BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG PHÁT HIỆN ĐÁM ĐÔNG HỖ TRỢ CẢNH BÁO NGUY CƠ LÂY NHIỄM VIRUS CORONA

Sinh viên thực hiện: **Nguyễn Minh Trường**

Lớp, khoa: Công Nghệ Thông Tin 3 – Công Nghệ Thông Tin

Sinh viên thực hiện: Cù Việt Hưng

Lớp, khoa: Công Nghệ Thông Tin 3 – Công Nghệ Thông Tin

Sinh viên thực hiện: Đường Ngọc Khanh

Lớp, khoa: Công Nghệ Thông Tin 3 – Công Nghệ Thông Tin

Sinh viên thực hiện: Quách Phương Thảo

Lớp, khoa: Công Nghệ Thông Tin 1 – Công Nghệ Thông Tin

Người hướng dẫn: Th.s. Nguyễn Lan Anh

**Hà Nội – Tháng 5/2022**

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG PHÁT HIỆN ĐÁM ĐÔNG HỖ TRỢ CẢNH BÁO NGUY CƠ LÂY NHIỄM VIRUS CORONA

|  |  |
| --- | --- |
| **Sinh viên thực hiện: Nguyễn Minh Trường** | **Giới tính: Nam** |
| **Dân tộc: Kinh** |  |
| **Lớp, khoa: Công Nghệ Thông Tin 3 – Công Nghệ Thông Tin** | **Năm thứ: 3** |
| **Ngành học: Công Nghệ Thông Tin** |  |
| Sinh viên thực hiện: Cù Việt Hưng | Giới tính: Nam |
| Dân tộc: Kinh |  |
| Lớp, khoa: Công Nghệ Thông Tin 3 – Công Nghệ Thông Tin | Năm thứ: 3 |
| Ngành học: Công Nghệ Thông Tin |  |
| Sinh viên thực hiện: Đường Ngọc Khanh | Giới tính: Nam |
| Dân tộc: Kinh |  |
| Lớp, khoa: Công Nghệ Thông Tin 3 – Công Nghệ Thông Tin | Năm thứ: 3 |
| Ngành học: Công Nghệ Thông Tin |  |
| Sinh viên thực hiện: Quách Phương Thảo | Giới tính: Nữ |
| Dân tộc: Kinh |  |
| Lớp, khoa: Công Nghệ Thông Tin 1 – Công Nghệ Thông Tin | Năm thứ: 4 |
| Ngành học: Công Nghệ Thông Tin |  |

Người hướng dẫn: Th.s. Nguyễn Lan Anh

**Hà Nội – Tháng 5/2022**

Mục lục

[DANH MỤC TỪ VIẾT TẮT 1](#_Toc103904415)

[DANH MỤC BẢNG BIỂU 2](#_Toc103904416)

[DANH MỤC HÌNH ẢNH 3](#_Toc103904417)

[MỞ ĐẦU 5](#_Toc103904418)

[Đề tài và lý do lựa chọn đề tài 5](#_Toc103904419)

[Mục tiêu của đề tài 5](#_Toc103904420)

[Ý nghĩa thực tiễn của đề tài 6](#_Toc103904421)

[Bố cục báo cáo 6](#_Toc103904422)

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ BÀI TOÁN 6](#_Toc103904423)

[1.1. Khái niệm nhóm người, đám đông, tập trung đông người trong xử lý ảnh 7](#_Toc103904424)

[1.2. Phạm vi của đề tài 7](#_Toc103904425)

[1.3. Các bài toán con cần giải quyết 8](#_Toc103904426)

[1.3.1. Phát hiện đối tượng (Object Detection) 8](#_Toc103904427)

[1.3.2. Đo lường đối tượng thời gian thực 9](#_Toc103904428)

[CHƯƠNG 2: CÁC NGHIÊN CỨU CÓ LIÊN QUAN 10](#_Toc103904429)

[2.1. Phát hiện đối tượng (Object Detection) 10](#_Toc103904430)

[2.1.1. Họ R-CNN 10](#_Toc103904431)

[2.1.2. YOLO 15](#_Toc103904432)

[2.2. Đo lường đối tượng thời gian thực 15](#_Toc103904433)

[2.2.1. Đo khoảng cách thực bằng tính năng phát hiện của Trí tuệ nhân tạo 15](#_Toc103904434)

[2.2.2. Tính khoảng cách dựa vào kích thước có sẵn 16](#_Toc103904435)

[CHƯƠNG 3: THUẬT TOÁN 18](#_Toc103904436)

[3.1. Phát hiện vật thể 18](#_Toc103904437)

[3.1.1. Tổng quan 18](#_Toc103904438)

[3.1.2. YOLOv4 18](#_Toc103904439)

[3.1.3. YOLOv5 28](#_Toc103904440)

[3.2. Đo lường đối tượng thời gian thực 30](#_Toc103904441)

[CHƯƠNG 4: THỰC NGHIỆM 31](#_Toc103904442)

[4.1. Chuẩn bị bộ dữ liệu chung 31](#_Toc103904443)

[4.1.1. Cấu trúc thư mục dữ liệu 32](#_Toc103904444)

[4.1.2. Tệp cấu hình 32](#_Toc103904445)

[4.2. Training 33](#_Toc103904446)

[4.2.1. Transfer learning 33](#_Toc103904447)

[4.2.2. Khai thác tính năng 34](#_Toc103904448)

[4.3. Đo lường đối tượng thời gian thực 37](#_Toc103904449)

[CHƯƠNG 5: TỔNG KẾT 39](#_Toc103904450)

[TÀI LIỆU THAM KHẢO 40](#_Toc103904451)

Đề tài: Nghiên cứu và xây dựng hệ thống phát hiện đám đông hỗ trợ cảnh báo nguy cơ lây nhiễm virus Corona.

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| R-CNN | Region-based Convolutional Neural Networks |
| YOLO | You only look once |
| CNN | Convolutional Neural Networks |
| CSP | Cross Stage Partial |
| PANet | Path Aggregation Network |
| FPN | Feature Pyramid Networks |
| RPN | Region Proposal Network |
| SPP | Spatial Pyramid Pooling |
| FCN | Fully convolution network |
| ROI | Region of Interest |

DANH MỤC BẢNG BIỂU

[Bảng 2. 1. Bảng so sánh tốc độ xử lý và độ chính xác của các lớp model object detection 14](#_Toc103902672)

[Bảng 3. 1. Bảng so sánh 3 mạng đường trục. (Huang, và cộng sự, 2018) 20](#_Toc103902683)

DANH MỤC HÌNH ẢNH

[Hình 2. 1. Sơ đồ pipeline xử lý trong mô hình mạng R-CNN 10](#_Toc103904903)

[Hình 2. 2. Kiến trúc single model Fast R-CNN 12](#_Toc103904904)

[Hình 2.3. Kiến trúc mô hình Faster R-CNN 14](#_Toc103904905)

[Hình 2.4. Mối quan hệ giữa một số đối tượng và sự phản chiếu của nó trên máy ảnh 16](#_Toc103904906)

[Hình 2.5. Ảnh minh họa 17](#_Toc103904907)

[Hình 3. 1. Hai khái niệm về phát hiện đối tượng kiến trúc. (Solawetz, 2020) 19](#_Toc103902108)

[Hình 3. 2. Sơ đồ các ý tưởng đổi mới tiên tiến nhất được tác giả áp dụng cho từng phần của kiến trúc YOLOv4. 19](#_Toc103902109)

[Hình 3. 3. Kiến trúc DenseNet với 3 khối Dense. Các lớp giữa 2 khối được gọi là các lớp chuyển tiếp. (Huang, và cộng sự, 2018) 20](#_Toc103902110)

[Hình 3. 4. Quá trình xử lý đầu vào trong một khối dày đặc. (Wang, et al., 2019) 21](#_Toc103902111)

[Hình 3. 5. Quá trình xử lý đầu vào trong một khối đặc một phần. (Wang, et al., 2019) 21](#_Toc103902112)

[Hình 3. 6. Kết nối khối dày đặc và kiến trúc YOLOv2 dựa trên Kim tự tháp không gian. (Shah, 2020) 22](#_Toc103902113)

[Hình 3. 7. Khối SPP cổ điển. (Huang, et al., 2019) 23](#_Toc103902114)

[Hình 3. 8. Khối SPP mới thích ứng với YOLO. (Huang, et al., 2019) 24](#_Toc103902115)

[Hình 3. 9. (a) Kiến trúc FPN ban đầu. (b) Kiến trúc FPN đã sửa đổi được sử dụng trong YOLOv3. (Gochoo, 2020) 25](#_Toc103902116)

[Hình 3. 10. Kiến trúc PANnet bao gồm (a) đường trục FPN, (b) tăng cường đường dẫn từ dưới lên, (c) tổng hợp tính năng thích ứng. (Liu và cộng sự, 2018) 25](#_Toc103902117)

[Hình 3. 11. (a) Kiến trúc FPN. (b) Kiến trúc PAN. (c) Kết nối trong đường dẫn nâng từ dưới lên. (Solawetz, 2020) 26](#_Toc103902118)

[Hình 3. 12. (a) PAN gốc. (b) PAN được sửa đổi cho YOLOv4. (Bochkovskiy, và cộng sự, 2020) 27](#_Toc103902119)

[Hình 3. 13. PAN đã sử dụng ROI Align để tích hợp thích ứng và các lớp được kết nối đầy đủ để kết hợp các tính năng từ tất cả các giai đoạn. (Liu và cộng sự, 2018) 27](#_Toc103902120)

[Hình 3. 14. Nhiều phương pháp tối ưu hóa hiện đại đã được thử nghiệm cho từng lĩnh vực. 28](#_Toc103902121)

[Hình 3. 15. Ảnh minh họa 31](#_Toc103902122)

[Hình 3. 16. Ảnh minh họa 31](#_Toc103902123)

[Hình 4. 1. Ảnh minh họa 33](#_Toc103902164)

[Hình 4. 2. Kết quả dữ liệu 37](#_Toc103902165)

[Hình 4. 3. Kết quả dự đoán của mô hình trên tập huấn luyện 38](#_Toc103902166)

[Hình 4. 4. Kết quả dự đoán của mô hình trên tập huấn luyện 38](#_Toc103902167)

[Hình 4. 5. Kết quả dự đoán của mô hình 39](#_Toc103902168)

[Hình 4. 6. Kết quả dự đoán của mô hình 40](#_Toc103902169)

[Hình 4. 7. Kết quả dự đoán của mô hình 40](#_Toc103902170)

MỞ ĐẦU

Đề tài và lý do lựa chọn đề tài

Công nghệ thông tin ngày càng phát triển và có vai trò hết sức quan trọng trong cuộc sống hiện đại. Con người ngày càng tạo ra những cỗ máy thông minh có khả năng tự nhận biết và xử lý được các công việc một cách tự động, phục vụ những lợi ích chung. Kèm theo đó là sự bùng nổ của các thiết bị ghi hình như: điện thoại di động, máy tính bảng, xách tay, máy ảnh, các hệ thống camera quan sát, … Trong những năm gần đây, nhiều tiến bộ trong lĩnh vực nghiên cứu trí tuệ nhân tạo, học máy và học sâu đã đưa ra giải pháp cho các bài toán nhận dạng như nhận dạng vật thể, nhận dạng chữ viết, nhận dạng khuôn mặt, nhận dạng hình dáng, nhận dạng giọng nói, … với độ chính xác rất cao. Những công nghệ trên hoàn toàn có khả năng để ứng dụng vào giải quyết các vấn đề xã hội và đặc biệt là y tế.

Trong thời gian diễn ra bênh dịch toàn cầu mang tên COVID-19, việc hạn chế tụ tập đông người là thực sự quan trọng. Tuy nhiên, vẫn tồn tại những nhân tố chưa chấp hành nội quy giãn cách, vì vậy, việc kiểm tra, rà soát thường xuyên là điều vô cùng cần thiết. Dẫu vậy, công việc của các lực lượng chức năng chưa bao giờ là dễ dàng, bởi luôn tồn tại những yếu tố ngoại cảnh như thời tiết hay khoảng cách địa lý, sự hạn chế về nhân lực, ... gây cản trở công tác phòng chống dịch. Những tác nhân này đã góp phần khiến cho việc quản lý người dân chưa được chặt chẽ. Vì vậy, nhóm chúng tôi đã lựa chọn đề tài “***Nghiên cứu và xây dựng hệ thống phát hiện đám đông hỗ trợ cảnh báo nguy cơ lây nhiễm virus Corona”****,* với mong muốn góp phần loại bỏ những nhược điểm trên, giúp việc kiểm soát dịch bệnh nhanh chóng và hiệu quả hơn.

Việc thực hiện nghiên cứu đề tài cũng góp phần thúc đẩy tinh thần cho các bạn sinh viên khác cùng chí hướng mạnh dạn tham gia nghiên cứu khoa học. Trang bị thêm cho các bạn sinh viên các kiến thức chuyên sâu, làm hành trang tri thức giúp các bạn dễ dàng tìm kiếm các cơ hội nghề nghiệp sau khi ra trường. Đóng góp vào nền khoa học chung những giải pháp công nghệ cho những bài toán thực tế trong tương lai.

Mục tiêu của đề tài

Nghiên cứu, phân tích, thiết kế, cài đặt nhằm mục tiêu xây dựng thử nghiệm hệ thống phát hiện phát hiện đám đông hỗ trợ cảnh báo nguy cơ lây nhiễm virus Corona.

Ý nghĩa thực tiễn của đề tài

Hệ thống phát hiện đám đông, nếu được xây dựng thành công, có thể hỗ trợ cảnh báo nguy cơ lây nhiễm virus Corona trong cộng đồng, có ý nghĩa quan trọng trong thời điểm dịch bệnh như hiện tại. Việc tự động hóa phát hiện đám đông không chỉ giúp cho các cán bộ tham gia phòng chống dịch dễ dàng tìm ra những sai phạm trong giai đoạn duy trì giãn cách xã hội, mà còn khiến mỗi người dân cảnh giác hơn và nâng cao ý thức tự giác phòng chống dịch. Vì vậy mà phần nào giúp giảm thiểu nguy cơ lây lan nhiễm virus.

Trong tương lai gần, sau khi thực hiện thành công nghiên cứu này, nhóm sẽ hướng đến xây dựng hệ thống phát hiện đám đông trong thời gian thực, có thể đưa vào áp dụng trong nhiều trường hợp, nhiều địa điểm công cộng như bến xe bus, rạp chiếu phim, ...

Bố cục báo cáo

Báo cáo nghiên cứu khoa học gồm 5 chương.

- Chương 1: Giới thiệu tổng quan về khái niệm nhóm người, đám đông, tập trung đông người trong xử lý ảnh, đưa ra phạm vi nghiên cứu phù hợp cho đề tài và định nghĩa các bài toán con khác cần phải giải quyết.

- Chương 2: Tổng hợp các nghiên cứu có liên quan về các phương pháp giải quyết từng bài toán con trong đề tài.

- Chương 3: Tập trung vào giới thiệu phương pháp tiếp cận của từng bài toán con.

- Chương 4: Đưa ra các thông tin về mặt thực nghiệm và các kết quả đạt được trong việc giải quyết các bài toán con và xây dựng hệ thống thử nghiệm.

- Chương 5: Là chương cuối cùng, nhằm tổng kết kết quả đạt được và đưa ra các điểm thiếu sót, các hướng phát triển, mở rộng đề tài nghiên cứu trong tương lai.

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ BÀI TOÁN

1.1. Khái niệm nhóm người, đám đông, tập trung đông người trong xử lý ảnh

Theo giáo sư Vittorio Murino, đại học Verona – nguyên giám đốc Viện công nghệ Ý (PAVIS), dưới góc độ quan điểm xã hội học, ta có thể đưa ra một số khái niệm như sau:

- Nhóm người là một thực thể gồm nhiều hơn một người, cạnh bên nhau và có một số hoạt động như sau:

* Di chuyển cùng nhau và có hướng chuyển động tương đồng;
* Có hoạt động giống nhau, chẳng hạn như băng qua đường, đang chờ xe, đang trò chuyện với nhau, ….

- Tập trung đông người là hành vi khi có một nhóm từ 2 người trở lên cùng hiện diện và có hoạt động tương tác với nhau. Việc xác định hành vi tập trung đông người thường dựa vào các tiêu chí sau:

* Sự hiện diện của một số lượng người;
* Có hình thức tương tác về mặt xã hội, hoặc có các kiểu tương tác với nhau bằng tay;
* Tính chất của việc thiết lập hoạt động tập trung đông người (địa điểm riêng tư hay nơi công cộng).

- Đám đông là sự hình thành bởi sự tập trung của một số lượng người rất lớn, có thể được nhận diện khi mật độ người lớn đến mức không thể xác định được từng cá thể hay nhóm người [(Hình 1.2](#_heading=h.lnxbz9)).

Hoạt động tập trung đông người được phân chia theo số lượng người hiện diện tại một không gian cụ thể như: tập trung số lượng ít (2 đến 6 người) ở những nơi riêng tư như nhà, vườn nhà, trong ô tô; tập trung số lượng trung bình (có thể từ 7 đến 12 người) ở lớp học, văn phòng, câu lạc bộ, khu vực đãi tiệc); tập trung số lượng lớn (có thể từ 13 người trở lên) ở những nơi như quảng trường, công viên, đường phố, nhà ga, …

1.2. Phạm vi của đề tài

Phát hiện đám đông phục vụ việc bảo đảm giãn cách xã hội thực sự là một toán rất rộng. Có nhiều phương pháp tiếp cận khác nhau, tuỳ thuộc vào điều kiện hoạt động khác nhau và tuỳ thuộc vào các trang thiết bị phần cứng khác nhau mà có từng phương pháp phù hợp riêng. Trong thực tế, việc phát hiện đám đông còn dựa trên nhiều yếu tố như khoảng cách; tư thế của cơ thể (mặt đối mặt, đối xứng); đầu, hướng nhìn, giao tiếp bằng ánh mắt (tập trung bằng thị giác); điệu bộ, tư thế (loại tương tác), ... Xét trên lượng kiến thức, kinh nghiệm, thời gian và nguồn lực phần cứng có hạn nên nhóm sẽ giới hạn bài toán này trong khuôn khổ sau đây:

* Việc xác định đám đông sẽ được định nghĩa bằng số người đứng cạnh nhau, cụ thể số lượng lớn hơn hoặc bằng 2 người, khoảng cách giữa các đối tượng người sẽ được tính theo giá trị pixel điểm ảnh vì việc tính khoảng cách chính xác là rất khó, có thể làm phức tạp bài toán dẫn đến việc không tối ưu được kết quả.
* Hệ thống có thể chạy thời gian thực nhưng việc đào tào dữ liệu đòi hỏi cần GPU mạnh và có dung lượng lớn nên phải sử dụng phần mềm ảo để đào tạo.
* Việc thực hiện chạy chương trình sẽ dùng trên laptop MSI modern 14 với GPU NVIDIA MX450 và camera điện thoại hoặc có thể một số thiết bị camera khác.

1.3. Các bài toán con cần giải quyết

1.3.1. Phát hiện đối tượng (Object Detection)

Cho tới thời điểm hiện tại, các thuật toán phát hiện vật thể vẫn là một trong những thuật toán quan trọng nhất trong phân ngành Thị giác máy tính (Computer Vision). Bởi, nó là tiền đề để giải quyết rất nhiều bài toán thực tế khác nhau. Các thuật toán phát hiện vật thể thường được chia thành 2 nhóm chính là one-stage object detection và two-stage object detection

* Two-stage object detection: nhóm các thuật toán (R-CNN, Fast R-CNN, Faster R-CNN, ...) chia việc phát hiện vật thể ra làm 2 bước. Ở bước một, mô hình sẽ xử lý ảnh và trích xuất ra các vùng có khả năng chứa vật thể từ bức ảnh. Ví dụ, với Faster R-CNN thì trong bước 1, mô hình sẽ đưa ra 1 sub-network gọi là RPN (Region Proposal Network) với nhiệm vụ trích xuất các vùng trên ảnh có khả năng chứa đối tượng dựa vào các anchor. Sau khi đã thu được các vùng đặc trưng từ RPN, mô hình Faster R-CNN sẽ thực hiện tiếp bước 2 là phân loại đối tượng và xác định vị trí.
* One-stage object detection: là các thuật toán (SSD, YOLO, ...) không có phần trích chọn các vùng đặc trưng như RPN của Faster R-CNN. Các mô hình thuộc nhóm one-stage object detection này coi việc định vị đối tượng (object localization) như một bài toán hồi quy (regression problem) với 4 tọa độ offset, ví dụ như x, y, w, h, và cũng dựa trên các pre-define box gọi là anchor để làm việc đó. Bằng việc chỉ chạy một thuật toán một lần duy nhất, các thuật toán thuộc nhóm này thường đạt tốc độ nhanh hơn tuy nhiên độ chính xác của mô hình thường kém hơn so với các thuật toán two-stage object detection.

Trong bài toán này, mô hình của nhóm chúng tôi cần phát hiện được đối tượng Người trong hình ảnh, xác định được bounding box bao quanh vật thể. Từ đó lấy ra tọa độ tâm để phục vụ cho bài toán tính toán khoảng cách.

1.3.2. Đo lường đối tượng thời gian thực

Công nghệ trí tuệ nhân tạo đang phát triển nhanh chóng trong những năm gần đây. Để đo khoảng cách đến một đối tượng bằng cách sử dụng điều này. Để đo khoảng cách, người ta sẽ chụp hai ảnh riêng biệt ở cùng một góc của vật. Nó trích xuất kích thước cho cùng một đối tượng trong hai hình ảnh. Để làm được điều này trong thời gian thực, công nghệ phát hiện đối tượng của Trí tuệ nhân tạo trên điện thoại di động đã được sử dụng.

CHƯƠNG 2: CÁC NGHIÊN CỨU CÓ LIÊN QUAN

2.1. Phát hiện đối tượng (Object Detection)

2.1.1. Họ R-CNN

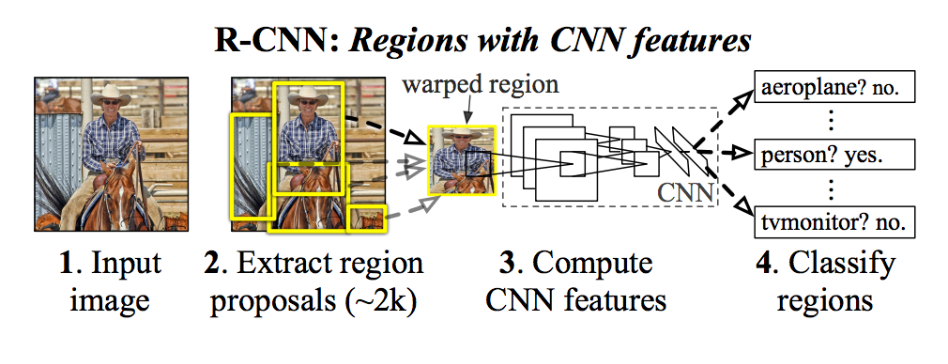
a. R-CNN

R-CNN được giới thiệu lần đầu vào 2014 bởi Ross Girshick và các cộng sự. Nó có thể là một trong những ứng dụng nền móng đầu tiên của mạng nơ-ron tích chập đối với vấn đề định vị, phát hiện và phân đoạn đối tượng. Cách tiếp cận đã được chứng minh trên các bộ dữ liệu điểm chuẩn, đạt được kết quả tốt nhất trên bộ dữ liệu VOC-2012 và bộ dữ liệu phát hiện đối tượng ILSVRC-2013 gồm 200 lớp.

Kiến trúc của R-CNN gồm 3 thành phần đó là:

* Vùng đề xuất hình ảnh (Region proposal): Có tác dụng tạo và trích xuất các vùng đề xuất chứa vật thể được bao bởi các bounding box.
* Trích lọc đặc trưng (Feature Extractor): Trích xuất các đặc trưng giúp nhận diện hình ảnh từ các region proposal thông qua các mạng deep convolutional neural network.
* Phân loại (classifier): Dựa vào input là các features ở phần trước để phân loại hình ảnh chứa trong region proposal về đúng nhãn.

Kiến trúc của mô hình được mô tả trong biểu đồ bên dưới:



Hình 2. 1. Sơ đồ pipeline xử lý trong mô hình mạng R-CNN

Ta có thể nhận thấy các hình ảnh con được trích xuất tại bước 2 với số lượng rất lớn (khoảng 2000 region proposals). Tiếp theo đó áp dụng một mạng deep CNN để tính toán các feature tại bước 3 và trả ra kết quả dự báo nhãn ở bước 4 như một tác vụ image classification thông thường.

Một kỹ thuật được sử dụng để đề xuất các region proposal hoặc các bounding box chứa các đối tượng tiềm năng trong hình ảnh được gọi là “selective search”, các region proposal có thể được phát hiện bởi đa dạng những thuật toán khác nhau. Nhưng điểm chung là đều dựa trên tỷ lệ IoU giữa bounding box và ground truth box.

Một nhược điểm của phương pháp này là chậm, đòi hỏi phải vượt qua nhiều module độc lập trong đó có trích xuất đặc trưng từ một mạng CNN học sâu trên từng region proposal được tạo bởi thuật toán đề xuất vùng chứa ảnh. Đây là một vấn đề chính cần giải quyết vì bài viết mô tả mô hình hoạt động trên khoảng 2000 vùng được đề xuất cho mỗi hình ảnh tại thời điểm thử nghiệm.

b. Fast R-CNN

Dựa trên thành công của R-CNN, Ross Girshick đề xuất một mở rộng để giải quyết vấn đề của R-CNN trong một bài báo vào năm 2015 với tiêu đề rất ngắn gọn [Fast R-CNN](https://arxiv.org/abs/1504.08083).

Bài báo chỉ ra những hạn chế của R-CNN đó là:

* Training qua một pipeline gồm nhiều bước: Pipeline liên quan đến việc chuẩn bị và vận hành ba mô hình riêng biệt.
* Chi phí training tốn kém về số lượng bounding box và thời gian huấn luyện: Mô hình huấn luyện một mạng CNN học sâu trên rất nhiều region proposal cho mỗi hình ảnh nên rất chậm.
* Phát hiện đối tượng chậm: Tốc độ xử lý không thể đảm bảo realtime.

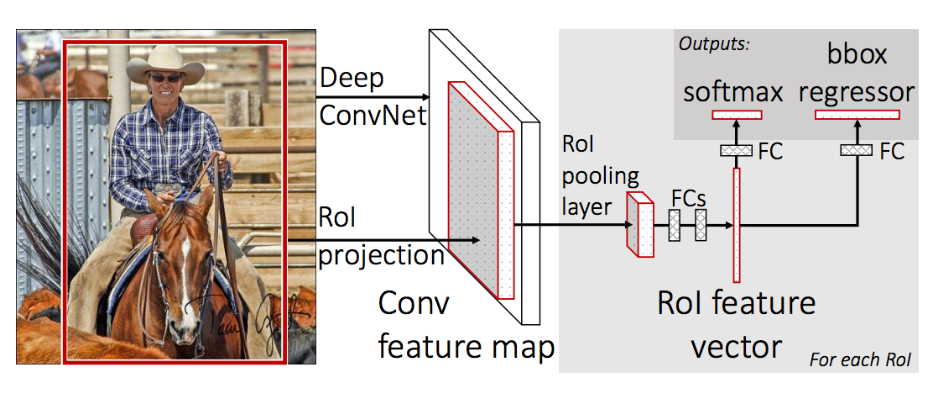
Trước đó một bài báo đã đề xuất phương pháp để tăng tốc kỹ thuật được gọi là [mạng tổng hợp kim tự tháp - Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition](https://arxiv.org/abs/1406.4729), hoặc SPPnets vào năm 2014. Phương pháp này đã tăng tốc độ trích xuất features nhờ lan truyền thuận trên bộ nhớ đệm.

Điểm đột phá của Fast R-CNN là sử dụng một single model thay vì pipeline để phát hiện region và classification cùng lúc.

Kiến trúc của mô hình trích xuất từ bức ảnh một tập hợp các region proposals làm đầu vào được truyền qua mạng deep CNN. Một pretrained-CNN, chẳng hạn VGG-16, được sử dụng để trích lọc features. Phần cuối của deep-CNN là một custom layer được gọi là layer vùng quan tâm (Region of Interest Pooling - RoI Pooling) có tác dụng trích xuất các features cho một vùng ảnh input nhất định.

Sau đó các features được kết bởi một lớp fully connected. Cuối cùng mô hình chia thành hai đầu ra, một đầu ra cho dự đoán nhãn thông qua một softmax layer và một đầu ra khác dự đoán bounding box (kí hiệu là bbox) dựa trên hồi quy tuyến tính. Quá trình này sau đó được lặp lại nhiều lần cho mỗi vùng RoI trong một hình ảnh.

Kiến trúc của mô hình được tóm tắt trong hình dưới đây, được lấy từ bài báo.



Hình 2. 2. Kiến trúc single model Fast R-CNN

Ở bước đầu ta áp dụng một mạng Deep CNN để trích xuất ra feature map. Thay vì warp image của region proposal như ở R-CNN chúng ta xác định ngay vị trí hình chiếu của của region proposal trên feature map thông qua phép chiếu RoI projection. Vị trí này sẽ tương đối với vị trí trên ảnh gốc. Sau đó tiếp tục truyền output qua các layer RoI pooling layer và các Fully Connected layers để thu được RoI feature véc tơ. Sau đó kết quả đầu ra sẽ được chia làm 2 nhánh. Một nhánh giúp xác định phân phối xác suất theo các class của 1 vùng quan tâm RoI thông qua hàm softmax và nhánh còn xác định tọa độ của bounding box thông qua hồi quy các offsets.

Mô hình này nhanh hơn đáng kể cả về huấn luyện và dự đoán, tuy nhiên vẫn cần một tập hợp các region proposal được đề xuất cùng với mỗi hình ảnh đầu vào.

c. Faster R-CNN

Kiến trúc mô hình đã được cải thiện hơn nữa về cả tốc độ huấn luyện và phát hiện được đề xuất bởi Shaoqing Ren.

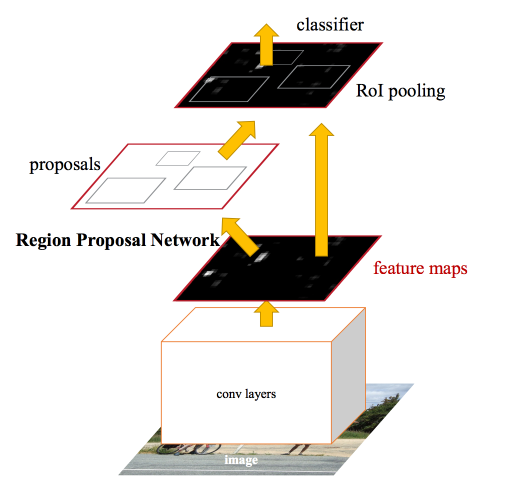
Kiến trúc này mang lại độ chính xác cao nhất đạt được trên cả hai nhiệm vụ phát hiện và nhận dạng đối tượng tại các cuộc thi ILSVRC-2015 và MS COCO-2015.

Kiến trúc được thiết kế để đề xuất và tinh chỉnh các region proposals như là một phần của quá trình huấn luyện, được gọi là mạng đề xuất khu vực (Region Proposal Network), hoặc RPN. Các vùng này sau đó được sử dụng cùng với mô hình Fast R-CNN trong một thiết kế mô hình duy nhất. Những cải tiến này vừa làm giảm số lượng region proposal vừa tăng tốc hoạt động trong thời gian thử nghiệm mô hình lên gần thời gian thực với hiệu suất tốt nhất. Tốc độ là 5fps trên một GPU.

Mặc dù là một mô hình đơn lẻ duy nhất, kiến trúc này là kết hợp của hai modules:

* Mạng đề xuất khu vực (Region Proposal Network). Mạng CNN để đề xuất các vùng và loại đối tượng cần xem xét trong vùng.
* Fast R-CNN: Mạng CNN để trích xuất các features từ các region proposal và trả ra các bounding box và nhãn.

Cả hai modules hoạt động trên cùng một output của một mạng deep CNN. Mạng RPN hoạt động như một cơ chế attention cho mạng Fast R-CNN, thông báo cho mạng thứ hai về nơi cần xem hoặc chú ý.



Hình 2.3. Kiến trúc mô hình Faster R-CNN

Ở giai đoạn sớm sử dụng một mạng deep CNN để tạo ra một feature map. Khác với Fast R-CNN, kiến trúc này không tạo RoI ngay trên feature map mà sử dụng feature map làm đầu vào để xác định các region proposal thông qua một RPN network. Đồng thời feature maps cũng là đầu vào cho classifier nhằm phân loại các vật thể của region proposal xác định được từ RPN network.

RPN hoạt động bằng cách lấy đầu ra của một mạng pretrained deep CNN, chẳng hạn như VGG-16, và truyền feature map vào một mạng nhỏ và đưa ra nhiều region proposals và nhãn dự đoán cho chúng. Region proposals là các bounding boxes, dựa trên các anchor boxes hoặc hình dạng được xác định trước được thiết kế để tăng tốc và cải thiện khả năng đề xuất vùng. Dự đoán của nhãn được thể hiện dưới dạng nhị phân cho biết region proposal có xuất hiện vật thể hoặc không.

Một quy trình huấn luyện xen kẽ được sử dụng trong đó cả hai mạng con được đào tạo cùng một lúc. Điều này cho phép các tham số trong feature detector của deep CNN được tinh chỉnh cho cả hai tác vụ cùng một lúc.

Tại thời điểm viết, kiến trúc Faster R-CNN này là đỉnh cao của họ model R-CNN và tiếp tục đạt được kết quả gần như tốt nhất trong các nhiệm vụ nhận diện đối tượng.

2.1.2. YOLO

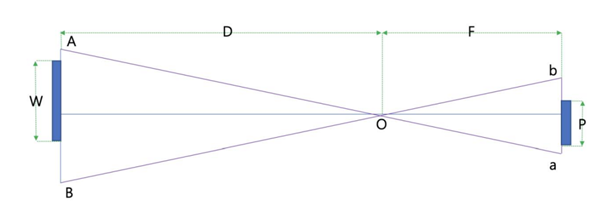
YOLO đã xuất hiện trên thị trường máy tính với bài báo năm 2015 của Joseph Redmon “You Only Look Once: Unified, Real-Time Object Detection”, và ngay lập tức được các nhà nghiên cứu thị giác máy tính đồng nghiệp chú ý. YOLO là một cách tiếp cận mới để phát hiện nhiều đối tượng hiện diện trên một hình ảnh theo thời gian thực, bằng cách vẽ các bounding boxes xung quanh chúng. Nó chuyển hình ảnh qua thuật toán CNN “một lần duy nhất” để lấy ra output, điều này giải thích cho cái tên của YOLO. Mặc dù tương đối giống với R-CNN, song, trên thực tế, YOLO chạy nhanh hơn rất nhiều so với Faster R-CNN vì kiến ​​trúc đơn giản hơn của nó. Không giống như Faster R-CNN, YOLO có thể phân loại và thực hiện hồi quy bounding box cùng một lúc. Với YOLO, lớp nhãn chứa các đối tượng, vị trí của chúng có thể được dự đoán trong nháy mắt. Hoàn toàn đi lệch khỏi pipeline CNN điển hình, YOLO coi việc phát hiện đối tượng như một vấn đề hồi quy bằng cách tách các bounding box theo không gian và xác suất lớp liên quan của chúng, được dự đoán bằng cách sử dụng một mạng nơ-ron duy nhất. Quá trình thực hiện cả dự đoán bounding box và tính toán xác suất lớp là một kiến ​​trúc mạng thống nhất mà YOLO đã giới thiệu ban đầu.

Thuật toán YOLO mở rộng các phương trình GoogleNet để được sử dụng như việc tính toán vận chuyển chuyển tiếp cơ sở của chúng, được cho là lý do đằng sau tốc độ và độ chính xác của việc phát hiện đối tượng theo thời gian thực của YOLO. So với kiến ​​trúc R-CNN, không giống như chạy một bộ phân loại trên một hộp giới hạn tiềm năng, sau đó đánh giá lại điểm xác suất, YOLO dự đoán các hộp giới hạn và xác suất lớp cho các hộp giới hạn đó đồng thời. Điều này tối ưu hóa thuật toán YOLO và là một trong những lý do quan trọng tại sao YOLO rất nhanh và ít có khả năng xảy ra lỗi để có thể sử dụng cho các dự đoán đối tượng trong thời gian thực.

2.2. Đo lường đối tượng thời gian thực

2.2.1. Đo khoảng cách thực bằng tính năng phát hiện của Trí tuệ nhân tạo

Các phương tiện tự hành, nhiều máy bay không người lái và rô bốt đều là những công nghệ mới đang trải qua sự bùng nổ sau những phát triển gần đây của Trí tuệ nhân tạo. Tất cả những công nghệ mới này về cơ bản đòi hỏi công nghệ để máy tính có thể nhận ra môi trường xung quanh. Trong số các công nghệ cho phép họ làm như vậy, một trong những công nghệ quan trọng nhất là công nghệ cho phép máy tính nhận biết / đo khoảng cách với các vật thể xung quanh khác. Hiện nay, Radar và Lidar được sử dụng phổ biến nhất để đạt được điều này. Radar tính toán khoảng cách thông qua sóng điện từ phản xạ. Radar cho phép máy tính tính toán khoảng cách ở một mức độ khá chính xác ngay cả trong thời gian ban đêm, nơi có ít ánh sáng. Tuy nhiên, các vật thể càng ở xa, các phép đo Radar càng có nhiều sai số. Lidar xác định khoảng cách, hình dạng và chất liệu của môi trường xung quanh nó thông qua sự phản xạ của các chùm tia của nó. Lidar có khả năng chống chọi với các điều kiện thời tiết khắc nghiệt và cũng có bước sóng ngắn hơn Radars. Tùy thuộc vào vị trí / lĩnh vực mà các công nghệ này được áp dụng, độ chính xác và tốc độ của các phép đo được yêu cầu có thể khác nhau rất nhiều. Bài báo này nhằm mục đích phân tích hình ảnh do máy ảnh nhập vào - với sự trợ giúp của công nghệ Trí tuệ nhân tạo - để đo khoảng cách giữa các đối tượng và máy ảnh.



Hình 2.4. Mối quan hệ giữa một số đối tượng và sự phản chiếu của nó trên máy ảnh

Phép đo khoảng cách giữa máy ảnh và một đối tượng nhất định được tính toán thông qua các nguyên tắc hoạt động của máy ảnh. Nếu kích thước thực của vật thể được đưa ra, thì việc tính toán khoảng cách giữa vật thể đó và máy ảnh rất đơn giản và dễ dàng. Điều này là do rất dễ hiểu cách máy ảnh hoạt động. Hình 2 là một lời giải thích đơn giản về cách hoạt động của máy ảnh. Bên trái của hình 2 cho biết một đối tượng thực tế và bên phải cho biết đối tượng đó xuất hiện như thế nào trên máy ảnh

2.2.2. Tính khoảng cách dựa vào kích thước có sẵn

Chúng ta có thể sử dụng việc số đo của các vật thể có sẵn để tính ra khoảng cách trong không gian thực bằng thư viện Computer Vision

A picture containing text, tiled

Description automatically generated

Hình 2.5. Ảnh minh họa

Ví dụ như *Hình 2.5*, việc xác định được tờ giấy A4 sẽ giúp xác định được đơn vị khoảng cách của tờ giấy, Dựa vào các điểm pixel chúng ta có thể hoàn toàn tính được giá trị đơn vị khoảng cách trên mỗi đơn vị pixel từ đó có thể tính xấp xỉ khoảng cách hay đơn vị khoảng cách của vật

CHƯƠNG 3: THUẬT TOÁN

3.1. Phát hiện vật thể

3.1.1. Tổng quan

Về độ chính xác thì YOLO có thể không phải là thuật toán tốt nhất nhưng nó là thuật toán nhanh nhất trong các lớp mô hình object detection. Nó có thể đạt được tốc độ gần như real time mà độ chính xác không quá giảm so với các model thuộc top đầu. YOLO là thuật toán object detection nên mục tiêu của mô hình không chỉ là dự báo nhãn cho vật thể như các bài toán classification mà nó còn xác định location của vật thể. Do đó YOLO có thể phát hiện được nhiều vật thể có nhãn khác nhau trong một bức ảnh, thay vì chỉ phân loại duy nhất một nhãn cho một bức ảnh.

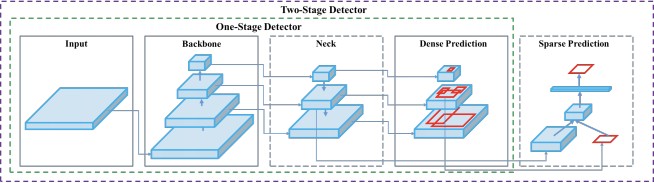
Trong đề tài này, nhóm chúng tôi sẽ ứng dụng một trong những thế hệ mới nhất của thuật toán YOLO ở thời điểm hiện tại – YOLOv5. Mô hình YOLOv5 được phát hành sau khi công bố phiên bản thứ tư khoảng 1 tháng, cả hai nhà nghiên cứu là Glenn Jocher (tác giả chính của YOLOv5) và Alexey Bochkovsky (tác giả chính của YOLOv4) đều áp dụng những sáng tạo tiên tiến nhất trong lĩnh vực thị giác máy tính vào thời điểm đó. Điều đó làm cho kiến ​​trúc của YOLOv4 và YOLOv5 rất giống nhau và nó khiến nhiều người không hài lòng với cái tên YOLOv5 (thế hệ thứ 5 của YOLO) khi nó không chứa đựng nhiều cải tiến vượt trội so với phiên bản YOLOv4 trước đó. Bên cạnh đó, Glenn đã không công bố bất kỳ bài báo nào cho YOLOv5, khiến nhiều người nghi ngờ hơn về YOLOv5. Tuy nhiên, sau một thời gian, YOLOv5 đã chứng tỏ hiệu suất cao hơn YOLOv4 trong một số hoàn cảnh nhất định và phần nào tạo được niềm tin trong cộng đồng thị giác máy tính bên cạnh YOLOv4.

3.1.2. YOLOv4

Thuật toán YOLO ban đầu được viết bởi Joseph Redmon. Sau 5 năm nghiên cứu và phát triển đến thế hệ thứ 3 của YOLO (YOLOv3), Joseph Redmon tuyên bố rút lui khỏi lĩnh vực thị giác máy tính và ngừng phát triển thuật toán YOLO. Vào tháng 4 năm 2020, Alexey Bochkovsky, dựa trên ý tưởng lý thuyết của Joseph Redmon, đã hợp tác với Chien Yao và Hon-Yuan để xuất bản YOLOv4. (Bochkovskiy, 2020)

a. Kiến trúc phát hiện đối tượng

Cùng với sự phát triển của YOLO, nhiều thuật toán phát hiện đối tượng với các cách tiếp cận khác nhau cũng đã đạt được những thành tựu đáng kể. Từ đó, hình thành hai khái niệm phát hiện đối tượng kiến trúc: Dò một tầng và Dò hai giai đoạn *(Hình 3.1).*



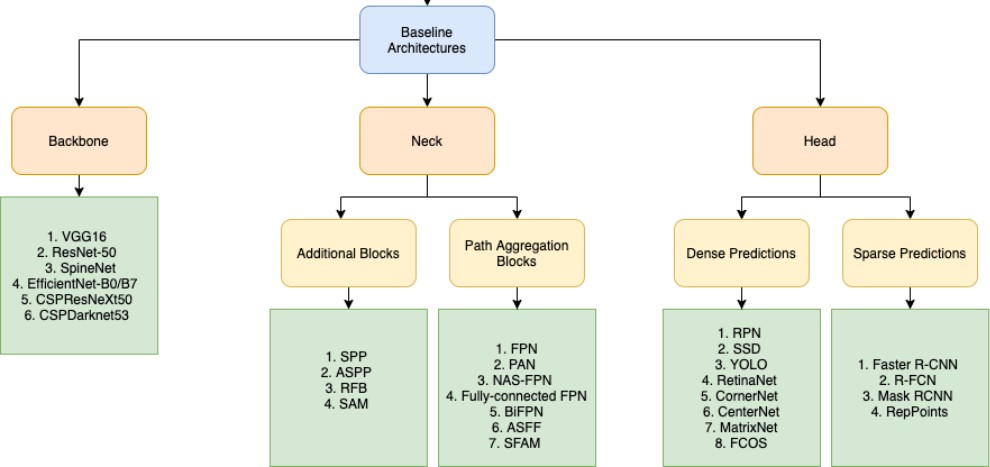
**Head**

Hình 3.1. Hai khái niệm về phát hiện đối tượng kiến trúc. (Solawetz, 2020)

Điểm chung của tất cả các kiến ​​trúc phát hiện đối tượng là các đặc điểm hình ảnh đầu vào sẽ được nén xuống thông qua bộ trích đặc tính (Backbone) rồi chuyển tiếp đến bộ phát hiện đối tượng (bao gồm cả Detection Neck và Detection Head) như trong *Hình 3.1.* Detection Neck (or Neck) hoạt động là một tập hợp đặc trưng có nhiệm vụ trộn và kết hợp các đặc trưng được hình thành trong Backbone để chuẩn bị cho bước phát hiện trong Detection Head (or Head).

Sự khác biệt xuất hiện ở đây mà Head chịu trách nhiệm phát hiện bao gồm bản địa hóa và phân loại cho mỗi hộp giới hạn. Máy dò hai giai đoạn thực hiện hai nhiệm vụ này riêng biệt và kết hợp các kết quả của chúng sau đó (Phát hiện thưa thớt), trong khi máy dò một giai đoạn thực hiện đồng thời (Phát hiện dày đặc) như trong *Hình 3.1* (Solawetz, 2020). YOLO là một máy dò một giai đoạn, do đó, *“Bạn Chỉ Nhìn Một lần”.*

Tác giả YOLOv4 đã thực hiện một loạt các thử nghiệm với nhiều ý tưởng đổi mới tiên tiến nhất của thị giác máy tính cho từng phần của kiến ​​trúc *(Hình 3.2).* (Bochkovskiy, và cộng sự, 2020)



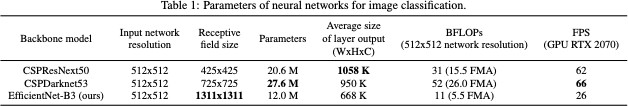
(One-stage)

(Two-stage)

Hình 3.2. Sơ đồ các ý tưởng đổi mới tiên tiến nhất được tác giả áp dụng cho từng phần của kiến trúc YOLOv4.

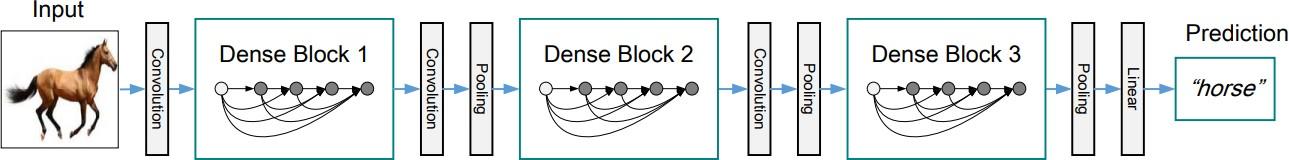
b. Backbone – CSPDarknet53

BackboneBackbone (bộ trích xuất tính năng) của mô hình YOLOv4 được các tác giả cân nhắc trong số 3 lựa chọn: CSPResNext53, CSPDarknet53 và EfficientNet-B3, mạng tích chập tiên tiến nhất tại thời điểm đó. Dựa trên lý thuyết và nhiều thử nghiệm, mạng nơ-ron CSP Darknet53 được xác định là mô hình tối ưu nhất *(Hình 3.3).*



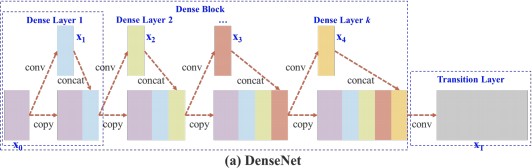
Bảng 3. 1. Bảng so sánh 3 mạng đường trục. (Huang, và cộng sự, 2018)

CSPResNext50 và CSPDarknet53 (CSP là viết tắt của Cross Stage Partial) đều có nguồn gốc từ kiến trúc DenseNet sử dụng đầu vào trước đó và nối nó với đầu vào hiện tại trước khi chuyển sang lớp dày đặc (Huang, et al., 2018). DenseNet được thiết kế để kết nối các lớp trong một mạng nơ-ron rất sâu với mục đích giảm bớt các vấn đề về vanishing gradient.



Hình 3. 3. Kiến trúc DenseNet với 3 khối Dense. Các lớp giữa 2 khối được gọi là các lớp chuyển tiếp. (Huang, và cộng sự, 2018)

Để làm rõ, mỗi giai đoạn của DenseNet bao gồm một khối dày đặc và một lớp chuyển tiếp *(Hình 3.3)*, mỗi khối dày đặc được xây dựng bằng 𝑘 lớp dày đặc. Đầu vào sau khi đi qua khối dày đặc sẽ đến lớp chuyển tiếp để thay đổi kích thước (mẫu xuống hoặc mẫu lên) thông qua tích chập và tổng hợp *(Hình 3.4)*. Đầu ra của lớp *ith* dày đặc sẽ kết hợp với đầu vào của chính nó để tạo thành đầu vào cho lớp (𝑖 + 1)*th* tiếp theo. Ví dụ, tại lớp dày 1*𝑠𝑡*, đầu vào 𝑥𝑜 sau khi chuyển tiếp qua các lớp tích chập đã tạo ra đầu ra 𝑥1. Sau đó, đầu ra 𝑥1 ghép với đầu vào của chính nó 𝑥𝑜 và kết quả được ghép đó trở thành đầu vào của lớp dày đặc 2𝑛𝑑 *(Hình 3.4).*

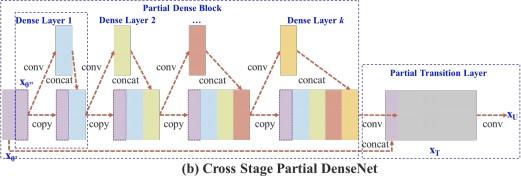


**Convolution**

**Pooling**

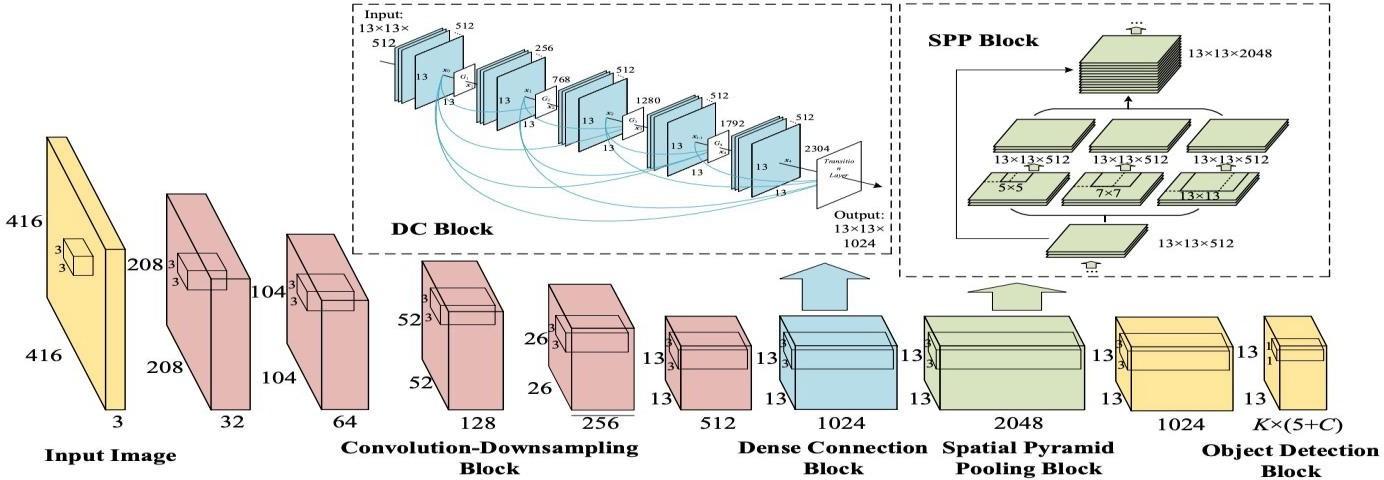
Hình 3. 4. Quá trình xử lý đầu vào trong một khối dày đặc. (Wang, et al., 2019)

CSP (Cross Stage Partial) dựa trên nguyên tắc tương tự của DenseNet ở trên ngoại trừ việc thay vì sử dụng bản đồ tính năng đầu vào kích thước đầy đủ ở lớp cơ sở, đầu vào sẽ được tách thành 2 phần. Một phần sẽ được chuyển tiếp qua khối dày đặc như bình thường và một phần khác sẽ được chuyển thẳng sang giai đoạn tiếp theo mà không cần xử lý *(Hình 3.5).* Điều này sẽ dẫn đến các lớp dày đặc khác nhau liên tục tìm hiểu thông tin gradient được sao chép. (Wang, et al., 2019)



Hình 3. 5. Quá trình xử lý đầu vào trong một khối đặc một phần. (Wang, et al., 2019)

Kết hợp những ý tưởng này với kiến trúc Darknet-53 trong YOLOv3, các khối còn lại được thay thế bằng các khối dày đặc. CSP duy trì các tính năng thông qua việc truyền bá, khuyến khích mạng sử dụng lại các tính năng và giảm số lượng tham số mạng, giúp bảo toàn các tính năng chi tiết để chuyển tiếp đến các lớp sâu hơn một cách hiệu quả hơn. Xét rằng sự gia tăng quá mức của các lớp chập kết nối dày đặc có thể dẫn đến giảm tốc độ phát hiện, chỉ khối chập cuối cùng có thể trích xuất các đặc điểm ngữ nghĩa phong phú hơn trong mạng đường trục Darknet-53 mới được cải thiện thành một khối dày đặc *(Hình 3.6).* (Huang, et al., 2019)

****

Hình 3. 6. Kết nối khối dày đặc và kiến trúc YOLOv2 dựa trên Kim tự tháp không gian. (Shah, 2020)

c. Neck – Additional block – SPP block

Trước khi chuyển tiếp đến kiến trúc tổng hợp tính năng ở neck, các bản đồ tính năng đầu ra của đường trục CSPDarknet53 đã được gửi đến một khối bổ sung (khối Spatial Pyramid Pooling) để tăng trường tiếp nhận và tách ra các tính năng quan trọng nhất *(Hình 3.6).*

Nhiều mô hình dựa trên CNN (mạng nơ ron tích chập) chứa các lớp được kết nối đầy đủ chỉ chấp nhận hình ảnh đầu vào có kích thước cụ thể. SPP ra đời với mục đích tạo ra đầu ra có kích thước cố định không phân biệt kích thước đầu vào. Không chỉ vậy, SPP còn giúp trích xuất các tính năng quan trọng bằng cách gộp các phiên bản đa quy mô của chính nó. Như trong *Hình 3.7*, các bản đồ tính năng đầu vào đã được sao chép thành 𝑛 phiên bản (trong trường hợp này là 𝑛 = 3) trong đó mỗi phiên bản được tiến hành gộp tối đa với các hạt nhân có kích thước khác nhau. Bằng cách đó, khối SPP đồng thời trích xuất 𝑛 các loại tính năng quan trọng khác nhau.

Diagram

Description automatically generated

Hình 3. 7. Khối SPP cổ điển. (Huang, et al., 2019)

Vì các lớp được kết nối đầy đủ đã bị xóa khỏi YOLOv2, thuật toán YOLO đã trở thành mô hình dựa trên FCN (mạng tích chập hoàn toàn) cho phép hình ảnh đầu vào có các kích thước khác nhau. Bên cạnh đó, YOLO phải đưa ra dự đoán và bản địa hóa về tọa độ của các ô giới hạn dựa trên ô lưới 𝑆 × 𝑆 được vẽ trên hình ảnh. Do đó, việc chuyển đổi các bản đồ đặc trưng hai chiều thành một vectơ một chiều có kích thước cố định không nhất thiết phải là mong muốn.

Vì lý do đó, khối SPP đã được sửa đổi để bảo toàn kích thước không gian đầu ra *(Hình 3.8).* Khối SPP mới được đặt liền kề với backbone *(Hình 3.6).* Để giảm số lượng bản đồ tính năng đầu vào được gửi đến khối SPP (từ 1024 xuống 512), một phép tích chập 1 × 1 được sử dụng giữa backbone và khối SPP. Sau đó, các bản đồ đối tượng đầu vào được sao chép và gộp lại theo các tỷ lệ khác nhau dựa trên cùng một nguyên tắc của khối SPP cổ điển ngoại trừ việc phần đệm được sử dụng để giữ kích thước không đổi của các bản đồ đối tượng đầu ra, khi đó 3 bản đồ đối tượng sẽ giữ nguyên kích thước của 𝑠𝑖𝑧𝑒𝑓𝑚𝑎𝑝 × 𝑠𝑖𝑧𝑒𝑓𝑚𝑎𝑝 × 512.

Khác với khối SPP cổ điển trong đó các bản đồ đối tượng *(Hình 3.7)* được chuyển đổi thành vectơ một chiều sau khi tiến hành gộp tối đa nhiều tỷ lệ, khối SPP mới *(Hình 3.8)* ghép 3 bản đồ đối tượng này được gộp chung với kích thước là 𝑠𝑖𝑧𝑒𝑓𝑚𝑎𝑝 × 𝑠𝑖𝑧𝑒𝑓𝑚𝑎𝑝 × 512 và bao gồm các bản đồ tính năng đầu vào để tránh mất các tính năng quan trọng trong trường hợp gộp tối đa 3 tỷ lệ là không đủ. Do đó, đầu vào không chỉ trích xuất các tính năng quan trọng giúp đào tạo dễ dàng hơn mà còn giữ nguyên kích thước không gian. (Wang, et al., 2019)

Diagram

Description automatically generated

Hình 3. 8. Khối SPP mới thích ứng với YOLO. (Huang, et al., 2019)

d. Neck – Feature Aggregation – PANet

Ảnh đầu vào sau khi chuyển tiếp qua backbone, các đặc trưng của ảnh được xử lý thành các đặc trưng ngữ nghĩa (hay các đặc trưng đã học). Nói cách khác, từ các lớp cấp thấp, hình ảnh đầu vào đi qua càng sâu, độ phức tạp của các đối tượng ngữ nghĩa sẽ tăng lên nhiều hơn trong khi độ phân giải không gian của bản đồ đối tượng địa lý sẽ giảm hơn do lấy mẫu xuống. Điều này dẫn đến mất thông tin không gian cũng như các tính năng chi tiết. Để bảo tồn những đặc điểm chi tiết này, Joseph Redmon đã áp dụng ý tưởng về kiến trúc Mạng Kim tự tháp đặc trưng (FPN) cho phần neck của YOLOv3.

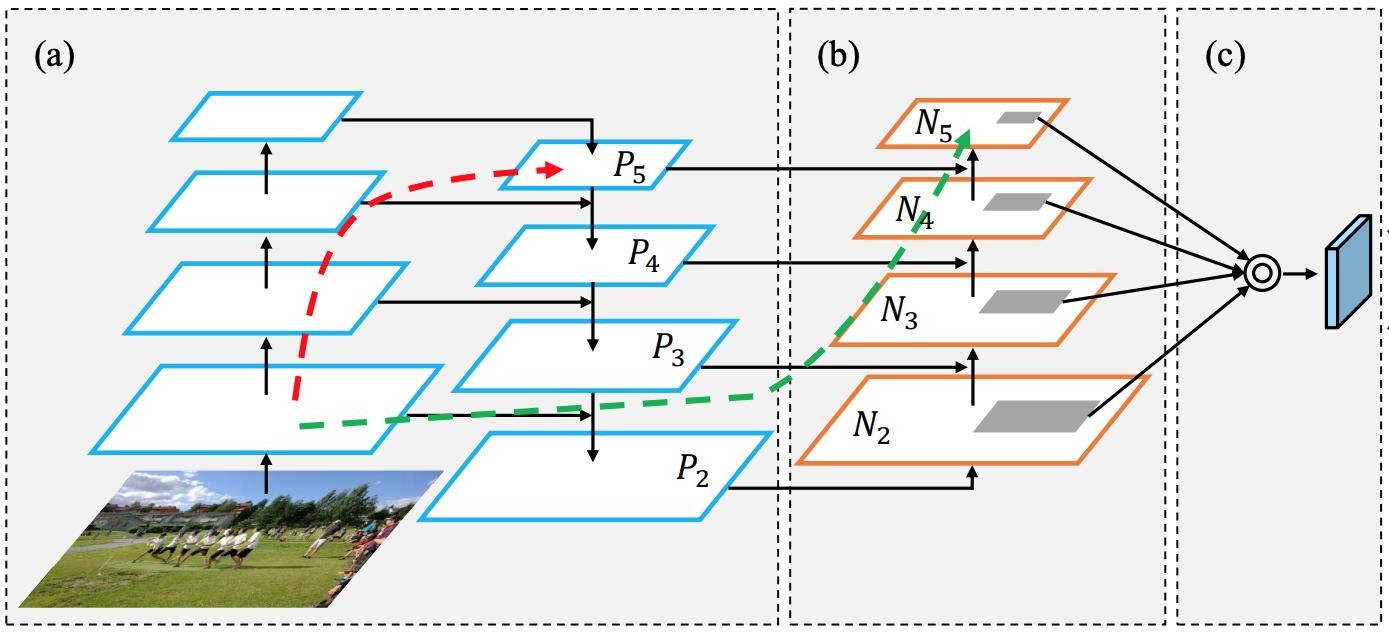
Kiến trúc FPN đã triển khai một đường dẫn từ trên xuống để chuyển các tính năng ngữ nghĩa (từ lớp cấp cao) và sau đó nối chúng với các tính năng chi tiết (từ lớp cấp thấp trong backbone) để dự đoán các đối tượng nhỏ trong máy dò quy mô lớn *(Hình 3.9)*. (Hui, 2020)

Diagram

Description automatically generated

Hình 3. 9. (a) Kiến trúc FPN ban đầu. (b) Kiến trúc FPN đã sửa đổi được sử dụng trong YOLOv3. (Gochoo, 2020)

Mạng tổng hợp đường dẫn (PAN) là một phiên bản nâng cao của FPN. Bởi vì luồng trong kiến trúc FPN là đường dẫn từ trên xuống, do đó chỉ bộ dò quy mô lớn từ các lớp cấp thấp trong FPN mới có thể đồng thời nhận các đặc trưng ngữ nghĩa từ các lớp cấp cao và các tính năng chi tiết từ các lớp cấp thấp ở lateral backbone *(Hình 3.10a)*. Hiện tại, máy dò quy mô nhỏ từ các lớp cấp cao trong FPN chỉ sử dụng các tính năng ngữ nghĩa để phát hiện đối tượng. Để cải thiện hiệu suất cho máy dò quy mô vừa và nhỏ, ý tưởng kết hợp các tính năng ngữ nghĩa và các tính năng chi tiết ở các lớp cấp cao đã được xem xét.



Backbone

FPN

Fine-grained features

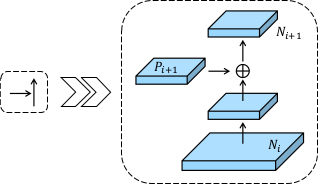
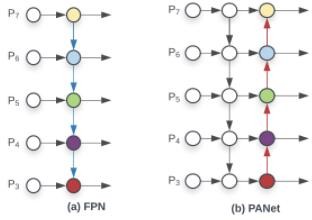
Semantic features

Hình 3. 10. Kiến trúc PANnet bao gồm (a) đường trục FPN, (b) tăng cường đường dẫn từ dưới lên, (c) tổng hợp tính năng thích ứng. (Liu và cộng sự, 2018)

Tuy nhiên, đối với mạng nơ-ron sâu ngày nay, backbone của chúng chứa rất nhiều lớp (có thể hơn 100 lớp). Do đó, trong FPN, các đối tượng địa lý chi tiết phải trải qua một chặng đường dài để di chuyển từ các lớp cấp thấp đến cấp cao (đường dẫn màu đỏ trong *Hình 3.10* hoặc đường dẫn màu đỏ trong *Hình 3.11a*).

Các tác giả của kiến ​​trúc PAN đã đề xuất thêm một đường dẫn nâng từ dưới lên bên cạnh đường dẫn từ trên xuống được sử dụng trong FPN *(Hình 3.10b).* Do đó, một “shortcut” đã được tạo ra để kết nối trực tiếp các tính năng chi tiết từ các lớp cấp thấp đến các lớp trên cùng (đường dẫn màu xanh lá cây trong *Hình 3.10* hoặc đường dẫn màu xanh lá cây trong *Hình 3.11b*). “Shortcut” này bao gồm ít hơn 10 lớp, cho phép dòng thông tin dễ dàng hơn (Liu, et al., 2018).

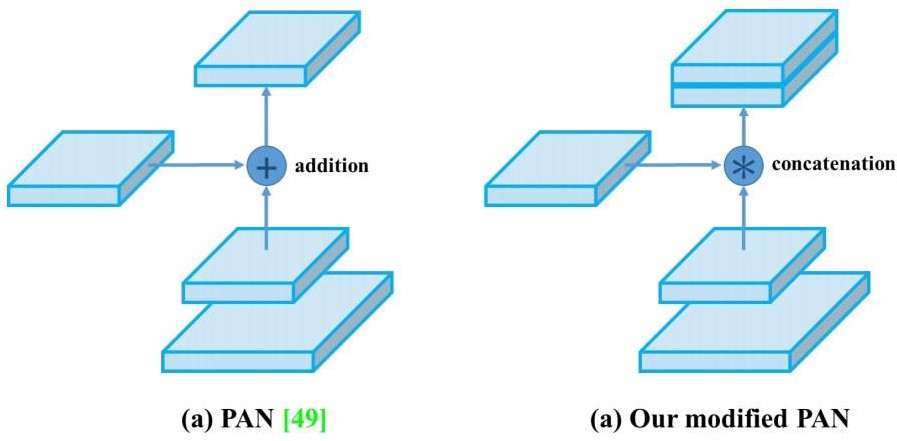
Như trong *Hình 3.10*, kiến ​​trúc backbone và FPN được trình bày giống nhau. Nhưng trên thực tế, backbone chứa rất nhiều lớp, 4 lớp được vẽ đại diện cho các lớp được nối với các bộ dò đa tỷ lệ trong kiến ​​trúc FPN, không đại diện cho toàn bộ backbone. Đó là lý do tại sao con đường màu xanh lá cây "shortcut" *(Hình 3.10)* đi qua ít hơn 10 lớp mặc dù dài hơn con đường màu đỏ trong cùng hình ảnh.



(c) Lateral connection

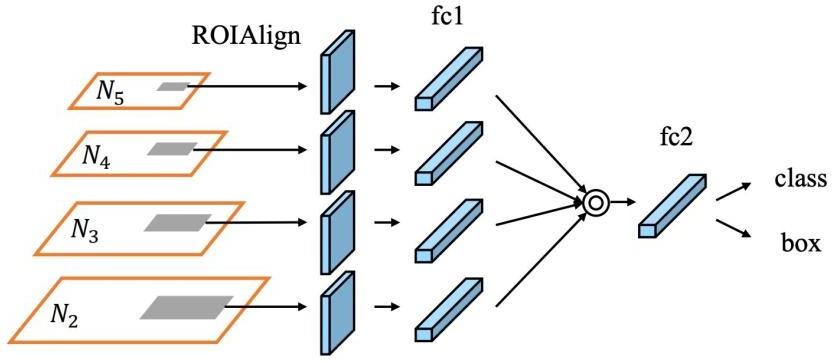
Hình 3. 11. (a) Kiến trúc FPN. (b) Kiến trúc PAN. (c) Kết nối trong đường dẫn nâng từ dưới lên. (Solawetz, 2020)

Đường dẫn tăng từ dưới lên có thể được coi là một bản sao của đường dẫn từ trên xuống FPN với mỗi giai đoạn chứa các lớp tạo ra bản đồ đối tượng có cùng kích thước không gian. Các bản đồ đặc trưng này được kết nối với kiến trúc bên bằng phép toán cộng phần tử *(Hình 3.12a),* trong khi trong kiến trúc PAN đã sửa đổi cho YOLOv4, các tác giả đã thay thế nó bằng phép toán nối *(Hình 3.12b)*. Điều này giúp luồng thông tin không bị mất trên các tính năng FPN cũng như các tính năng đường dẫn tăng từ dưới lên.



Hình 3. 12. (a) PAN gốc. (b) PAN được sửa đổi cho YOLOv4. (Bochkovskiy, và cộng sự, 2020)

Trong FPN, các dự đoán được thực hiện riêng biệt và độc lập ở các cấp độ quy mô khác nhau *(Hình 3.9).* Điều này có thể tạo ra các dự đoán trùng lặp và không sử dụng thông tin từ các bản đồ đối tượng địa lý khác. PAN đã hợp nhất tất cả các bản đồ tính năng đầu ra của kim tự tháp tăng từ dưới lên bằng cách sử dụng ROI (Vùng quan tâm) căn chỉnh và kết nối đầy đủ các lớp với hoạt động tối đa của phần tử (*Hình 3.10c* và *Hình 3.13*). Vì vậy, tất cả các biến thể của bản đồ đối tượng được tổng hợp và sử dụng cho các dự đoán. (Liu và cộng sự, 2018)



Hình 3. 13. PAN đã sử dụng ROI Align để tích hợp thích ứng và các lớp được kết nối đầy đủ để kết hợp các tính năng từ tất cả các giai đoạn. (Liu và cộng sự, 2018)

e. Head – YOLOv3

Trong trường hợp của máy dò một giai đoạn, chức năng của head là thực hiện các dự đoán dày đặc. Dự đoán dày đặc là dự đoán cuối cùng bao gồm một vectơ có chứa tọa độ hộp giới hạn dự đoán (tâm, chiều cao, chiều rộng), điểm tin cậy của dự đoán và các lớp xác suất.

YOLOv4 triển khai đầu giống hệt như YOLOv3 để phát hiện với các bước phát hiện dựa trên mỏ neo và ba cấp độ chi tiết phát hiện (Solawetz, 2020).

3.1.3. YOLOv5

a. Tổng quan

Bên cạnh đó, Glenn Jocher cũng là người phát minh ra phương pháp tăng cường dữ liệu Mosaic và được Alexey Bochkovsky thừa nhận trong bài báo YOLOv4 (Bochkovskiy, et al., 2020). Tuy nhiên, mô hình YOLOv5 của ông đã gây ra nhiều tranh cãi trong cộng đồng thị giác máy tính vì tên gọi và những cải tiến của nó.

YOLOv5 sở hữu những lợi thế về kỹ thuật. YOLOv5 được viết bằng ngôn ngữ lập trình Python thay vì C như các phiên bản trước. Điều đó làm cho việc cài đặt và tích hợp trên các thiết bị IoT dễ dàng hơn. Ngoài ra, cộng đồng PyTorch cũng lớn hơn cộng đồng Darknet, đồng nghĩa với việc PyTorch sẽ nhận được nhiều đóng góp và tiềm năng phát triển hơn trong tương lai. Do được viết bằng 2 ngôn ngữ khác nhau trên 2 framework khác nhau nên việc so sánh hiệu năng giữa YOLOv4 và YOLOv5 rất khó chính xác. Nhưng sau một thời gian, YOLOv5 đã chứng tỏ hiệu suất cao hơn YOLOv4 trong một số hoàn cảnh nhất định và phần nào tạo được niềm tin trong cộng đồng thị giác máy tính bên cạnh YOLOv4.

b. Điểm đáng chú ý ở YOLOv5

Như đã đề cập ở trên, kiến trúc YOLOv5 đã tích hợp những cải tiến mới nhất tương tự như kiến trúc YOLOv4, do đó không có nhiều sự khác biệt về mặt lý thuyết. Tác giả đã không xuất bản bài báo chi tiết mà chỉ đưa ra một kho lưu trữ trên Github và cập nhật các cải tiến ở đó. Bằng cách mổ xẻ mã cấu trúc của nó trong tệp yaml, mô hình YOLOv5 có thể được tóm tắt như sau (Jocher, 2020):

- Backbone: Focus structure, CSP network

- Neck: SPP block, PANet

- Head: YOLOv3 head sử dụng GIoU-loss

Điểm đáng chú ý mà tác giả YOLOv5 đề cập là sự khác biệt về mặt kỹ thuật. Joseph Redmon đã giới thiệu cấu trúc hộp neo trong YOLOv2 và quy trình chọn hộp neo có kích thước và hình dạng gần giống với hộp giới hạn chân lý mặt đất trong tập huấn luyện. Bằng cách sử dụng thuật toán phân cụm k-mean với các giá trị 𝑘 khác nhau, các tác giả đã chọn 5 hộp neo phù hợp nhất cho tập dữ liệu COCO (chứa 80 lớp) và sử dụng chúng làm mặc định. Điều đó làm giảm thời gian đào tạo và tăng độ chính xác của mạng.

Tuy nhiên, khi áp dụng 5 hộp neo này cho một tập dữ liệu duy nhất (chứa một lớp không thuộc 80 lớp trong tập dữ liệu COCO), các hộp neo này không thể nhanh chóng thích ứng với các hộp giới hạn chân lý cơ bản của tập dữ liệu duy nhất này. Ví dụ: tập dữ liệu về hươu cao cổ phù hợp với các hộp neo có hình dạng mỏng và cao hơn hộp hình vuông. Để giải quyết vấn đề này, các kỹ sư thị giác máy tính thường chạy thuật toán phân cụm k-mean trên tập dữ liệu duy nhất để có được các hộp neo phù hợp nhất cho dữ liệu trước tiên. Sau đó, các tham số này sẽ được cấu hình thủ công trong kiến ​​trúc YOLO.

Glenn Jocher đề xuất tích hợp quy trình lựa chọn hộp neo vào YOLOv5. Do đó, mạng không phải xem xét bất kỳ tập dữ liệu nào được sử dụng làm đầu vào, nó sẽ tự động "học" các hộp neo tốt nhất cho tập dữ liệu đó và sử dụng chúng trong quá trình đào tạo. (Solawetz, 2020)

3.2. Đo lường đối tượng thời gian thực

Đối với bài toàn xác định nhóm người thì việc khoảng cách giữa các cá đối tượng người là điều kiện để xác định thế nào là một nhóm người. Ở đây chúng tôi không sử dung tính khoảng cách cụ thể ngoài thực tế.

Điều này có thể làm phức tạp bài toán, việc xác định khoảng cách chiều sâu nếu 2 người bị đè Bbox là rất khó khăn vì phải xử lí cả chiều sâu của ảnh, làm cho việc xử lí thông tin chậm chạp hơn. Chúng tôi đã sử dụng cách tiếp cận đơn giản hơn dựa trên khái niệm đám đông của nhóm. Đối với môi trường lí tưởng, góc quay phù hợp thì việc xác định đám đông chỉ cần dựa vào điểm ảnh pixel.

**A picture containing text, outdoor, road, way

Description automatically generated**

Hình 3. 15. Ảnh minh họa

A picture containing outdoor, person, athletic game, person

Description automatically generated

Hình 3. 16. Ảnh minh họa

Việc xây dựng hàm khoảng cách này khá đơn giản, dựa vào đầu ra của yolov5, tôi đã lấy các giá trị xy tọa độ của Bbox để thực hiện, kết hợp với thư viện OpenCV, là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.   
Dựa vào các điểm ảnh pixel chúng tôi thực nghiệm và tìm ra độ dài pixel trong các ảnh phù hợp với khoảng cách góc quay, điều này làm cho khoảng cách pixel trong gần tương đương khoảng cách trong thời gian thực

CHƯƠNG 4: THỰC NGHIỆM

4.1. Chuẩn bị bộ dữ liệu chung

Để đào tạo mô hình phát hiện người, chúng tôi đã tạo tập dữ liệu về người của riêng mình, bằng cách chuẩn bị khoảng 1000 hình ảnh về người từ web. Để đạt được mô hình YOLOv5 mạnh mẽ, chúng tôi sẽ thêm tối đa 10% ảnh nền để giảm lỗi xác định sai. Vì tập dữ liệu của tôi chưa đủ lớn, nên tôi sẽ thu hẹp quy trình đào tạo bằng cách sử dụng các kĩ thuật transfer learning.

A picture containing outdoor, building, bus, road

Description automatically generated

Hình 4. 1. Ảnh minh họa

Hầu hết các nền tảng chú thích đều hỗ trợ xuất ở định dạng ghi label YOLO, cung cấp một tệp văn bản chú thích cho mỗi hình ảnh. Mỗi tệp văn bản chứa một chú thích hộp giới hạn cho mỗi đối tượng trong hình ảnh. Các chú thích được chuẩn hóa theo kích thước hình ảnh và nằm trong phạm vi từ 0 đến 1. Chúng được thể hiện ở định dạng sau:

<object-class-ID> <X center> <Y center> <Chiều rộng hộp> <Chiều cao hộp>

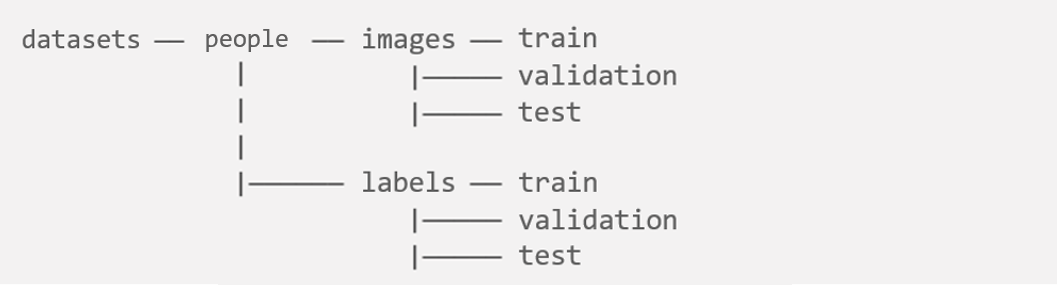
Nếu có hai đối tượng trong hình ảnh, nội dung của tệp văn bản chú thích YOLO có thể trông giống như sau:

Background pattern

Description automatically generated

4.1.1. Cấu trúc thư mục dữ liệu

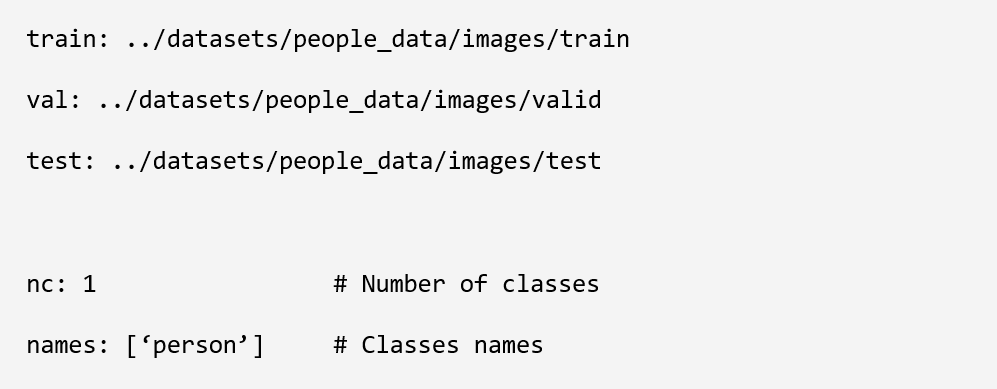
Để tuân thủ cấu trúc thư mục Ultralytics, dữ liệu được cung cấp theo cấu trúc sau:



****4.1.2. Tệp cấu hình****

Các cấu hình cho khóa đào tạo được chia thành ba tệp YAML, được cung cấp cùng với chính repo. Chúng tôi sẽ tùy chỉnh các tệp này tùy thuộc vào nhiệm vụ, để phù hợp với nhu cầu mong muốn của chúng tôi.

Tệp **cấu hình dữ liệu** mô tả các tham số tập dữ liệu. Vì chúng tôi đang đào tạo về bộ dữ liệu người tùy chỉnh của mình, chúng tôi sẽ chỉnh sửa tệp này và cung cấp: đường dẫn, bộ dữ liệu xác thực và kiểm tra (tùy chọn); số lớp (nc); và tên của các lớp theo cùng thứ tự với chỉ mục của chúng. Trong báo cáo này, chúng tôi chỉ có một lớp, có tên là “person”. Chúng tôi đặt tên tệp cấu hình dữ liệu tùy chỉnh của mình là custom\_data.yaml và đặt nó trong thư mục “data”. Nội dung của tệp YAML này như sau:



Khi quá trình huấn luyện được khởi tạo từ các trọng số đã được huấn luyện trước như trong hướng dẫn này, không cần phải chỉnh sửa tệp cấu hình mô hình vì mô hình sẽ được trích xuất với các trọng số đã được huấn luyện trước.

****4.2. Training****

Đối với vấn đề cần độ chính xác tốt và tốc độ xử lí nhanh, chúng tôi sẽ đào tạo mô hình kích thước tham số vừa phải và đủ mạnh là YOLOv5m, mặc dù có thể sử dụng mô hình lớn hơn để cải thiện kết quả. Các cách tiếp cận đào tạo khác nhau có thể được xem xét cho các tình huống khác nhau và ở đây chúng tôi sẽ ưu tiên việc vừa mô hình vừa phải có tốc độ cao vừa phải có độ chính xác nhanh.

4.2.1. Transfer learning

Bắt đầu đào tạo từ mô hình trước:

Vì tập dữ liệu về người của chúng tôi tương đối nhỏ (~ 1000 hình ảnh), việc transfer learning được mong đợi sẽ tạo ra kết quả tốt hơn so với huấn luyện từ đầu. Mô hình mặc định của Ultralytic đã được đào tạo trước qua tập dữ liệu COCO, mặc dù cũng có hỗ trợ cho các mô hình được đào tạo trước khác (VOC, Argoverse, VisDrone, GlobalWheat, xView, Objects365, SKU-110K). COCO là một tập dữ liệu phát hiện đối tượng với hình ảnh từ các cảnh hàng ngày. Nó chứa 80 lớp, bao gồm lớp người liên quan, Mô hình của chúng tôi sẽ được khởi tạo với các trọng số từ một mô hình COCO được đào tạo trước, bằng cách chuyển tên của mô hình cho đối số 'weights'. Mô hình được đào tạo trước sẽ được tự động tải xuống.

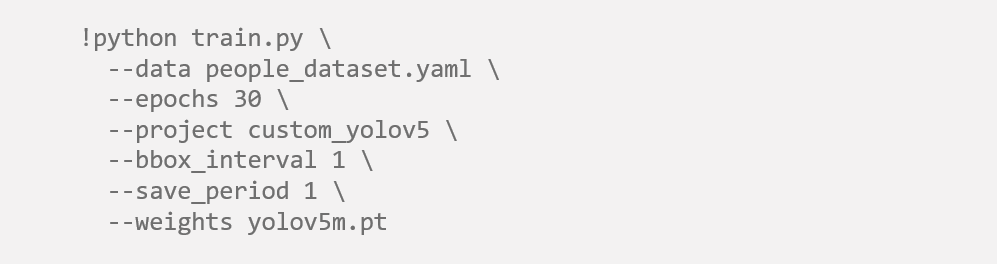
Weight and Bias

Đối với việc đào tạo mô hình việc không có một thiết bị phù hợp hay một GPU không đủ khỏe thì việc đào tạo mô hình là không thể. Vì thế chúng tôi quyết định đảo tạo mô hình trên Google Colaboratory có GPU NVIDIA Tesla P100 được cung cấp miễn phí kết hợp với phần mềm thứ 3 là Weight&Bias.

Weights & Biases là nền tảng học máy dành cho các nhà phát triển để xây dựng các mô hình tốt hơn nhanh hơn. Sử dụng các công cụ nhẹ, có thể tương tác của W&B để nhanh chóng theo dõi các thử nghiệm, phiên bản và lặp lại trên tập dữ liệu, đánh giá hiệu suất mô hình, tái tạo mô hình, trực quan hóa kết quả và hồi quy tại chỗ, đồng thời chia sẻ phát hiện với đồng nghiệp.

4.2.2. Khai thác tính năng

Các mô hình bao gồm hai phần chính: các lớp backbone đóng vai trò như một bộ trích xuất đặc tính và các lớp đầu tính toán các dự đoán đầu ra. Để bù đắp thêm cho kích thước tập dữ liệu nhỏ, chúng tôi sẽ sử dụng backbone giống như mô hình COCO được đào tạo trước và chỉ đào tạo phần đầu của mô hình.



YOLOv5 wandb đã được tích hợp sẵn, vì vậy tất cả những gì tôi cần làm là định cấu hình ghi nhật ký bằng các đối số dòng lệnh.

* --project đặt dự án W & B mà chúng tôi đang kết nối
* --epochs là số kỷ nguyên
* --upload\_dataset kết nối wandb, yêu cầu tải tập dữ liệu lên dưới dạng bảng. Tại các khoảng thời gian đều đặn được thiết lập bởi --bbox\_interval, đầu ra của mô hình trên tập hợp xác nhận cũng sẽ được ghi vào W&B.
* --save\_period đặt số lượng kỷ nguyên chờ giữa việc logging các checkpoint của mô hình. Nếu không được đặt, chỉ mô hình được đào tạo cuối cùng mới được ghi lại.
* --weights mô hình sử dụng để train

Ngay cả khi không có các đối số này, các chỉ số mô hình cơ bản và một số kết quả đầu ra của mô hình vẫn sẽ được lưu vào W&B.

Sau khi train thì kết quả dữ liệu được hiển thị dưới dạng các bảng trực quá như dưới:

Graphical user interface, chart

Description automatically generated

Hình 4. 2. Kết quả dữ liệu

Để hiểu rõ hơn về kết quả, hãy cùng tóm tắt các số liệu và tổn thất của YOLOv5. Chức năng mất YOLO bao gồm ba phần:

1. **box\_loss** - mất hồi quy hộp giới hạn (Lỗi bình phương trung bình).
2. **obj\_loss** - độ tin cậy của sự hiện diện của đối tượng là mất đối tượng (Binary Cross Entropy).
3. **cls\_loss** - mất phân loại (Cross Entropy).

Vì dữ liệu của chúng tôi chỉ có một lớp, nên không có sự phân biệt sai lớp nào và lỗi phân loại liên tục bằng 0.

**Precision** đo lường mức độ chính xác của các dự đoán bbox (True Postitive / (True Positive + False Positive)) và **Recall** đo lường bao nhiêu bbox thực được dự đoán chính xác (True Postitive / (True Postitive + False Negative)). 'mAP\_0,5' là Độ chính xác trung bình trung bình (mAP) tại ngưỡng IoU (Giao điểm qua Liên minh) là 0,5. 'mAP\_0,5: 0,95' là mAP trung bình trên các ngưỡng IoU khác nhau, nằm trong khoảng từ 0,5 đến 0,95. 

A group of people standing in front of a sign

Description automatically generated with medium confidence

Hình 4. 3. Kết quả dự đoán của mô hình trên tập huấn luyện

A picture containing text, building, outdoor, street

Description automatically generated

Hình 4. 4. Kết quả dự đoán của mô hình trên tập huấn luyện

A picture containing text, road, outdoor, tree

Description automatically generated

Hình 4. 5. Kết quả dự đoán của mô hình

Như vậy, mô hình hoạt động rất tốt trên điều kiện của bài toán. Mặc dù tất nhiên vẫn còn một số bất tiện do vị trí góc quay của máy quay khiến cho người bị khất, đôi khi không bắt được. Nhưng điều này hoàn toàn có thể giải quyết sử dụng các góc máy khác nhau cũng như một số loại camera chuyên dụng khác nhau.

4.3. Đo lường đối tượng thời gian thực

Với đầu ra của YOLOv5 thì việc chuyền tham số vào hàm xây dựng trước là việc rất đơn giản, các giá trị này bao gồm tạo độ của Bbox vì thế ta hoàn toàn có thể dựa vào đó để tính khoảng cách giữa các điểm ảnh. Khoảng cách chúng tôi chọn ở đây là khoảng từ 200 đến 250 nếu lớn hơn khoảng cách này thì sẽ không thông báo, nếu khoảng cách giữa 2 người nhỏ hơn 200 thì đây được gọi là đám đông.

A group of people stand outside a bus stop

Description automatically generated with low confidence

Hình 4. 6. Kết quả dự đoán của mô hình

Diagram

Description automatically generated

Hình 4. 7. Kết quả dự đoán của mô hình

Như vậy việc xác định đám đông bằng độ dài của điểm ảnh là hoàn toàn khải thi khi đặt camera ở môi trường lí tưởng.

CHƯƠNG 5: TỔNG KẾT

Qua quá trình học hỏi, nghiên cứu, thử nghiệm và đánh giá. Bằng việc tận dụng các nghiên cứu cũng như các mô hình mã nguồn mở về xử lý ảnh, phát hiện đối tượng đã giúp nhóm hoàn thành đề tài ***Nghiên cứu và xây dựng hệ thống phát hiện đám đông hỗ trợ cảnh báo nguy cơ lây nhiễm virus Corona.***

Nhóm đã tìm hiểu, nghiên cứu, cài đặt thử nghiệm các thuật toán học sâu tiên tiến liên quan đến các bài toán phát hiện đối tương, theo dõi đối tượng, đo lường khoảng cách. Ngoài ra, nhóm đã xây dựng và ứng dụng một bộ dữ liệu mới dành cho bài toán phát hiện đám đông. Nhóm đã kết hợp các sự phát minh mới trong sự phát triển của khoa học kĩ thuật hay AI nói riêng, đồng thời sử dụng các thuật toán khác áp dụng vào tình hình xã hội. Hệ thống phát hiện đám đông hoạt động tốt với độ chính xác trên 80% với các đối tượng đám đông di chuyển trong thực tế.

Nhóm xin chân thành cảm ơn cô Nguyễn Lan Anh đã hướng dẫn nhóm thực hiện đề tài này. Nghiên cứu khoa học này đã giúp nhóm củng cố và rèn luyện được rất nhiều kỹ năng trong việc xây dựng mô hình xử lý hình ảnh để ứng dụng vào thực tiễn. Đây chắc chắn sẽ là hành trang rất hữu ích trong sự nghiệp của chúng em trong tương lai.

Tuy nhiên, nhóm vẫn có có những thiếu sót do hệ thống vẫn còn rất sơ sài chưa thực sự tối ưu, chưa có giao diện tinh chỉnh và sử dụng dễ dàng. Vì vậy nên nhóm rất mong được các thầy cô đóng góp thêm các ý kiến để nhóm có thể cải thiện hệ thống này, tiến tới ứng dụng thực tiễn trong tương lai.

Nhóm xin trân trọng cảm ơn!

TÀI LIỆU THAM KHẢO

[1] Vittorio Murino, “Groups and Crowds: Detection, Tracking and Behavior

Analysis of People Aggregations,” trong *VISIGRAPP Conference*, Rome, Italy, 2016.

[2]. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

[3]. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, UC Berkeley, 2014. Rich feature hierarchies for accurate object detection and semantic segmentation

[4]. Ross Girshick, 2015. Fast R-CNN

[5]. Bochkovskiy, A., Wang, C.-Y., & Mark Liao, H.-Y. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

[6]. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional.

[7]. Huang, G., Liu, Z., & Maaten, L. v. (2018). Densely Connected Convolutional Networks.

[8]. Huang, Z., & Wang, J. (2019). DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection.

[9]. Hui, J. (2020). YOLOv4. Medium.

[10]. ODSC Medium, 2018. Overview of the YOLO Object Detection Algorithm