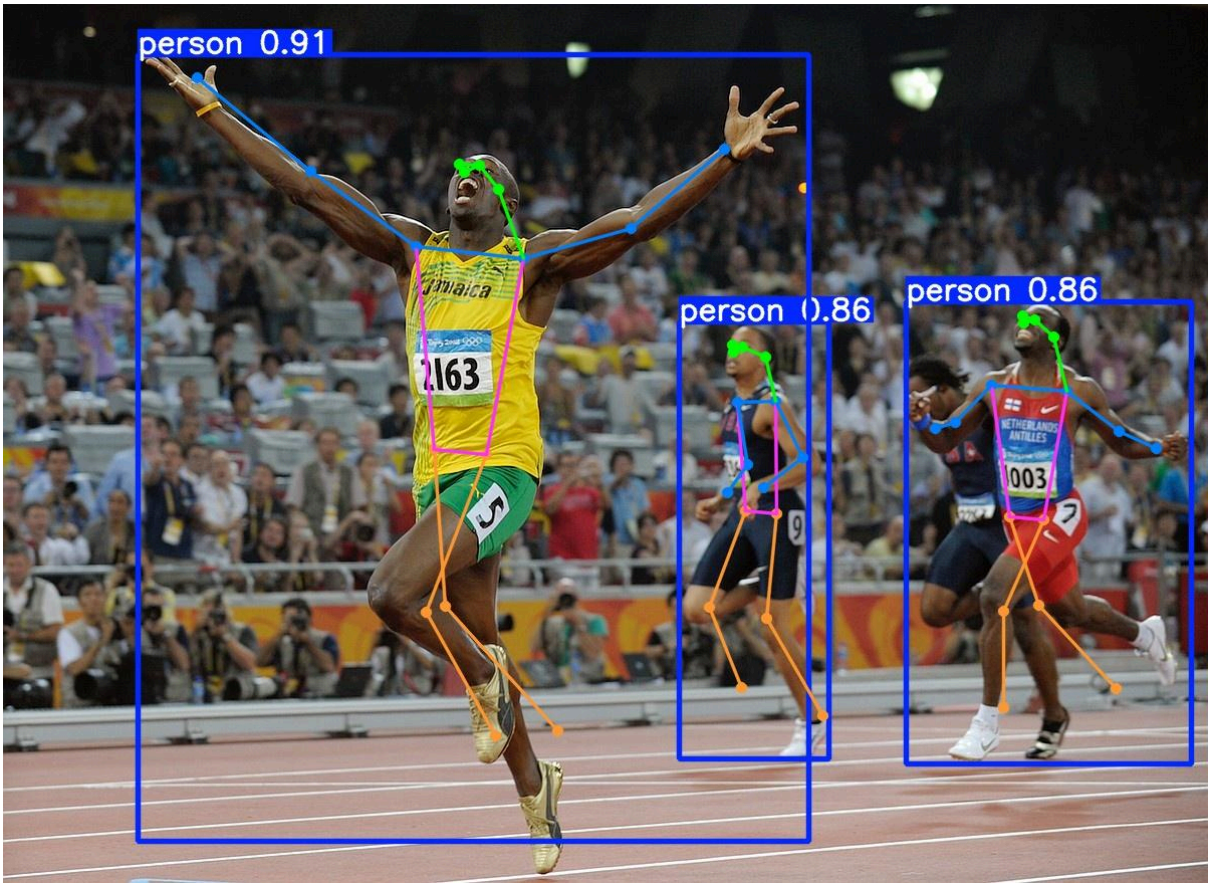
# Human Pose Estimation

## 1. Giới thiệu

### 1.1 Tổng quan dự án

Dự án này tập trung vào việc Human Pose Estimation, đặc biệt là nghiên cứu và hiểu rõ các vấn đề và mô hình liên quan đến nhiệm vụ thị giác máy tính này. Mục tiêu chính là huấn luyện và phân tích hai kiến trúc khác nhau: RCNN-ResNet50 và YOLOv11-Pose. Thông qua công việc này, em hướng đến việc hiểu sâu hơn về các loại dữ liệu, kiến trúc mô hình, và những thách thức cơ bản trong bài toán Pose Estimation




### 1.2 Mô tả vấn đề

Human Pose Estimation liên quan đến việc phát hiện và xác định vị trí các keypoints (điểm khớp) giải phẫu trên cơ thể con người trong ảnh hoặc khung hình video. Nhiệm vụ này đóng vai trò quan trọng cho nhiều ứng dụng bao gồm nhận dạng hành động, theo dõi chuyển động, thực tế tăng cường và tương tác người-máy. Thách

thức nằm ở việc phát hiện chính xác các keypoints qua nhiều tư thế, điều kiện ánh sáng, che khuất và tỷ lệ khác nhau.

### 1.3 Mục tiêu dự án

- Triển khai và huấn luyện các mô hình RCNN-ResNet50 và YOLOv11-Pose cho việc phát hiện keypoint
- Đánh giá và so sánh hiệu suất của cả hai phương pháp sử dụng các tiêu chí đánh giá tiêu chuẩn
- Xác định điểm mạnh, điểm yếu và các cân nhắc thực tế cho từng mô hình
- Hiểu rõ các thách thức cốt lõi trong bài toán Human Pose Estimationi

## 2. Phương pháp

### 2.1 Bộ dữ liệu

Đối với dự án này, em sử dụng bộ dữ liệu COCO 2017 cho bài toán Keypoint Detection, bao gồm hơn 118 nghìn hình ảnh với hơn 200.000 đối tượng người được gán nhãn với các điểm khớp. Sơ đồ keypoints của COCO xác định 17 keypoints cho mỗi người, bao gồm các keypoints như vai, khuỷu tay, cổ tay, hông, đầu gối và mắt cá chân.

#### 2.1.1 Tiền xử lý dữ liệu

em đã triển khai một lớp dữ liệu tùy chỉnh `CocoKeypoint` để xử lý các chú thích COCO và chuẩn bị dữ liệu cho việc huấn luyện:

Lớp này xử lý:

- Tải hình ảnh và tệp chú thích từ bộ dữ liệu COCO
- Lọc hình ảnh dựa trên yêu cầu keypoints
- Chuyển đổi định dạng keypoints từ mảng phẳng của COCO sang ma trận có cấu trúc
- Chuyển đổi định dạng bounding box từ XYWH sang XYXY
- Chuẩn bị biểu diễn tensor cho quá trình huấn luyện mô hình

### 2.2 Kiến trúc mô hình

#### 2.2.1 RCNN-ResNet50

Mô hình RCNN-ResNet50 là một framework phát hiện hai giai đoạn, đầu tiên xác định các đối tượng người sau đó dự đoán vị trí điểm khớp cho mỗi người được phát hiện.

**Kiến trúc ResNet50**

ResNet50 (Residual Network 50 layers) là một mạng nơ-ron tích chập sâu được giới thiệu bởi Microsoft Research vào năm 2015. Điểm đặc biệt của ResNet là việc sử dụng các "kết nối tắt" (skip connections) để giải quyết vấn đề tiêu biến gradient trong các mạng sâu. Kiến trúc này bao gồm:

- **Convolutional Layer đầu tiên**: Sử dụng bộ lọc 7x7 với stride 2, theo sau là max pooling
- **Residual Blocks**: 16 block dư thừa được tổ chức thành 4 nhóm (layers):
    - Layer 1: 3 blocks với 64 kênh
    - Layer 2: 4 blocks với 128 kênh
    - Layer 3: 6 blocks với 256 kênh
    - Layer 4: 3 blocks với 512 kênh
- **Bottleneck Architecture**: Mỗi block bao gồm 3 lớp phức hợp (1x1, 3x3, 1x1) được thiết kế để giảm và sau đó tăng số lượng kênh
- **Skip Connections**: Kết nối tắt cho phép gradient dễ dàng lưu thông qua mạng, giúp huấn luyện mạng sâu hơn mà không bị suy giảm hiệu suất
- **Global Average Pooling và Fully Connected Layer**: Cuối cùng, đặc trưng được trích xuất được xử lý thông qua pooling toàn cục và một lớp fully connected

**Kiến trúc RCNN (Region-based Convolutional Neural Network)**

RCNN đã phát triển qua nhiều phiên bản (R-CNN, Fast R-CNN, Faster R-CNN) với cải tiến dần về hiệu suất. Faster R-CNN, phiên bản được sử dụng với ResNet50, bao gồm các thành phần sau:

- **Backbone Network**: ResNet50 được sử dụng để trích xuất các đặc trưng từ hình ảnh đầu vào
- **Region Proposal Network (RPN)**: Một mạng nơ-ron chạy trên bản đồ đặc trưng để đề xuất các vùng có thể chứa đối tượng
    - Sử dụng "anchors" ở nhiều tỷ lệ và kích thước để đề xuất vùng tiềm năng
    - Đánh giá mỗi anchor là foreground hay background
    - Tinh chỉnh vị trí và kích thước của các vùng đề xuất
- **ROI Align**: Trích xuất đặc trưng có kích thước cố định cho mỗi vùng đề xuất
    - Sử dụng nội suy song tuyến để bảo toàn thông tin không gian
    - Tạo ra bản đồ đặc trưng có kích thước cố định cho mỗi vùng đề xuất
- **Classification và Bounding Box Regression**: Dự đoán lớp của đối tượng và tinh chỉnh bounding box
- **Keypoint Head**: Một thành phần bổ sung để dự đoán vị trí của 17 điểm khớp cho mỗi người được phát hiện
    - Sử dụng kiến trúc fully convolutional để tạo ra heatmap cho mỗi điểm khớp

    - Dự đoán độ tin cậy và vị trí chính xác của mỗi điểm khớp

Quy trình hoạt động của RCNN-ResNet50 cho ước lượng tư thế:

1. Hình ảnh đầu vào được đưa qua backbone ResNet50 để trích xuất bản đồ đặc trưng
2. RPN đề xuất các vùng có thể chứa người
3. ROI Align trích xuất đặc trưng có kích thước cố định cho mỗi vùng đề xuất
4. Phần phân loại xác định vùng nào chứa người
5. Đối với mỗi vùng được xác định là người, phần keypoint head dự đoán vị trí của 17 điểm khớp

### 2.2.2 YOLOv11-Pose

YOLOv11-Pose là một detector một giai đoạn, đồng thời dự đoán bounding box và các điểm khớp trong một lần chạy. Kiến trúc của YOLOv11-Pose bao gồm các thành phần sau:

#### Kiến trúc YOLOv11-Pose chi tiết

YOLOv11-Pose dựa trên kiến trúc YOLO (You Only Look Once), một trong những phương pháp phát hiện đối tượng một giai đoạn hiệu quả nhất. Phiên bản này được cải tiến đặc biệt cho nhiệm vụ ước lượng tư thế:

- **Backbone Network**: Sử dụng CSPDarknet (Cross Stage Partial Network) cải tiến
    - Tích hợp kiến trúc CSP (Cross Stage Partial) để tăng hiệu suất nhận dạng
    - Sử dụng các kết nối dư thừa (residual connections) để cải thiện luồng gradient
    - Áp dụng SPPF (Spatial Pyramid Pooling - Fast) để bắt được đặc trưng ở các tỷ lệ khác nhau
    - Tích hợp các khối Convolutional Block Attention Module (CBAM) để tập trung vào các vùng quan trọng
- **Neck Network**: Sử dụng kiến trúc PANet (Path Aggregation Network) cải tiến
    - Tạo ra Feature Pyramid Network (FPN) để xử lý đối tượng ở nhiều tỷ lệ
    - Tích hợp các đường dẫn bottom-up và top-down để cải thiện luồng thông tin
    - Sử dụng kết nối CSP để giảm khả năng quá khớp và tăng tốc độ
- **Detection Head**: Thiết kế đặc biệt cho nhiệm vụ phát hiện tư thế
    - Phương pháp anchor-free để dự đoán trực tiếp vị trí đối tượng
    - Phân loại nhãn thông qua mạng fully connected
    - Hồi quy bbox qua mạng fully connected
    - Head riêng biệt cho dự đoán keypoint heatmap và offset

    - Sử dụng encoding-decoding mechanism để cải thiện độ chính xác của keypoint
- **Pose Estimation Components**:
    - Dự đoán đồng thời vị trí người và 17 điểm khớp
    - Sử dụng biểu diễn heatmap cho mỗi điểm khớp
    - Áp dụng các kỹ thuật trung bình hóa trọ
    - Áp dụng các kỹ thuật trung bình hóa trọng số để tăng độ chính xác khi xác định tọa độ cuối cùng
    - Sử dụng cơ chế OKS (Object Keypoint Similarity) để đánh giá độ chính xác của dự đoán
    - Tích hợp kỹ thuật data association cho việc theo dõi keypoint trong trường hợp nhiều người
- **Kỹ thuật tối ưu hóa**:
    - Sử dụng phương pháp Focal Loss để giải quyết vấn đề mất cân bằng giữa foreground và background
    - Áp dụng Non-Maximum Suppression (NMS) thông minh để lọc các dự đoán trùng lặp
    - Kỹ thuật Mosaic và MixUp cho augmentation dữ liệu trong quá trình huấn luyện
    - Tối ưu hóa CUDA để tận dụng tối đa hiệu năng GPU
    - Áp dụng kỹ thuật anchor-free giúp giảm đáng kể số lượng tham số và tăng tốc độ xử lý

So với phương pháp hai giai đoạn như RCNN-ResNet50, YOLOv11-Pose cung cấp các lợi thế sau:

1. Tốc độ xử lý nhanh hơn đáng kể do kiến trúc một giai đoạn
2. Pipeline đơn giản hơn, loại bỏ nhu cầu về mạng đề xuất vùng riêng biệt
3. Xử lý end-to-end, giảm thiểu lỗi tích lũy giữa các giai đoạn
4. Khả năng xử lý thời gian thực trên các thiết bị có tài nguyên hạn chế
5. Phương pháp học tập trực tiếp từ dữ liệu, giảm thiểu thiết kế thủ công

## 2.3 Thiết lập huấn luyện

Do giới hạn thời gian GPU nên em mới chỉ thực hiện train model RCNN, model YoloV11 em lấy bộ pretrain thông qua package ultralytics

#### 2.3.1 Thông số huấn luyện

Các mô hình được huấn luyện với cấu hình sau:

- Số lượng epoch: 42
- Kích thước batch: 8
- Số lượng điểm khớp: 17

#### 2.3.2 Tối ưu hóa

- Optimizer: SGD với các thông số sau:
    - Tốc độ học (Learning rate): 0.02
    - Momentum: 0.9
    - Weight decay: 1e-4
- Bộ lập lịch tốc độ học (Learning rate scheduler):
    - Multi-step decay tại epoch 36 và 43
    - Hệ số giảm (gamma): 0.1

#### 2.3.3 Công cụ triển khai

Quá trình triển khai sử dụng các framework và thư viện sau:

- Python làm ngôn ngữ lập trình
- PyTorch và torchvision cho việc triển khai và huấn luyện mạng nơ-ron
- Ultralytics package cho việc triển khai YOLOv11-Pose. Em lựa chọn model yolov11-nano để có thời gian infer nhanh nhất, bù lại độ chính xác sẽ bị giảm đi
- COCO API (pycocotools) cho việc xử lý bộ dữ liệu

# 3. Kết quả và phân tích

## 3.1 Kết quả định lượng

#### 3.1.1 Các chỉ số đánh giá độ chính xác

Bảng 1: So sánh hiệu suất của RCNN-ResNet50 và YOLOv11-Pose trên tập kiểm định COCO

| | mAP | mAP50 | Time |
|---|---|---|---|
| RCNN | 50 | 81 | 0.5 |
| Yolo | 54.9 | 82.5 | 9s |

## 3.2 Phân tích

#### 3.2.1 RCNN-ResNet50

Mô hình RCNN-ResNet50 thể hiện độ chính xác tuyệt vời trong việc xác định vị trí điểm khớp, đặc biệt là cho các đối tượng được nhìn thấy rõ ràng trong điều kiện ánh sáng tốt. Phương pháp phát hiện hai giai đoạn góp phần vào độ chính xác cao với chi phí là tốc độ. Mô hình thể hiện các đặc điểm sau:

- Điểm mạnh:
    - Hiệu suất ổn định cho nhiều người trong cùng một cảnh
- Điểm yếu:
    - Thời gian suy luận chậm đáng kể
    - Yêu cầu tính toán cao hơn
    - Thỉnh thoảng bỏ sót phát hiện trong các tư thế hoặc điều kiện ánh sáng khó khăn

#### 3.2.2 YOLOv11-Pose

Mô hình YOLOv11-Pose mang lại tốc độ ấn tượng trong khi vẫn duy trì độ chính xác cạnh tranh. Phương pháp phát hiện một giai đoạn cung cấp giải pháp hiệu quả hơn nhiều phù hợp cho các ứng dụng thời gian thực. Mô hình thể hiện:

- Điểm mạnh:
    - Tốc độ suy luận nhanh hơn đáng kể (nhanh hơn 18 lần so với RCNN)
    - Độ chính xác tương đương với phương pháp hai giai đoạn
- Điểm yếu:
    - Định vị điểm khớp kém chính xác hơn một chút trong các cảnh phức tạp
    - Nhạy cảm hơn đối với các trường hợp bị che khuất

### 3.3 Thách thức gặp phải

Trong quá trình triển khai và thử nghiệm, một số thách thức đã được gặp phải:

1. **Hiệu quả xử lý dữ liệu**: Kích thước và độ phức tạp của bộ dữ liệu COCO đòi hỏi quy trình tải và tiền xử lý dữ liệu hiệu quả để tránh trở thành nút thắt cổ chai trong quá trình huấn luyện..
2. **Xử lý các điểm bị che khuất**: Cả hai mô hình đều gặp khó khăn với các keypoints bị che khuất nặng, mặc dù RCNN thể hiện độ mạnh mẽ tốt hơn một chút trong các tình huống này.
3. **Hạn chế bộ nhớ**: Yêu cầu bộ nhớ của mô hình RCNN hạn chế kích thước batch, điều này ảnh hưởng đến thời gian huấn luyện và sự ổn định.

## 4. Thảo luận

### 4.1 So sánh mô hình

Kết quả thể hiện sự cân bằng cổ điển giữa độ chính xác và tốc độ trong các nhiệm vụ thị giác máy tính. RCNN-ResNet50 cung cấp độ chính xác cao hơn một chút cho việc định vị điểm khớp nhưng với chi phí đáng kể về thời gian suy luận. YOLOv11-Pose hy sinh một chút độ chính xác để cải thiện đáng kể về tốc độ, làm cho nó phù hợp hơn nhiều cho các ứng dụng thời gian thực.

Đối với các ứng dụng mà độ chính xác tuyệt đối là quan trọng và tài nguyên tính toán ít bị hạn chế hơn, RCNN có thể là lựa chọn ưu tiên. Tuy nhiên, đối với hầu hết các trường hợp sử dụng thực tế, YOLOv11-Pose cung cấp sự cân bằng tốt hơn nhiều giữa độ chính xác và hiệu quả.

## 4.2 Cân nhắc kiến trúc

Sự khác biệt kiến trúc cơ bản giữa phương pháp hai giai đoạn RCNN và phương pháp một giai đoạn YOLO làm nổi bật các cân nhắc thiết kế quan trọng:

1. **Trích xuất đặc trưng**: Cả hai mô hình đều sử dụng mạng backbone mạnh, nhưng các giai đoạn đề xuất vùng và ước lượng điểm khớp riêng biệt của RCNN cho phép trích xuất đặc trưng chuyên biệt hơn ở mỗi bước.
2. **Tính bất biến tỷ lệ**: Phương pháp kim tự tháp đặc trưng của YOLOv11-Pose dường như xử lý các tỷ lệ khác nhau hiệu quả hơn, đặc biệt là đối với các đối tượng nhỏ hơn.
3. **Thông tin ngữ cảnh**: Phương pháp hai giai đoạn của RCNN có khả năng nắm bắt nhiều thông tin ngữ cảnh hơn, giúp xử lý các điểm khớp bị che khuất hoặc không rõ ràng.

## 4.3 Ý nghĩa thực tiễn

Từ góc độ thực tiễn, cải thiện tốc độ gấp 18 lần do YOLOv11-Pose cung cấp có thể sẽ quan trọng hơn so với độ chính xác thấp hơn một chút trong hầu hết các ứng dụng thực tế. Lợi thế về tốc độ này cho phép:

- Theo dõi tư thế thời gian thực trong video
- Tích hợp với các thiết bị có tài nguyên hạn chế
- Xử lý khối lượng dữ liệu lớn hơn
- Chi phí tính toán thấp hơn cho việc triển khai