Object Tracking

Assignment Questions

PW SKILLS

Object Tracking

1. What is object tracking, and how does it differ from object detection?
2. Explain the basic working principle of a Kalman Filter.
3. What is YOLO, and why is it popular for object detection in real-time applications
4. How does DeepSORT improve object tracking?
5. Explain the concept of state estimation in a Kalman Filter.
6. What are the challenges in object tracking across multiple frames?
7. Describe the role of the Hungarian algorithm in DeepSORT.
8. What are the advantages of using YOLO over traditional object detection methods?
9. How does the Kalman Filter handle uncertainty in predictions?
10. What is the difference between object tracking and object segmentation?
11. How can YOLO be used in combination with a Kalman Filter for tracking?
12. What are the key components of DeepSORT?
13. Explain the process of associating detections with existing tracks in DeepSORT.
14. Why is real-time tracking important in many applications?
15. Describe the prediction and update steps of a Kalman Filter.
16. What is a bounding box, and how does it relate to object tracking?
17. What is the purpose of combining object detection and tracking in a pipeline?
18. What is the role of the appearance feature extractor in DeepSORT?
19. How do occlusions affect object tracking, and how can Kalman Filter help mitigate this?
20. Explain how YOLO's architecture is optimized for speed.
21. What is a motion model, and how does it contribute to object tracking?
22. How can the performance of an object tracking system be evaluated?
23. What are the key differences between DeepSORT and traditional tracking algorithms?

Practical

1. Implement a Kalman filter to predict and update the state of an object given its measurements.
2. Write a function to normalize an image array such that pixel values are scaled between 0 and 1.
3. Create a function to generate dummy object detection data with confidence scores and bounding boxes. Filter the detections based on a confidence threshold.
4. Write a function that takes a list of YOLO detections and extracts a random 128-dimensional feature vector for each detection.
5. Write a function to re-identify objects by matching feature vectors based on Euclidean distance.
6. Write a function to track object positions using YOLO detections and a Kalman Filter.
7. Implement a simple Kalman Filter to track an object's position in a 2D space (simulate the object's movement with random noise).