ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

GIẢI PHÁP ỨNG DỤNG AI TÓM TẮT CÁC TRẬN ĐẤU THỂ THAO TENNIS VÀ CẦU LÔNG

BÁO CÁO HỌC PHẦN
THỰC TẬP DOANH NGHIỆP

Ngành : Trí Tuệ Nhân Tạo

Sinh viên: Nguyễn Văn Thân - 22022596

Người hướng dẫn: Bùi Văn Sơn

HÀ NỘI, 2025

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

GIẢI PHÁP ỨNG DỤNG AI TÓM TẮT CÁC TRẬN ĐẤU THỂ THAO TENNIS VÀ CẦU LÔNG

BÁO CÁO HỌC PHẦN
THỰC TẬP DOANH NGHIỆP

Ngành : Trí Tuệ Nhân Tạo

Sinh viên: **Nguyễn Văn Thân - 22022596**

Người hướng dẫn: **Bùi Văn Sơn**

HÀ NỘI, 2025

Lời cảm ơn

Trong suốt quá trình thực tập tại Công ty phát triển phần mềm VNPT-Media (VNPT-Media Software), tôi đã nhận được sự hỗ trợ nhiệt tình từ ban lãnh đạo, anh chị trong nhóm Computer Vision, đặc biệt là anh Bùi Văn Sơn - người đã trực tiếp hướng dẫn tôi. Nhờ sự chỉ bảo tận tình, tôi đã có cơ hội tiếp cận một dự án thực tế, mở rộng kiến thức và rèn luyện kỹ năng nghiên cứu, lập trình và kỹ năng làm việc nhóm trong lĩnh vực Trí tuệ nhân tạo (AI).

Tôi cũng xin gửi lời cảm ơn đến các thầy cô giảng dạy và nghiên cứu trong Viện Trí Tuệ Nhân Tạo, Trường Đại học Công nghệ, ĐHQGHN, đã trang bị cho tôi nền tảng kiến thức vững chắc để tôi có thể áp dụng trong quá trình thực tập.

Mặc dù đã rất cố gắng, nhưng báo cáo khó có thể tránh khỏi những thiếu sót. Kính mong nhận được sự góp ý của thầy cô và anh chị để tôi có thể hoàn thiện hơn trong học tập và công việc sau này.

Tóm tắt

Báo cáo trình bày dự án phát triển một ứng dụng AI dùng Computer Vision nhằm tự động tóm tắt các trận đấu tennis và cầu lông từ video. Mục tiêu dài hạn là xây dựng pipeline gồm: phát hiện sân, theo dõi người chơi, phát hiện keypoint/tư thế, nhận diện hành vi quan trọng và sinh bản tóm tắt highlight.

Tại thời điểm báo cáo này, nhóm đã hoàn thành các bước tiền đề quan trọng: phát hiện sân, gán nhãn dữ liệu, phát triển mô-đun keypoint detection theo hướng regression và xây dựng thành công mô hình phát hiện và trích xuất nội dung bảng điểm. Mô-đun keypoint của nhóm được triển khai dưới dạng bài toán hồi quy trực tiếp trên tọa độ keypoint, thử nghiệm với nhiều backbone khác nhau (ví dụ ResNet, EfficientNet và các biến thể nhẹ hơn dùng cho inference thời gian thực). Việc gán nhãn tuân theo định dạng keypoint tùy chỉnh (tương tự COCO keypoint) để hỗ trợ huấn luyện.

Về mặt kỹ thuật, các điểm chính thực hiện gồm:

- Mô hình: Regression-based keypoint heads gắn trên backbone trích xuất đặc trưng (ResNet/EfficientNet/...); hàm mất mát chính là SmoothL1 trên tọa độ keypoint với trọng số cho các keypoint mất mát khác nhau khi cần.
- Tiền xử lý và augmentation: chuẩn hóa kích thước khung, crop theo vùng sân/ người chơi, xoay/flip, thay đổi ánh sáng để tăng tính kháng nhiễu.
- Huấn luyện: chia train/val, sử dụng early stopping và checkpointing; điều chỉnh tốc độ học, weight decay và cỡ batch để tối ưu hội tụ.
- Đánh giá: dùng hai chỉ số chính:
  - PCK (Percentage of Correct Keypoints): tỉ lệ keypoint dự đoán nằm trong bán kính cho phép so với ground truth (bán kính có thể chuẩn hóa theo kích thước người/khung). PCK phản ánh độ chính xác vị trí ở mức ngưỡng cụ thể.
  - OKS (Object Keypoint Similarity): điểm tương đồng giữa bộ keypoint dự đoán và ground truth, chuẩn hóa theo kích thước đối tượng và độ phân bố sai số cho từng keypoint; OKS cho phép so sánh tổng hợp chất lượng dự đoán keypoint.

Báo cáo mô tả chi tiết quy trình gán nhãn, cấu hình mô hình thử nghiệm (danh sách backbone, learning rate schedule, loss), cũng như kết quả đánh giá sơ bộ bằng PCK và OKS trên tập validation. Những thách thức hiện gặp phải bao gồm: che khuất người chơi,

góc quay và phối cảnh khác nhau, khung hình có độ phân giải thấp, và cân bằng nhãn cho các keypoint ít xuất hiện.

Các bước tiếp theo đề ra trong phần còn lại của dự án gồm: tích hợp module tracking để duy trì ID người chơi giữa các pha và hỗ trợ bình luận theo từng người, phát triển mô-đun action recognition / event detection dựa trên chuỗi khung hình , xây dựng thuật toán segmentation cho từng pha (rally/point) và gán saliency score để chọn highlight, và tối ưu hóa mô hình keypoint cho inference thời gian thực (pruning/quantization hoặc dùng kiến trúc nhẹ).

Việc hoàn thiện phần phát hiện sân, gán nhãn, mô-đun phát hiện cấu trúc sân, cùng với phát hiện và trích xuất thông tin bảng điểm tạo nền tảng vững chắc cho các bước cao hơn trong pipeline tóm tắt trận đấu. Báo cáo cũng đề xuất tăng cường dữ liệu (data augmentation nâng cao, thu thập thêm mẫu đa dạng), và thử nghiệm ensemble/backbone selection nhằm cải thiện PCK và OKS trên các điều kiện thực tế đa dạng.

Từ khóa: computer vision, keypoint detection, regression-based keypoint, PCK, OKS, phát hiện sân, gán nhãn dữ liệu, tennis, cầu lông.

Hà Nội, tháng 08 năm 2025

Mục lục

Chương 1

Giới thiệu

1.1 Đề tài thực tập

Trong thời gian từ ngày 07/07/2025 đến nay, tôi thực tập tại nhóm Computer Vision của công ty VNPT Media, dưới sự hướng dẫn trực tiếp của anh Bùi Văn Sơn và anh Trịnh Văn Hậu. Đề tài thực tập là Ứng dụng Thị giác máy tính trong phát hiện highlight thể thao (cầu lông, tennis).

Mục tiêu của đề tài là xây dựng hệ thống phân tích video thể thao, trong đó video đầu vào sẽ được chia thành các rally (đoạn bóng qua lại giữa các vận động viên), sau đó phát hiện các tình huống nổi bật (highlight) như ghi điểm, đập mạnh, hoặc đánh qua lại kéo dài. Đây là bài toán phức tạp, đòi hỏi sự kết hợp của nhiều tác vụ con trong lĩnh vực Thị giác máy, nhằm tự động hóa việc trích xuất các khoảnh khắc hấp dẫn từ video trận đấu dài, giúp người xem tiết kiệm thời gian và tăng trải nghiệm xem thể thao. Các tác vụ chính bao gồm:

- Phát hiện cấu trúc sân: Nhận diện cấu trúc sân thi đấu với 14 keypoints cho tennis và 22 keypoints cho cầu lông, giúp chuẩn hóa góc nhìn và hỗ trợ theo dõi đối tượng.
- Phát hiện bảng điểm: Phát hiện bảng điểm trên màn hình và trích xuất nội dung bằng công nghệ OCR (Optical Character Recognition), để theo dõi sự thay đổi điểm số thời gian thực.
- Theo dõi đối tượng: Theo dõi quỹ đạo của bóng/cầu và người chơi, sử dụng các thuật toán như ByteTrack hoặc StrongSORT để duy trì ID đối tượng qua các frame.
- Phân đoạn sự kiện: Phân đoạn video để xác định thời điểm bắt đầu và kết thúc từng rally, dựa trên các sự kiện như phát bóng hoặc ghi điểm.

- Highlight Generation: Kết hợp các tiêu chí như độ dài rally, tốc độ smash, hoặc ghi điểm quan trọng để tạo video highlight ngắn gọn.

1.2 Kế hoạch dự án

Dự án được chia thành 4 giai đoạn chính, mỗi giai đoạn bao gồm các tác vụ nhỏ cụ thể, với mục tiêu rõ ràng và đầu ra đo lường được. Kế hoạch này giúp đảm bảo tiến độ và chất lượng dự án, đồng thời cho phép điều chỉnh linh hoạt dựa trên kết quả từng giai đoạn. Các giai đoạn bao gồm:

- **Giai đoạn 1: Nghiên cứu lý thuyết và khảo sát** (07/07-20/07/2025)
  - Xác định các dạng highlight phổ biến trong tennis và cầu lông, chẳng hạn như smash quyết định, rally dài hơn 20 lần chạm, hoặc ghi điểm ngoạn mục.
  - Khảo sát các nghiên cứu liên quan, bao gồm các mô hình như YOLO cho phát hiện đối tượng, SORT cho theo dõi, và Pose Estimation cho phân tích tư thế người chơi.
- **Giai đoạn 2: Thu thập và gán nhãn dữ liệu** (21/07–04/08/2025)
  - Thu thập tối thiểu 30–50 trận tennis và 30–50 trận cầu lông chất lượng cao từ các nguồn mở như YouTube hoặc kho dữ liệu thể thao.
  - Gán nhãn đối tượng: bóng, cầu, người chơi, bảng điểm, và sân thi đấu, sử dụng công cụ như Roboflow để đảm bảo độ chính xác cao.
  - Chuẩn hóa dữ liệu theo định dạng YOLO hoặc COCO, hỗ trợ huấn luyện các mô hình deep learning.
  - Đầu ra: Dataset khoảng 1500 ảnh mỗi môn, được chia thành train/val/test theo tỷ lệ 70/15/15, sẵn sàng cho huấn luyện.
- **Giai đoạn 3: Tracking object và xác định ghi điểm** (05/08–03/10/2025)
  - Huấn luyện YOLOv8 kết hợp với ByteTrack hoặc StrongSORT để theo dõi bóng và cầu, tập trung vào độ chính xác trong môi trường động.
  - Phát triển mô hình phát hiện sân và bảng điểm, kết hợp OCR để đọc điểm số tự động.

- Nghiên cứu kỹ thuật cắt video trực tuyến sử dụng FFmpeg cho xử lý file video, GStreamer cho streaming, và buffer để quản lý dữ liệu thời gian thực.
- Đầu ra: Mô hình tracking đạt FPS > 10 và IoU > 0.5; script Python xác định thời điểm ghi điểm và cắt clip highlight tự động.

- **Giai đoạn 4: Tạo highlight và đánh giá** (04/10–30/12/2025)
  - Sinh video highlight từ các rally dựa trên tiêu chí như độ dài, smash, hoặc ghi điểm đẹp, sử dụng các công cụ chỉnh sửa video tự động.
  - Đánh giá chất lượng highlight thông qua feedback từ mentor và so sánh với highlight thủ công từ các kênh thể thao chuyên nghiệp.
  - Hoàn thiện hệ thống, tinh chỉnh mô hình dựa trên đánh giá, bàn giao sản phẩm đầy đủ, và viết báo cáo tổng kết chi tiết.

1.3 Phạm vi công việc của tôi

Trong thời gian thực tập, tôi tập trung vào các công việc cụ thể để hỗ trợ dự án tổng thể, bao gồm:

1. Thực hiện nghiên cứu trong giai đoạn 1: nghiên cứu, tìm hiểu bài toán, tổng hợp các giải pháp khả thi từ các tài liệu và nghiên cứu liên quan, lựa chọn các giải pháp phù hợp với dự án.

2. Tham gia thu thập và gán nhãn dữ liệu trong giai đoạn 2: đảm bảo dữ liệu đa dạng và chất lượng cao để huấn luyện mô hình, ưu tiên dữ liệu mới và các trận đấu hay.

3. Triển khai và huấn luyện các mô hình phát hiện sân, phát hiện và trích xuất nội dung bảng điểm trong giai đoạn 3, với trọng tâm vào việc tối ưu hóa hiệu suất và tích hợp vào hệ thống tổng thể.

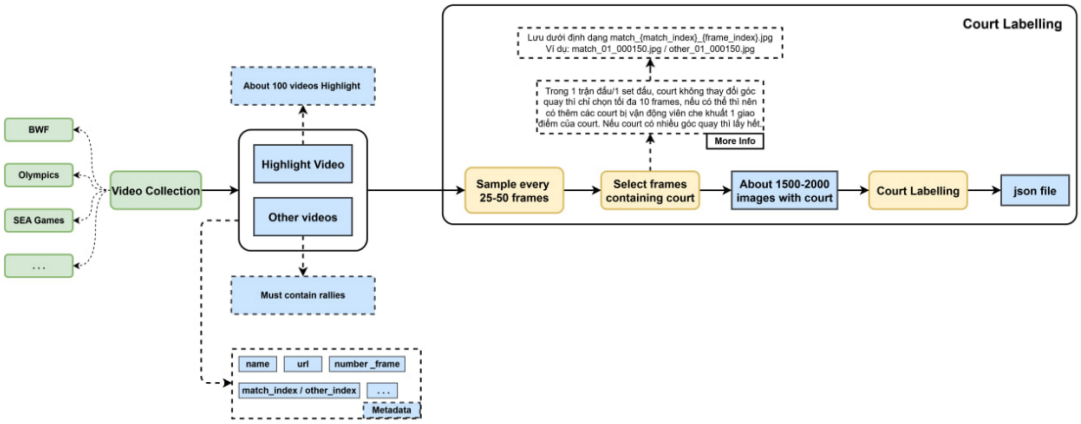
Chương 2

Triển khai dự án

Trong chương này, tôi trình bày chi tiết các công việc chính mà tôi đã tham gia trong dự án phát hiện highlight thể thao tại VNPT Media, bao gồm: thu thập và gán nhãn dữ liệu, xây dựng và huấn luyện mô hình phát hiện cấu trúc sân, nghiên cứu triển khai mô hình phát hiện và trích xuất thông tin bảng điểm.

2.1 Thu thập và gán nhãn dữ liệu

Mục tiêu của phần này là xây dựng một bộ dữ liệu chất lượng cao bằng cách gán nhãn các keypoints trên hình ảnh chứa sân cầu lông và sân tennis, phục vụ trực tiếp cho bài toán phát hiện cấu trúc sân và hỗ trợ các giai đoạn sau như theo dõi và phát hiện đối tượng.



Hình 2.1: Minh họa quá trình thu thập và gán nhãn dữ liệu

2.1.1 Nguồn dữ liệu

Dữ liệu bao gồm các hình ảnh có độ phân giải HD-720p (1280x720) hoặc Full HD-1080p (1920x1080), được chọn lọc từ các video trận đấu cầu lông và tennis (bao gồm video full trận, highlight, hoặc tổng hợp), đảm bảo mỗi hình ảnh phải chứa đầy đủ hoặc một phần sân đấu để tăng tính đa dạng. Nhóm đã thu thập dữ liệu từ nhiều nguồn khác nhau:

- Các giải đấu tennis quốc tế lớn như ATP Tour, Wimbledon, US Open, từ các kênh YouTube chính thức hoặc nền tảng chia sẻ video chuyên về thể thao.
- Các trận cầu lông quốc tế từ BWF (Liên đoàn Cầu lông Thế giới), SEA Games, Thomas Cup, với trọng tâm vào video chất lượng cao để tránh nhiễu.
- Video highlight có sẵn từ các kênh như ESPN hoặc YouTube Sports, dùng để tham khảo tiêu chí highlight thực tế và bổ sung dữ liệu đa dạng.

Mục tiêu dữ liệu:

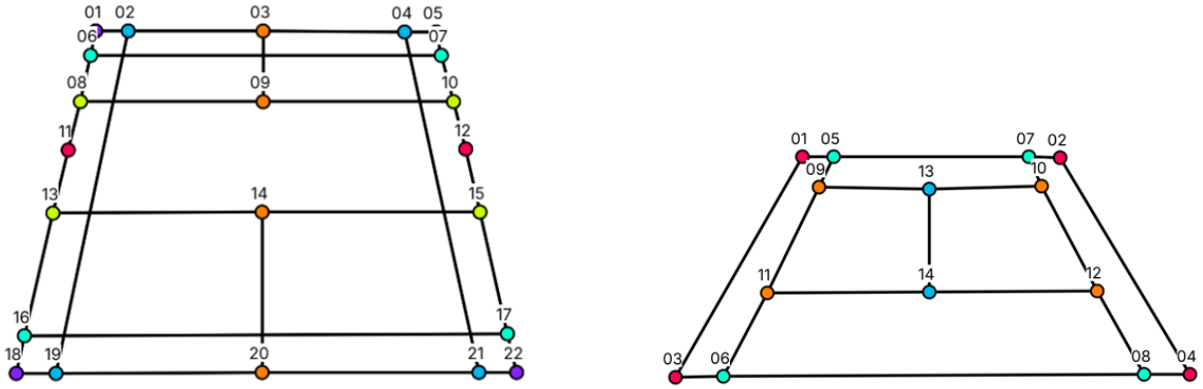
- Cầu lông: 1500 hình ảnh, đa dạng góc sân, màu sắc sân, giải đấu, chất lượng, và các trường hợp sân bị che khuất bởi người chơi.
- Tennis: 1200 hình ảnh với tiêu chí tương tự.
- Bảng điểm: 2000 hình ảnh với sự đa dạng về màu sắc, giải đấu, thiết kế, người chơi.

Gán nhãn dữ liệu:

1. Một đối tượng sân cầu lông có 22 keypoints.
2. Một đối tượng sân tennis có 14 keypoints.
3. Một đối tượng bảng điểm gồm 1 bounding box.

2.1.2 Phân chia dữ liệu

Để đảm bảo tính đa dạng và giảm thiểu overfitting, dữ liệu được phân chia thành các bộ train, validation, và test sử dụng thư viện FAISS (Facebook AI Similarity Search) – một thư viện mã nguồn mở chuyên về tìm kiếm tương đồng vector, được phát triển bởi Facebook AI Research. FAISS hỗ trợ các thuật toán như k-means clustering và approximate nearest neighbor search, giúp phân cụm dữ liệu hình ảnh dựa trên đặc trưng vector một cách hiệu quả, đặc biệt với dataset lớn. Phương pháp này vượt trội hơn



(a) Sân cầu lông với 22 keypoints (b) Sân tennis với 14 keypoints

Hình 2.2: Cấu trúc sân cầu lông và sân tennis

phân chia ngẫu nhiên vì nó đảm bảo tập test chứa các mẫu "khó"hoặc outlier (ít tương đồng với train), từ đó tăng tính tổng quát hóa của mô hình trên dữ liệu thực tế.
Quy trình phân chia chi tiết như sau:

- Trước hết, trích xuất đặc trưng từ các hình ảnh bằng mô hình CNN pretrained như ResNet18 từ thư viện PyTorch, đây là một framework deep learning phổ biến hỗ trợ huấn luyện mô hình trên GPU, tạo ra embeddings vector cho từng ảnh (kích thước thường là 512 hoặc 1024 chiều).

- Sau đó, áp dụng FAISS để thực hiện phân cụm k-means với số lượng cụm là 150. Số cụm này được chọn dựa trên kích thước dataset, nhằm đảm bảo mỗi cụm đại diện cho một nhóm hình ảnh tương đồng về đặc trưng như góc quay, màu sắc sân, hoặc điều kiện ánh sáng (ví dụ: sân trong nhà so với ngoài trời).

- Đối với **tennis** (1316 ảnh): Sau phân cụm, chọn 50 cụm chứa ít ảnh nhất (đại diện cho các trường hợp hiếm gặp hoặc outlier, như góc quay lạ hoặc sân bị che khuất nhiều), và từ mỗi cụm lọc tay lấy 2-3 ảnh chất lượng tốt nhất (kiểm tra thủ công để tránh ảnh lỗi hoặc nhiễu), tạo thành tập test với khoảng 125 ảnh (trung bình 2.5 ảnh/cụm). 100 cụm còn lại được sử dụng cho train và validation, với tỷ lệ phân chia 70% train và 15% validation (tổng tỷ lệ train/val/test 70/15/15). Quy trình này giúp mô hình học từ dữ liệu đa dạng và được đánh giá trên các mẫu độc lập, giảm bias.

- Đối với **cầu lông** (1508 ảnh): Tương tự, phân thành 150 cụm, chọn 50 cụm ít ảnh nhất, lọc tay để tạo tập test 125 ảnh. 100 cụm còn lại chia train/val theo tỷ lệ 70/15. Việc lọc tay đảm bảo tập test không chứa ảnh nhiễu hoặc gán nhãn kém,

tăng độ tin cậy của đánh giá.

Phương pháp sử dụng FAISS không chỉ giúp tránh rò rỉ dữ liệu giữa các tập mà còn cải thiện hiệu suất mô hình trên dữ liệu thực tế, vì tập test đại diện cho các tình huống đa dạng và khó khăn hơn, giống như môi trường triển khai thực tế.

2.2 Phát hiện sân

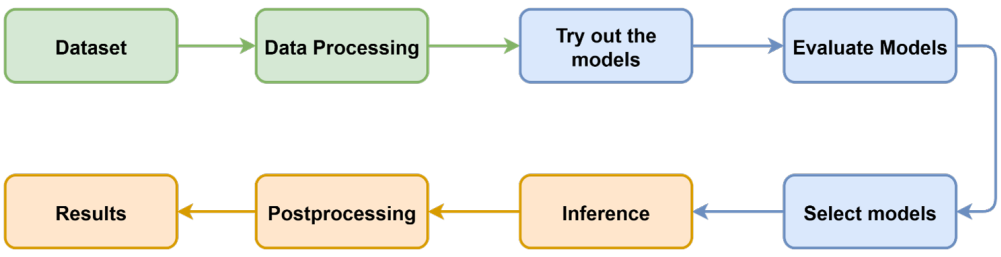
2.2.1 Mục tiêu

Court Detection là bước nền tảng để phân đoạn rally và xác định sự kiện ghi điểm. Mục tiêu là dự đoán chính xác các keypoints đặc trưng trên sân cầu lông và tennis, từ đó xây dựng phép chiếu hình học (homography) nhằm chuẩn hóa góc nhìn và phục vụ các bước xử lý sau.

2.2.2 Dữ liệu

- Sân cầu lông: 1508 ảnh, độ phân giải 1920×1080, mỗi ảnh gán nhãn với 22 keypoints.
- Sân tennis: 1316 ảnh, độ phân giải 1280×720, mỗi ảnh chứa 14 keypoints.
- Định dạng nhãn:
  - *Regression-based*: Biểu diễn tọa độ keypoints dưới dạng vector (28 chiều cho tennis, 44 chiều cho cầu lông).
  - *Heatmap-based*: Tạo heatmap cho từng keypoint với độ phân giải 64×64 hoặc 128×128.
  - *YOLO-Pose*: Bounding box bao quanh sân kết hợp với vector keypoints.

2.2.3 Các hướng tiếp cận



Hình 2.3: Pipeline giải quyết bài toán Court Detection

Hai nhóm phương pháp chính:

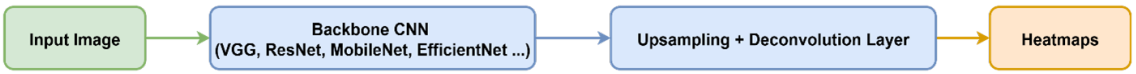
- Two-stage (Top-down): Sử dụng mô hình phát hiện đối tượng (YOLOv8, Faster R-CNN) để khoanh vùng bounding box của sân, sau đó crop ảnh và dùng keypoint estimation để dự đoán tọa độ các điểm mốc.
- Single-stage (Bottom-up): Dự đoán toàn bộ keypoints trực tiếp trên ảnh.

Hai hướng tiếp cận mô hình:

- Regression-based: Dự đoán trực tiếp tọa độ (x,y) cho từng keypoint, trả về vector kích thước 2K (hoặc 3K nếu có confidence/visibility).
  - YOLO-Pose: Dự đoán bounding box + keypoints trực tiếp.
  - Backbone CNN + Fully Connected: Sử dụng backbone CNN kết hợp lớp fully connected.

Hình 2.4: Cấu trúc mô hình regression-based dựa trên backbone CNN và FC

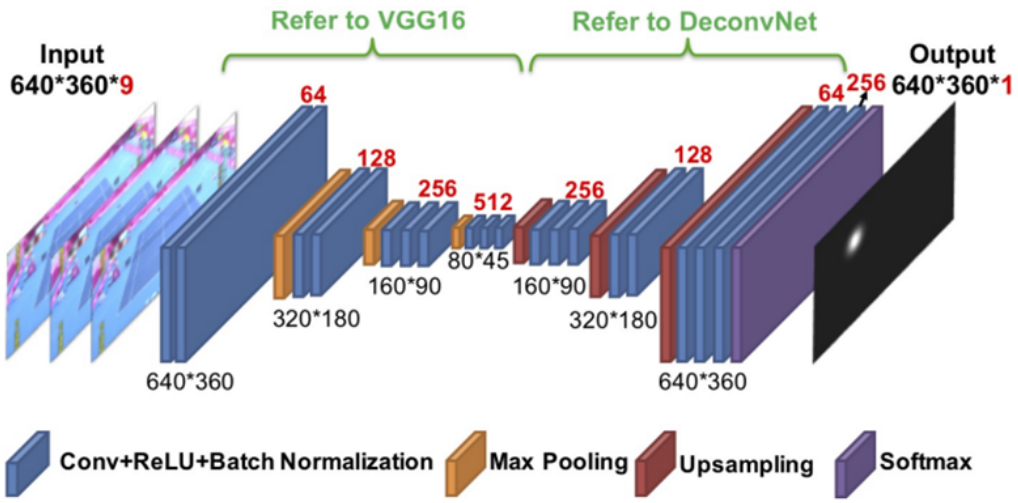
- Heatmap-based: Dự đoán heatmap cho mỗi keypoint, giá trị pixel thể hiện xác suất vị trí keypoint.



Hình 2.5: Cấu trúc mô hình heatmap-based

Thử nghiệm các mô hình

- Thay đổi backbone CNN trong mô hình regression-based: ResNet (18, 34, 50), MobileNet (v2, v3-small), EfficientNet (b0, b1), VGG (16, 19).
- Thử nghiệm đầu ra regression-based với 3K (x, y, confidence).
- Thử nghiệm mô hình YOLO-Pose: Ưu tiên các phiên bản nhanh như YOLOv8-pose và YOLOv11n-pose, tập trung vào tốc độ FPS cao cho video thời gian thực.
- Thay đổi backbone và cấu trúc trong mô hình heatmap-based.



Hình 2.6: Minh họa mô hình heatmap-based với đầu ra là K heatmap cho K keypoint

2.2.4 Hậu xử lý Keypoints với Hough Line Transform

Để cải thiện độ chính xác của các keypoints dự đoán, một bước hậu xử lý (refinement) được áp dụng sau khi mô hình dự đoán tọa độ keypoints. Phương pháp này sử dụng Hough Line Transform từ thư viện OpenCV để tinh chỉnh vị trí keypoints dựa trên các đặc trưng hình học của sân thi đấu.

Quy trình hậu xử lý bao gồm các bước sau:

1. **Cắt vùng lân cận**: Đối với mỗi keypoint dự đoán, cắt một vùng hình vuông kích thước 40x40 pixel xung quanh vị trí tọa độ (x,y) của keypoint. Kích thước này được chọn để bao quát đủ đặc trưng cục bộ (như đường vạch trên sân) mà không bao gồm quá nhiều nhiễu từ các khu vực khác.

2. Phát hiện đường thẳng: Áp dụng Hough Line Transform (cụ thể là `cv2.HoughLinesP`) trên vùng đã cắt để phát hiện các đường thẳng đại diện cho các vạch kẻ trên sân. Thuật toán Hough Line Transform chuyển đổi không gian hình ảnh sang không gian Hough, cho phép xác định các đường thẳng dựa trên các điểm có cường độ pixel cao (thường là các vạch trắng hoặc sáng trên sân).

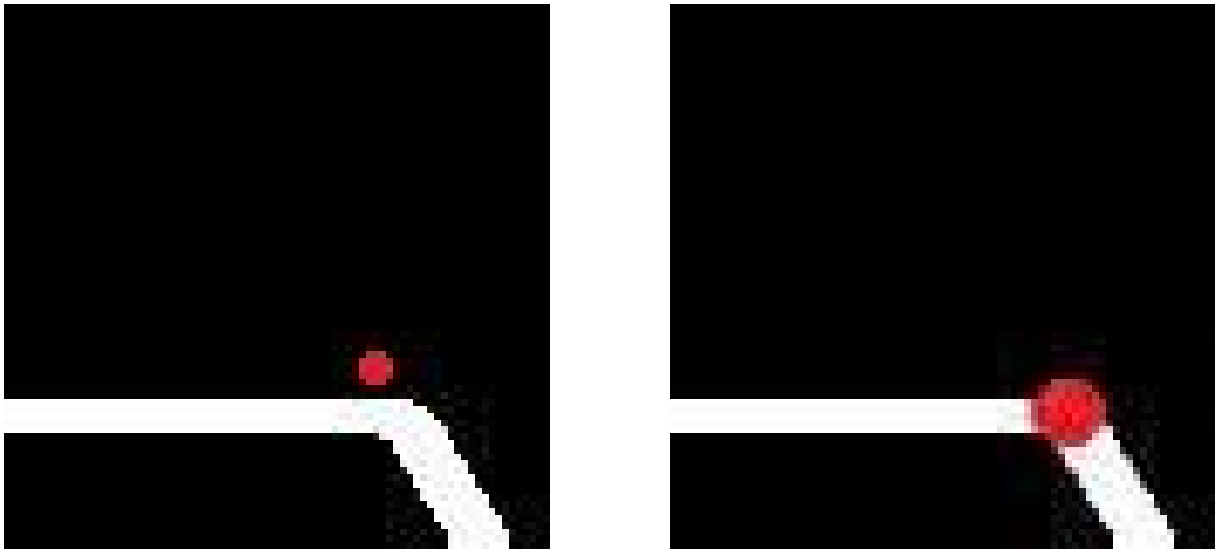
3. Tính giao điểm: Xác định giao điểm của hai đường thẳng gần nhất với vị trí keypoint ban đầu. Giao điểm này được tính bằng cách giải hệ phương trình tuyến tính của các đường thẳng:

$$y = m_1 x + c_1 \quad \text{và} \quad y = m_2 x + c_2$$

Trong đó m_1, m_2 là độ dốc và c_1, c_2 là tung độ gốc của hai đường thẳng. Giao điểm

$(x_{\text{refined}}, y_{\text{refined}})$ được sử dụng làm vị trí keypoint đã tinh chỉnh.

4. Cập nhật keypoints: Thay thế tọa độ keypoint ban đầu bằng tọa độ giao điểm mới, giúp cải thiện độ chính xác, đặc biệt trong các trường hợp keypoint bị lệch do nhiễu, góc quay camera, hoặc che khuất một phần.



(a) Trước tinh chỉnh (b) Sau tinh chỉnh

Hình 2.7: Minh họa hậu xử lý refine keypoints bằng Hough Line Transform

Phương pháp này đặc biệt hiệu quả trong các trường hợp sân có các vạch kẻ rõ ràng, như sân tennis hoặc cầu lông, vì Hough Line Transform tận dụng được các đặc trưng hình học mạnh mẽ của sân.

2.2.5 Chỉ số đánh giá

Để đánh giá mô hình phát hiện cấu trúc sân, nhóm sử dụng ba chỉ số: PCK, OKS, và RMSE kết hợp với inference time để cân bằng giữa độ chính xác và hiệu suất của mô hình.

PCK (Percentage of Correct Keypoints)

PCK đo tỷ lệ keypoint nằm trong bán kính cho phép của ground truth:

$$PCK(\alpha) = \frac{\text{True Keypoints}}{\text{Total Keypoints}} \times 100\%, \quad \text{với True Keypoint nếu } \frac{\|\mathbf{p}_{\text{pred}} - \mathbf{p}_{\text{gt}}\|_2}{L_{\text{norm}}} \leq \alpha$$

Trong đó:

- L_{norm}: Khoảng cách chéo giữa keypoints góc trên trái và dưới phải của sân ground truth.
- $\alpha = 0.02$, đảm bảo sai số Euclidean trong khoảng 20-30 pixel.

OKS (Object Keypoint Similarity)

OKS đánh giá mức độ tương đồng giữa tập keypoints dự đoán và ground truth:

$$\text{OKS} = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2\kappa_i^2}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}$$

Trong đó:

- d_i: Khoảng cách Euclidean giữa keypoint dự đoán và ground truth thứ i.
- $s = \sqrt{0.53 \cdot \text{area}}$: Tham số chuẩn hóa, với area là diện tích bounding box của sân.
- κ_i: Hằng số scale, phản ánh độ khó khi gán nhãn. Đối với sân tennis (14 keypoints):

$$[0.075, 0.075, 0.060, 0.060, 0.0725, 0.0725, 0.0575, 0.0575, 0.07, 0.07, 0.055, 0.055, 0.055, 0.04]$$

- v_i: Cờ hiển thị keypoint (1 nếu tồn tại, 0 nếu không).
- $\delta(v_i > 0)$: Hàm chỉ báo.

RMSE (Root Mean Squared Error)

Bao gồm RMSE trung bình trên tất cả keypoints, trên tất cả ảnh, và chuẩn hóa với độ dài đường chéo sân.

Inference Time

Tốc độ dự đoán (FPS) được xem xét để đảm bảo mô hình phù hợp cho ứng dụng thời gian thực.

2.3 Phát hiện bảng điểm và trích xuất thông tin

2.3.1 Mục tiêu

Mục tiêu của mô-đun bảng điểm là:

- Tự động phát hiện vị trí bảng điểm xuất hiện trong khung hình video (bounding box).
- Trích xuất văn bản (OCR) từ vùng bảng điểm và chuyển thành thông tin có cấu trúc: set hiện tại, tên vận động viên, điểm số.
- Cung cấp dữ liệu thời gian thực cho module tracking và event detection để xác định thời điểm ghi điểm.

2.3.2 Dữ liệu và gán nhãn

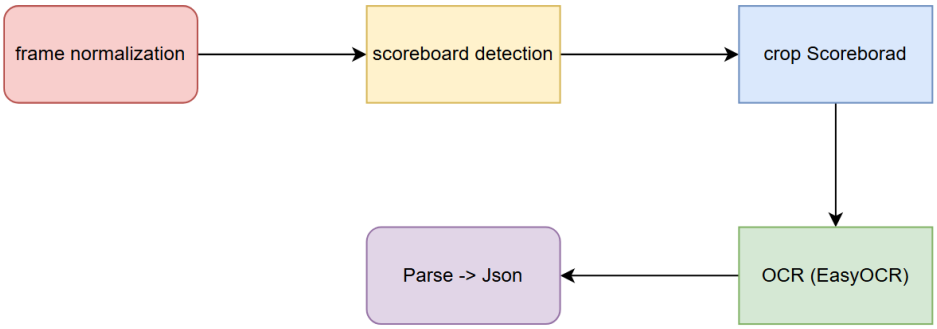
- Dữ liệu huấn luyện: ảnh chụp từ các khung video có hiển thị bảng điểm. Mỗi ảnh được gán nhãn bounding box cho bảng điểm theo định dạng YOLO.
- Annotation: bounding box chứa bảng điểm.
- Tỷ lệ train/val/test: 70/15/15

2.3.3 Kiến trúc và cấu hình kỹ thuật

Cho detection, nhóm sử dụng YOLO (ví dụ `yolov11n`) do cân bằng giữa tốc độ và độ chính xác. Các thông số chính thử nghiệm:

- Kích thước ảnh huấn luyện: 640.
- Batch size: tùy GPU (khuyến nghị batch=8 hoặc nhỏ hơn; inference chạy per-image khi VRAM hạn chế).
- Tiền xử lý crop trước OCR: tăng tương phản, threshold adaptive, resize up-sampling cho text nhỏ.
- OCR engine: `EasyOCR` .

2.3.4 Pipeline xử lý (đồ họa)



Hình 2.8: Pipeline phát hiện bảng điểm và trích xuất thông tin.

2.3.5 Hậu xử lý và rules để lấy thông tin cấu trúc

- Dò pattern numeric: `\d{1,2}[-:/]\d{1,2}` để trích score.
- Mỗi bảng điểm có 2 dòng, mỗi dòng thể hiện tên vận động viên, sét đấu, và điểm số, quá trình trích xuất thông tin trên bảng điểm, chia bảng điểm làm 2 phần theo chiều ngang để dễ lấy thông tin.

2.3.6 Chỉ số đánh giá cho module bảng điểm

Bên cạnh metrics detection chuẩn (mAP/IoU), cần thêm metrics cho OCR/trích xuất:

- Detection: AP@0.5 (Average Precision), Recall.
- OCR: Character Accuracy (CHAR_ACC), Word Accuracy (WORD_ACC).
- Structured extraction accuracy:
  - *Score Extraction Accuracy*: tỉ lệ crops mà score (ví dụ `6-3`) được trích chính xác.
  - *Set Detection Accuracy*: tỉ lệ set number trích đúng.
  - *Player Name Match Rate*: tỉ lệ trích đúng tên (hoặc đúng với fuzzy threshold).
- Latency: thời gian inference + OCR trung bình (ms/frame).

Chương 3

Thực nghiệm và kết quả

3.1 Thiết lập thực nghiệm

Trong giai đoạn 2, nhóm thu thập và gán nhãn dữ liệu:

- Cầu lông: 1508 ảnh, gán nhãn 22 keypoints.
- Tennis: 1316 ảnh, gán nhãn 14 keypoints.

Mô hình được huấn luyện với:

- Batch size: 16–32.
- Learning rate: 2×10^{-4}.
- Epoch: 120 epochs cho mỗi mô hình.
- Tỷ lệ train/val/test: 70/15/15.

Chỉ số đánh giá: PCK, OKS (xem Mục 2.2.5), RMSE.

3.2 Tối ưu hóa siêu tham số với Optuna

Để tối ưu hóa mô hình phát hiện cấu trúc sân, nhóm sử dụng framework Optuna, tích hợp với PyTorch để tìm kiếm siêu tham số hiệu quả.

3.2.1 Không gian tìm kiếm

Không gian siêu tham số được định nghĩa rộng để bao quát các yếu tố ảnh hưởng đến hiệu suất:

- Learning rate: Phạm vi $[1 \times 10^{-5}, 5 \times 10^{-3}]$ với phân phối log-uniform để ưu tiên giá trị nhỏ.
- Batch size: Các giá trị rời rạc $\{16, 32, 64\}$ để cân bằng giữa tốc độ huấn luyện và ổn định gradient.
- **Image size**: $\{256, 384, 512, 640\}$ để tối ưu giữa độ phân giải chi tiết và thời gian xử lý.
- Weight decay: $[1 \times 10^{-6}, 1 \times 10^{-3}]$ cho regularization, tránh overfitting.
- Dropout rate: $[0.1, 0.4]$ để ngẫu nhiên loại bỏ neuron, tăng tính tổng quát.

3.2.2 Thuật toán tối ưu hóa

- Sampler: TPESampler với 10 startup trials để khởi tạo không gian tìm kiếm ban đầu.
- Pruner: MedianPruner để dừng sớm các trial kém hiệu quả dựa trên metric trung vị của các trial trước.
- Objective: Tối đa hóa OKS trên tập validation, với direction là maximize.

3.2.3 Kết quả tối ưu hóa

Sau 100 trials (mất khoảng 20 giờ tính toán), bộ siêu tham số tốt nhất được tìm thấy, cải thiện đáng kể so với baseline:

- Batch size: 32
- Image size: 512×512 pixels
- Learning rate: 2×10^{-4}
- Weight decay: 1×10^{-4}
- Dropout: 0.25

3.3 Kết quả mô hình phát hiện cấu trúc sân

Bảng 3.1 thể hiện kết quả huấn luyện các mô hình CNN:

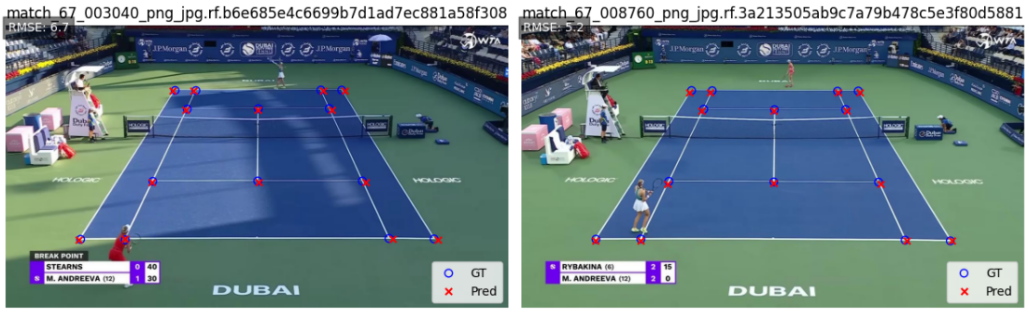
Bảng 3.1: Kết quả huấn luyện cấu trúc sân với các kiến trúc CNN

Model	Epochs	PCK	OKS	RMSE	Inference Time (s)
ResNet18	120	0.837	0.882	11.707	0.0130
ResNet34	120	0.851	0.895	10.758	0.0123
ResNet50	120	0.884	0.907	9.888	0.0111
EfficientNet-B0	120	0.634	0.786	17.566	0.0200
EfficientNet-B1	120	0.705	0.811	15.899	0.0185
MobileNetV2	120	0.795	0.850	13.370	0.0150
MobileNetV3-Small	120	0.680	0.802	16.388	0.0180
MobileNetV3-Large	120	0.706	0.817	15.230	0.0170
VGG16	120	0.822	0.879	11.200	0.0160
VGG19	120	0.888	0.918	8.790	0.0200

Nhận xét:

- Các kiến trúc ResNet và VGG vượt trội hơn EfficientNet và MobileNet nhờ khả năng trích xuất đặc trưng mạnh trên dữ liệu đa dạng về góc quay và điều kiện ánh sáng.
- VGG19 đạt PCK = 0.888 và OKS = 0.918 (120 epoch)
- EfficientNet và MobileNet có ưu điểm về tốc độ suy luận nhưng độ chính xác thấp hơn, phù hợp cho ứng dụng cần tốc độ cao.

Trực quan kết quả của VGG19 và Resnet50 với 120 epochs:



Hình 3.1: Kết quả trực của VGG19 trên tập test

Hình 3.2: Kết quả trực quan của Resnet50 trên tập test

3.3.1 Kết quả module phát hiện bảng điểm và trích xuất thông tin

Bảng 3.2: Kết quả cho module phát hiện bảng điểm và trích xuất thông tin.

Model pipeline	**AP@0.5**	**Score Acc.**	**Char Acc.**	**Latency (ms)**
yolov11n + EasyOCR	0.82	0.76	0.88	140
yolov8n + EasyOCR	0.78	0.72	0.86	110

Trong bảng trên:

- AP@0.5: Average Precision với IoU threshold = 0.5 cho việc phát hiện bounding-box bảng điểm.
- Score Acc: Tỉ lệ crops mà cặp điểm được trích chính xác (exact match).
- Char Acc: Độ chính xác ký tự của kết quả OCR (ứng với EasyOCR).
- Latency: Thời gian trung bình cho một frame (detection + crop + OCR), tính bằng milliseconds.

Hình 3.3: Ví dụ phát hiện bảng điểm

Kết quả thử nghiệm cho thấy detector đạt AP khá cao (trên 0.8 trong ví dụ), OCR có *character accuracy* tốt trên crops sạch (độ tương phản tốt, không nghiêng). Tuy nhiên, các trường hợp gây lỗi phổ biến gồm:

- Phông nền và font lẫn màu tương tự → cần tăng tương phản (CLAHE) và threshold adaptive.
- Thường xuyên trích xuất nhầm thông tin giữa điểm số và sét đấu do khoảng cách giữa chúng nhỏ.
- Văn bản nhỏ (font size nhỏ) → cần up-sampling crop trước OCR.
- Một số bảng điểm trong các giải đấu mới thường được trang trí gây khó khăn cho quá trình phát hiện bảng điểm.

3.4 Hướng cải tiến trong tương lai

Để nâng cao hiệu quả của hệ thống, một số hướng cải tiến có thể được xem xét:

- Sử dụng mô hình nhẹ hơn: Tối ưu hóa các mô hình như MobileNetV3 hoặc EfficientNet để giảm độ phức tạp tính toán, phù hợp cho triển khai trên thiết bị có tài nguyên hạn chế như camera thể thao thời gian thực.
- Data augmentation nâng cao: Áp dụng các kỹ thuật như random crop, rotation, hoặc synthetic data generation để tăng tính đa dạng của dataset.

- Tích hợp temporal information: Sử dụng mô hình dựa trên video như TimeSformer hoặc SlowFast để khai thác thông tin thời gian, giúp phân đoạn rally chính xác hơn trong các video dài.

- Bổ sung các bước tiền xử lý cho ảnh crop bảng điểm như *deskew*, *CLAHE* và *up-sampling* để cải thiện chất lượng đầu vào OCR, đồng thời áp dụng *temporal smoothing* (majority vote trên cửa sổ khung hình) nhằm ổn định kết quả trích xuất điểm số.

- Tự động hóa pipeline end-to-end: Xây dựng pipeline tích hợp từ phát hiện cấu trúc sân, phát hiện và trích xuất thông tin bảng điểm đến Highlight Generation, sử dụng các công cụ như TensorRT để tối ưu hóa tốc độ suy luận và triển khai trên môi trường sản xuất.

Chương 4

Kết luận

Báo cáo đã trình bày quá trình thực tập tại VNPT Media, tập trung vào phát triển hệ thống phát hiện highlight thể thao sử dụng Thị giác máy tính. Các kết quả chính bao gồm:

- Nghiên cứu các giải pháp, phương hướng cho bài toán.
- Xây dựng dataset chất lượng cao: 1508 ảnh cầu lông và 1316 ảnh tennis, gán nhãn 22 và 14 keypoints, 2300 ảnh các trận đấu cầu lông và tennis cho phát hiện bảng điểm.
- Huấn luyện mô hình phát hiện cấu trúc sân với nhiều mô hình để chọn ra mô hình tốt nhất, tiền đề cho phát triển sau này.
- Tích hợp module phát hiện bảng điểm và OCR giúp tự động hóa việc ghi nhận điểm số theo thời gian thực, làm nền tảng cho việc xác định sự kiện ghi điểm chính xác hơn.
- Tối ưu hóa siêu tham số với Optuna, cải thiện hiệu suất mô hình đáng kể.

4.1 Bài học từ quá trình thực tập

Qua quá trình thực tập, tôi đã học được nhiều bài học quý giá không chỉ về kỹ thuật mà còn về cách tiếp cận và quản lý một dự án thực tế:

- Cách tiếp cận dự án: Tầm quan trọng của việc lập kế hoạch chi tiết và chia nhỏ dự án thành các giai đoạn cụ thể, với mục tiêu rõ ràng và có thể đo lường được. Việc xác định các tác vụ con như phát hiện cấu trúc sân, theo dõi đối tượng, và phân

đoạn sự kiện ngay từ đầu giúp nhóm tập trung nguồn lực và điều chỉnh linh hoạt khi gặp khó khăn.

- Phân công công việc: quá trình làm việc trong nhóm Computer Vision tại VNPT Media giúp tôi hiểu cách phân chia công việc dựa trên thế mạnh và năng lực của từng thành viên, đảm bảo tiến độ dự án được duy trì.

- Kỹ năng giao tiếp và báo cáo: Việc thường xuyên thảo luận với người hướng dẫn và trình bày tiến độ dự án giúp tôi cải thiện kỹ năng giao tiếp kỹ thuật, trình bày ý tưởng rõ ràng và nhận phản hồi hiệu quả. Tôi học được văn hóa doanh nghiệp, hiểu rõ chức năng của các phòng ban, và kỹ năng làm việc nhóm.

- Tư duy giải quyết vấn đề: Đối mặt với các thách thức như dữ liệu nhiễu hoặc hiệu suất mô hình chưa tối ưu, tôi học được cách thử nghiệm nhiều phương pháp (ví dụ: regression-based vs. heatmap-based) và sử dụng các công cụ như Optuna để tìm giải pháp tốt nhất.

Tài liệu tham khảo

[1] Jaided AI. Easyocr: Ready-to-use ocr with 80+ supported languages, 2020. https://github.com/JaidedAI/EasyOCR.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. *arXiv preprint arXiv:1907.10902*, 2019.

[3] LearnOpenCV. Object keypoint similarity in keypoint detection, 2023.

[4] Banoth Thulasya Naik, Mohammad Farukh Hashmi, and Neeraj Dhanraj Bokde. A comprehensive review of computer vision in sports: Open issues, future trends and research directions. *Applied Sciences*, 12(9):4429, 2022.

[5] Roboflow. Roboflow - computer vision data platform, 2020. https://roboflow.com.

[6] Ultralytics. Ultralytics yolo: Real-time object detection and segmentation. 2023. https://github.com/ultralytics/ultralytics.

[7] Yijian Wu, Zewen Chen, Hongxing Zhang, Yulin Yang, and Weichao Yi. Enhanced pose estimation for badminton players via improved yolov8-pose with efficient local attention. *Sensors*, 25(14):4446, 2025.