



ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교

# Chương 6. Dò tìm/Phát hiện đối tượng (Object Detection)

TS. PHẠM NGUYỄN MINH NHỰT

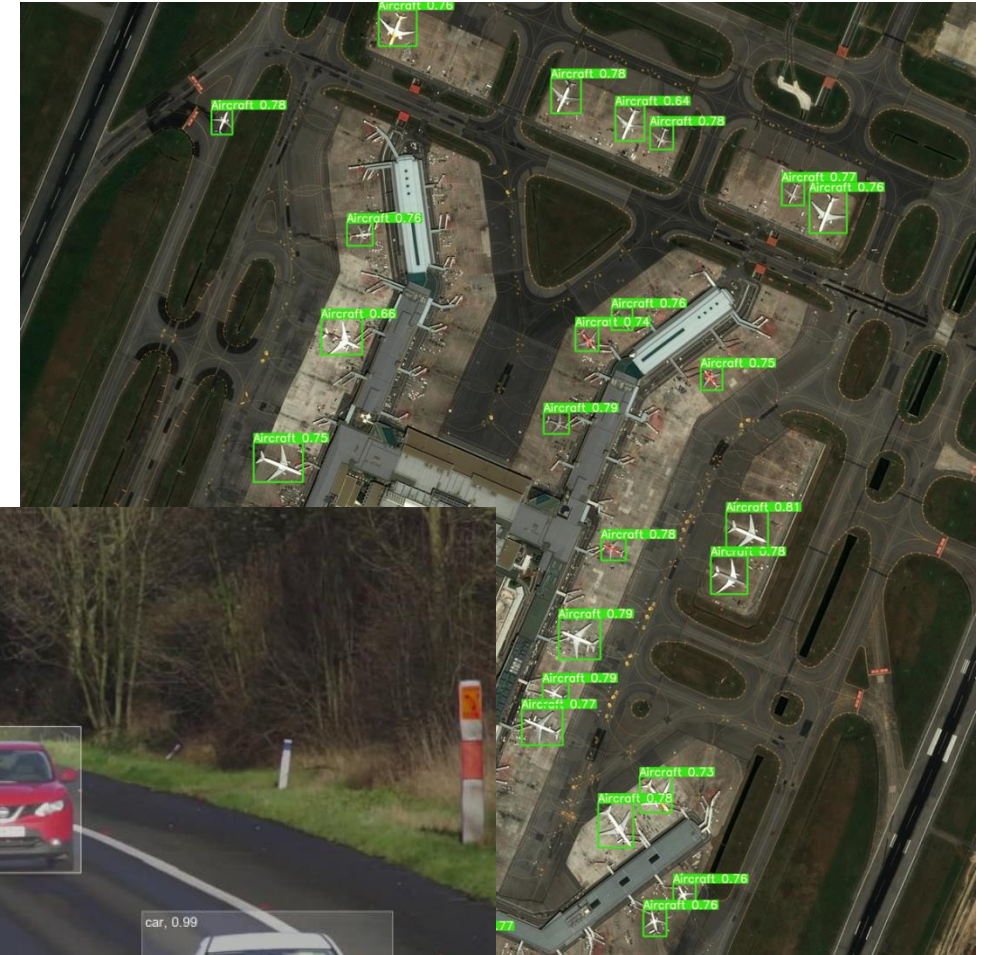
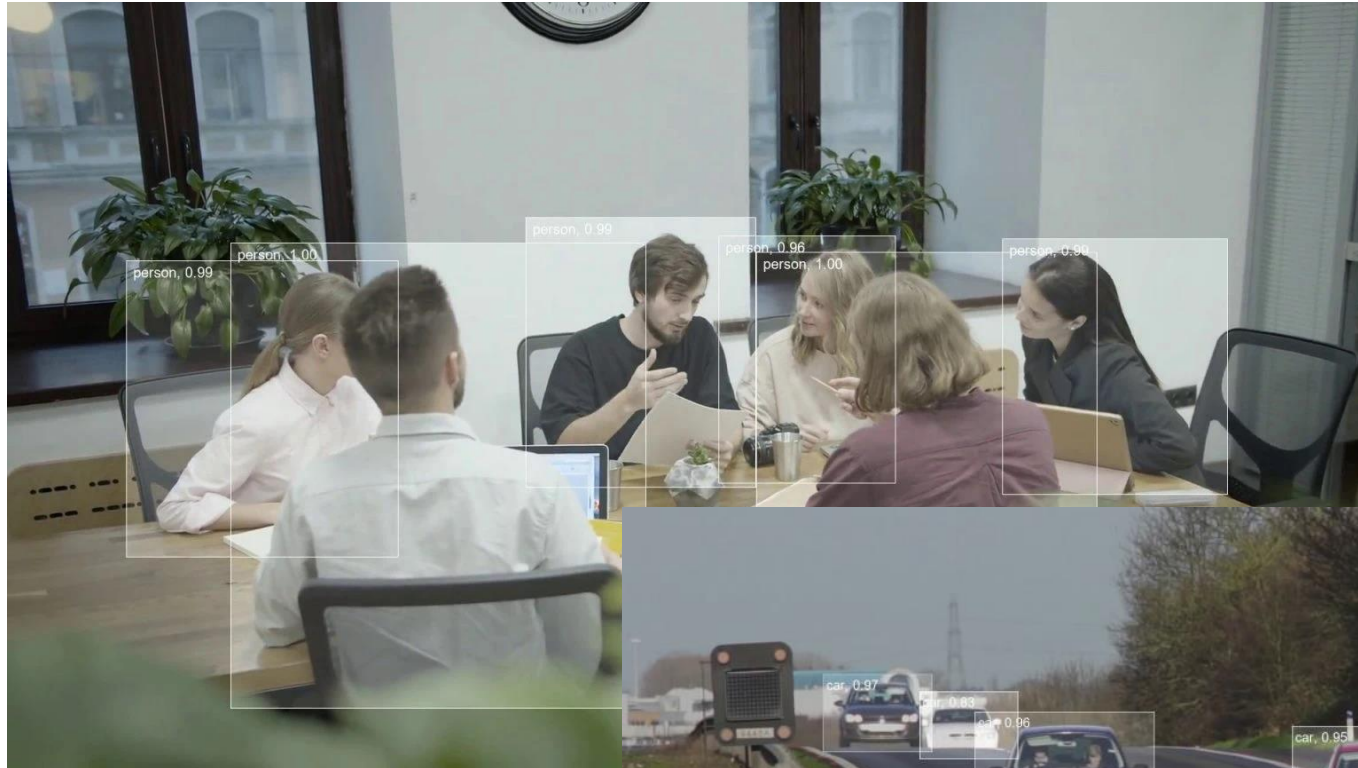
Email: [pnmnhut@vku.udn.vn](mailto:pnmnhut@vku.udn.vn)

Phone: 0903.501.421

- Phát hiện đối tượng là gì và nó đã phát triển như thế nào trong 20 năm qua
- Các phương pháp phát hiện đối tượng
- Ứng dụng phát hiện đối tượng
- Các thuật toán phát hiện đối tượng

- Phát hiện đối tượng là một nhiệm vụ thị giác máy tính quan trọng
- Được sử dụng để phát hiện các đối tượng
  - Ví dụ: con người, động vật, ô tô hoặc tòa nhà...
  - Trong ảnh số hoặc khung hình video
- Mục tiêu của phát hiện đối tượng là phát triển các mô hình tính toán
  - Cung cấp thông tin cần thiết cho các ứng dụng thị giác máy tính
  - Trả lời cho câu hỏi “Đối tượng đó là gì và ở đâu?”





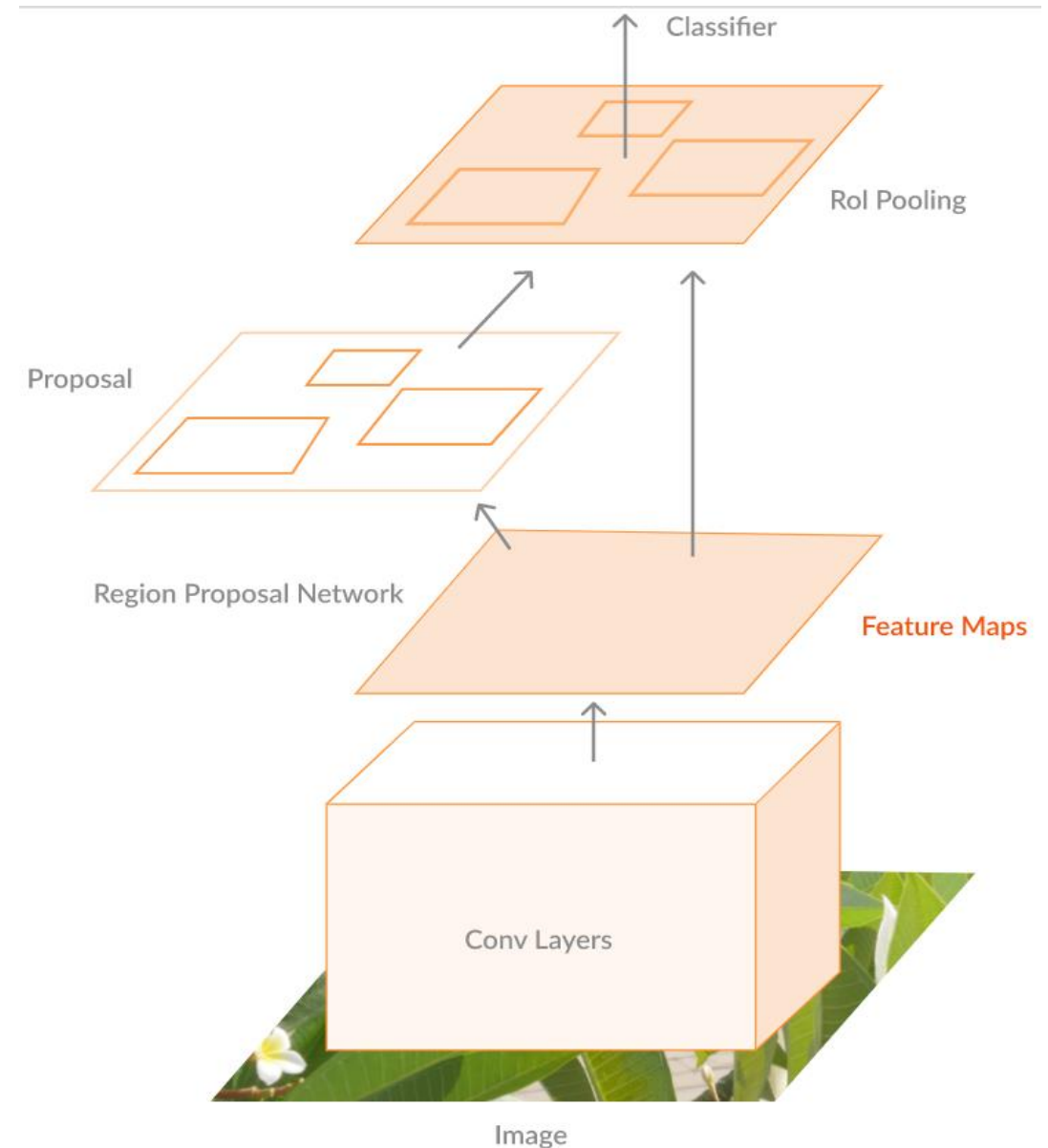
- Phát hiện đối tượng có thể được thực hiện bằng:
  - Kỹ thuật xử lý ảnh truyền thống
  - Machine Learning/Deep Learning
- Kỹ thuật xử lý ảnh không yêu cầu dữ liệu để đào tạo và về bản chất không được giám sát.
  - Ưu điểm: Không cần ảnh có gán nhãn, gán nhãn thủ công.
  - Nhược điểm: Bị hạn chế ở nhiều yếu tố, như: các tình huống phức tạp (không có nền đơn sắc), che khuất (đối tượng bị ẩn một phần), hiệu ứng chiếu sáng và đổ bóng ...
- Phương pháp Deep Learning thường học có giám sát hoặc không giám sát
  - Ưu điểm: Phát hiện đối tượng bằng Deep Learning rất hiệu quả đối với các cảnh và ánh sáng phức tạp
  - Nhược điểm: Cần có một lượng lớn dữ liệu đào tạo; quá trình gán nhãn ảnh tốn kém

- Trước năm 2014 – Giai đoạn Phát hiện đối tượng truyền thống:
  - Viola-Jones Detector (2001), công trình tiên phong phát triển các phương pháp phát hiện đối tượng truyền thống
  - HOG Detector (2006), bộ mô tả đặc trưng để phát hiện đối tượng trong thị giác máy tính và xử lý ảnh
  - DPM (2008), giới thiệu kỹ thuật hồi quy hộp giới hạn (bounding box regression) để xác định vùng ảnh có khả năng chứa đối tượng
- Sau năm 2014 – Giai đoạn Phát hiện đối tượng sử dụng Deep Learning:
  - Các thuật toán phát hiện đối tượng hai giai đoạn (two-stage): RCNN, SPPNet (2014) Fast RCNN, Faster RCNN (2015), Mask R-CNN (2017), Pyramid Networks/FPN (2017), G-RCNN (2021)...
  - Các thuật toán phát hiện đối tượng một giai đoạn (one-stage): YOLO (2016), SSD (2016), RetinaNet (2017), YOLOv3 (2018), YOLOv4 (2020), YOLOR (2021), YOLOv7 (2022)...



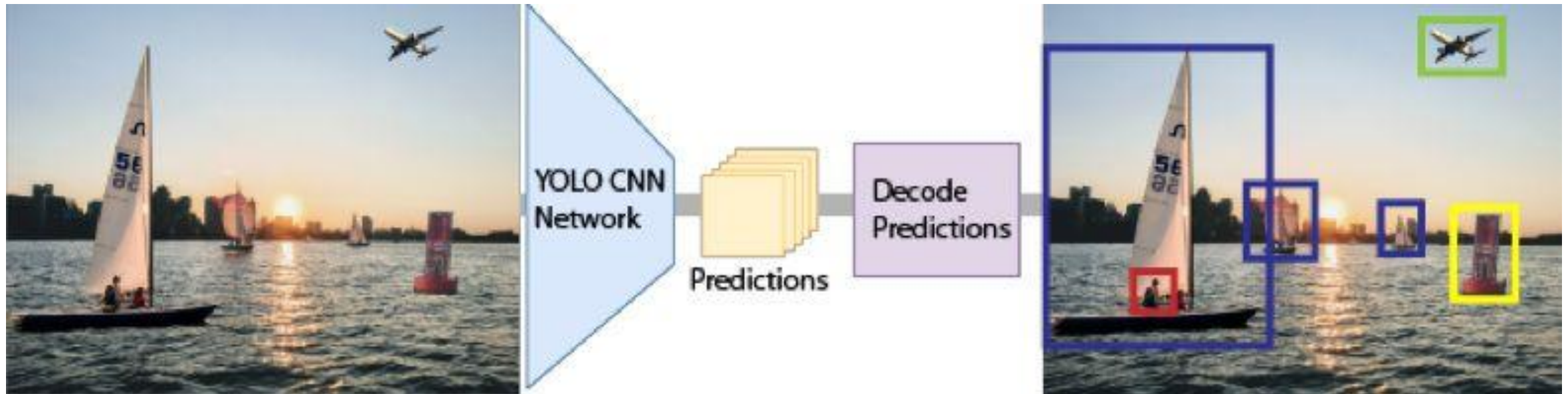
- Bộ phát hiện đối tượng dựa trên deep learning trích xuất đặc trưng từ hình ảnh hoặc khung hình video → Giải quyết 02 nhiệm vụ tiếp theo:
  - (1) Tìm số lượng đối tượng
  - (2) Phân loại đối tượng và ước tính kích thước của nó bằng hộp giới hạn (bounding box)
- Two-stage detector:
  - Giai đoạn 1: Đề xuất các vùng đối tượng gần đúng
  - Giai đoạn 2: Sử dụng hồi quy hộp giới hạn (bounding box regression) để xác định vùng đối tượng chính xác trong số các vùng đối tượng gần đúng ở giai đoạn 1 và Phân loại ảnh đối tượng.
  - Độ chính xác dự đoán cao, nhưng thời gian thực hiện chậm
- One-stage detector:
  - Dự đoán các hộp giới hạn trên ảnh mà không cần qua giai đoạn đề xuất các vùng đối tượng gần đúng
  - Quá trình này tiêu tốn ít thời gian hơn, do đó được sử dụng trong ứng dụng thời gian thực

- Ví dụ: Two-stage detector: Kiến trúc mạng Faster R-CNN
  - Lớp Tích chập (Conv) tạo các bản đồ đặc trưng ảnh (Feature Maps) từ ảnh đầu vào
  - Mạng đề xuất vùng (Region Proposal Network) được áp vào Feature Maps để trả về các vùng đề xuất (Proposal) cùng với Score của đối tượng.
  - Lớp RoI Pooling được áp dụng cho các Proposal này để tạo ra Feature Map có kích thước nhỏ hơn
  - Các Feature Map này (chứa các Proposal) được chuyển vào lớp được kết nối đầy đủ (Full Connection) và sử dụng hàm Softmax để phân lớp cho các đối tượng và tạo ra các Bounding Box cho các đối tượng.

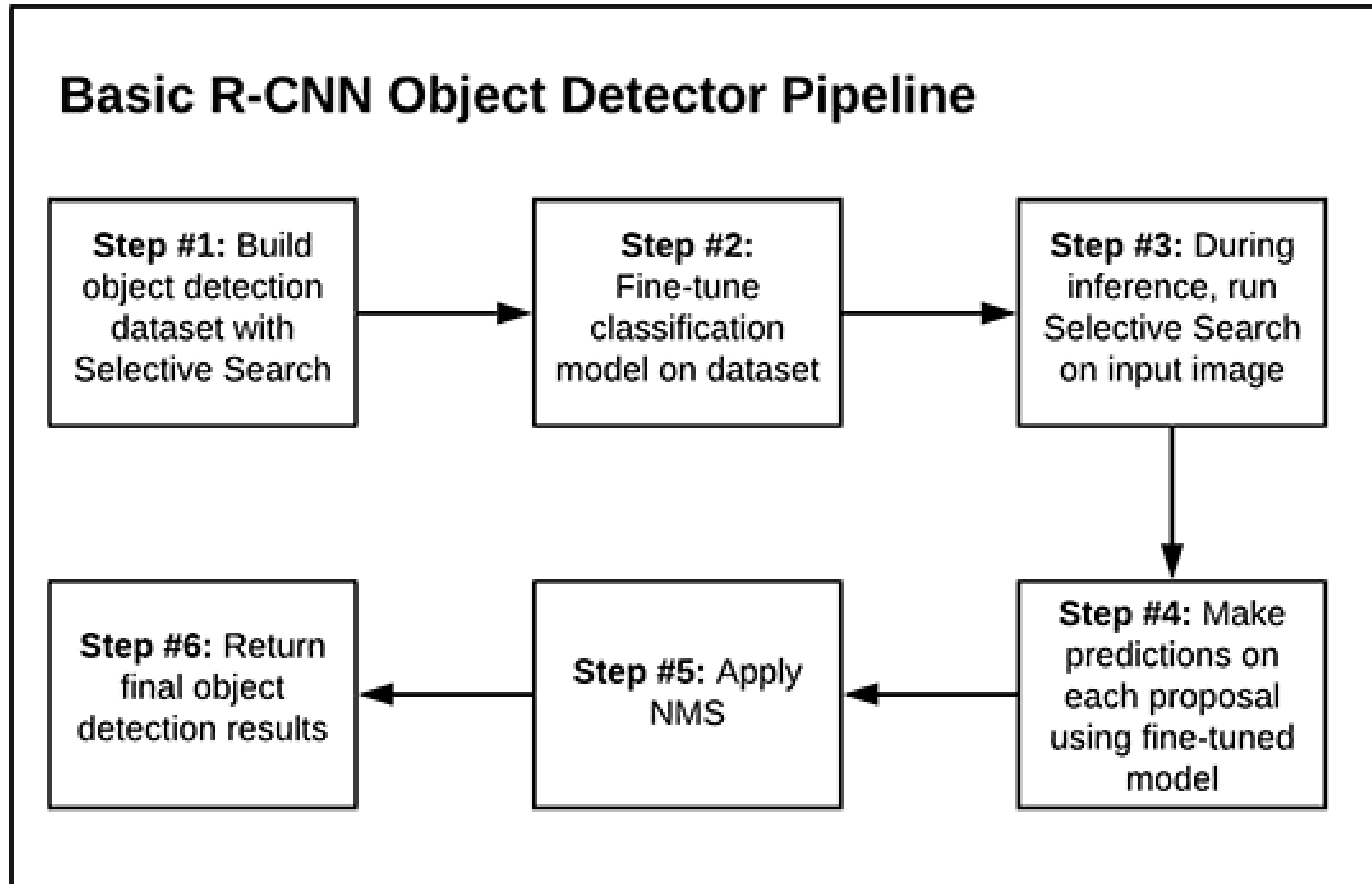




- Ví dụ: One-stage detector: Kiến trúc mạng Yolo



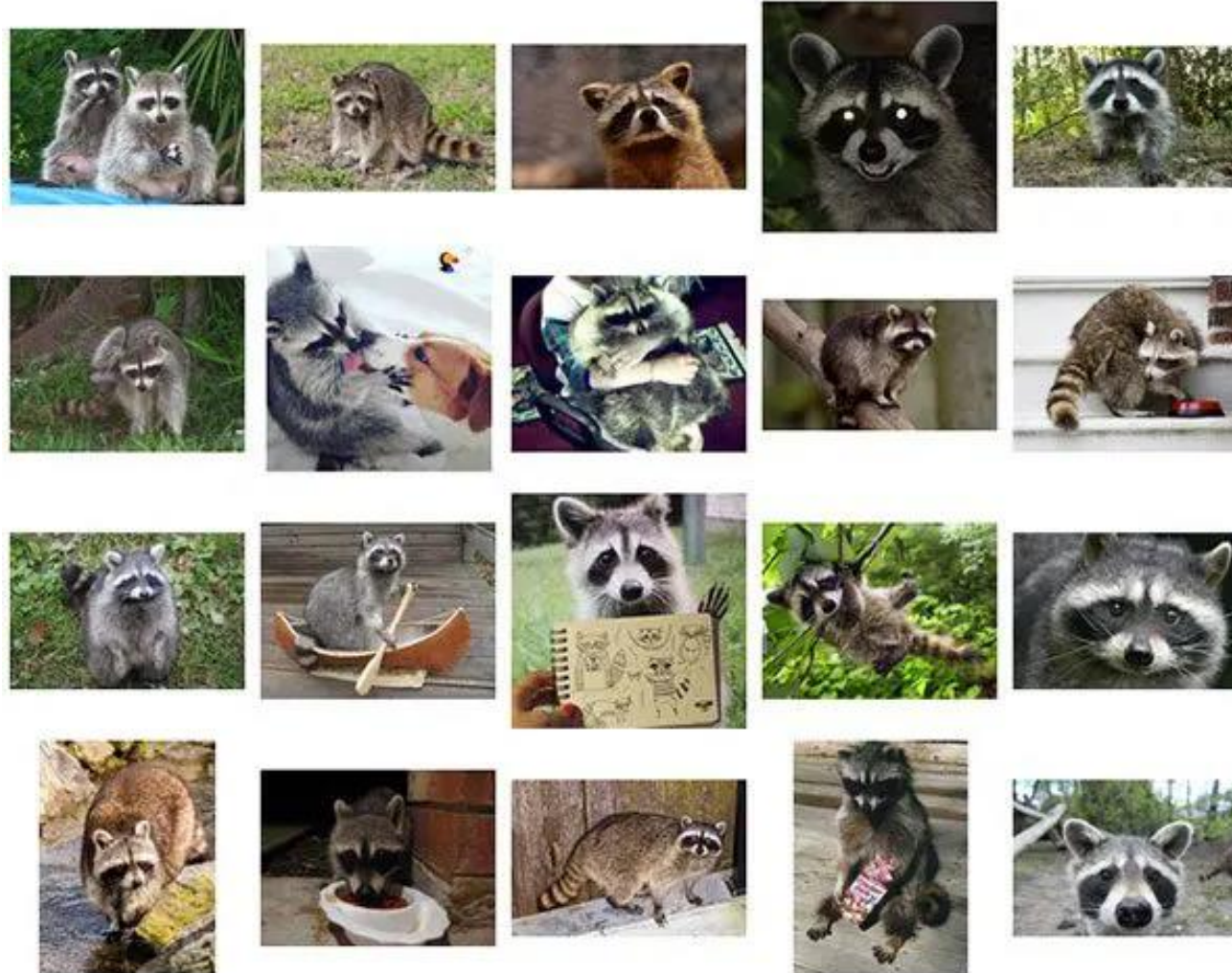
- Sơ đồ các bước thực hiện



- **Bước 1:** Xây dựng tập dữ liệu
- **Bước 2:** Tinh chỉnh mạng phân loại (được đào tạo ban đầu trên ImageNet) để phát hiện đối tượng
- **Bước 3:** Tạo tập lệnh suy luận phát hiện đối tượng sử dụng Tìm kiếm chọn lọc để đề xuất các vùng có *thể* chứa đối tượng mà chúng ta muốn phát hiện
- **Bước 4:** Sử dụng mạng tinh chỉnh để phân loại từng khu vực được đề xuất thông qua Tìm kiếm chọn lọc
- **Bước 5:** Áp dụng triệt tiêu không cực đại để triệt tiêu các hộp giới hạn yếu, chồng chéo
- **Bước 6:** Trả về kết quả phát hiện đối tượng cuối cùng

## 4. Phát hiện đối tượng bằng thuật toán R-CNN

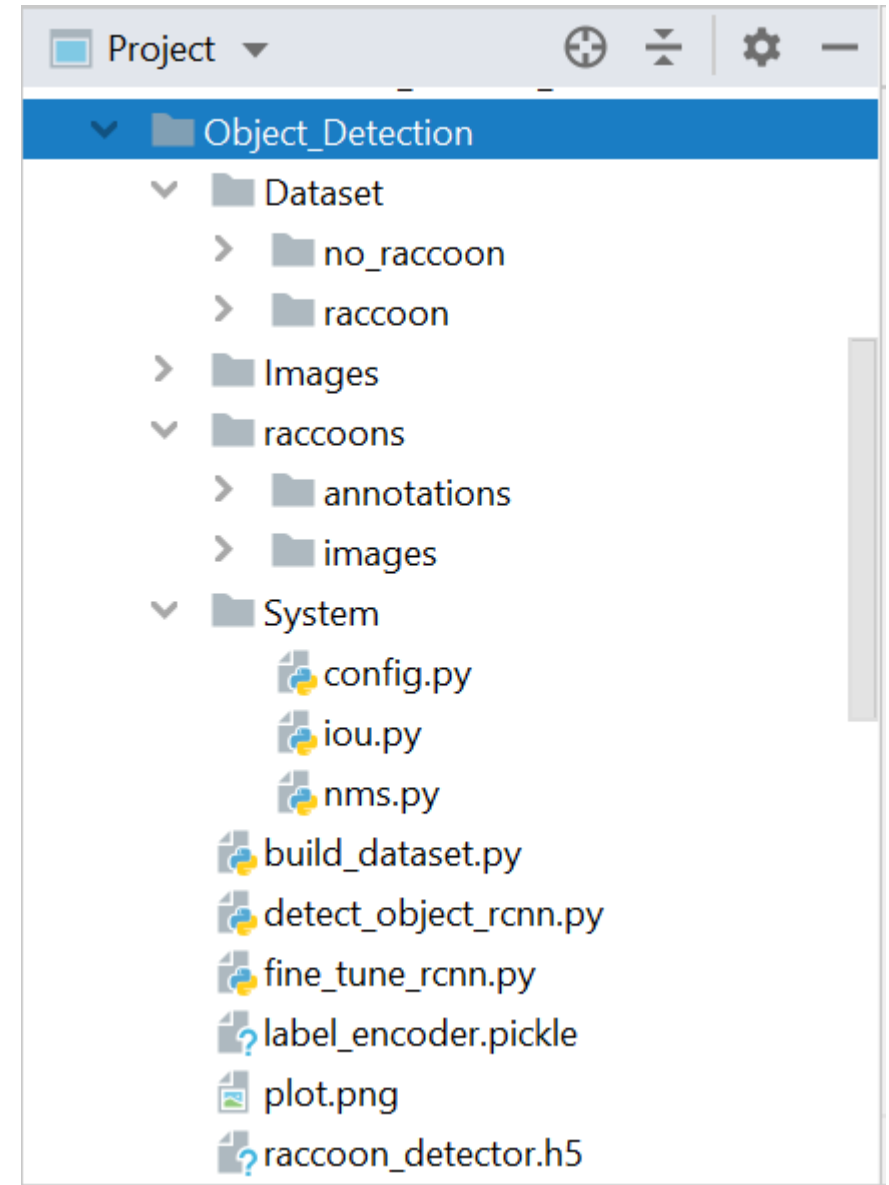
- Bộ dữ liệu này chứa **200 hình ảnh** với **tổng số 217 con gấu trúc** (một số hình ảnh chứa nhiều hơn một con gấu trúc).





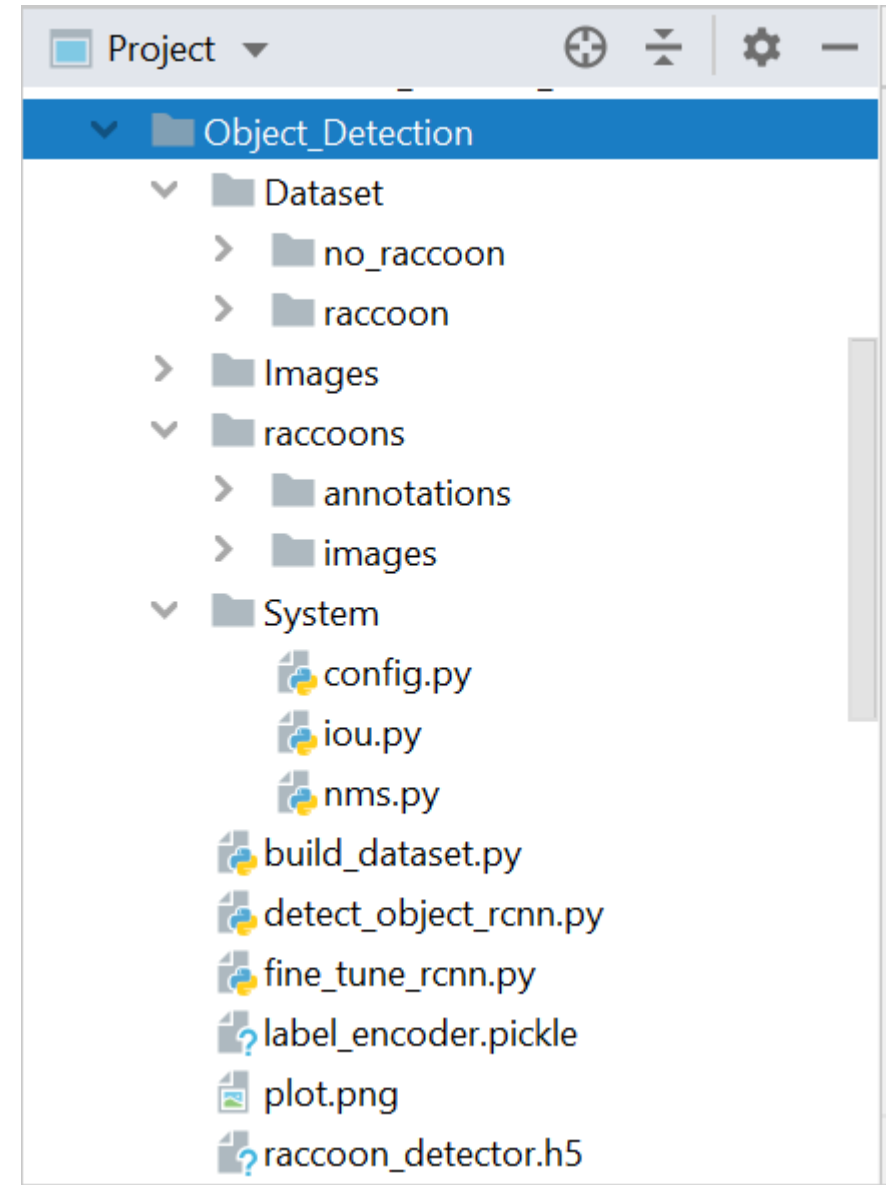
- Cấu trúc dự án

- Dataset:** chứa dữ liệu để huấn luyện (tinh chỉnh) model phân lớp (2 lớp), dùng mạng MobileNet V2 đã được huấn luyện trước bởi dữ liệu ImageNet
- Images:** chứa các file ảnh để thực hiện Object Detection
- Raccoons:** Chứa ảnh đã được găng nhãn (chú thích tọa độ của đối tượng (raccoon) trong ảnh)
- iou.py:** Tính toán chỉ số Intersection over Union (IoU) , đánh giá thuật toán phát hiện đối tượng
- config.py:** cấu hình
- nms.py:** Thực hiện thuật toán non-maxima suppression (NMS) để loại bỏ các hộp chồng chéo xung quanh các đối tượng



- Cấu trúc dự án

- **build\_dataset.py**: Tách dữ liệu no\_raccoon và raccoon từ bộ dữ liệu gốc raccoon để huấn luyện (tinh chỉnh) model MobileNet V2 được đào tạo trước trên bộ dữ liệu ImageNet
- **fine\_tune\_rcnn.py**: Huấn luyện bộ phân loại (2 lớp) bằng phương pháp tinh chỉnh
- **detect\_object\_rcnn.py**: Kết hợp các phần lại với nhau để thực hiện phát hiện đối tượng R-CNN.



- **Bước 1: Xây dựng tập dữ liệu**

- Sao chép tập dữ liệu raccoons, bao gồm:
  - Lưu File ảnh vào **folder image**
  - Lưu File XML theo chuẩn Pascal Voc vào **folder annotations**: Chứa các chú thích phát hiện đối tượng trong ảnh
- Thực thi file **build\_dataset.py** để tách dữ liệu no\_raccoon và raccoon từ bộ dữ liệu gốc raccoons để huấn luyện (tinh chỉnh) model MobileNet V2
- ➔ 1.560 hình ảnh “raccoon” và 2.200 hình ảnh “no\_raccoon”

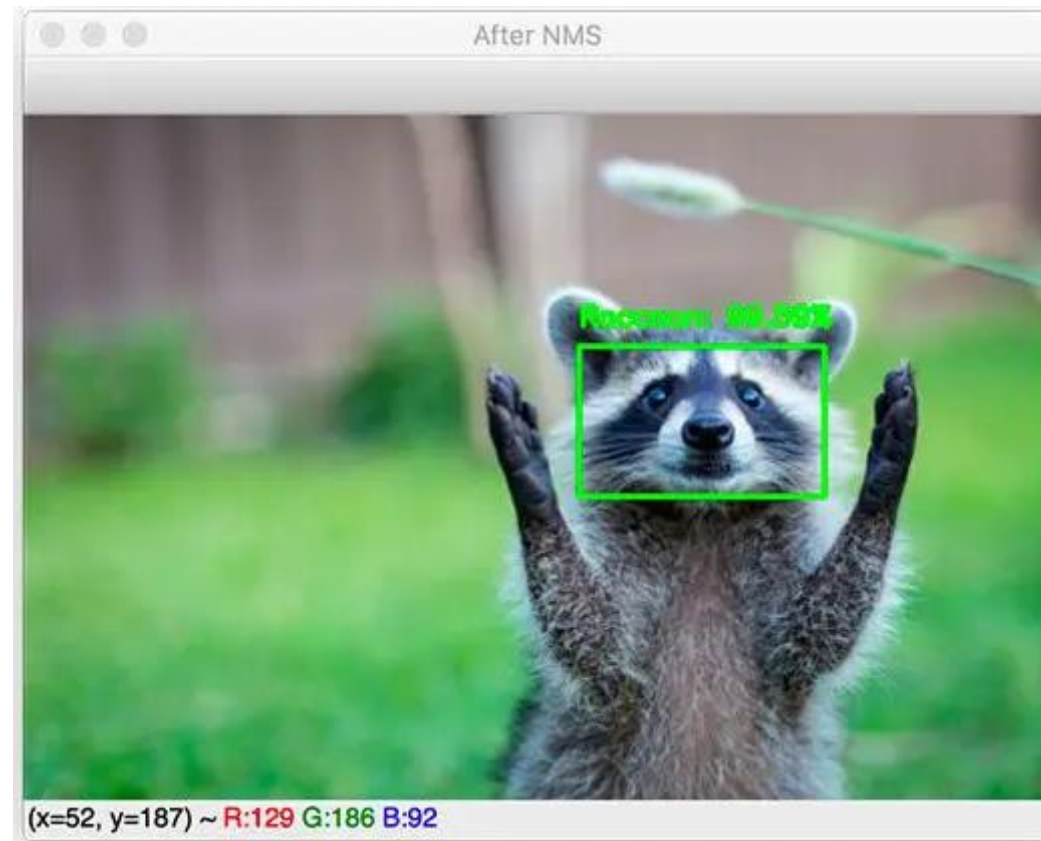
- **Bước 2:** Huấn luyện model MobileNetV2 (bằng phương pháp tinh chỉnh)
  - Thực thi file **`fine_tune_rcnn.py`**: Huấn luyện bộ phân loại (2 lớp) bằng phương pháp tinh chỉnh
  - Kết quả





- **Bước 3,4,5,6:**

- Thực thi **file detect\_object\_rcnn.py**: Kết hợp các phần lại với nhau để thực hiện phát hiện đối tượng bằng model R-CNN.



- **Labelling** dùng để đánh nhãn vật thể trong hình để training các model detection
- Cài đặt Labelling:
  - **Bước 1:** Clone source code từ repo: <https://github.com/tzutalin/labelling>.
  - **Bước 2:** Cài PyQt5 bằng cmd: `pip install PyQt5`
  - **Bước 3:** Tiếp theo là cài lxml: `pip install lxml`
  - **Bước 4:** Copy các file cần thiết bằng cmd: `pyrcc5 -o libs/resources.py resources.qrc`
- Thực thi Labelling: `labelling.py`
- Tham khảo: <https://pypi.org/project/labelling/1.4.0/>