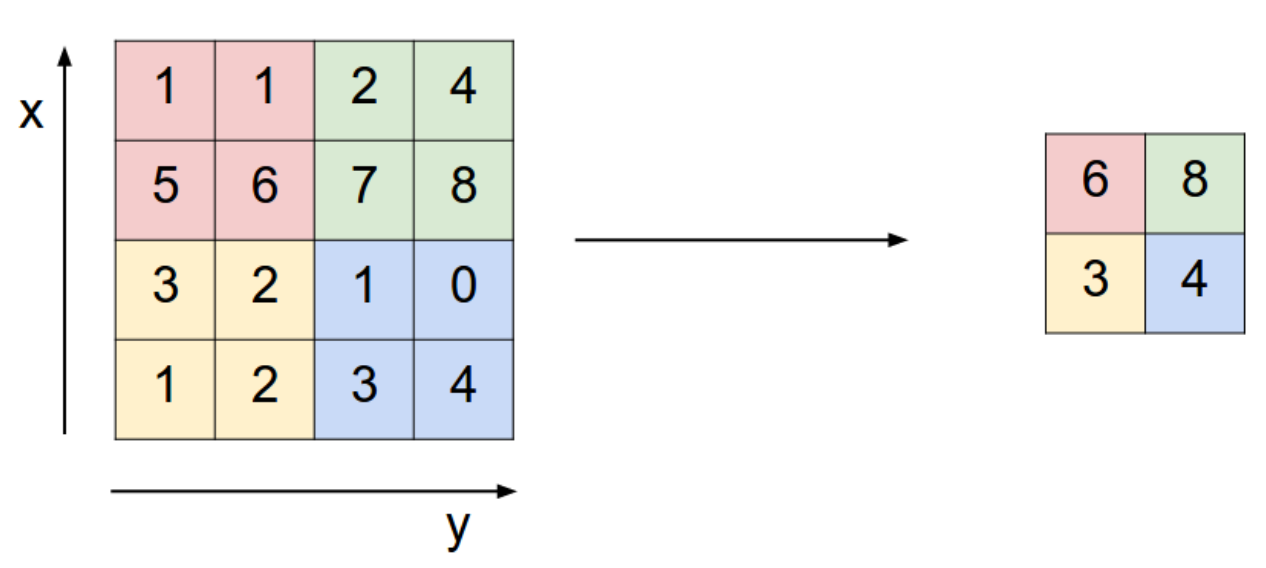
Điểm quan trọng nhất để phân biệt mạng nơ-ron tích chập (CNN) và mạng nơ-ron kết nối đầy đủ truyền thống chính là cách quản lý trọng số. Trong mạng nơ-ron kết nối đầy đủ, mỗi nút đầu vào có một trọng số riêng để tính toán giá trị ẩn tương ứng. Trái lại, trong mạng tích chập, các trọng số này được chia sẻ (shared) trên toàn mạng. Điều này mang lại lợi thế đáng kể trong việc huấn luyện mạng tích chập. Khi sử dụng mạng tích chập đa tầng, mỗi điểm trên bản đồ đặc trưng của tầng sau sẽ ứng với một vùng trên bản đồ đặc trưng của tầng trước. Điều này cho phép mạng tích chập đa tầng có khả năng trích xuất các đặc trưng từ các mức cơ bản đến các mức trừu tượng. Hơn nữa, các cửa sổ được sử dụng để rút trích đặc trưng trong mạng tích chập được học tự động, trong khi việc trích xuất đặc trưng thủ công với một số lượng cửa sổ cố định thường không hiệu quả. Do đó, mạng CNN đa tầng có lợi thế rõ ràng hơn.



*Các tính chất của mạng tích chập*

**3.2.2 Lấy mẫu (Subsampling hay pooling)**

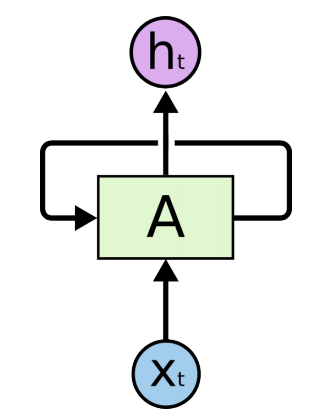
Mạng CNN thường không bất biến đối với các phép biến đổi như xoay và co giãn. Để đảm bảo mạng học sâu hoạt động hiệu quả và tổng quát hóa tốt, thường có sự kết hợp của các lớp lấy mẫu xen kẽ giữa các lớp CNN. Với đầu vào là bản đồ đặc trưng được tạo ra từ lớp trước đó, ta thực hiện việc lấy mẫu bằng cách chia thành các cửa sổ kích thước 2 x 2 hoặc 3 x 3, và có thể là các kích thước khác nữa. Quá trình lấy mẫu có thể được thực hiện bằng cách chọn giá trị lớn nhất hoặc tính giá trị trung bình của các giá trị trong cửa sổ. Ví dụ, hình 3-11 miêu tả việc thực hiện lấy mẫu theo hàm max trên một ma trận 4x4. Chúng ta sử dụng cửa sổ kích thước 2x2 và thực hiện lấy mẫu với bước trượt là 2. Mỗi lần trượt cửa sổ, hàm lấy mẫu sẽ chọn giá trị lớn nhất trong vùng 2x2 đó. Kết quả sau khi lấy mẫu là một ma trận có kích thước 2x2 như hình 3-11. Quá trình lấy mẫu này có tác dụng làm giảm kích thước đầu ra và giữ lại những giá trị quan trọng.



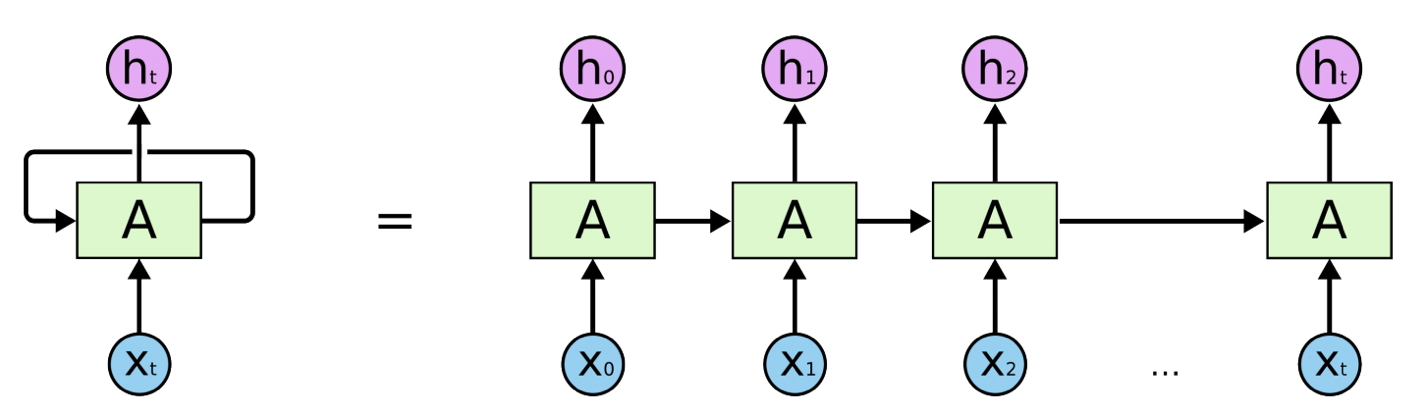
Hình 3-11: Minh họa về lớp Max Pooling với kích thước cửa sổ 2x2 và độ dịch chuyển là 2

**3.3 Mạng nơ-ron đệ quy**

Mạng nơ-ron đệ quy (Recurrent Neural Network - RNN) là một loại mạng nơ-ron, trong đó mỗi nút trong các lớp ẩn kết nối với chính nó. Đây là kết nối quan trọng tạo ra các trạng thái ẩn của kiến trúc mạng, cho phép mô hình hoá các chuỗi có độ dài khác nhau. Mạng RNN thường được biểu diễn dưới dạng lặp, như được minh họa trong hình 3-12. Tuy nhiên, để dễ dàng hiểu hơn, kiến trúc của mạng nơ-ron đệ quy cũng có thể được biểu diễn dưới dạng không lặp, như hình 3-13.



**Hình 3-12:** *Mạng nơ-ron đệ quy biểu diễn dưới dạng lặp*



**Hình 3-13:** *Mạng nơ-ron đệ quy biểu diễn dưới dạng không lặp*

Một mạng nơ-ron đệ quy có thể nhận vào một chuỗi có chiều dài bất kỳ và tạo ra một

chuỗi nhãn có chiều dài tương ứng nên kiến trúc này rất phù hợp với bài toán nhận diện

chuỗi, theo vết đối tượng. Như mô tả trong hình 3-13, tại mỗi thời điểm *t* một RNN điển

hình *A* sẽ tạo ra một vector *ht* chứa toàn bộ thông tin của các dữ liệu đầu vào từ *X*0 tới

*Xt* . Công thức tổng quát của mạng RNN được định nghĩa như sau:

*ht* = *fW* (*ht −* 1*, xt*) (3.6)

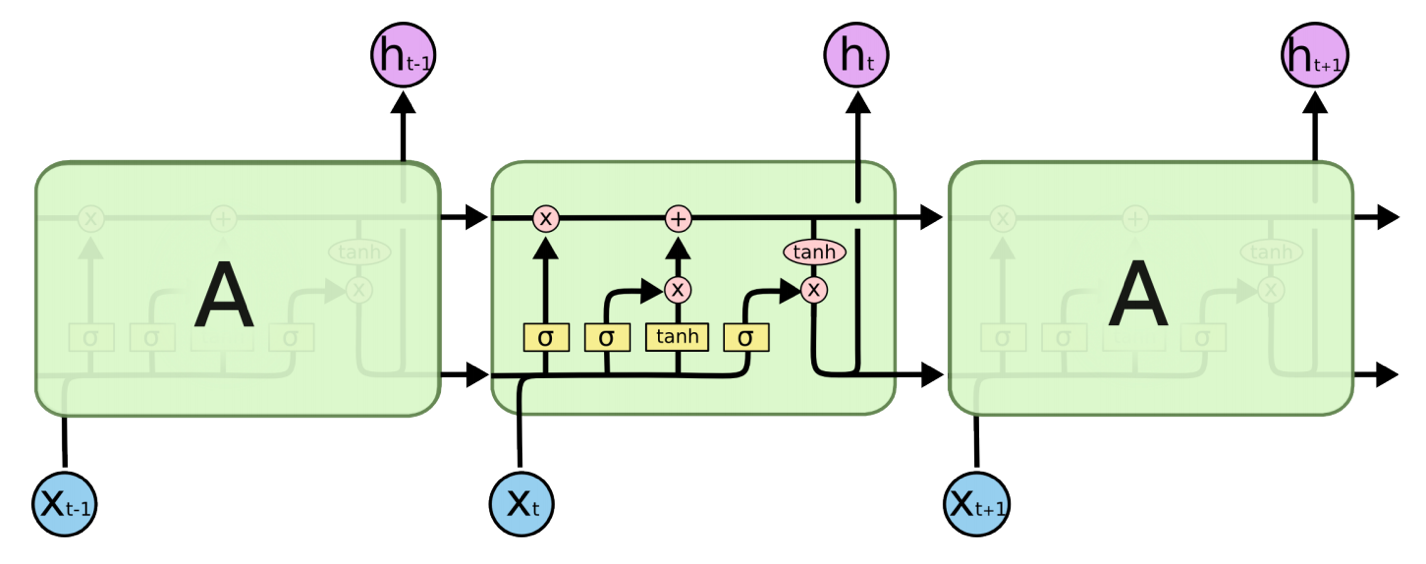
Với *ht* là trạng thái ẩn, *fW* là một hàm số với tham số *W*, *xt* là dữ liệu đầu vào của

mạng. Trong trường hợp hàm *fW* là hàm *tanh* ta có công thức như sau:

*ht* = *tanh*(*Whhht −* 1 + *Wxhxt*)*yt* = *W hyht*  (3.7)

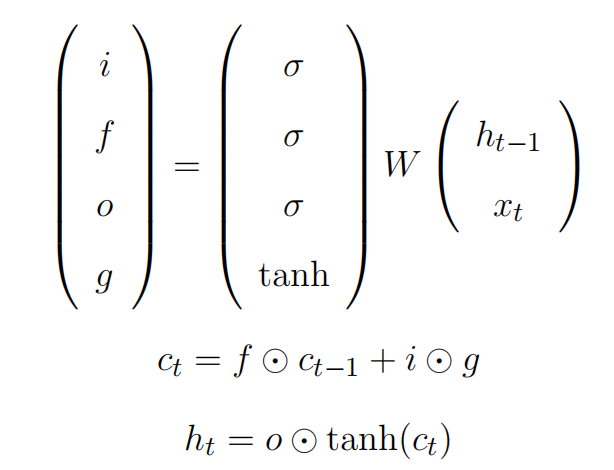
Tuy nhiên, kiến trúc RNN truyền thống thường gặp khó khăn trong việc áp dụng thực tế do vấn đề liên quan đến việc mất mát hoặc bùng nổ giá trị (vanishing and exploding gradient) khi cập nhật trọng số của mạng trong quá trình học, đặc biệt khi mô hình hoá các chuỗi có độ dài lớn.

Để giải quyết vấn đề này, Long Short Term Memory (LSTM) [15], một biến thể nổi tiếng của mạng nơ-ron hồi quy, đã được đề xuất. Điểm đặc biệt của kiến trúc LSTM là sự sử dụng các ô nhớ với các cổng, cho phép lưu trữ hoặc truy xuất thông tin. Các cổng này bao gồm cổng ghi đè (input gate), cổng loại bỏ thông tin không cần thiết (forget gate) và cổng truy xuất thông tin (output gate) từ bên trong các ô nhớ. Cấu trúc của một phần tử LSTM và các công thức tổng quát liên quan được mô tả như sau:

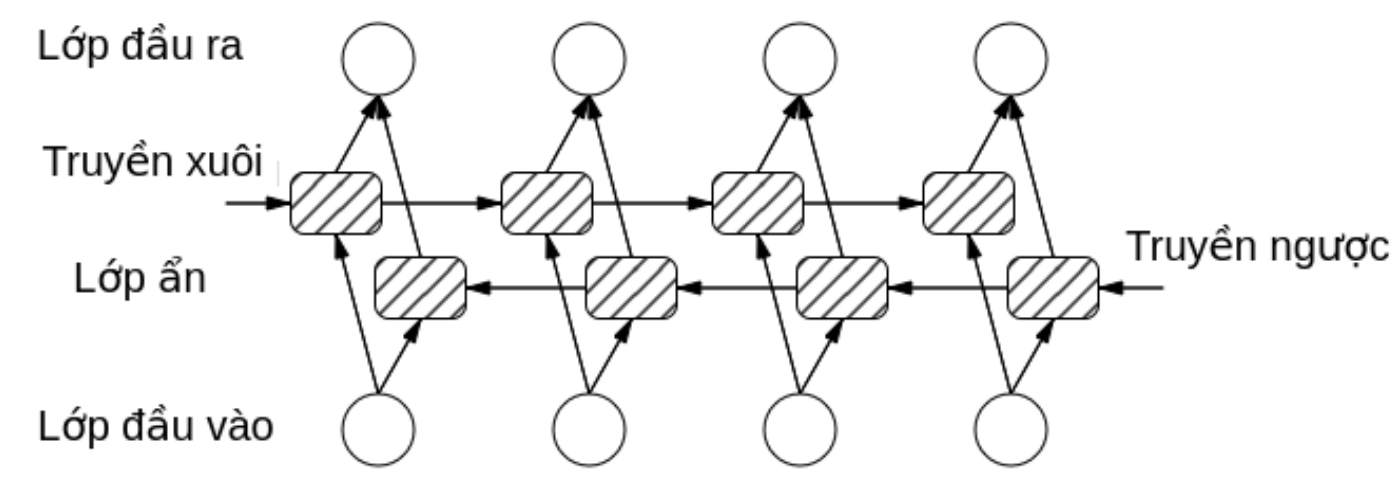


**Hình 3-14:** *Cấu trúc của mạng LSTM*

Về mặt toán học, một phần tử LSTM được định nghĩa như sau:



Đối với nhiều bài toán nhận dạng chuỗi, việc dự đoán ký tự hiện tại thường đòi hỏi sử dụng thông tin từ các ký tự trước đó. Tuy nhiên, để đạt được độ chính xác tốt hơn, thông tin từ các ký tự sau cũng cần được tích hợp vào quá trình dự đoán. Để thực hiện điều này, mô hình Bidirectional LSTM (BiLSTM) đã được đề xuất. Đây là một mạng LSTM bao gồm hai phần tử hoạt động độc lập. Một phần tử sẽ mô hình hóa chuỗi dữ liệu theo hướng từ trái sang phải, trong khi phần tử còn lại sẽ thực hiện quá trình ngược lại, từ phải sang trái.



**Hình 3-15:** *Mạng Bidirectional LSTM*

**CHƯƠNG 4: MÔ HÌNH ĐỀ XUẤT**

**4.1 Tổng quan**

Đề tài xây dựng một mô hình tích hợp giữa phát hiện và nhận dạng biển số xe. Mô hình này bỏ qua bước phân đoạn ký tự và nhận diện trực tiếp chuỗi ký tự từ ảnh biển số hoặc từ bản đồ đặc trưng của phần mạng phát hiện. Mô hình gồm hai mạng chính: mạng phát hiện và mạng nhận dạng. Trong đó mạng phát hiện sử dụng kiến trúc mạng YOLOv8, với kích thước đầu vào là 640x480, model pre-trained là YOLOv8n. Mạng nhận diện sử dụng kiến trúc ……….. Tùy theo các thí nghiệm khác nhau mà đề tài sẽ có những thay đổi kiến trúc của từng mạng phát hiện và nhận dạng. Kiến trúc mô hình được đề xuất trong đề tài là kiến trúc tuần tự.

Kiến trúc tuần tự được mô tả theo như hình ………. . Trình tự các bước từ ảnh đầu vào cho tới kết quả đầu ra như sau:

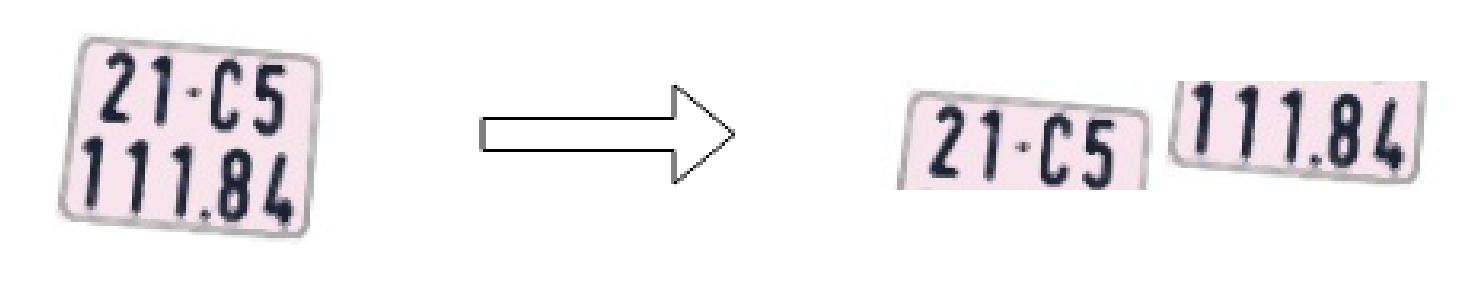
* **Bước 1:** Đầu tiên ảnh đầu vào sẽ được cho qua lớp mạng YOLO để trích xuất đặc trưng. Ảnh đầu vào là ảnh màu (BGR) và có kích thước bất kỳ. Trước khi đưa vào lớp trích xuất đặc trưng ảnh sẽ được đưa về kích thước 640x480. Tại đầu ra của lớp này, ảnh sẽ biến đổi thành các bản đồ đặc trưng với kích thước lần lượt là ……… .
* **Bước 2:** Các bản đồ đặc trưng từ bước một gồm ……… kích thước khác nhau sẽ được đưa qua lần lượt các lớp mạng của YOLO để phát hiện vị trí của biển số xe cũng như phân loại biển số. Chi tiết về bước phát hiện biển số xe cũng như kiến trúc mạng sẽ được trình bày trong mục ……… .
* **Bước 3:** Tại bước này vùng biển số xe sẽ được cắt ra từ ảnh đầu vào. Sau đó các bản đồ đặc trưng cho vùng biển số này, tùy vào loại biển số là ngắn hay dài sẽ được xử lý tương ứng và đưa về kích thước chung là……… . Vùng biển số sau xử lý sẽ được đưa vào mạng nhận diện phía sau. Mục ……… sẽ trình bày chi tiết về cách trích xuất biển số xe.
* **Bước 4:** Đây là bước quan trọng nhất trong toàn bộ hệ thống. Vùng biển số sau khi được trích xuất sẽ đưa qua lớp mạng nhận diện để ra chuỗi ký tự trực tiếp mà không cần qua bước phân đoạn ký tự. Mạng nhận diện ở mô hình tich hợp là mạng CRNN. Chi tiết về bước phát hiện được trình bày trong mục ……… .
* **Bước 5:** Bước cuối cùng trong mô hình là bước tổng hợp các kết quả từ hai lớp mạng phát hiện và nhận diện để cho kết quả sau cùng. Tại bước này các vùng biển số sẽ được lọc bớt dựa theo kết quả của việc nhận diện.

<hình vẽ kiến trúc>

**4.3 Trích xuất vùng biển số xe**

Với kiến trúc tuần tự vùng biển số sẽ được cắt ra từ ảnh gốc. Vùng biển số được cắt ra

và dựa trên loại biển số là ngắn hay dài mà sẽ được chia làm đôi. Với biển số ngắn sau khi được chia làm đôi sẽ được ghép nối tiếp với nhau theo chiều ngang. Sau đó tất cả biển số sẽ được đưa về kích thước cố định là ………. . Hình ………. minh họa cho thao tác trích xuất vùng biển số xe.



**Hình 4-3:** *Biển số sau khi cắt ra sẽ được ghép lại*

**4.4 Nhận dạng biển số xe**

**Chương 5: Edge Computing Trên thiết bị nhúng**

5.1. Giới thiệu xu hướng Edge Computing

Trong thời đại số hóa ngày càng phát triển, xu hướng Edge Computing đang trỗi dậy mạnh mẽ. Edge Computing là một mô hình tính toán mà xử lý dữ liệu diễn ra ngay tại nguồn thu thập dữ liệu, thay vì gửi dữ liệu lên một trung tâm dữ liệu trung ương (server) để xử lý. Điều này giúp giảm độ trễ, tăng cường bảo mật và hiệu năng, và phù hợp cho các ứng dụng yêu cầu thời gian thực.

5.2. Giới thiệu cấu tạo và chức năng của Raspberry Pi 4

Raspberry Pi 4 là một ví dụ tiêu biểu cho việc triển khai Edge Computing. Đây là một bo mạch nhỏ gọn nhưng mạnh mẽ với CPU ARM, RAM, cổng USB, và kết nối mạng. Raspberry Pi 4 có khả năng kết nối với nhiều thiết bị đầu cuối, như camera và cảm biến, và có thể chạy các ứng dụng xử lý dữ liệu phức tạp. Điểm mạnh của Raspberry Pi 4 là tính nhỏ gọn, tiết kiệm năng lượng, và giá thành rất thấp.

5.3. Quy trình chuyển dữ liệu đã xử lý từ Raspberry Pi 4 về server

Khi Raspberry Pi 4 thu thập và xử lý dữ liệu tại điểm thu thập (Edge), dữ liệu có thể được gửi trực tiếp đến server. Quy trình này bao gồm việc truyền dữ liệu qua mạng, sử dụng giao thức mạng như MQTT hoặc HTTP. Tại server, dữ liệu có thể được lưu trữ, phân tích, và sử dụng để ra quyết định hoặc tạo ra báo cáo. Đánh giá ưu điểm và nhược điểm của Edge Computing sử dụng Raspberry Pi 4 so với kiến trúc truyền thống xử lý tại server

Ưu điểm:

Giảm độ trễ: Edge Computing giúp giảm độ trễ trong xử lý dữ liệu, phù hợp cho các ứng dụng thời gian thực như IoT và xử lý video trực tiếp.

Bảo mật tăng cường: Dữ liệu không cần phải di chuyển xa, giảm nguy cơ tiết lộ thông tin quan trọng trong quá trình truyền tải.

Tiết kiệm băng thông: Dữ liệu được xử lý và gửi đến server chỉ khi cần thiết, giảm tải mạng.

Tiết kiệm năng lượng: Raspberry Pi 4 tiêu thụ ít năng lượng hơn so với các máy chủ trung tâm dữ liệu lớn.

Nhược điểm:

Hạn chế về xử lý: Raspberry Pi 4 có giới hạn về tài nguyên tính toán, không phù hợp cho các ứng dụng yêu cầu xử lý dữ liệu lớn.

Hạn chế về dung lượng lưu trữ: Dung lượng lưu trữ trên Raspberry Pi 4 có thể hạn chế, yêu cầu chuyển dữ liệu đến server để lưu trữ lâu dài.

Yêu cầu kiến thức kỹ thuật: Triển khai và quản lý Raspberry Pi 4 đòi hỏi kiến thức kỹ thuật, không phù hợp cho người không chuyên.

Edge Computing sử dụng Raspberry Pi 4 đang là một lựa chọn hấp dẫn cho các ứng dụng yêu cầu hiệu năng và thời gian thực, nhưng cần xem xét kỹ càng về yêu cầu cụ thể của dự án trước khi triển khai.

**CHƯƠNG 5: THÍ NGHIỆM**

Từ mô hình đã nêu ở phần trên, đề tài thực hiện một vài thí nghiệm nhằm đánh giá độ

hiệu quả của mô hình cũng như tìm ra được các siêu tham số phù hợp nhất với mô hình

đề ra. Trình tự các bước được mô tả như sau

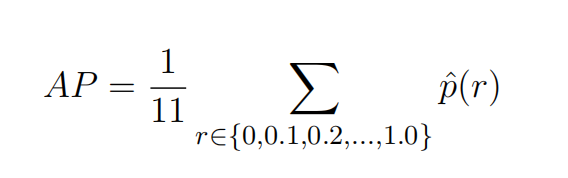
**5.1 Tiêu chí đánh giá**

Phương pháp đề xuất sẽ tập trung vào đánh giá việc phát hiện biển số dựa trên độ đo

chính là Average Precision (AP).

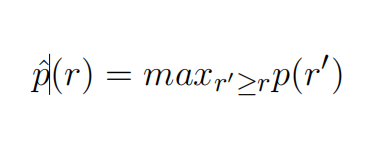
Độ đo AP được định nghĩa là độ chính xác trung bình của một tập 11 giá trị truy hồi

*r ∈ {*0*,* 0*.*1*,* 0*.*2*, . . . ,* 1*.*0*}*:



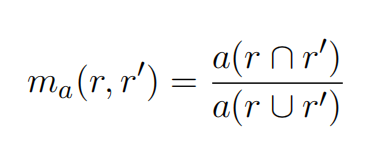
Trong đó độ chính xác tại giá trị truy hồi thứ i được lấy là độ chính xác tối đa tại những

giá trị truy hồi *r0* lớn hơn *r*.



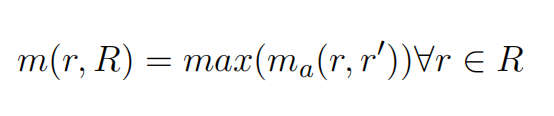
mAP là trung bình AP của từng lớp.

Độ trùng khớp giữa 2 vùng bao chữ nhật:

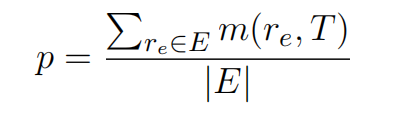


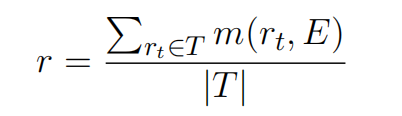
Trong đó a(r) là diện tích vùng chữ nhật r. Giá trị trùng khớp tốt nhất cho một vùng

chữ nhật r trong tập chữ nhật R được định nghĩa:



Độ chính xác và độ truy hồi được định nghĩa:





Trong đó E và T tương ứng là tập hợp các đường bao chữ nhật ước lượng và ground

truth. *rt* và *re* là các đường bao chữ nhật thuộc các tập hợp trên.

Độ chính xác nhận diện một chuỗi: