**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 14: XÂY DỰNG HỆ THỐNG XÁC NHẬN ĐỐI TƯỢNG VÀ ĐẾM ĐỐI TƯỢNG TRONG ẢNH**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20211343 | Nguyễn Duy Công | DCCNTT12.10.5 |
| **2** | 20211492 | Nguyễn Văn Hiệp | DCCNTT12.10.5 |
| **3** | 20211257 | Nguyễn Văn Sinh Hùng | DCCNTT12.10.5 |
| **4** | 20211267 | Phạm Thế Sơn | DCCNTT12.10.5 |
| **5** | 20211341 | Phùng Quang Tuân | DCCNTT12.10.5 |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 14: XÂY DỰNG HỆ THỐNG XÁC NHẬN ĐỐI TƯỢNG VÀ ĐẾM ĐỐI TƯỢNG TRONG ẢNH**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20211343 | Nguyễn Duy Công | DCCNTT12.10.5 |
| **2** | 20211492 | Nguyễn Văn Hiệp | DCCNTT12.10.5 |
| **3** | 20211257 | Nguyễn Văn Sinh Hùng | DCCNTT12.10.5 |
| **4** | 20211267 | Phạm Thế Sơn | DCCNTT12.10.5 |
| **5** | 20211341 | Phùng Quang Tuân | DCCNTT12.10.5 |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 14**  **Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**  **Lớp Tín chỉ:**  **XATGMT.03.K12.05.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *Lương Thị Hồng Lan* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Nguyễn Duy Công** | **Nguyễn Văn Hiệp** | **Nguyễn Văn Sinh Hùng** | **Phạm Thế Sơn** | **Phùng Quang Tuân** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 20211343 | 20211492 | 20211257 | 20211267 | 20211341 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |  |

**Mục Lục**

[**Chương 1: Tổng quan** 7](#_Toc184646133)

[1.1. Bài toán nhận dạng: 7](#_Toc184646134)

[1.1.1. Bài toán nhận dạng đối tượng: 7](#_Toc184646135)

[1.1.2.Một số ứng dụng: 8](#_Toc184646136)

[1.2. Tổng quan các kỹ thuật áp dụng cho bài toán nhận dạng 10](#_Toc184646137)

[1.2.1. Lớp mô hình họ R-CNN 10](#_Toc184646138)

[1.2.2. Lớp các mô hình họ YOLO 20](#_Toc184646139)

[1.3. Ngôn ngữ lập trình 25](#_Toc184646140)

[**Chương 2: Xây dựng hệ thống nhận dạng** 30](#_Toc184646141)

[2.1. Mô tả bài toán: 30](#_Toc184646142)

[2.1.1 Bài toán nhận diện đối tượng 30](#_Toc184646143)

[2.1.2 Bài toán phân lớp và đếm đối tượng 31](#_Toc184646144)

[2.1.3 Phân tích luồng webcam 31](#_Toc184646145)

[2.2. Xây dựng hệ thống 33](#_Toc184646146)

[2.2.1. YOLOV3 33](#_Toc184646147)

[2.2.2.Đếm 39](#_Toc184646148)

[**Chương 3: Kết quả thực nghiệm** 41](#_Toc184646149)

[3.1 Cơ sở dữ liệu 41](#_Toc184646150)

[3.1.1 Dữ liệu 41](#_Toc184646151)

[3.1.2Tiền xử lý dữ liệu: 41](#_Toc184646152)

[3.2.Kết quả thực nghiệm 42](#_Toc184646153)

[**Tài liệu tham khảo** 45](#_Toc184646154)

# **Chương 1: Tổng quan**

## **1.1. Bài toán nhận dạng:**

### 1.1.1. Bài toán nhận dạng đối tượng:

Nhận dạng đối tượng là một thuật ngữ chung để mô tả một tập hợp các nhiệm vụ thị giác máy tính có liên quan liên quan đến việc xác định các đối tượng trong ảnh kỹ thuật số.

Phân loại hình ảnh liên quan đến việc dự đoán lớp của một đối tượng trong một hình ảnh. Định vị vật thể đề cập đến việc xác định vị trí của một hoặc nhiều đối tượng trong một hình ảnh và vẽ bounding box xung quanh chúng. Phát hiện đối tượng kết hợp hai nhiệm vụ trên và thực hiện cho một hoặc nhiều đối tượng trong hình ảnh. Chúng ta có thể phân biệt giữa ba nhiệm vụ thị giác máy tính cơ bản trên thông qua input và output của chúng như sau:

- Phân loại hình ảnh: Dự đoán nhãn của một đối tượng trong một hình ảnh.

+ Input: Một hình ảnh với một đối tượng, chẳng hạn như một bức ảnh.

+ Output: Nhãn lớp (ví dụ: một hoặc nhiều số nguyên được ánh xạ tới nhãn lớp).

- Định vị đối tượng: Xác định vị trí hiện diện của các đối tượng trong ảnh và cho biết vị trí của chúng bằng bounding box.

+ Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.

+ Output: Một hoặc nhiều bounding box được xác định bởi tọa độ tâm, chiều rộng và chiều cao.

- Phát hiện đối tượng: Xác định vị trí hiện diện của các đối tượng trong bounding box và nhãn của các đối tượng nằm trong một hình ảnh.

+ Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.

+ Output: Một hoặc nhiều bounding box và nhãn cho mỗi bounding box.

Một số định nghĩa khác cũng rất quan trọng trong computer vision là phân đoạn đối tượng (object segmentation), trong đó các đối tượng được nhận dạng bằng cách làm nổi bật các pixel cụ thể của đối tượng thay vì bounding box. Và image captioning kết hợp giữa các kiến trúc mạng CNN và LSTM để đưa ra các lý giải về hành động hoặc nội dung của một bức ảnh.

### 1.1.2.Một số ứng dụng:

- Nhận dạng biển số – sử dụng cả công nghệ phát hiện đối tượng và nhận dạng ký tự quang học (OCR) để nhận dạng các ký tự chữ và số trên biển số xe. Tính năng phát hiện đối tượng được sử dụng thông qua việc lưu giữ lại hình ảnh và phát hiện các đối tượng cụ thể như xe cộ, phương tiện đi lại trên bức ảnh đó. Sau khi đã xác định được tọa độ của phương tiện trên ảnh, chúng ta có thể sử dụng các mô hình Object Detection khác để phát hiện biển số xe và các ký tự trên biển số. Từ đó, áp dụng các mô hình OCR hoặc image classification để nhận dạng ký tự trên hình, ánh xạ từ dạng dữ liệu hình ảnh sang dạng chữ (text).

A group of cars driving on a highway

Description automatically generated

- Phát hiện và nhận dạng khuôn mặt – một trong những ứng dụng chính của phát hiện đối tượng là nhận diện và phát hiện khuôn mặt. Với sự trợ giúp của các thuật toán hiện đại, chúng ta có thể phát hiện khuôn mặt người trong một hình ảnh hoặc video.

A person looking at camera

Description automatically generated

- Theo dõi đối tượng – Công nghệ này có thể ứng dụng để theo dõi chuyển động của một đối tượng hay đồ vật cụ thể. Chẳng hạn, trong khi xem một trận bóng chày hoặc cricket, quả bóng có thể bị đập ra xa. Trong những tình huống này, thuật toán giúp chúng ta phát hiện, nhận dạng và theo dõi (tracking) vị trí và đường bay của quả bóng hiện tại.

A screenshot of a crosswalk with people walking

Description automatically generated

- Ô tô tự lái – đối với ô tô tự lái, điều quan trọng là phải nghiên cứu các yếu tố môi trường xung quanh ô tô khi lái xe. Một mô hình phát hiện đối tượng được đào tạo trên nhiều đối tượng để phát hiện các thực thể khác nhau cản trở việc lái xe nhằm đảo bảo an toàn cho người lái.

A car dashboard with green lines

Description automatically generated with medium confidence

- Người máy – Hiện nay thuật toán phát hiện đối tượng thường được ứng dụng trong robot để nhận diện và di chuyển các vật nặng, giảm thiểu những công việc về thể chất cho con người và tự động hóa công việc.

A robot carrying boxes in a warehouse

Description automatically generated

## **1.2. Tổng quan các kỹ thuật áp dụng cho bài toán nhận dạng**

### 1.2.1. Lớp mô hình họ R-CNN

1.2.1.1. R-CNN (2014)

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 Extraction): 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:

A diagram of a person

Description automatically generated

**Hình 1:** Sơ đồ pipeline xử lý trong mô hình mạng R-CNN (được trích xuất từ bài báo gốc). 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 mà bạn đọc sẽ được tìm hiểu ở bài viết tiếp theo giới thiệu về mạng SSD.

Trích xuất đặc trưng về bản chất là một mạng CNN học sâu, ở đây là AlexNet, mạng đã giành chiến thắng trong cuộc thi phân loại hình ảnh ILSVRC-2012. Đầu ra của CNN là một vectơ 4096 chiều mô tả nội dung của hình ảnh được đưa đến một mô hình SVM tuyến tính để phân loại.

Đây là một ứng dụng tương đối đơn giản và dễ hiểu của CNN đối với vấn đề object localization và object detection. 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.

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Phát hiện đối tượng chính xác : R-CNN cung cấp khả năng phát hiện đối tượng chính xác bằng cách tận dụng các tính năng tích chập dựa trên vùng. Nó vượt trội trong các tình huống mà việc định vị và nhận dạng đối tượng chính xác là rất quan trọng. | Độ phức tạp tính toán : R-CNN có khối lượng tính toán lớn. Nó bao gồm việc trích xuất các đề xuất vùng, áp dụng CNN cho từng đề xuất, sau đó chạy các tính năng đã trích xuất thông qua một bộ phân loại. Quá trình nhiều giai đoạn này có thể chậm và đòi hỏi nhiều tài nguyên. |
| Độ bền trước các biến thể đối tượng : Mô hình R-CNN có thể xử lý các đối tượng có nhiều kích thước, hướng và tỷ lệ khác nhau, phù hợp với các tình huống thực tế có nhiều đối tượng và bối cảnh phức tạp. | Suy luận chậm : Do xử lý tuần tự các đề xuất vùng, R-CNN tương đối chậm trong quá trình suy luận. Các ứng dụng thời gian thực có thể thấy độ trễ này không thể chấp nhận được. |
| Tính linh hoạt : R-CNN là một khuôn khổ linh hoạt có thể được điều chỉnh cho nhiều tác vụ phát hiện đối tượng khác nhau, bao gồm phân đoạn thể hiện và theo dõi đối tượng. Bằng cách sửa đổi các lớp cuối cùng của mạng, bạn có thể tùy chỉnh R-CNN để phù hợp với nhu cầu cụ thể của mình. | Đề xuất vùng chồng lấn : R-CNN có thể tạo ra nhiều đề xuất vùng chồng lấn đáng kể, dẫn đến tính toán dư thừa và có khả năng ảnh hưởng đến hiệu suất phát hiện. |

**1.2.1.2. Fast R-CNN (2015)**

Dựa trên thành công của R-CNN, Ross Girshick (lúc này đã chuyển sang Microsoft Research) 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 qui 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.

A diagram of a diagram of a structure

Description automatically generated

**Hình 2**: Kiến trúc single model Fast R-CNN (được trích xuất từ bài báo gốc). 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 dịnh 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. 1 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 qui 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.

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| **Tốc độ nhanh hơn:** Fast R-CNN cải thiện đáng kể tốc độ bằng cách xử lý một lần ảnh đầu vào và trích xuất đặc trưng cho tất cả các vùng đề xuất. | **Yêu cầu chọn vùng:** Fast R-CNN vẫn phụ thuộc vào việc chọn vùng đề xuất (region proposal) để hoạt động, thường sử dụng Selective Search, có thể không tối ưu. |
| **Chất lượng chính xác cao:** Độ chính xác được cải thiện nhờ việc kết hợp toàn bộ ảnh vào quá trình học và phân loại vùng. | **Không hoàn toàn tự động:** Vẫn cần một bước ngoại vi để tạo ra các vùng đề xuất, không thể tự động như Faster R-CNN. |
| **Hỗ trợ học một giai đoạn:** Cho phép việc học một giai đoạn, giúp giảm số lượng tham số và thời gian huấn luyện. | **Khó khăn trong việc xử lý ảnh lớn:** Với ảnh kích thước lớn, Fast R-CNN có thể gặp khó khăn trong việc xử lý do giới hạn về kích thước đầu vào. |
| **Giảm thiểu bộ nhớ:** Sử dụng các kỹ thuật tối ưu như ROI Pooling để giảm lượng bộ nhớ cần thiết cho việc lưu trữ thông tin vùng. | **Cần điều chỉnh kích thước:** Kích thước của vùng đề xuất cần được điều chỉnh để phù hợp với quy trình phân loại, có thể làm giảm tính linh hoạt. |

1.2.1.3. Faster R-CNN (2016)

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 và các cộng sự tại Microsoft Research trong bài báo năm 2016 có tiêu đề [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks](https://arxiv.org/abs/1506.01497). Dịch nghĩa là “Faster R-CNN: Hướng tới phát hiện đối tượng theo thời gian thực với các mạng đề xuất khu vực”.

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, viết tắT là RPN). 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.

Kiến trúc của mô hình được tổng kết thông qua sơ đồ bên dưới:

A diagram of a network

Description automatically generated

**Hình 3**: Kiến trúc mô hình Faster R-CNN (được trích xuất từ bài báo gốc). 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 dectector 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. Một mô hình mở rộng hỗ trợ cho phân đoạn hình ảnh, được mô tả trong bài báo năm 2017 có tựa đề [Mask R-CNN](https://arxiv.org/abs/1703.06870).

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Hiệu suất cao:Faster R-CNN đạt độ chính xác vượt trội trong phát hiện và nhận dạng đối tượng nhờ khả năng kết hợp Region Proposal Network (RPN) và Fast R-CNN vào cùng một kiến trúc. | Tốc độ chưa đạt mức thời gian thực hoàn toàn: Mặc dù nhanh hơn các mô hình trước, tốc độ 5 fps trên GPU vẫn chưa đủ để đáp ứng các ứng dụng đòi hỏi thời gian thực như xe tự lái hay camera giám sát. |
| Tích hợp Region Proposal Network (RPN)**:** RPN giúp giảm đáng kể số lượng các region proposals không liên quan, đồng thời tập trung vào các khu vực có khả năng chứa đối tượng.  Khả năng đề xuất vùng theo thời gian thực nhờ việc tối ưu hóa qua các anchor boxes, giảm độ phức tạp và thời gian xử lý. | Yêu cầu phần cứng mạnh: Faster R-CNN cần GPU hiệu suất cao để đạt tốc độ tối ưu, điều này có thể là rào cản trong các môi trường tính toán hạn chế. |
| Thiết kế kiến trúc liền mạch:Cả RPN và Fast R-CNN đều sử dụng chung feature map, tiết kiệm tài nguyên tính toán so với các phương pháp trước đó như R-CNN và Fast R-CNN. Mô hình duy nhất nhưng xử lý được cả hai nhiệm vụ: xác định vùng và phân loại đối tượng. | Độ phức tạp trong thiết kế và huấn luyện:Quy trình huấn luyện xen kẽ (RPN và Fast R-CNN) đòi hỏi sự đồng bộ và tối ưu hóa cẩn thận, khiến việc triển khai phức tạp hơn so với các mô hình đơn giản hơn như YOLO hoặc SSD. |
| Khả năng mở rộng:Faster R-CNN có thể được mở rộng để hỗ trợ các nhiệm vụ như phân đoạn đối tượng (Mask R-CNN), giúp mở rộng ứng dụng vào các lĩnh vực như y tế, giao thông, và robot. | Không phù hợp với các ứng dụng yêu cầu tốc độ rất cao: Các mô hình như YOLO (You Only Look Once) hoặc SSD (Single Shot MultiBox Detector) thường được chọn cho các ứng dụng cần tốc độ cực nhanh (lên đến hàng chục fps) dù độ chính xác có thể thấp hơn. |
| Xử lý thời gian thực (tương đối):Tốc độ cải thiện rõ rệt, đạt 5 fps trên GPU, nhanh hơn so với các phương pháp R-CNN và Fast R-CNN ban đầu. | Phụ thuộc vào anchor boxes: Cơ chế anchor boxes cần phải được thiết kế tốt và phù hợp với từng bộ dữ liệu cụ thể. Điều này làm tăng khối lượng công việc để điều chỉnh mô hình cho các ứng dụng khác nhau. |

### 1.2.2. Lớp các mô hình họ YOLO

1.2.2.1. YOLO (2015)

Mô hình YOLO được mô tả lần đầu tiên bởi Joseph Redmon, và các cộng sự. trong bài viết năm 2015 có tiêu đề [Bạn chỉ nhìn một lần: Phát hiện đối tượng theo thời gian thực - You Only Look Once: Unified, Real-Time Object Detection](https://arxiv.org/abs/1506.02640). Trong công trình này thì một lần nữa Ross Girshick, người phát triển mạng R-CNN, cũng là một tác giả và người đóng góp khi ông chuyển qua [Facebook AI Research](https://research.fb.com/category/facebook-ai-research/).

Phương pháp chính dựa trên một mạng neural network duy nhất được huấn luyện dạng end-to-end model. Mô hình lấy input là một bức ảnh và dự đoán các bounding box và nhãn lớp cho mỗi bounding box. Do không sử dụng region proposal nên kỹ thuật này có độ chính xác thấp hơn (ví dụ: nhiều lỗi định vị vật thể - localization error hơn), mặc dù hoạt động ở tốc độ 45 fps (khung hình / giây) và tối đa 155 fps cho phiên bản tối ưu hóa tốc độ. Tốc độ này còn nhanh hơn cả tốc độ khung hình của máy quay phim thông thường chỉ vào khoảng 24 fps.

Mô hình hoạt động bằng cách trước tiên phân chia hình ảnh đầu vào thành một lưới các ô (grid of cells), trong đó mỗi ô chịu trách nhiệm dự đoán các bounding boxes nếu tâm của nó nằm trong ô. Mỗi grid cell (tức 1 ô bất kì nằm trong lưới ô) dự đoán các bounding boxes được xác định dựa trên tọa độ x, y (thông thường là tọa độ tâm, một số phiên bản là tọa độ góc trên cùng bên trái) và chiều rộng (width) và chiều cao (height) và độ tin cậy (confidence) về khả năng chứa vật thể bên trong. Ngoài ra các dự đoán nhãn cũng được thực hiện trên mỗi một bonding box.

Ví dụ: một hình ảnh có thể được chia thành lưới 7 × 7 và mỗi ô trong lưới có thể dự đoán 2 bounding box, kết quả trả về 98 bounding box được đề xuất. Sau đó, một sơ đồ xác suất nhãn (gọi là class probability map) với các confidence được kết hợp thành một tợp hợp bounding box cuối cùng và các nhãn. Hình ảnh được lấy từ bài báo dưới đây tóm tắt hai kết quả đầu ra của mô hình.

A diagram of a bicycle

Description automatically generated

**Hình 4**: Các bước xử lý trong mô hình YOLO (hình ảnh trích xuất từ bài báo gốc). Đầu tiên mô hình chia hình ảnh thành một grid search kích thước $S \times S$. Trên mỗi một grid cell ta dự báo một số lượng $B$ bounding boxes và confidence cho những boxes này và phân phối xác suất của $C$ classes. Như vậy output các dự báo là một tensor kích thước $S \times S \times (B \times 5 + C)$. Giá trị 5 là các tham số của offsets của bounding box gồm $x, y ,w, h$ và confidence. $C$ là số lượng tham số của phân phối xác suất.

Ưu nhược điểm của YOLO

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Tốc độ xử lý nhanh: YOLO có thể đạt tốc độ xử lý rất nhanh, lên đến 45 khung hình/giây, phù hợp với các ứng dụng yêu cầu xử lý nhanh như nhận dạng đối tượng trong thời gian thực. | Khó tối ưu hóa siêu tham số: Việc tối ưu hóa siêu tham số của YOLO có thể phức tạp và tốn thời gian hơn so với các mô hình khác. |
| Hiệu suất cao: YOLO sử dụng mạng nơ-ron tích hợp, có thể thực hiện cùng lúc việc phát hiện và phân loại đối tượng, giúp cải thiện hiệu suất. | Không phù hợp với các đối tượng nhỏ: YOLO có khả năng kém hơn trong việc phát hiện các đối tượng nhỏ trong ảnh. |
| Dễ triển khai: YOLO có cấu trúc đơn giản, dễ triển khai trên các thiết bị nhúng và thiết bị di động. | Dễ bị ảnh hưởng bởi sự che chắn: YOLO có thể gặp khó khăn trong việc phát hiện đối tượng khi chúng bị che chắn một phần. |
| Độ chính xác tương đối: Mặc dù không đạt được độ chính xác cao như các mô hình phát hiện đối tượng khác, YOLO vẫn có độ chính xác tương đối tốt, đủ cho nhiều ứng dụng thực tế. | Độ chính xác thấp hơn so với các mô hình phát hiện đối tượng khác: YOLO không đạt được độ chính xác cao bằng các mô hình như Faster R-CNN, Mask R-CNN, v.v. |

1.2.2.2. YOLOv2 (2016)

Mô hình YOLOv2 được Joseph Redmon và Ali Farhadi cập nhật nhằm cải thiện hơn nữa hiệu suất trong bài báo năm 2016 có tựa đề là [YOLO9000: Better, Faster, Stronger](https://arxiv.org/abs/1612.08242).

Mặc dù biến thể của YOLO được gọi là YOLOv2, một instance của mô hình theo như mô tả đã được đào tạo trên hai bộ dữ liệu nhận dạng đối tượng, và có khả năng dự đoán lên tới 9000 loại đối tượng khác nhau, do đó được đặt tên là YOLO9000. Với con số này thì mô hình này đã tiến xa hơn rất nhiều so với mọi mô hình trước đó về số lượng các loại đối tượng có khả năng phát hiện.

Một số thay đổi về huấn luyện và kiến trúc đã được thực hiện, chẳng hạn như việc sử dụng batch normalization cho hàng loạt và hình ảnh đầu vào phân giải cao.

Giống như Faster R-CNN, mô hình YOLOv2 sử dụng anchor boxes, bounding box được xác định trước với hình dạng và kích thước hợp lý được tùy chỉnh trong quá trình huấn luyện. Sự lựa chọn các bounding boxes cho hình ảnh được xử lý trước bằng cách sử dụng thuật toán phân cụm k-mean trên tập dữ liệu huấn luyện.

Điều quan trọng, các predicted bounding box được tinh chỉnh để cho phép các thay đổi nhỏ có tác động ít hơn đến các dự đoán, dẫn đến mô hình ổn định hơn. Thay vì dự đoán trực tiếp vị trí và kích thước, các offsets (tức tọa độ tâm, chiều dài và chiều rộng) được dự đoán để di chuyển và định hình lại các pre-defined anchor boxes tại mỗi một grid cell thông qua hàm logistic.

A diagram of a number of boxes

Description automatically generated

**Hình 5**: Sơ đồ giúp tạo prior bounding box có chiều rộng $p\_w$ và chiều cao $p\_h$ đã xác định từ grid cell có tọa độ $(c\_x, c\_y)$. Khi đó tọa độ tâm $(b\_x, b\_y)$ được tính theo mức độ tịnh tiến hàm sigmoid. Đồng thời, chiều rộng và chiều cao $(b\_w, b\_h)$ được tính như công thức scale số mũ của cơ số tự nhiên $e$.

Ưu điểm của yolo:

Một trong những ưu điểm chính của YOLO là tốc độ suy luận nhanh, cho phép nó xử lý hình ảnh theo thời gian thực. Nó rất phù hợp cho các ứng dụng như giám sát bằng video, ô tô tự lái và thực tế tăng cường.

Nhược điểm:

1. Độ chính xác thấp hơn trong các trường hợp đối tượng nhỏ hoặc dày đặc

YOLO sử dụng các ô lưới (grid cells) để chia ảnh và dự đoán vị trí đối tượng. Điều này khiến mô hình gặp khó khăn khi đối tượng:

Có kích thước nhỏ và rơi vào nhiều ô lưới khác nhau.

Nằm gần nhau hoặc chồng chéo, đặc biệt trong các cảnh đông đúc.

2. Khả năng nhận diện các đối tượng bất đối xứng hoặc phức tạp

YOLO gặp hạn chế trong việc nhận diện các đối tượng có hình dạng không chuẩn, phức tạp hoặc đối tượng bị biến dạng.

3. Cân bằng giữa tốc độ và độ chính xác

Mặc dù nhanh, YOLO thường không đạt độ chính xác cao bằng các mô hình khác như Faster R-CNN trong các tác vụ yêu cầu độ chính xác cực kỳ cao.

4. Phân chia ô lưới làm giảm hiệu quả

Một đối tượng lớn có thể rơi vào một ô lưới duy nhất, trong khi các đối tượng nhỏ có thể rơi vào nhiều ô lưới, dẫn đến dự đoán không chính xác.

Việc phân chia cố định (fixed grid) không phù hợp với các ảnh có bố cục hoặc mật độ đối tượng đa dạng.

5. Khó nhận diện đối tượng ẩn hoặc bị che khuất

YOLO không được thiết kế tối ưu để xử lý các đối tượng bị che khuất một phần, do mô hình tập trung vào các đặc trưng dễ nhận diện.

6. Không linh hoạt trong việc xử lý độ phân giải và tỉ lệ khung hình

YOLO yêu cầu một kích thước đầu vào cố định. Việc chuyển đổi kích thước ảnh để phù hợp với mô hình có thể làm mất thông tin chi tiết hoặc làm méo hình ảnh.

7. Hạn chế trong việc học các đặc trưng nhỏ và hiếm

Mô hình thường bỏ qua các đối tượng hoặc chi tiết nhỏ, đặc biệt nếu chúng không thường xuyên xuất hiện trong tập dữ liệu huấn luyện.

## **1.3. Ngôn ngữ lập trình**

1.3.1.Python

là một ngôn ngữ lập trình hướng đối tượng, cấp cao và vô cùng mạnh mẽ. Điều đặc biệt ở ngôn ngữ này là sự đa năng, đa công dụng. Mang các tính năng ưu việt như tạo kiểu dữ liệu tự động và cơ chế cung cấp bộ nhớ tự động. Ngoài ra, Python còn được coi là một ngôn ngữ lập trình có cấu trúc dữ liệu cao cấp và mạnh mẽ nhất.

Các tính năng chính thể hiện sự ưu việt của Python có thể liệt kê như:

* Cực kỳ đơn giản và dễ tiếp thu cũng như thực hành: sự đơn giản của ngôn ngữ này thể hiện ở các câu lệnh dễ học, dễ hiểu. Python biến những “cú pháp code khó nhằn” trở nên dễ dàng hơn bằng những giải pháp.
* Hoàn toàn miễn phí và là mã nguồn mở: nhà lập trình có thể thoải mái di chuyển Python thậm chí sử dụng nó với mục đích thương mại hóa. Với đặc trưng là một ngôn ngữ lập trình mã nguồn mở. Bạn hoàn toàn có thể thay đổi các mã nguồn mở của Python ở bất cứ các ứng dụng phần mềm nào.
* Là một ngôn ngữ di động: di động có nghĩa là nó rất dễ di chuyển từ nền tảng này sang các nền tảng khác. Đồng thời, bạn có thể chạy chương trình của Python ở mọi nền tảng mà không sợ có bất kỳ những thay đổi nào ảnh hưởng.
* Python là ngôn ngữ phiên dịch cao cấp:khi chạy chương trình Python trên bất cứ hệ điều hành hay mạng Internet nào. Python có thể hoàn toàn tự động dịch và chạy theo đúng chương trình dữ liệu của máy tính đó.

A logo for a computer company

Description automatically generated

**Ưu Điểm:**

* Python có cấu trúc đơn giản, rõ ràng và rất dễ học. Đây được coi là một trong những ngôn ngữ nên học cho những người mới bắt đầu tiếp cận đến lập trình.
* Python có mặt trên tất cả các hệ điều hành phổ biến như: Windows, MacOS, MS-DOS, Unix,… đây cũng thể hiện được sự đa năng của ngôn ngữ này.
* Tương thích với thư viện lớn nhất hiện nay với 400 triệu người sử dụng như data mining Scikit-learn, Pandas…
* Tốc độ xử lý cũng là một trong những lí do tạo nên ưu điểm của Python. Ngôn ngữ này có thể tạo ra những script siêu nhỏ tới những phần mềm cực lớn như Blender 3D.

**Nhược Điểm:**

* Các cấu trúc của Python đòi hỏi nhiều không gian bộ nhớ hơn, ngôn ngữ này không thể phát huy hết tác dụng trong điều kiện bộ nhớ hạn chế.
* Python không phải là ngôn ngữ được ưu tiên để phát triển ứng dụng di động hay trò chơi do tiêu tốn nhiều bộ nhớ hơn và tốc độ xử lý chậm so với ngôn ngữ khác.
* Rất khó để chạy các bài kiểm tra trên mã được viết bằng Python, các lỗi trên này thường xuất hiện trong thời gian chạy, điều này khiến việc kiểm tra trở nên khó khăn hơn.

**1.3.2.Thư viện sử dụng:**

OpenCV là tên viết tắt của open source computer vision library – có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực.

OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các learning machine khác.

Thư viện OpenCV được phát hành với giấy phép BDS. Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV sở hữu giao diện thiên thiện với mọi loại ngôn ngữ lập trình, ví dụ như C++, C, Python hay Java… Ngoài ra, nó cũng dễ dàng tương thích với các hệ điều hành khác nhau, bao gồm từ Windows, Linux, Mac OS, iOS cho đến cả Android.

Kể từ lần đầu xuất hiện từ năm 1999, giờ đây OpenCV đã sở hữu đội ngũ người dùng hùng hậu, con số ước tính có thể lên tới 47.000 người. Tất cả là nhờ những ưu điểm vượt trội của OpenCV.

A close-up of a head

Description automatically generated

**Các module được dùng trong OpenCV**

OpenCV có cấu trúc module, tức là nó bao gồm cả những thư viện liên kết tĩnh lẫn thư viện liên kết động. Nắm rõ các module của OpenCV sẽ giúp bạn đọc hoàn toàn thấu hiểu OpenCV là gì.

* Core functionality (core): Module này sở hữu cơ chế rất nhỏ gọn. Nó được dùng để định hình các cấu trúc của cơ sở dữ liệu cơ bản, bao gồm cả những mảng đa chiều. Ngoài ra nó còn xác định các chức năng của những module đi kèm khác nữa.
* Image Processing (imgproc): Đây là module được dùng cho quá trình xử lý hình ảnh. Nó cho phép người dùng thực hiện các hoạt động như lọc hình ảnh tuyến tính và phi tuyến, thực hiện phép biến hình, thay đổi không gian màu, xây dựng biểu đồ và rất nhiều thao tác khác liên quan.
* Video Analysis (video): Giống như tên gọi của nó, module này cho phép phân tích các video. Kết quả được trả về bao gồm các ước tính chuyển động, thực hiện tách nền và các phép toán theo dõi vật thể.
* Camera Calibration and 3D Reconstruction (calib3d): Module này cung cấp các thuật toán hình học đa chiều cơ bản và hiệu chuẩn máy ảnh single và stereo. Ngoài ra nó còn đưa ra các dự đoán kiểu dáng của đối tượng và sử dụng thuật toán thư tín âm thanh nổi cùng các yếu tố tái tạo 3D.
* 2D Features Framework (features2d): Module này giúp phát hiện các tính năng nổi trội của bộ nhận diện, bộ truy xuất thông số và thông số đối chọi.
* Ngoài ra còn có rất nhiều module khác với đa dạng tính năng, ví dụ như: FLANN, Google test wrapper

**Ứng dụng của OpenCV**

* OpenCV giúp định vị hình ảnh
* OpenCV được cho là một phần mềm đa nhiệm. Nó được ứng dụng trong rất nhiều trường hợp khác nhau. Ví dụ, ta sẽ nói về các phần mềm định vị, bản đồ nói chung. Hẳn rằng trong chúng ta ai cũng đã có ít nhất một lần cần sử dụng đến các map online đúng không. Bạn sử dụng các map để tìm đường, tra cứu tình hình giao thông hoặc đơn giản là xem xét các hình ảnh thực tế của địa điểm cần đến. Những lúc như vậy, OpenCV đóng vai trò là nhà cung cấp dữ liệu hình ảnh cho các app về Map này. OpenCV sẽ đem đến cho người dùng hình ảnh về đường phố hay các căn nhà, con người xung quanh địa điểm được chỉ định.
* OpenCV còn được dùng để khởi tạo ra những hình ảnh 3 chiều phức tạp. Hoạt động này rất được yêu thích, nhất là trong thời đại trí tuệ nhân tạo AI phát triển như thế này.

Đối với các công nghệ hiện đại, OpenCV cũng là một yếu tố không thể thiếu. Tất cả những ứng dụng công nghệ như robot, xe tự lái, bảng cảm ứng thông minh… đều có sự góp mặt của OpenCV trong khâu xử lý hình ảnh. Ví dụ gần gũi nhất trong cuộc sống có thể kể đến hệ thống mở khóa điện thoại bằng cách nhận diện khuôn mặt người dùng

# **Chương 2: Xây dựng hệ thống nhận dạng**

## **2.1. Mô tả bài toán:**

Bài toán nhận diện và đếm đối tượng trong ảnh là một vấn đề quan trọng trong lĩnh vực thị giác máy tính, với mục tiêu tự động phân tích hình ảnh hoặc video để phát hiện, phân loại, và đếm số lượng các đối tượng xuất hiện. Đây là một ứng dụng phổ biến trong nhiều lĩnh vực như giám sát an ninh, giao thông thông minh, và phân tích dữ liệu thị giác.

Bài toán được chia thành ba phần chính:

● *Nhận diện đối tượng:* Phát hiện và định vị các đối tượng trong ảnh hoặc video dựa trên tọa độ của hộp giới hạn (bounding box).

● *Phân loại đối tượng:* Gán nhãn cụ thể cho các đối tượng phát hiện được, chẳng hạn như "người", "ô tô", "xe đạp", "chó", v.v., sử dụng các lớp có sẵn trong tập dữ liệu.

● *Đếm đối tượng:* Tính toán số lượng mỗi loại đối tượng được nhận diện trong ảnh hoặc video.

### 2.1.1 Bài toán nhận diện đối tượng

Sử dụng mô hình YOLOv3 để phát hiện và nhận diện đối tượng:

- Đặc điểm chính:

- Phát hiện đối tượng trong thời gian thực với độ chính xác cao.

- Sử dụng toàn bộ hình ảnh đầu vào để xác định vị trí và nhãn các đối tượng trong một lần chạy mô hình (one-shot detection).

- Quy trình hoạt động:

- Lớp Convolutional: Phát hiện đặc trưng của đối tượng như hình dạng, màu sắc, và kết cấu.

- Lớp Fully Connected: Xác định tọa độ bounding box và nhãn đối tượng dựa trên các đặc trưng đã phát hiện.

Đầu ra của mô hình:

- Bounding box và nhãn cho từng đối tượng phát hiện được.

- Độ tin cậy (confidence score) cho mỗi nhãn.

### 2.1.2 Bài toán phân lớp và đếm đối tượng

Phân loại đối tượng:

- Dựa trên các bounding box phát hiện được, gán nhãn cho từng đối tượng dựa trên tập lớp có sẵn (VD: người, xe, động vật).

- Sử dụng các đặc trưng của bounding box để so khớp với nhãn đối tượng.

Đếm đối tượng:

- Tổng hợp các nhãn để tính toán số lượng đối tượng thuộc mỗi loại trong ảnh hoặc luồng webcam.

Đầu ra của mô hình:

- Danh sách các đối tượng được phát hiện và số lượng tương ứng.

### 2.1.3 Phân tích luồng webcam

Sử dụng webcam để thu thập dữ liệu thời gian thực và thực hiện nhận diện:

- Phát hiện đối tượng:

- Áp dụng YOLOv3 trên từng khung hình của luồng webcam để phát hiện các đối tượng.

- Hiển thị kết quả:

- Vẽ bounding box và nhãn lên từng khung hình trong luồng video.

- Hiển thị số lượng các đối tượng phát hiện được theo thời gian thực.

Đầu ra:

- Luồng video có hiển thị bounding box và nhãn cho các đối tượng phát hiện được.

- Bảng tổng hợp số lượng đối tượng thuộc từng loại trong luồng video.

\*ví dụ:

**A yellow square with red arrows

Description automatically generated**

\* Input

Hình ảnh đối tượng:

- Hình ảnh có thể là ảnh chụp một người hoặc một nhóm người, được tải lên từ máy tính hoặc thu trực tiếp từ webcam.

- Ảnh có thể là ảnh chân dung hoặc toàn cảnh, không giới hạn số lượng đối tượng trong một ảnh.

Yêu cầu xử lý:

- Phát hiện đối tượng trong ảnh:

- Nhận diện các đối tượng xuất hiện trong ảnh, bao gồm vị trí (bounding box) và nhãn của đối tượng.

- Gán nhãn cho từng đối tượng dựa trên các lớp đã định nghĩa (VD: người, xe, động vật, v.v.).

- Đếm đối tượng:

- Tính toán số lượng các đối tượng thuộc từng loại được phát hiện trong ảnh.

- Phân tích dữ liệu từ webcam:

- Thực hiện các bước tương tự như trên với dữ liệu luồng trực tiếp từ webcam.

\* Output

=>Kết quả hiển thị:

- số lượng đối tượng có trong ảnh

- nhãn của các đối tượng có trong ảnh đó

## **2.2. Xây dựng hệ thống**

### 2.2.1. YOLOV3

YOLOv3 (You Only Look Once version 3) là một trong những mô hình phát hiện đối tượng mạnh mẽ và phổ biến nhất trong lĩnh vực thị giác máy tính. Được phát triển bởi Joseph Redmon và cộng sự, YOLOv3 đã cải thiện đáng kể về tốc độ và độ chính xác so với các phiên bản trước. Dưới đây là một cái nhìn chi tiết và sâu sắc hơn về lý thuyết, cấu trúc, cách hoạt động, cải tiến kỹ thuật, ứng dụng thực tiễn và so sánh với các mô hình khác.

2.2.1.1. **Kiến Trúc Mạng Nơ-ron**

YOLOv3 có một kiến trúc mạng nơ-ron phức tạp, được xây dựng dựa trên Darknet-53, với nhiều thành phần quan trọng.

A. **Darknet-53**

* **Cấu Trúc**: Darknet-53 bao gồm 53 lớp, trong đó có nhiều lớp convolutional và các khối residual. Các lớp này giúp trích xuất đặc trưng từ hình ảnh một cách hiệu quả.
* **Residual Connections**: Sử dụng các kết nối residual giúp cải thiện khả năng học của mạng, cho phép các thông tin đi qua các lớp mà không bị biến dạng. Điều này giúp ngăn chặn hiện tượng gradient vanishing, giúp mạng học tốt hơn.
* **Kích Thước Kernel**: YOLOv3 sử dụng các kích thước kernel 3x3 và 1x1, cho phép mạng học được các đặc trưng ở nhiều cấp độ khác nhau.

B. **Feature Pyramid Network (FPN)**

* **Nhiều Kích Thước Đầu Ra**: YOLOv3 tạo ra ba đầu ra cho các kích thước lưới khác nhau (13x13, 26x26, 52x52), cho phép phát hiện các đối tượng ở nhiều kích thước khác nhau.
* **Chia Sẻ Thông Tin**: Các đặc trưng từ các lớp khác nhau được chia sẻ để cải thiện khả năng phát hiện. Điều này cho phép YOLOv3 nhận diện các đối tượng nhỏ hơn mà các mô hình khác có thể bỏ qua.

C. **Anchor Boxes**

* **Tối Ưu Hóa Kích Thước**: Sử dụng thuật toán K-means clustering để tối ưu hóa kích thước anchor box, giúp chúng phù hợp hơn với các đối tượng trong tập dữ liệu huấn luyện.
* **Dự Đoán Nhiều Bounding Boxes**: Mỗi ô trong lưới có thể dự đoán nhiều bounding boxes, giúp cải thiện khả năng phát hiện với độ chính xác cao hơn.

2.1.1.2. **Cách Thức Hoạt Động**

A. **Tiền Xử Lý Hình Ảnh**

* **Resizing**: Hình ảnh đầu vào được thay đổi kích thước thành 416x416 pixel (hoặc 608x608) để phù hợp với đầu vào của mạng. Việc này giúp giảm thiểu độ phức tạp và tăng tốc độ xử lý.
* **Normalization**: Các giá trị pixel được chuẩn hóa để nằm trong khoảng [0, 1], giúp mạng học hiệu quả hơn.

B. **Quá Trình Dự Đoán**

* **Forward Pass**: Hình ảnh được đưa qua mạng và các giá trị đầu ra được tính toán cho mỗi ô trong lưới. Mỗi ô sẽ dự đoán vị trí của các bounding boxes và xác suất cho mỗi lớp đối tượng.
* **Bounding Boxes và Xác Suất**: Mỗi ô dự đoán vị trí của các bounding boxes và xác suất cho mỗi lớp. Mô hình sẽ tạo ra nhiều bounding boxes cho từng ô, mỗi bounding box có một độ tin cậy (confidence score).

C. **Post-Processing**

* **Non-Maximum Suppression (NMS)**: Sau khi dự đoán, NMS được sử dụng để loại bỏ các bounding boxes trùng lặp. Chỉ giữ lại những bounding boxes có độ tin cậy cao nhất, giúp cải thiện độ chính xác của các dự đoán.

2.1.1.3. **Hàm Mất Mát**

Hàm mất mát của YOLOv3 bao gồm ba thành phần chính:

* **Mất Mát Vị Trí**: Đo lường độ chính xác của vị trí các bounding boxes dự đoán.
* **Mất Mát Nhãn**: Đo lường độ chính xác của nhãn đối tượng.
* **Mất Mát Độ Tin Cậy**: Đo lường độ tin cậy của các dự đoán. Công thức tổng quát cho hàm mất mát của YOLOv3 là:

Loss=Mất mát vị trí +Mất mát nhãn + Mất mát độ tin cậy

2.2.1.4. **Cải Tiến Kỹ Thuật**

YOLOv3 đã cải thiện nhiều điểm so với các phiên bản trước:

* **Sử Dụng Darknet-53**: Cải thiện khả năng trích xuất đặc trưng và độ chính xác, nhờ vào việc sử dụng các khối residual.
* **Dự Đoán Nhiều Kích Thước**: Khả năng phát hiện đối tượng nhỏ và lớn nhờ vào việc sử dụng nhiều kích thước đầu ra. Điều này giúp mô hình hoạt động hiệu quả hơn trong các tình huống thực tế.
* **Mô Hình Đa Nhiệm**: Có thể dự đoán nhiều đối tượng trong một hình ảnh mà không cần phải chạy nhiều mô hình, tiết kiệm thời gian và tài nguyên.

2.2.1.5. **Ứng Dụng Thực Tiễn**

YOLOv3 được áp dụng trong nhiều lĩnh vực:

* **Giám Sát An Ninh**: Phát hiện và theo dõi người và phương tiện trong video giám sát. Các hệ thống an ninh hiện đại sử dụng YOLOv3 để phát hiện các hành vi đáng ngờ trong thời gian thực.
* **Giao Thông Thông Minh**: Phát hiện và phân loại xe cộ, người đi bộ và các chướng ngại vật khác. Điều này rất quan trọng trong các ứng dụng như xe tự lái và hệ thống quản lý giao thông.
* **Thương Mại Điện Tử**: Tự động nhận diện sản phẩm trong hình ảnh để cải thiện trải nghiệm người dùng. Các nền tảng thương mại điện tử có thể sử dụng YOLOv3 để tìm kiếm sản phẩm nhanh chóng và chính xác.

2.2.1.6. **So Sánh Với Các Mô Hình Khác**

A. **YOLOv3 vs. Faster R-CNN**

* **Tốc Độ**: YOLOv3 nhanh hơn nhiều so với Faster R-CNN, phù hợp cho các ứng dụng yêu cầu tốc độ cao. Faster R-CNN thường mất nhiều thời gian hơn vì nó phải thực hiện hai giai đoạn: đề xuất vùng và sau đó phân loại.
* **Độ Chính Xác**: Faster R-CNN thường có độ chính xác cao hơn trong một số trường hợp, nhưng YOLOv3 cung cấp sự cân bằng tốt giữa tốc độ và độ chính xác, đặc biệt trong các ứng dụng thời gian thực.

B. **YOLOv3 vs. SSD (Single Shot MultiBox Detector)**

* **Khả Năng Phát Hiện**: YOLOv3 thường tốt hơn trong việc phát hiện các đối tượng nhỏ nhờ vào FPN. SSD có thể gặp khó khăn trong việc phát hiện các đối tượng nhỏ hơn do chỉ sử dụng một kích thước đầu ra duy nhất.
* **Tốc Độ**: Cả hai mô hình đều nhanh, nhưng YOLOv3 có thể xử lý nhiều đối tượng hơn trong một lần chạy, làm cho nó trở thành lựa chọn lý tưởng cho các ứng dụng thời gian thực.

*2.2.1.7.****Lợi Ích và Hạn Chế***

A. **Lợi Ích**

* **Nhanh**: Thích hợp cho các ứng dụng yêu cầu thời gian thực. YOLOv3 có thể xử lý hàng trăm khung hình mỗi giây trên phần cứng tiêu chuẩn.
* **Đơn Giản**: Dễ dàng triển khai và sử dụng, với nhiều thư viện hỗ trợ như TensorFlow, PyTorch và OpenCV.

B. **Hạn Chế**

* **Độ Chính Xác**: Có thể không đạt được độ chính xác cao như các mô hình hai giai đoạn trong một số tình huống, đặc biệt khi phát hiện các đối tượng chồng chéo.
* **Khó Khăn Khi Huấn Luyện**: Cần nhiều dữ liệu và tài nguyên tính toán để huấn luyện mô hình. Việc thu thập và chuẩn bị dữ liệu cũng có thể là một thách thức lớn.

2.2.1.8. **Tương Lai và Cải Tiến**

* **YOLOv4 và YOLOv5**: Các phiên bản tiếp theo của YOLO đã được phát triển, với nhiều cải tiến về độ chính xác và tốc độ. YOLOv4 sử dụng một số kỹ thuật mới như Mosaic data augmentation và Self-adversarial training để cải thiện hiệu suất.
* **Ứng Dụng AI Tích Hợp**: YOLOv3 và các phiên bản tiếp theo đang được tích hợp vào nhiều ứng dụng AI khác nhau, từ nhận diện khuôn mặt đến phân tích video, giúp mở rộng khả năng của thị giác máy tính.

2.2.1.9. Các bước thực hiện của hệ thống

Hệ thống nhận dạng và đếm đối tượng trong ảnh được xây dựng với các bước chính sau:

\*Bước 1: Phát hiện đối tượng trong ảnh

Mục tiêu:

Xác định các vị trí của các đối tượng trong ảnh và khoanh vùng các vùng chứa đối tượng.

Phương pháp:

Sử dụng mô hình YOLOv3:

- Cấu trúc mạng:

+ Là một mạng sâu (Deep Neural Network) chuyên dùng để phát hiện đối tượng trong ảnh.

+ YOLOv3 chia ảnh đầu vào thành các lưới (grid cells).

+ Mỗi ô trong lưới dự đoán bounding box, confidence score và nhãn (class probabilities) của đối tượng.

**-** Nguyên tắc hoạt động:

+ Ảnh đầu vào được chia thành các lưới (grid cells).

+ Mỗi lưới dự đoán bounding box, confidence score và class probabilities.

+ Sử dụng Non-Maximum Suppression (NMS) để loại bỏ các bounding box trùng lặp hoặc không liên quan.

Yêu cầu đầu vào:

- Ảnh đầu vào được resize về kích thước cố định (ví dụ: 416x416 pixels).

- Giá trị pixel được chuẩn hóa về khoảng [0, 1].

Kết quả đầu ra:

- Danh sách các bounding box, confidence score và nhãn (class) của các đối tượng được phát hiện.

\*Bước 2: Phân loại đối tượng (Detection and Classification)

Mục tiêu:

Gán nhãn đối tượng và đếm số lượng từng loại đối tượng trong ảnh.

Phương pháp:

- Sử dụng YOLOv3:

- Từ đầu ra của mô hình YOLOv3, dựa vào nhãn (class probabilities) để gán nhãn cho từng đối tượng.

- Lưu bounding box, nhãn và số lượng vào một cấu trúc dữ liệu để hiển thị kết quả.

Nguyên tắc hoạt động:

- Với mỗi bounding box:

- Lấy nhãn của đối tượng có xác suất cao nhất.

- Đếm số lượng các nhãn để tổng hợp số lượng từng loại đối tượng trong ảnh.

Kết quả đầu ra:

- Nhãn của từng đối tượng, số lượng đối tượng theo từng loại.

- Kết quả hiển thị trên ảnh với bounding box, nhãn và số lượng.

\*Bước 3: Phân tích qua Webcam

Mục tiêu:

Phát hiện và đếm đối tượng trong luồng video thời gian thực từ webcam.

Phương pháp:

- Sử dụng mô hình YOLOv3 như bước phát hiện đối tượng trong ảnh tĩnh.

- Xử lý từng khung hình (frame) từ webcam, phát hiện đối tượng và hiển thị kết quả trên màn hình.

Nguyên tắc hoạt động:

- Mỗi khung hình:

- Resize và chuẩn hóa ảnh.

- Dự đoán các bounding box, confidence score và nhãn.

- Sử dụng Non-Maximum Suppression để lọc kết quả.

- Hiển thị kết quả trực tiếp trên màn hình với bounding box và nhãn.

Kết quả đầu ra:

- Luồng video trực tiếp với các đối tượng được phát hiện và gán nhãn.

\*Công cụ và tệp sử dụng:

- Mô hình YOLOv3:

- Cấu trúc mạng: `yolov3.cfg`.

- Trọng số: `yolov3.weights`.

- Danh sách nhãn: `coco.names`.

- Thư viện:

- OpenCV: Dùng để xử lý ảnh và luồng video.

- Flask: Xây dựng giao diện web và cung cấp API phát hiện đối tượng.

### 2.2.2.Đếm

2.2.2.1. Khái Niệm Đếm Đối Tượng

Đếm đối tượng là một quá trình quan trọng trong thị giác máy tính, cho phép chúng ta xác định và ghi lại số lượng các đối tượng khác nhau trong một bức ảnh hoặc video. Trong ngữ cảnh của ứng dụng phát hiện đối tượng sử dụng mô hình YOLO (You Only Look Once), việc đếm này không chỉ giúp hiểu rõ hơn về nội dung của bức ảnh mà còn có thể ứng dụng trong nhiều lĩnh vực như giám sát an ninh, phân tích giao thông, và nghiên cứu sinh thái.

2.2.2.2. Cấu Trúc Dữ Liệu Sử Dụng

Để thực hiện việc đếm đối tượng, đoạn mã sử dụng một cấu trúc dữ liệu gọi là từ điển (dictionary) trong Python. Cấu trúc này rất hữu ích vì nó cho phép lưu trữ các cặp khóa-giá trị, trong đó:

* **Khóa**: Là tên của lớp đối tượng (ví dụ: "person", "car", "dog").
* **Giá trị**: Là số lượng của đối tượng tương ứng đã được phát hiện.

**Ví dụ:**

object\_count = {

"person": 3,

"car": 2,

"dog": 1

}

Trong ví dụ này, từ điển cho thấy rằng có 3 người, 2 xe, và 1 chó trong bức ảnh.

2.2.2.3. Quy Trình Đếm Đối Tượng

Quá trình đếm đối tượng diễn ra qua các bước sau:

2.2.2.3.1. Khởi Tạo Từ Điển Đếm

* **Tạo Từ Điển**: Một từ điển object\_count được khởi tạo để lưu trữ số lượng các đối tượng phát hiện được theo từng loại.

**Ví dụ:** object\_count = {}

2.2.2.3.2. Phân Tích Kết Quả Phát Hiện

* **Lặp Qua Các Chỉ Số**: Sau khi áp dụng Non-Maximum Suppression (NMS), mã sẽ lặp qua danh sách các chỉ số của các hộp giới hạn còn lại.

**Ví dụ:** for i in indices.flatten():

2.2.2.3.3. Lấy Thông Tin Từ Các Hộp Giới Hạn

* **Lấy Vị Trí và Nhãn**: Với mỗi chỉ số i, mã sẽ lấy vị trí (x, y) và kích thước (w, h) của hộp giới hạn, cùng với nhãn lớp của đối tượng.

Ví dụ:

x, y, w, h = boxes[i]

label = str(classes[class\_ids[i]])

2.2.2.3.4. Cập Nhật Số Lượng Đối Tượng

* **Kiểm Tra Nhãn**: Mã kiểm tra xem nhãn label đã tồn tại trong từ điển object\_count chưa.
  + **Nếu đã tồn tại**: Tăng giá trị của nhãn đó lên 1.
  + **Nếu chưa tồn tại**: Thêm nhãn vào từ điển với giá trị khởi tạo là 1.

**ví dụ:**

if label in object\_count:

object\_count[label] += 1

else:

object\_count[label] = 1

2.2.2.3.5. Tính Tổng Số Đối Tượng

* **Tính Tổng**: Sau khi lặp qua tất cả các đối tượng, tổng số đối tượng được tính bằng cách sử dụng hàm sum() trên các giá trị của từ điển object\_count.

Ví dụ: total\_objects = sum(object\_count.values())

2.2.2.3.6. Hiển Thị Kết Quả

* **Hiển Thị Tổng Số Đối Tượng**: Tổng số đối tượng được hiển thị trên bức ảnh đã xử lý, giúp người dùng dễ dàng nhận biết số lượng đối tượng trong ảnh.

**Ví dụ:**

count\_text = "Total Objects: " + str(total\_objects)

cv2.putText(image, count\_text, (10, 30), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

# **Chương 3: Kết quả thực nghiệm**

## **3.1 Cơ sở dữ liệu**

### 3.1.1 Dữ liệu

Dữ liệu sử dụng trong thực nghiệm được lấy từ các nguồn sau:

* Dữ liệu đầu vào:
  + Dữ liệu hình ảnh chứa nhiều đối tượng khác nhau, có thể là ảnh chân dung, ảnh toàn cảnh hoặc ảnh nhóm với nhiều đối tượng.
  + Hình ảnh được thu thập từ các nguồn như: camera an ninh, ảnh công khai từ Internet hoặc các bộ dữ liệu công khai như COCO[6], PASCAL VOC[7].
    1. Tiền xử lý dữ liệu:
  + Chuẩn hóa ảnh đầu vào:  
    Hình ảnh được chuẩn hóa về kích thước đồng nhất (416x416 pixels) để phù hợp với yêu cầu của mô hình YOLOv3.
  + Chuẩn hóa màu sắc:  
    Các giá trị pixel của ảnh được chuẩn hóa theo giá trị trung bình [0, 0, 0] và chuẩn hóa tỷ lệ (0-1).
  + Cắt và thu nhỏ:  
    Các đối tượng được phát hiện trong ảnh sẽ được cắt và xử lý để phù hợp với mô hình nhận diện đối tượng.
* Phân chia dữ liệu:  
  Dữ liệu được chia thành hai phần:
  + Tập huấn luyện (Training Set): 70% của dữ liệu dùng để huấn luyện mô hình.
  + Tập kiểm tra (Test Set): 30% còn lại dùng để kiểm tra hiệu quả của mô hình.

## **3.2.Kết quả thực nghiệm**

Ví dụ kết quả:

* A group of girls taking a selfie

  Description automatically generatedĐối tượng 1: "Người"
* Đối tượng 2: "Xe"

A car parked on the road

Description automatically generated

* Đối tượng 3: "Mèo"

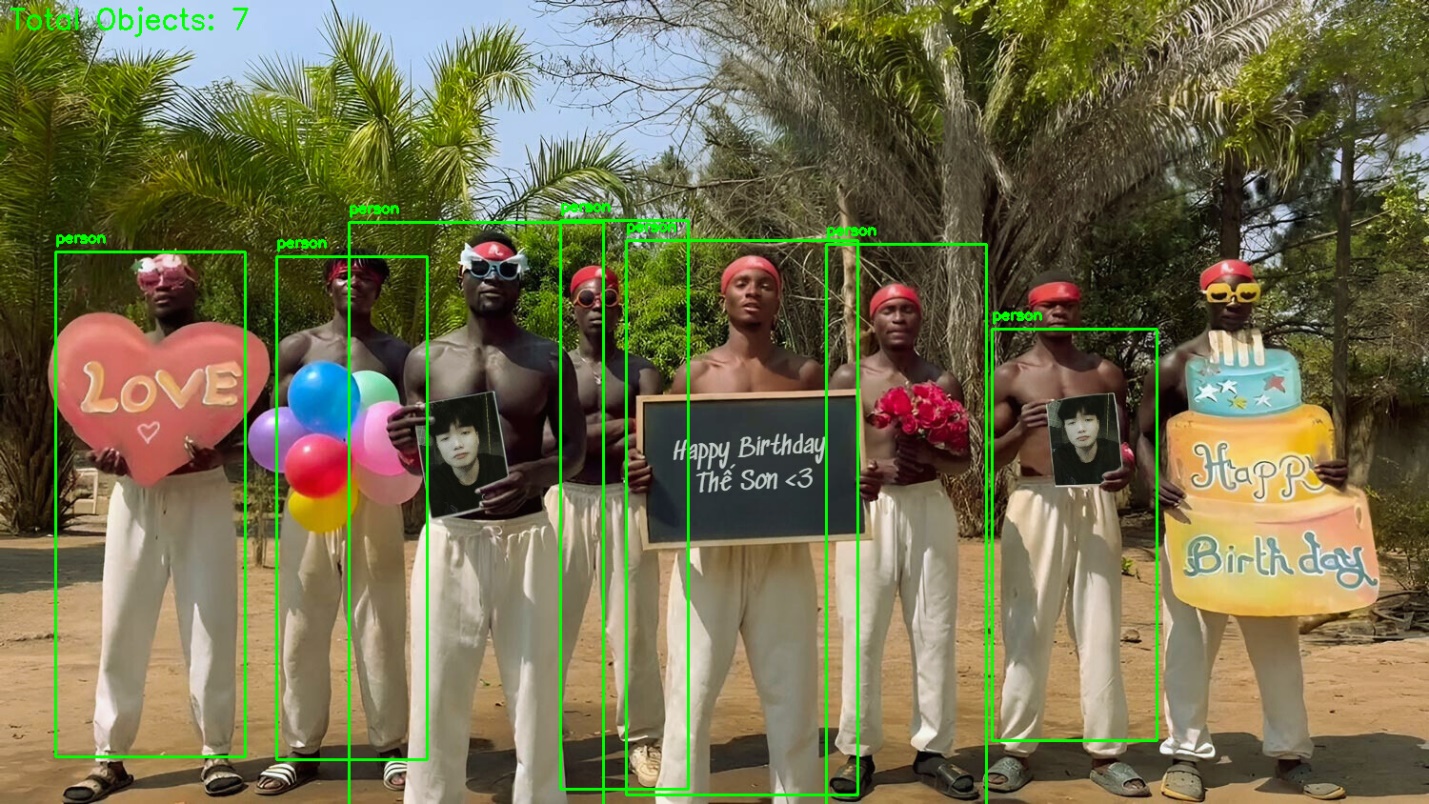
**Hiển thị trực quan:**

Kết quả dự đoán được hiển thị trực quan trên ảnh đầu ra với các đặc điểm sau:

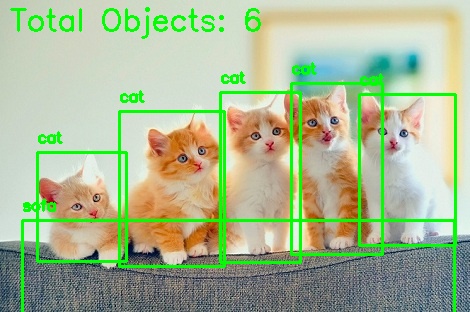
* **Khung hình chữ nhật:**  
  Các đối tượng được bao quanh bởi khung hình chữ nhật màu xanh lá cây, giúp dễ dàng nhận diện vị trí của đối tượng trong ảnh.
* **Nhãn đối tượng:**  
  Mỗi đối tượng được gán nhãn tên của đối tượng và hiển thị thông tin về số lượng đối tượng tương ứng. Ví dụ:
  + **Khuôn mặt 1:** "Person"
  + **Khuôn mặt 2:** "Car"
  + **Khuôn mặt 3:** "Cat"

Điều này giúp người dùng dễ dàng xác định thông tin về loại đối tượng trong ảnh và có thể theo dõi số lượng các đối tượng được phát hiện.

**Kết quả thực nghiệm:**

****

*Hình 3.1 Ảnh người*

**

*Hình 3.2 Ảnh mèo*

**

*Hình 3.3 Ảnh điện thoại*

# **Tài liệu tham khảo**

[1] Bradski, G. (2000). The OpenCV Library. OpenCV Documentation.

[2] Chollet, F. (2017). Deep Learning with Python. Manning Publications.

[3] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524.

[4] OpenCV Documentation (2021). OpenCV Library. https://opencv.org/.

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640.

[6] https://docs.ultralytics.com/vi/datasets/detect/coco

[7] https://docs.ultralytics.com/vi/datasets/detect/pascalvoc