스포츠 영상 탐지 시스템

**스포츠 플레이 3D로 표현**

* 시각 장애인의 경우 스포츠 경기를 실시간으로 즐기는 것에 어려움이 있다
* 플레이어와 공을 실시간으로 시청할 수 있도록 판넬로 표현한다
* 코치와 선수가 자신의 플레이를 실시간으로 분석할 수 있다
* **스포츠 종목 : 테니스**

**시스템 동작 방식**

* 실시간 영상 입력 수집
* YOLO 기반 객체 탐지 모델을 활용한 주요 이벤트 탐지
* 사용자가 직관적으로 경기 상황을 이해할 수 있도록 최적화된 정보 제공

**데이터셋**

* 플래이어 2명, 네트, 필드, 공
  + <https://universe.roboflow.com/sportsanalysis-3mrde/tennis-ball-and-court-detection-4qpqa/dataset/1>
* 테니스 동작 및 경기영상 데이터
  + <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=71612>

**모델 - YOLOv8**

* Yolov8은 object detection과 함께 이미지나 동영상의 image segmentation 또한 동일한 api를 이용해서 구현할 수 있다

[yolov8-comparison-plots.avif](https://prod-files-secure.s3.us-west-2.amazonaws.com/c09f8228-29c7-4dcb-8ca3-1de7d3988fab/ec4f3e43-b62e-4871-87fa-09aad8cd0ce3/yolov8-comparison-plots.avif)

* <https://docs.ultralytics.com/ko/models/yolov8/#overview>
* <https://m.blog.naver.com/beyondlegend/223050797442>

**모델 - YOLOv11**

* YOLO 시리즈의 최신 버전
* 실시간 객체 탐지 분야에서 정확도, 속도, 효율성 측면에서 발전을 이루었다
* <https://docs.ultralytics.com/ko/models/yolo11/>

**3D 모델링**

**MiDaS -** 단일 이미지에서 깊이를 추정하는 모델

* <https://github.com/isl-org/MiDaS>
* YOLOv8의 2D 디텍션 결과를 기반으로 **Z축(깊이) 정보**를 추가.
* 변환 행렬을 통해 top-down view로 시점 변경.
* Matplotlib 또는 Open3D로 시각화

**deepsort**

* 이전 프레임에 등장한 개체를 이용하여 다음 프레임의 개체의 위치를 예측하고 측정
  + <https://github.com/MuhammadMoinFaisal/YOLOv8-DeepSORT-Object-Tracking?tab=readme-ov-file>
  + <https://dohyeon.tistory.com/11>
* [https://pointer81.tistory.com/entry/YOLOv8을-이용한-객체-감지와-MP4-파일로-저장하기](https://pointer81.tistory.com/entry/YOLOv8%EC%9D%84-%EC%9D%B4%EC%9A%A9%ED%95%9C-%EA%B0%9D%EC%B2%B4-%EA%B0%90%EC%A7%80%EC%99%80-MP4-%ED%8C%8C%EC%9D%BC%EB%A1%9C-%EC%A0%80%EC%9E%A5%ED%95%98%EA%B8%B0)

모델 : Yolov8

<https://universe.roboflow.com/sportsanalysis-3mrde/tennis-ball-and-court-detection-4qpqa>

<https://universe.roboflow.com/scratchpad/tennis-ball-and-court-detection-fbmbi>

<https://universe.roboflow.com/testingdataset-tz9ou/tennis-ball-detection-v2-ih0f0-swjta>

<https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=71612>

<https://www.youtube.com/watch?v=QCG8QMhga9k>

<https://www.youtube.com/watch?v=aBVGKoNZQUw&list=LL&index=2>

[\*\*](https://www.linkedin.com/posts/yolovx_vision-analytics-for-sports-with-tactical-activity-7286626742151086080-N1Ej?utm_source=share&utm_medium=member_desktop)[https://www.linkedin.com/posts/yolovx\_vision-analytics-for-sports-with-tactical-activity-7286626742151086080-N1Ej?utm\_source=share&utm\_medium=member\_desktop\*\*](https://www.linkedin.com/posts/yolovx_vision-analytics-for-sports-with-tactical-activity-7286626742151086080-N1Ej?utm_source=share&utm_medium=member_desktop**)

* 다양한 실제 경기를 실시간으로 추적하는 것이 최종 목표
* 실제 경기 이미지가 있는 데이터셋을 활용하여 학습 진행
* 공만 학습했을 때의 결과를 확인한 후 데이터셋을 추가할지 삭제할지 결정한다
* opencv 3d 툴
* depth 모델이 정확하지 않을 때
* 정확하게 결과가 나오지 않았을 때
* 시각장애인이 경기를 즐기는 방법
* 공의 움직임이 시각장애인에게 중요할까

**목표**

* 장애를 가진 분들을 위해서의 목표가 가까워진다
* 이것을 가지고 데이터셋을 만들면 차별화를 얻을 수 있다
* 테니서 해설 데이터셋 예시를 몇개 만들 수 있다
* chat gpt api를 사용해서 멀티모달을 구현할 수 있다

**테니스**

<https://github.com/yastrebksv/TennisProject/tree/main?tab=readme-ov-file>

**영상 트래킹 코드**

# results = model(source='tennis.mp4', show=True, save=True, tracker="bytetrack.yaml", conf=0.3)

**테니스 공만 있는 데이터셋**

<https://universe.roboflow.com/testingdataset-tz9ou/tennis-ball-detection-v2-ih0f0-swjta> 3000장

<https://universe.roboflow.com/viren-dhanwani/tennis-ball-detection/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true> 570장

**ver1**

사용 모델 : YOLOv11

사용 데이터셋 :

<https://universe.roboflow.com/scratchpad/tennis-ball-and-court-detection-fbmbi(4.9K)>

**ver2**

사용 모델 : YOLOv11

사용 데이터셋 :

<https://universe.roboflow.com/sportsanalysis-3mrde/tennis-ball-and-court-detection-4qpqa(6.1K)>

**ver3**

사용 모델 : YOLOv8

<https://github.com/neowizard2018/neowizard/blob/master/DeepLearningProject/YOLOv8_Object_Detection_Roboflow_Aquarium_Data.ipynb>

[YOLOv8\_yes.ipynb](https://prod-files-secure.s3.us-west-2.amazonaws.com/c09f8228-29c7-4dcb-8ca3-1de7d3988fab/564000ea-c3bc-4c1e-9891-9cce9c5239b3/YOLOv8_yes.ipynb)

**ver4**

사용모델: YOLOv11

데이터 셋:

<https://universe.roboflow.com/viren-dhanwani/tennis-ball-detection/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true> 570장 중 100장(Train 80, Val: 15, Test 5)

<https://universe.roboflow.com/sportsanalysis-3mrde/tennis-ball-and-court-detection-4qpqa> 6090

**농구**

<https://github.com/brettfazio/CVBallTracking?form=MG0AV3>

<https://github.com/avishah3/AI-Basketball-Shot-Detection-Tracker?form=MG0AV3>

<https://github.com/Basket-Analytics/BasketTracking?form=MG0AV3>

* 데이터셋
  + 공만
    - <https://universe.roboflow.com/gswt/basketball-detection-1mtj3-4ad5o-cxnux>
    - <https://universe.roboflow.com/workspace-a3bxu/basketball-detection-unwvv>
  + 클래스 2개 이상
    - <https://universe.roboflow.com/sharangs/basketball-idlvv-drpzm>
      * **(이 아래 둘 중 하나를 사용하는 게 좋을 것 같습니다!)**
    - <https://universe.roboflow.com/cpen-355/score-keep-ml>
    - <https://universe.roboflow.com/cv-8scak/cv-cnfd4>

**농구 ver1**

사용 모델 : yolov8

사용 데이터셋 : 3번 <https://universe.roboflow.com/cv-8scak/cv-cnfd4>

사용모델: yolov11

사용데이터셋: 2번 <https://universe.roboflow.com/cpen-355/score-keep-ml>

Train: epoch=10, Test: conf=0.3

<https://github.com/mov-z/LLM-Tutorials/blob/main/OpenAI/OpenAI_API_call.ipynb>

**테니스 해설**

이미지 프롬프트

[프롬프트]

테니스 경기 릴레이가 진행되고 있는 중이다.

각 상황에 맞춰 시각장애인이 이해할 수 있도록 해설위원처럼 해설해줘.

사실을 기반으로 해설해줘.

[추가 설명(디텍팅 이미지에서만)]

빨간색 점은 코드의 꼭짓점 부분을 나타낸다.

파란색 박스는 각각의 플레이들을 나타낸다.

초록색은 공의 위치와 궤도를 나타낸다.

화면 오른쪽 위에 있는 탑 뷰 이미지로 파란색은 플레이어의 위치이다.

노란색은 공이 바닥에 튕긴 부분을 점으로 나타낸 것이다.

* 테니스 경기이다 해설 위원이 해설하는 것처럼 순서대로 있는 사진을 보고 해설해줘
* 개선사항
  + 경기 도중인데 서브하는 것으로 인지

[프롬프트]

테니스 경기의 프레임을 추출한 연속적 이미지이다.

이미지를 이름순으로 정렬하여 이미지를 시각화 한 뒤,

각 상황에 맞춰 시각장애인이 이해할 수 있도록 해설위원처럼 해설해줘

[추가 설명(디텍팅 이미지에서만)]

빨간색 점은 코드의 꼭짓점 부분을 나타낸다

파란색 박스는 각각의 플레이들을 나타낸다

초록색은 공의 위치와 궤도를 나타낸다

화면 오른쪽 위에 있는 탑 뷰 이미지로 파란색은 플레이어의 위치.

노란색은 공이 바닥에 튕긴 부분을 점으로 나타낸 것이다

[요청사항]

연속 장면의 흐름을 반영할 것

사실을 기반으로 해설하며 추론하지 말 것

공의 속도, 방향, 궤적, 선수의 움직임을 위주로 서술할 것

→ 너무 딱딱

개선사항

* 해설위원같이 설명하지 않음

VER2

[프롬프트] 테니스 경기의 프레임을 추출한 연속적 이미지이다. 이미지를 이름순으로 정렬하여 이미지를 시각화 한 뒤, 각 상황에 맞춰 시각장애인이 이해할 수 있도록 해설위원처럼 해설해줘 [추가 설명(디텍팅 이미지에서만)] 빨간색 점은 코드의 꼭짓점 부분을 나타낸다 파란색 박스는 각각의 플레이들을 나타낸다 초록색은 공의 위치와 궤도를 나타낸다 화면 오른쪽 위에 있는 탑 뷰 이미지로 파란색은 플레이어의 위치. 노란색은 공이 바닥에 튕긴 부분을 점으로 나타낸 것이다 [요청사항] 연속 장면의 흐름을 반영할 것 사실을 기반으로 해설하며 추론하지 말 것 공의 속도, 방향, 궤적, 선수의 움직임을 위주로 서술할 것

**TrackNet이란? (스포츠 경기에서 공 추적에 특화된 모델)**

TrackNet은 **딥러닝 기반의 스포츠 공 추적 모델**로, 주로 배드민턴, 탁구, 테니스 같은 스포츠에서 **작은 공을 프레임 단위로 지속적으로 추적**하는 데 특화된 알고리즘

**CNN + RNN 기반의 구조**

* YOLO와 달리, TrackNet은 CNN을 사용한 후 **LSTM(RNN)** 을 추가하여 **시간 정보를 고려한 예측**을 수행함.
* 즉, 공이 일시적으로 보이지 않더라도(예: 선수 뒤에 가려질 경우) **이전 프레임을 참고하여 공의 위치를 예측**할 수 있음.

**YOLO vs. TrackNet 비교**

| **기능** | **YOLO** | **TrackNet** |
| --- | --- | --- |
| **작은 객체 탐지** | 어렵다 (특히 빠른 공) | 강하다 |
| **연속된 추적 가능?** | 불가능 (단일 프레임 기준) | 가능 (이전 프레임 고려) |
| **공 가림 예측 가능?** | 불가능 | 가능 |
| **속도** | 매우 빠름 (하지만 일관성 부족) | 비교적 빠름 (연속 추적 가능) |
| **적용 스포츠** | 농구, 축구 등 큰 물체 추적 | 테니스, 배드민턴, 탁구 등 작은 공 추적 |

**TrackNet을 사용할지 고민해야 할 점**

💡 **언제 TrackNet을 사용할까?**

✅ **작은 공을 정확하게 추적해야 하는 경우** (테니스, 배드민턴, 탁구)

✅ **연속된 프레임을 고려한 예측이 필요한 경우**

✅ **공이 가려지거나 일시적으로 사라지는 경우에도 추적해야 할 때**

💡 **언제 YOLO를 사용할까?**

✅ **큰 객체(선수, 자동차 등)를 탐지할 경우**

✅ **빠른 실시간 속도가 필요할 경우**

✅ **공을 단순히 "탐지"하는 것만 필요할 경우**

**Bounce Detection**

* 머신러닝에서 공의 바운스(튕김) 위치를 예측하는 방법 중의 하나임
* 바운스 탐지란 공이 어떤 표면에 닿아 튀어오르는 위치를 바운스하고 하고, 이를 탐지하여 게임 중 공의 궤적, 방향, 변화, 속도 등을 분석함

**CatBoostRegressor**

* CatBoostRegressor는 공의 바운스 위치FMF 연속적인 좌표(예: X, Y, Z)로 나타낸 회귀 모델임
* 공의 궤적 정보에 대해 PreTrained된 모델로 바로 적용하여 공의 바운스 위치 예측가능
  + 이전 단계에서 탐지된 공의 위치와 이동 경로
  + 바운스 표면, 중력, 속도
  + 바운스 위치 시각화 및 게임 진행 예측 등에 활용.

**Bounce Detection을 위한 대표적인 3가지 모델**

| **모델** | **특징** | **장점** | **단점** |
| --- | --- | --- | --- |
| **CatBoostRegressor** | Gradient Boosting 기반의 회귀 모델. 범주형 데이터와 비선형 데이터를 잘 처리. | - 범주형 데이터 자동 처리- 빠른 학습 속도- 적은 하이퍼파라미터 튜닝 필요 | - 메모리 사용량이 많을 수 있음 |
| **XGBoost** | Gradient Boosting의 확장판. 다양한 튜닝 옵션과 높은 정확도를 제공. | - 높은 정확도- 결측치 자동 처리- 커뮤니티와 튜토리얼이 풍부 | - 범주형 데이터 처리 시 인코딩 필요- 학습 속도가 느릴 수 있음 |
| **LSTM (RNN)** | 시계열 데이터를 처리하는 심층 학습 모델. 시간 순서가 중요한 데이터에 적합. | - 시간적 관계 학습 가능- 복잡한 패턴 처리 가능 | - 많은 데이터 필요- 계산 비용 높음- 학습 속도 느림 |

**Ball Detection vs. Bounce Detection: 주요 차이점**

| **항목** | **Ball Detection** | **Bounce Detection** |
| --- | --- | --- |
| **목적** | 공의 위치를 탐지 (현재 좌표) | 공이 바운스한 위치와 시점을 예측 |
| **입력 데이터** | 이미지 또는 영상 프레임 | 공의 궤적 데이터 (시간에 따른 위치 변화) |
| **출력 데이터** | 공의 위치 좌표 (x, y 또는 x, y, z) | 바운스 지점 좌표와 시점 (x, y 또는 x, y, z, t) |
| **사용 모델** | YOLO, SSD, Faster R-CNN 등 객체 탐지 모델 | CatBoost, XGBoost, LSTM 등 회귀 및 시계열 모델 |
| **난이도** | 영상 처리(컴퓨터 비전)의 정확도 문제 | 물리적 모델링(속도, 가속도, 충돌 데이터 등)의 정확도 문제 |
| **응용 사례** | 스포츠 경기 중 공의 위치 추적, 장애물 탐지 | 테니스/탁구 경기 분석, 공의 물리적 운동 예측, 전략 분석 |

* **Court detection**
  + 테니스 코트를 탐지하기 위한 모델로 코드내 14개 포인트를 탐지함

**Ball Detection vs. Bounce Detection: 주요 차이점**

| **항목** | **Ball Detection** | **Bounce Detection** |
| --- | --- | --- |
| **목적** | 공의 위치를 탐지 (현재 좌표) | 공이 바운스한 위치와 시점을 예측 |
| **입력 데이터** | 이미지 또는 영상 프레임 | 공의 궤적 데이터 (시간에 따른 위치 변화) |
| **출력 데이터** | 공의 위치 좌표 (x, y 또는 x, y, z) | 바운스 지점 좌표와 시점 (x, y 또는 x, y, z, t) |
| **사용 모델** | YOLO, SSD, Faster R-CNN 등 객체 탐지 모델 | CatBoost, XGBoost, LSTM 등 회귀 및 시계열 모델 |
| **난이도** | 영상 처리(컴퓨터 비전)의 정확도 문제 | 물리적 모델링(속도, 가속도, 충돌 데이터 등)의 정확도 문제 |
| **응용 사례** | 스포츠 경기 중 공의 위치 추적, 장애물 탐지 | 테니스/탁구 경기 분석, 공의 물리적 운동 예측, 전략 분석 |

* 코드

[모두 합치기.ipynb](https://prod-files-secure.s3.us-west-2.amazonaws.com/c09f8228-29c7-4dcb-8ca3-1de7d3988fab/e4b9e885-3d19-4a47-9b86-ac7f4639ad4d/%EB%AA%A8%EB%91%90_%ED%95%A9%EC%B9%98%EA%B8%B0.ipynb)

* + 모델 모음

[YOLO11\_TennisDetection\_CustomData\_FineTuning(Data\_600).ipynb](https://prod-files-secure.s3.us-west-2.amazonaws.com/c09f8228-29c7-4dcb-8ca3-1de7d3988fab/0cedcf42-baef-4607-9c7a-de0021a44349/YOLO11_TennisDetection_CustomData_FineTuning(Data_600).ipynb)

[YOLO11\_TennisDetection\_CustomData\_FineTuning(Class\_1).ipynb](https://prod-files-secure.s3.us-west-2.amazonaws.com/c09f8228-29c7-4dcb-8ca3-1de7d3988fab/db1df90c-f53c-4852-8af9-f3c4c3f51e2b/YOLO11_TennisDetection_CustomData_FineTuning(Class_1).ipynb)

[YOLO11\_TennisDetection\_CustomData\_FineTuning(Class\_4).ipynb](https://prod-files-secure.s3.us-west-2.amazonaws.com/c09f8228-29c7-4dcb-8ca3-1de7d3988fab/2868e45d-b11b-4c01-9374-557d5131221a/YOLO11_TennisDetection_CustomData_FineTuning(Class_4).ipynb)