

Deep association metric을 활용한 simple online and realtime tracking

이지호 김수연 이제희 박준하

KHUDA

목차

1. Object Detection과 Object Tracking

2. DeepSORT에서 사용되는 기술들
- Kalman Filter, 헝가리안 알고리즘, Mahalanobis distance, IOU, SORT

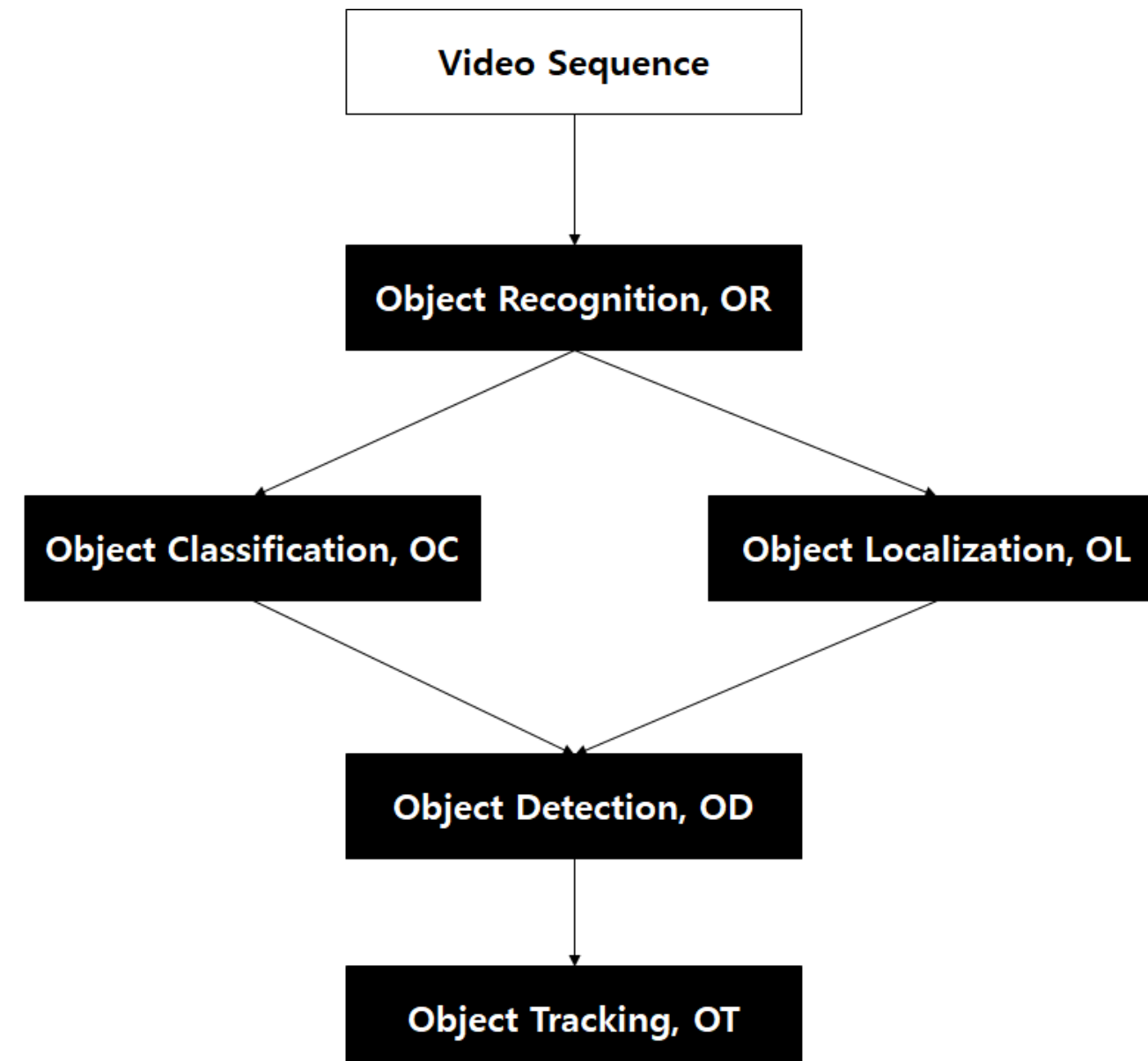
3. DeepSORT란



Object Detection 이란?

Object Detection

이미지나 동영상에서 사람, 동물, 차량 등 의미 있는 객체의 종류와 그 위치를 정확하게 찾기 위한 컴퓨터 비전기술.



일단 영상이 들어오면,

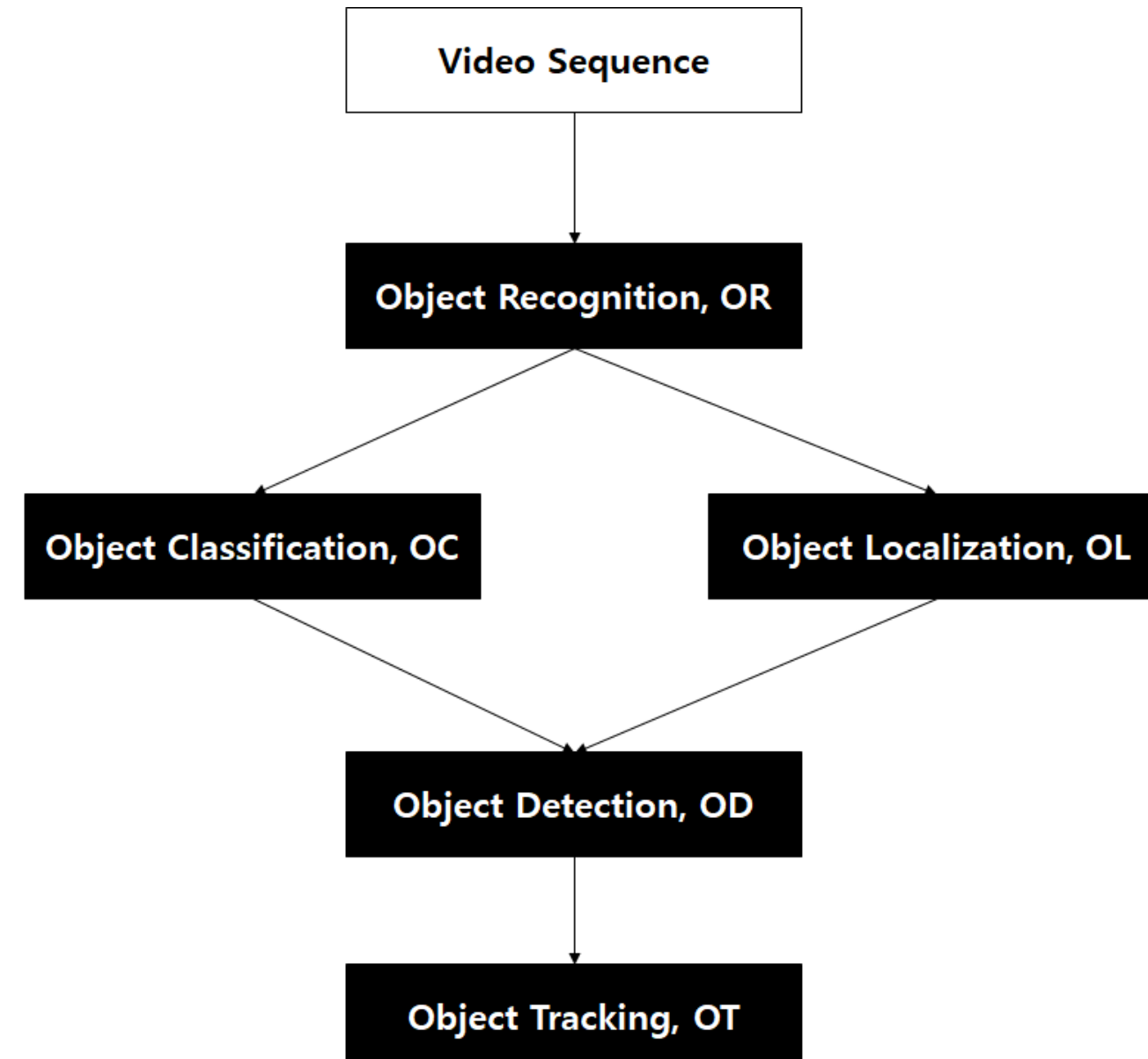
물체가 있다는 것을 인식 (Object Recognition) 하고,
그 물체가 무엇인지 분류 (Object Classification) 하고,
정확한 위치를 찍어줍니다 (Object Localization).

OC와 OL이 합쳐지면 Object Detection, OD가 됩니다.

Object Tracking 이란?

Object Tracking

이미지나 영상에서 객체를 찾아내는 것.



일단 영상이 들어오면,

물체가 있다는 것을 인식 (Object Recognition) 하고,
그 물체가 무엇인지 분류 (Object Classification) 하고,
정확한 위치를 찍어줍니다 (Object Localization).

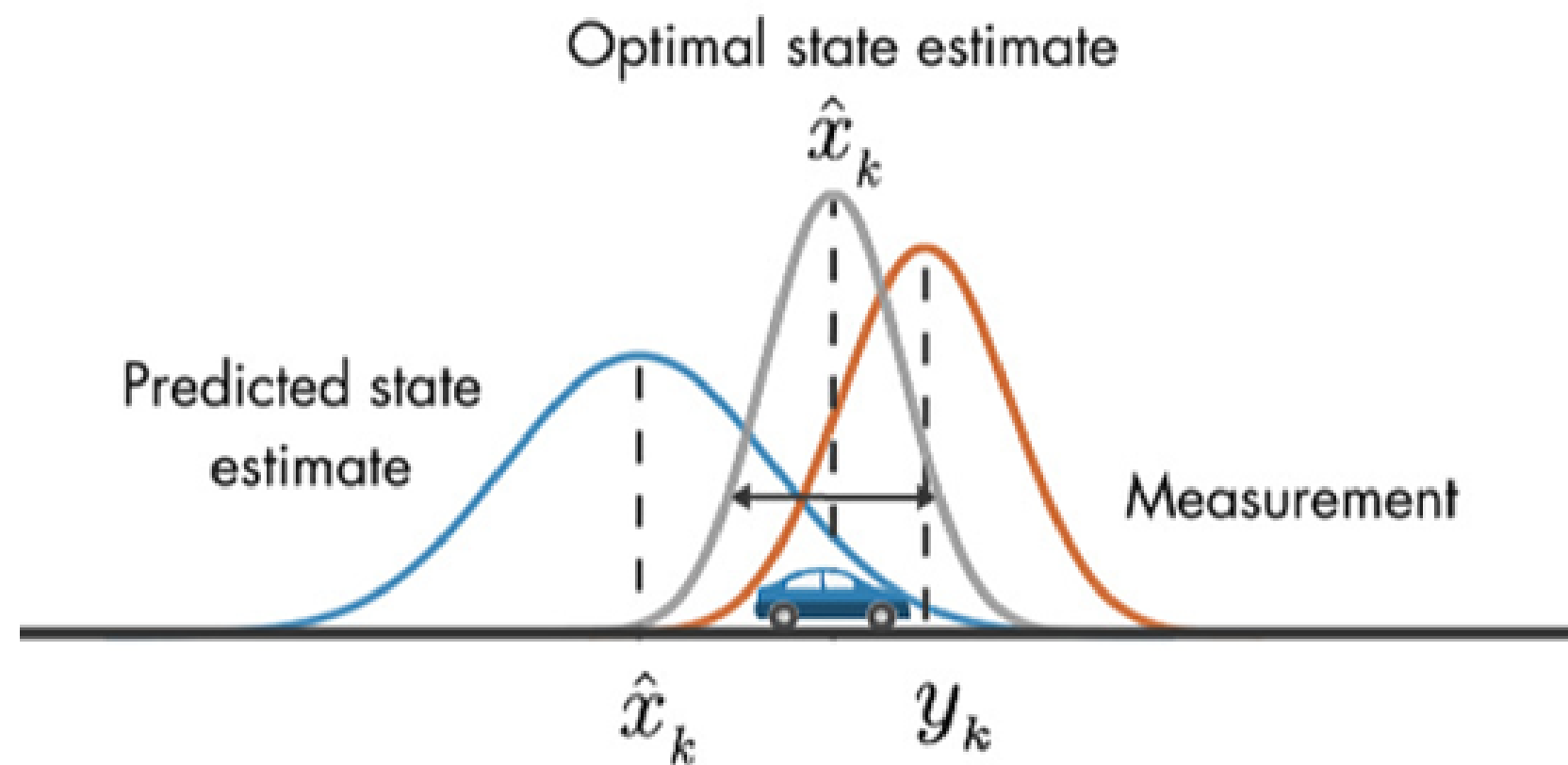
OC와 OL이 합쳐지면 Object Detection, OD가 됩니다.

추후 OD의 결과로 각 Box를 이전 Frame과 비교하여
ID를 매칭시키고 여러가지 기술들을 사용하는게 Object Tracking 입니다.

1. Kalman Filter

Kalman Filter

이전 프레임에 등장한 개체를 이용하여 다음 프레임의 개체의 위치를 예측하고 측정하는 것.



Predicted state estimate = 예측한 값(예측 모델)

Measurement = 실제 측정 값(측정 모델)

예측 값과 측정 값의 각각의 Gaussian Distribution을 이용해 상태를 업데이트 하여 최적의 추정값을 얻음.

두 모델을 가지고 '더 잘 추측'하기 위해 상태를 업데이트 함.

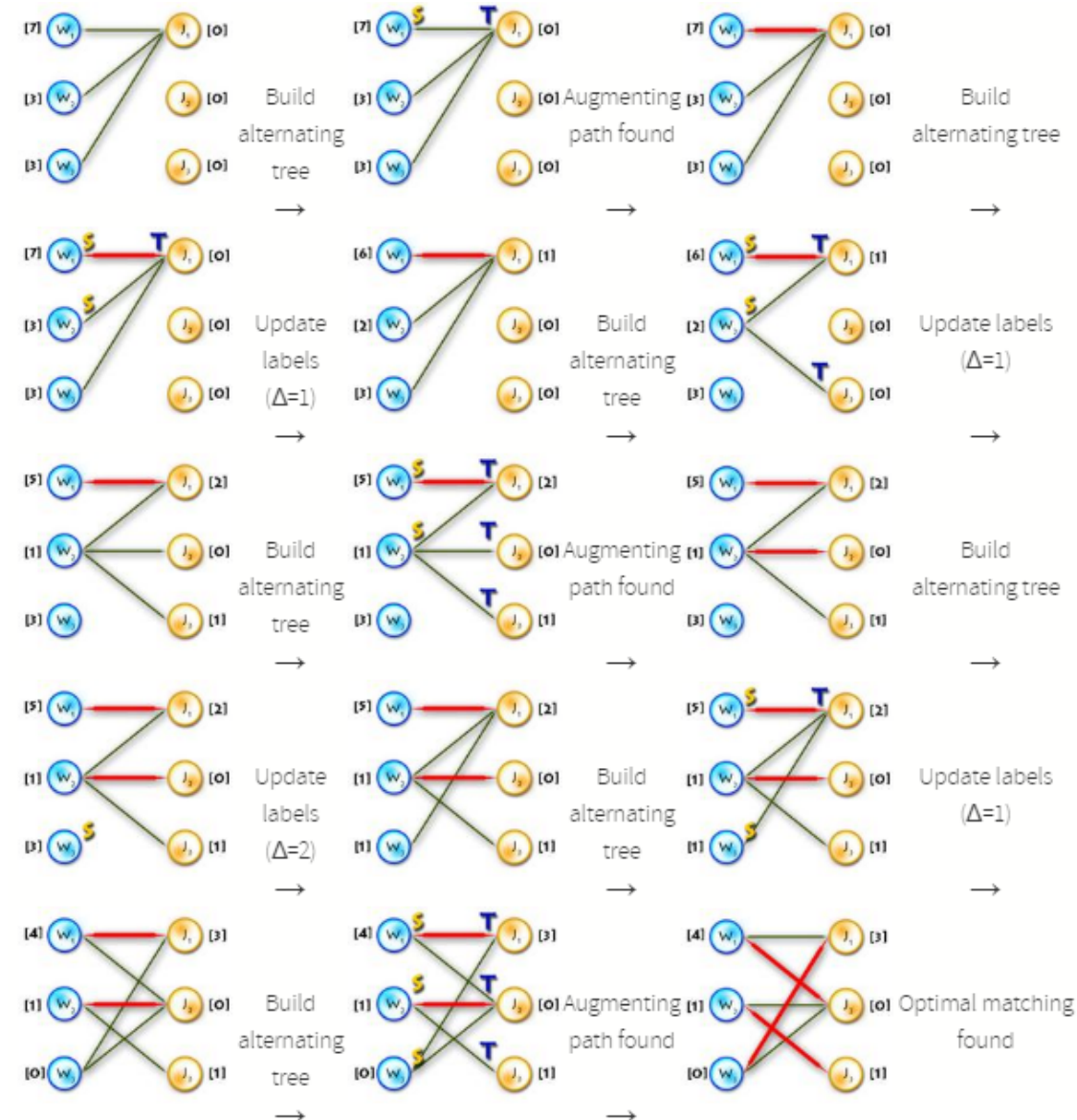
사용하는 이유?

- Detection 중 발생하는 Noise를 처리하는 데 도움을 줌.

선형성(물체가 순간적으로 사라지거나 나타나지 않음)을 나타내는 영상 Tracking에 적합함.

2. 헝가리안 알고리즘

헝가리안 알고리즘



- 행렬 형태로 작업과 작업자 간의 비용을 표현하고, 최적의 할당을 찾기 위해 행과 열을 순서대로 스캔하면서 매칭을 결정하는 방식을 사용함.

- 작업자와 작업의 수가 같을 때 가장 효과적으로 작동하며, 이러한 경우 최적의 할당을 찾아냄.

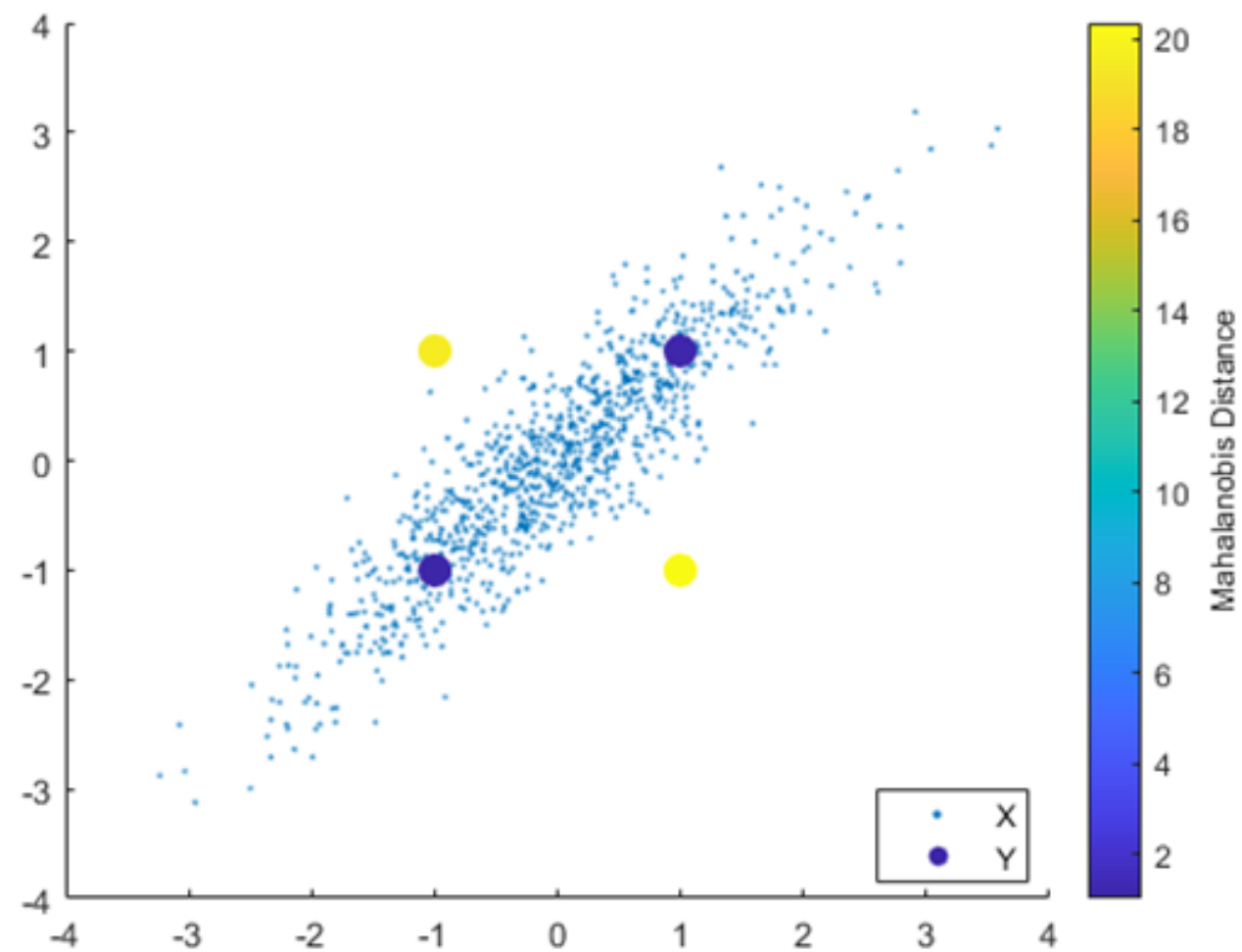
- 모든 가능한 할당 조합을 탐색하는 대신, 최적의 할당을 찾기 위해 이러한 단계를 거치며 계산량을 줄임.

- 시간 복잡도가 $O(n^3)$ 으로 상당히 효율적이며, 작업과 작업자의 수가 증가해도 비교적 빠른 속도로 문제를 해결할 수 있음

3. Mahalanobis distance

Mahalanobis distance

평균과의 거리가 표준 편차의 몇 배인지를 나타내는 값.



- 어떤 값이 얼마나 일어나기 힘든 값인지, 또는 얼마나 이상한 값인지를 수치화 하는 한 방법임.
- 주로 어떤 데이터가 가짜 데이터인지, 아니면 진짜 데이터인지를 구분하는 용도로 사용됨.


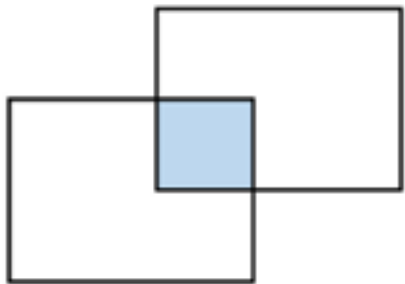
4. IOU

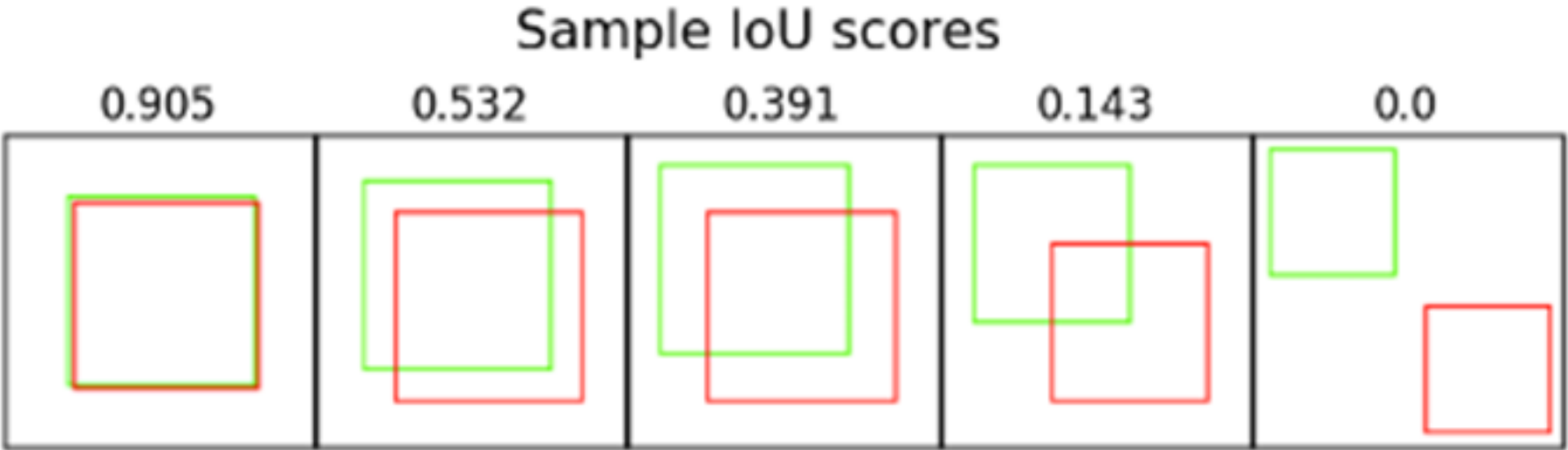
IOU 겹치는 영역에 대해 수치화한 값. 겹치는 영역이 커질수록 값이 높아짐.

IOU =

Overlapping Region

Combined Region





5. SORT

SORT

실시간 추적을 위해 object들을 효과적으로 연관지어주는 MOT(Multi Object Tracking).



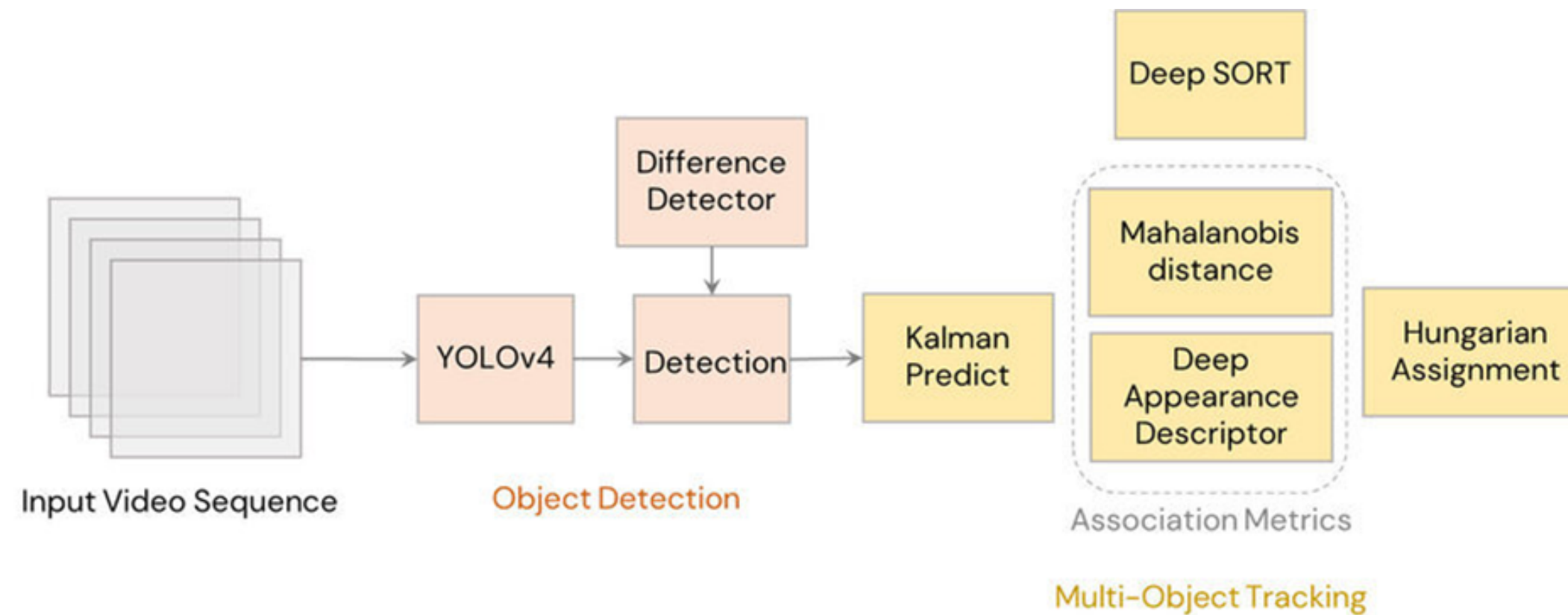
Fig. 3. Provided detections from the MOT17 benchmark video 10 [11]. The left detections are from DPM, and includes the partially occluded pedestrian but also includes several false positives. The right detections are from Faster-RCNN, which has high precision but fails to detect the occluded pedestrian.

- SORT = Simple Online and Realtime Tracking
- 프레임에 대한 정보 없이 과거와 현재 프레임의 객체 detection 정보만을 사용하여 연관 관계에 대한 Tracking을 수행하는 방식임.
- Occlusion(폐색, 가려짐) 문제와 다양한 객체들이 움직일 때, 서로의 ID 추적이 변경되는 ID Switching에 취약함.

6. DeepSORT

DeepSORT

SORT의 문제점인 ID Switching과 Occlusion 문제를 해결하고자 함.



- Deep Appearance Descriptor로 Re-identification 모델을 적용해서 ID Switching 문제를 해결함.
- Matching Cascade 로직으로 더 정확한 추적을 가능케 함.

6. DeepSORT

Process of DeepSORT

1. Detection

- 이미지를 input으로 받아 존재하는 물체에 대해 Bounding box 정보를 받는다.

2. Kalman Filter Predict

- 칼만 필터를 통해, 기존의 Track 정보로부터 다음 frame의 물체의 위치를 예측한다.

3. Track Check

- 해당 Track들이 충분한 근거를 가진 Track인지 3번이상 확인하고 Track으로 인정받을 경우 "Confirm", 그렇지 않을 경우 "Tentative"

4. Matching Cascade

Confirmed인 트랙들에 대해서 매칭을 진행한다.

5. IOU Matching

Match되지 않은 Track, Detection들에 대해서 IOU 매칭을 진행한다.

6. DeepSORT

Process of DeepSORT

6. Tracking Life Cycle

- 매칭 결과를 기반으로 Track 객체의 생애주기를 정해주는 부분이다.

7. Kalman Filter Update

- 현재 가진 Matched Track들을 다음 frame을 위해 Bounding box를 예측한다.
- 이때, Track의 정당성을 주기 위해, Track이 등장한 횟수(hit)가 3회 이상 나왔을 경우 상태를 "Confirmed"로 변경.

8. Reculsive

- 이 과정을 재귀적으로 진행한다.

6. DeepSORT

Result of DeepSORT

실험적 평가 결과, 이 확장된 방법을 사용하면 ID 변경 횟수가 45% 감소하고 고프레임 속도에서 전반적으로 경쟁력 있는 성능을 달성할 수 있었다.

이는 온라인 추적 중에도 객체를 오래 가리는 상황에서도 정확한 추적이 가능해진다는 의미이다.

이 연구는 다중 객체 추적과 관련된 중요한 문제를 다루며, 실시간 및 온라인 추적 시스템의 성능을 향상시킬 수 있는 유용한 접근 방식을 제시한다.

6. DeepSORT

pros and cons

장점

- 매우 빠른 Object Detector 덕분에 빠르게 추적할 수 있다.
- real-time 어플리케이션에 사용할 수 있다.
- 높은 정확도를 보이며, SORT에 비해 ID Switching이 줄어들었다.

단점

- CPU와 GPU가 모두 필요하다.
- CPU는 계산이 느려 실시간 처리에 적합하지 않고, GPU는 자체적으로 구동할 수 없어서 비용과 전력 소비 측면에서 다소 비싸진다.
- feature에 배경 정보가 너무 많아서 Object Detector의 bounding box가 너무 크게 잡힐 때 알고리즘의 효과가 저하된다.
- 어두운 환경에서 성능이 약간 저하된다.

감사합니다.

KHUDA