**NGHIÊN CỨU MẠNG CNN ĐỂ GIẢI BÀI TOÁN TỐI ƯU TỔ HỢP**

|  |  |
| --- | --- |
| Nguyễn Thị Mỹ Bình  [binhntm@haui.edu.vn](mailto:binhntm@haui.edu.vn)  Đại Học Công Nghiệp Hà Nội | Đỗ Viết Thịnh  [thinhdo19022002@gmail.com](mailto:thinhdo19022002@gmail.com)  Đại Học Công Nghiệp Hà Nội |
| Nguyễn Hoàng Minh  [minhnguyeny2002@gmail.com](mailto:minhnguyeny2002@gmail.com)  Đại Học Công Nghiệp Hà Nội | Nguyễn Thị Thu Thủy  [nguyenthithuthuy19122002@gmail.com](mailto:nguyenthithuthuy19122002@gmail.com)  Đại Học Công Nghiệp Hà Nội |
| Mai Ngọc Định  [ngocdinh2k1@gmail.com](mailto:ngocdinh2k1@gmail.com)  Đại Học Công Nghiệp Hà Nội | Hoàng Thị Sao Mai  hoangsaomai04@gmail.com  Đại Học Công Nghiệp Hà Nội |

**TÓM TẮT**

Bài toán tối ưu tổ hợp là một bài toán tối ưu hóa cổ điển, đã được nghiên cứu và ứng dụng nhiều trong đời sống, là một trong những trọng tâm nghiên cứu khoa học máy tính. Bên cạnh đó, phân tích kĩ thuật học sâu đang rất phát triển hiện nay, đã hé lộ những hướng phát triển thú vị và có nhiều điểm tương đồng trong giải quyết các bài toán với bài toán tối ưu tổ hợp. Từ những phân tích trên, chúng tôi đề xuất một phương pháp giải bài toán tối ưu tổ hợp thông qua việc nghiên cứu và sử dụng mạng CNN. Kết quả phân tích là một nguồn tham khảo cho việc giải bài toán tối ưu tổ hợp.

Từ khóa: tối ưu hóa tổ hợp; CNN.

1. MỞ ĐẦU

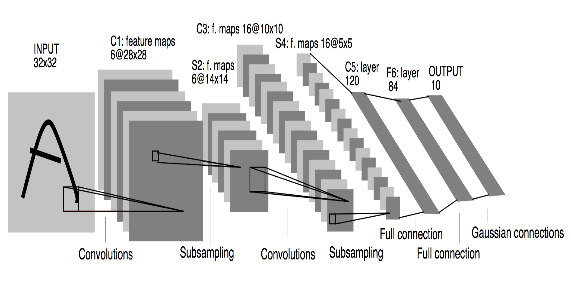
Phương pháp tối ưu hóa tổ hợp được nghiên cứu từ rất sớm, cùng với sự phát triển nhanh chóng của công nghệ thông tin, chúng đang được nhiều người quan tâm nghiên cứu và ứng dụng rộng rãi trong các bài toán thực tế đặc biệt là trong tin-sinh học. Trong tiếp cận truyền thống các bài toán và thuật toán giải phải tuân thủ nhiều điều kiện toán học khắt khe như phải được thiết lập đúng đắn (tồn tại duy nhất nghiệm và ổn định với điều kiện ban đầu), phải chứng minh được tính hội tụ hoặc ước lượng sai số/ tỷ lệ tối ưu. Vì các yêu cầu như vậy, những bài toán cần tối ưu ngày càng tăng nhưng các thuật toán được đề xuất không đáp ứng được nhu cầu trong ứng dụng cũng rất lớn. Các phương pháp dử dụng trí tuệ nhân tạo giải quyết các bài toán tối ưu hóa tổ hợp theo tiếp cận mềm dẻo hơn. Kết quả thực nghiệm cho thấy hiệu quả tốt của các tiếp cận này nên chúng đang thu hút nhiều người nghiên cứu, ứng dụng. Phương pháp giải bài toán tối ưu tổ hợp với mạng nơ-ron thường được áp dụng cho các bài toán tối ưu tô hợp cỡ lớn. Trong đó hiệu quả của các thuật toán được đánh giá bằng thực nghiệm. Phương pháp này có 2 hướng kết quả được hướng đến là tìm kiếm một lời giả đủ tốt, thường hướng tới cực trị địa phương và hướng tới cực trị toàn cục với thời gian chạy lâu hơn.

Trong bối cảnh đó, chúng tôi chọn chủ đề nghiên cứu “Nghiên cứu mạng CNN để giải bài toán tối ưu tổ hợp” với nội dung nghiên cứu trình bày về mạng nơ-ron tích chập (Convolutional Neural Network – CNN), và khả năng ứng dụng của nó để giải bài toán tối ưu tổ hợp cụ thể, mà ở đây là bài toán “chiếc răng”. Trong phạm vi bài báo, chúng tôi sẽ trình bày mô hình CNN và sử dụng các mô hình nhận dạng hình ảnh để đánh giá các đối tượng nằm sai vị trí trong tổ hợp các vị trí của chiếc răng với chất luọng lời giải và thời gian tính toán tốt hơn các thuật toán cổ điển.

2. PHƯƠNG PHÁP VÀ TƯ LIỆU

**2.1. Phân tích mạng CNN**

Trong các mạng nơ-ron truyền thống, mỗi nơ-ron trong lớp phía trước sẽ được kết nối đến tất cả các nơ-ron ở lớp phía sau, điều này khiến cho khối lượng tính toán trong mạng tăng mạnh khi tăng độ sâu của mô hình (tăng số lượng lớp) cho mô hình.

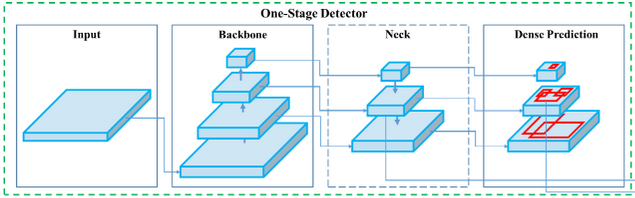
 *Hình 2.1. Sơ đồ khối mô tả mô hình mạng lenet-5 (1998).*

Mạng CNN đã giúp giải quyết vấn đề trên bằng cách sử dụng các vùng tiếp nhận cục bộ, tập trọng số chia sẻ và phương pháp lấy tích chập để trích xuất thông tin thay cho mạng nơ-ron cổ điển. Hình 2.1 phía trên biểu diễn mô hình mạng Lenet-5, mô hình CNN được Yan Lecun công bố đầu tiên vào năm 1998. Sau đây, chúng tôi sẽ đi vào chi tiết hơn về mạng nơ-ron tích chập và các tham số sẽ được nghiên cứu đánh giá. Lớp tích chập là một thành phần cốt lõi của mạng nơ-ron tích chập (CNN), sử dụng để trích xuất các thông tin đặc tính của hình ảnh (feature map) hỗ trợ cho quá trình “học” của mạng CNN. Phương thức hoạt động của lớp này được thực hiện thông qua quá trình trượt và lấy tích chập của bộ lọc (filter/kernel) trên toàn bộ ảnh. Kết quả đầu ra là đặc tính của ảnh tương ứng với bộ lọc đã sử dụng, với càng nhiều bộ lọc được sử dụng, chúng tôisẽ thu được càng nhiều đặc tính của ảnh tương ứng. Trong phạm vi bài báo này, chúng tôi sẽ xét đến các tham số có tác động đến lớp tích chập gồm: kích thước bộ lọc, bước trượt (stride). Lớp lấy mẫu xuống có tác dụng giảm kích thước của dữ liệu hình ảnh từ đó giúp cho mạng có thể học được các thông tin có tính chất khái quát hơn, đồng thời quá trình này giảm số lượng các thông số trong mạng. Các phương pháp lấy mẫu xuống thường được sử dụng là Max Pooling và Average Pooling. Lớp dropout là một kỹ thuật được sử dụng để hạn chế hiện tượng overfitting (hiện tượng mạng nơ-ron quá bám sát vào tập dữ liệu huận luyện và không đáp ứng được với các tập dữ liệu mới), thường gặp ở mạng CNN và giúp mô hình tính toán nhanh hơn. Dropout sử dụng phương pháp loại bỏ một số nơ-ron ngẫu nhiên trong mạng với một xác suất cho trước bằng cách thiết lập tất cả trọng số nơ-ron đó bằng 0, đồng nghĩa với các liên kết tới nơ-ron đó đều không có giá trị, khi đó mô hình sẽ phải cố gắng nhận dạng đúng trong khi thiếu thông tin từ các nơ-ron bị loại bỏ. Điều này sẽ giúp tăng tỉ lệ nhận dạng của mô hình nhưng không quá phụ thuộc vào dữ liệu huấn luyện. Đầu vào của lớp kết nối đầy đủ là đầu ra từ lớp lấy mẫu xuống hoặc lớp tích chập cuối cùng, nó được làm phẳng và sau đó được đưa vào lớp kết nối đầy đủ để chuyển tiếp. Lớp FC có nhiệm vụ tổng hợp thông tin đưa ra lớp quyết định (output) cho ra kết quả chính xác nhất.

**2.2. Phân tích mô hình YOLO**

**2.2.1. YOLOv5**

YOLO là một thuật toán phát hiện đối tượng theo thời gian thực, hiện đại do Joseph Redmon, Santosh Divvala, Ross Girshick và Ali Farhadi tạo ra vào năm 2015 và đã được đào tạo trước trên bộ dữ liệu COCO. Nó sử dụng một mạng thần kinh duy nhất để xử lý toàn bộ hình ảnh. Hình ảnh được chia thành các vùng và thuật toán dự đoán xác suất và hộp giới hạn cho từng vùng. Kiến trúc máy dò đối tượng một giai đoạn (như YOLO) bao gồm ba thành phần: Xương sống, Cổ và Đầu để đưa ra các dự đoán dày đặc như thể hiện trong hình dưới đây.



*Hình 2.2. Sơ đồ khối mô tả YOLO v5*

Xương sống là một mạng được đào tạo trước được sử dụng để trích xuất biểu diễn tính năng phong phú cho hình ảnh. Điều này giúp giảm độ phân giải không gian của hình ảnh và tăng độ phân giải tính năng (kênh) của nó. Cổ là mô hình được sử dụng để trích xuất các kim tự tháp đặc trưng. Điều này giúp mô hình khái quát hóa tốt cho các đối tượng ở các kích cỡ và tỷ lệ khác nhau. Đầu là mô hình được sử dụng để thực hiện các thao tác giai đoạn cuối. Nó áp dụng các hộp neo trên bản đồ tính năng và hiển thị đầu ra cuối cùng: các lớp, điểm đối tượng và hộp giới hạn. Tất cả các mô hình YOLOv5 đều bao gồm 3 thành phần giống nhau: CSP-Darknet53 làm xương

sống, SPP và PANet ở cổ mô hình và phần đầu được sử dụng trong YOLOv4. Việc chọn một chức năng kích hoạt là rất quan trọng đối với bất kỳ mô hình học sâu nào, đối với YOLOv5, các tác giả đã sử dụng chức năng kích hoạt SiLU và Sigmoid. SiLU là viết tắt của Sigmoid Linear Unit và nó còn được gọi là hàm kích hoạt swish. Nó đã được sử dụng với các hoạt động tích chập trong các lớp ẩn. Trong khi chức năng kích hoạt Sigmoid đã được sử dụng với các hoạt động tích chập trong lớp đầu ra.

**2.2.2. YOLOv8**

YOLOv8 là mô hình YOLO tiên tiến nhất có thể được sử dụng cho các tác vụ phát hiện đối tượng, phân loại hình ảnh và phân đoạn phiên bản. YOLOv8 được phát triển bởi [Ultralytics](https://ultralytics.com/?ref=roboflow-blog) , người cũng đã tạo ra mô hình YOLOv5 có sức ảnh hưởng và định hình ngành. YOLOv8 bao gồm nhiều thay đổi và cải tiến về kiến ​​trúc và trải nghiệm của nhà phát triển so với YOLOv5. YOLOv8 đang được phát triển tích cực tại thời điểm viết bài đăng này. YOLOv8 có những ưu điểm đặc biệt như:YOLOv8 có tỷ lệ chính xác cao được đo bằng COCO và Roboflow 100, YOLOv8 đi kèm với rất nhiều tính năng thuận tiện cho nhà phát triển, từ CLI dễ sử dụng đến gói Python có cấu trúc tốt, Có một cộng đồng lớn xung quanh YOLO và một cộng đồng đang phát triển xung quanh mô hình YOLOv8, nghĩa là có nhiều người trong giới thị giác máy tính có thể hỗ trợ bạn khi bạn cần hướng dẫn, YOLOv8 chưa có bài báo nào được xuất bản, vì vậy chúng tôi thiếu thông tin chi tiết trực tiếp về phương pháp nghiên cứu trực tiếp và các nghiên cứu tạo ra nó. Ở đây chúng tôi tóm tắt nhanh về các bản cập nhật mô hình có tác động và sau đó chúng tôi sẽ xem xét đánh giá của mô hình, YOLOv8 là một mô hình không có neo. Điều này có nghĩa là nó dự đoán trực tiếp tâm của một đối tượng thay vì phần bù từ [hộp neo](https://blog.roboflow.com/what-is-an-anchor-box/) đã biết, [các hộp neo](https://blog.roboflow.com/what-is-an-anchor-box/) là một phần khét tiếng phức tạp của các mô hình YOLO trước đó, vì chúng có thể đại diện cho việc phân phối các hộp của điểm chuẩn mục tiêu nhưng không phải là sự phân phối của tập dữ liệu tùy chỉnh, Chuyển đổi đầu tiên của gốc 6x6 được thay thế bằng 3x3, khối xây dựng chính đã được thay đổi và [C2f](https://github.com/ultralytics/ultralytics/blob/dba3f178849692a13f3c43e81572255b1ece7da9/ultralytics/nn/modules.py?ref=roboflow-blog#L196) thay thế [C3](https://github.com/ultralytics/yolov5/blob/cdd804d39ff84b413bde36a84006f51769b6043b/models/common.py?ref=roboflow-blog#L157) . Mô-đun được tóm tắt trong hình bên dưới, trong đó "f" là số lượng tính năng, "e" là tốc độ mở rộng và CBS là một khối bao gồm a Conva BatchNormvà a SiLUsau. Trong C2f, tất cả các đầu ra từ Bottleneck (tên ưa thích cho hai 3x3 convsvới các kết nối còn lại) được nối với nhau. Trong khi C3chỉ đầu ra của cái cuối cùng Bottleneckđược sử dụng C2f Mô-đun YOLOv8 mới Giống Bottlenecknhư trong YOLOv5 nhưng kích thước hạt nhân của chuyển đổi đầu tiên đã được thay đổi từ 1x1thành 3x3. Từ thông tin này, chúng ta có thể thấy rằng YOLOv8 đang bắt đầu trở lại khối ResNet được xác định vào năm 2015. Ở phần cổ, các tính năng được nối trực tiếp mà không bắt buộc các kích thước kênh giống nhau. Điều này làm giảm số lượng tham số và kích thước tổng thể của các tenxơ. YOLOv8 tăng cường hình ảnh trong quá trình đào tạo trực tuyến. Tại mỗi thời điểm, mô hình sẽ thấy một biến thể hơi khác nhau của hình ảnh mà nó đã được cung cấp. Một trong những phần mở rộng đó được gọi là [phần mở rộng khảm](https://blog.roboflow.com/advanced-augmentations/) . Điều này liên quan đến việc ghép bốn hình ảnh lại với nhau, buộc mô hình tìm hiểu các đối tượng ở các vị trí mới, che khuất một phần và dựa trên các pixel xung quanh khác nhau.

**2.3. Phân tích thuật toán K-mean**

Mục đích của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra center của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng. Giả sử thêm rằng mỗi điểm dữ liệu chỉ thuộc vào đúng một nhóm.

Đầu vào: Dữ liệu X và số lượng cluster cần tìm K.

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y.

Chọn K điểm bất kỳ làm các center ban đầu. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2. Quay lại bước 2. Chúng ta có thể đảm bảo rằng thuật toán sẽ dừng lại sau một số hữu hạn vòng lặp. Thật vậy, vì hàm mất mát là một số dương và sau mỗi bước 2 hoặc 3, giá trị của hàm mất mát bị giảm đi. Theo kiến thức về dãy số trong chương trình cấp 3: nếu một dãy số giảm và bị chặn dưới thì nó hội tụ! Hơn nữa, số lượng cách phân nhóm cho toàn bộ dữ liệu là hữu hạn nên đến một lúc nào đó, hàm mất mát sẽ không thể thay đổi, và chúng ta có thể dừng thuật toán tại đây.

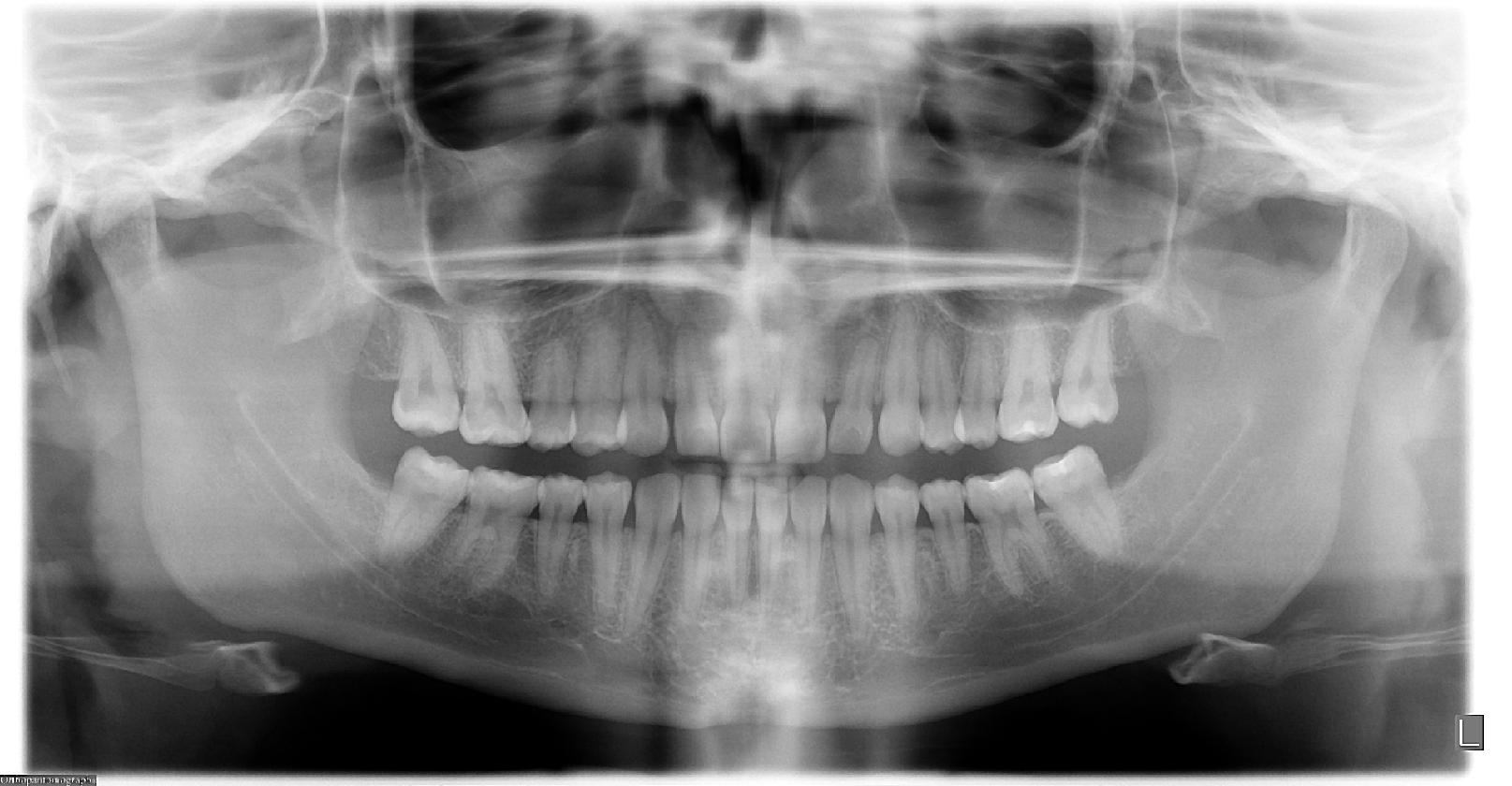
**2.4. Phương pháp thực hiện**

Để sử dụng thuật toán phân cụm để phân nhóm các răng, bạn có thể làm như sau: Trong phạm vi bài báo này, chúng tôi lấy bài toán “Chiếc răng” để mô tả chi tiết phương pháp giải bài toán tối ưu tổ hợp bằng việc sử dụng mạng CNN. Dữ liệu đầu vào của bài toán là một hình ảnh chụp X-quang. Mỗi chiếc răng cần phải được đặt vào đúng vị trí của nó trong tổ hợp vị trí các chiếc răng. Đầu ra của bài toán là những chiếc răng đang ở sai vị trí trong tổ hợp vị trí cá chiếc răng. Đầu tiên trong phương pháp này, chúng tôi trích xuất các thông tin về răng từ đầu ra của mô hình YOLO để thu thập tọa độ của các bounding box tương ứng với các răng và các thông tin liên quan khác như điểm trung tâm, chiều rộng và chiều dài của bounding box. Tiếp theo chúng tôi sử dụng thuật toán phân cụm để phân nhóm các răng. Có nhiều phương pháp phân cụm khác nhau, trong đó phương pháp K-means là phương pháp phổ biến và có thể áp dụng được cho bài toán này. Ta có thể chọn số lượng nhóm phù hợp tùy thuộc vào số lượng răng và độ phân chia mong muốn mà cụ thể ở đây chúng tôi chọn số lượng nhóm là 3. Tiếp theo, chúng tôi kiểm tra các nhóm răng được phân cụm, tính toán trung bình hoặc trung vị của các thông tin về tọa độ, chiều rộng và chiều dài của bounding box trong mỗi nhóm răng, so sánh các giá trị này giữa các nhóm răng để kiểm tra xem có sự lệch nhau hay không. Cuối cùng, nếu có răng nằm trong cùng một nhóm bị lệch so với các răng khác, chúng tôi có thể xử lý các răng này bằng cách sử dụng các phép biến đổi hình học để cân bằng kích thước và hướng của chúng, hoặc sử dụng các phương pháp học máy để dự đoán vị trí và hướng của các răng bị lệch.

**3.KẾT QUẢ VÀ THẢO LUẬN**

**3.1. KẾT QUẢ**

Dữ liệu là hỉnh ảnh chụp X-quang hàm răng của các bệnh nhân thông qua các nguồn trên internet. Hình ảnh được giấu tên, tuổi, thông tin của các bệnh nhân để đảm bảo yêu cầu quyền cá nhân. Mỗi bức ảnh được gán nhãn về răng.



*Hình 3.1 Hình ảnh đại diện cho bộ dữ liệu*

Qua quá trình thực hiện các phương pháp và thuật toán, chúng tôi đã mô tả phía trên, kết quả thu được là hình ảnh chụp X-quang răng đã được nhận diện và hình ảnh chụp X-quàn răng bị phát hiện sai lệch.



*Hình 3.2. Hình ảnh mô tả kết quả khi phát hiện răng*



*Hình 3.3. Hình ảnh mô tả kết quả phát hiện răng sai lệch qua thuật toán K-mean*

Tiếp theo trong chương này, chúng tôi sẽ trình bày các kết quả thực nghiệm phương pháp đã được nêu trên với các mô hình khác nhau trong mạng CNN. Các thực nghiệm đều thực hiện trên cùng một bộ dữ liệu được sinh ra một cách ngẫu nhiên. Cấu trúc của chương này được chia ra như sau:

* Kết quả thực nghiệm của chương trình với YOLO v5m
* Kết quả thực nghiệm của chương trình với YOLO v5n
* Kết quả thực nghiệm của chương trình với YOLO v5s
* Kết quả thực nghiệm của chương trình với YOLO v8m
* Kết quả thực nghiệm của chương trình với YOLO v8n

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Params  (M) | FLOPs (B) | Size | Speed  (ms) | mAP  50 | mAP  50-95 |
| YOLOv8n | 3.0 | 8.1 | 640 | 17.4 | 95% | 85% |
| YOLOv8s | 11.2 | 28 | 640 | 65.4 | 97% | 88% |
| YOLOv8m | 25.9 | 78.1 | 640 | 114.1 | 98% | 89% |
| YOLOv5n | 2.0 | 7.1 | 640 | 13.2 | 95% | 86% |
| YOLOv5s | 7.6 | 26.4 | 640 | 88.3 | 96% | 87% |
| YOLOv5m | 22.0 | 70.8 | 640 | 207.5 | 97% | 87% |

*Hình 3.1. Biểu đồ so sánh kết quả giữa các phương pháp*

Dựa trên kết quả thu được ta thấy rằng: Các mô hình đều ấp ứng dụng cho ra kết quả khá tốt, có độ chính xác cao. Tuy nhiên, thời gian thực hiện cũng tương đối lớn, tham số nhiều, cần lượng tài nguyên không nhỏ như YOLOv5m, YOLOv8m. Bên cạnh đó, với các mô hình như YOLO v5n, YOLOv8n cần tham số ít hơn, thời gian chạy ít hơn, kết quả có mức độ chính xác tuy không cao như YOLOv5m, YOLOv8m nhưng vẫn có thể chấp nhận được.

***Lời cảm ơn:*** Chúng em rất chân thành cảm ơn cô Nguyễn Thị Mỹ Bình, Khoa CNTT, Đại học Công nghiệp Hà Nội đã hướng dẫn thực hiện đề tài này.

**4.TÀI LIỆU THAM KHẢO**

[1] Hoffmann, J., Navarro, O., Kastner, F., Janßen, B., & Hubner, M. (2017). A survey on CNN and RNN implementations. In PESARO 2017: The Seventh International Conference on Performance, Safety and Robustness in Complex Systems and Applications (No. 3).

[2] Martello, S., Minoux, M., Ribeiro, C., & Laporte, G. (Eds.).

(2011). Surveys in combinatorial optimization. Elsevier.

[3] Oliveira, C. A., & Pardalos, P. M. (2005). A survey of combinatorial optimization problems in multicast routing. Computers & Operations Research, 32(8), 1953-1981.

[4] Schrijver, A. (2005). On the history of combinatorial optimization (till 1960). Handbooks in operations research and management science, 12, 1-68.