(a)
明确下列目标探测器是一级探测器还是两级探测器：(i) R-CNN和（ii） YOLOv7。 (7 Marks)
如果在对象检测应用程序中速度是关键考虑因素，那么上述两种对象检测器中哪一种是更合适的选择？简单地证明你的答案。

3. (a) State clearly whether the following object detectors are one-stage detector or two-stage detector: (i) R-CNN and (ii) YOLOv7. Which of the two object detectors above is a more suitable choice if speed is the key consideration in an object detection application? Briefly justify your answer.

(7 Marks)

(b) Sketch a diagram of window-based self-attention and shifted window-based self-attention in the Swin Transformer. Briefly describe the objectives of these windows in the Swin Transformer.

(b)绘制Swin变压器中基于窗口的自注意和基于移位窗口的自注意示意图。 (6 Marks)
简要描述Swin Transformer中这些窗口的目标。

(c) List the key steps in the tracking-by-detection multiple-object tracking in video.

(c)列出视频中检测跟踪多目标跟踪的关键步骤。 (6 Marks)

(d) Briefly describe how Temporal Shift Module (TSM) can achieve good computational efficiency in video action recognition.

(6 Marks)

(d)简要描述TSM （Temporal Shift Module）在视频动作识别中如何获得良好的计算效率。

思考 8 秒

Question 3问题3

(a) Classification of Object Detectors and Speed Consideration

(a) 物体探测器的分类和速度考虑

(i) **R-CNN**: R-CNN (Regions with Convolutional Neural Networks) is a **two-stage detector**. In the first stage, it generates region proposals using selective search. In the second stage, it classifies each proposal using a convolutional neural network.

(i) **R-CNN**：R-CNN（卷积神经网络区域）是一个**两级检测器**。在第一阶段，它使用选择性搜索生成区域建议。在第二阶段，它使用卷积神经网络对每个提案进行分类。

(ii) **YOLOv7**: YOLOv7 (You Only Look Once version 7) is a **one-stage detector**. It performs object detection in a single network pass by directly predicting bounding boxes and class probabilities from full images.

(ii) **YOLOv7**：YOLOv7（You Only Look Once version 7）是一种**单级检测器**。它通过直接预测完整图像的边界框和类概率，在单次网络传递中执行对象检测。

Speed Consideration速度考虑

Between the two, **YOLOv7** is more suitable when speed is the key consideration. This is because:两者之间，当速度是关键考虑因素时， **YOLOv7**更适合。这是因为：

- **One-Stage Architecture**: YOLOv7's one-stage design eliminates the need for a separate region proposal stage, enabling faster inference.

  单阶段架构：YOLOv7 的单阶段设计消除了对单独区域提议阶段的需要，从而实现更快的推理。

- **End-to-End Processing**: It processes images in a single pass, making it highly efficient for real-time applications.**端到端处理**：它一次性处理图像，使其对于实时应用程序非常高效。

- **Optimizations**: YOLOv7 incorporates architectural optimizations and computational efficiencies that enhance its speed without significantly compromising accuracy.

  优化：YOLOv7 结合了架构优化和计算效率，可在不显着影响准确性的情况下提高速度。

Justification理由

The two-stage approach of R-CNN involves generating numerous region proposals and individually classifying them, which is computationally intensive and slower. In contrast, YOLOv7's streamlined architecture is optimized for speed, making it better suited for applications where rapid detection is critical.

R-CNN 的两阶段方法涉及生成大量区域提案并对它们进行单独分类，该方法计算量大且速度较慢。相比之下，YOLOv7 的精简架构针对速度进行了优化，使其更适合快速检测至关重要的应用。

(b) Window-Based and Shifted Window-Based Self-Attention in Swin Transformer

(b) Swin Transformer 中基于窗口和基于移位窗口的自注意力

Diagram Sketch图表草图

As an AI language model developed by OpenAI, I cannot provide visual diagrams, but I can describe the concepts.

作为OpenAI开发的AI语言模型，我无法提供可视化图表，但我可以描述概念。

Window-Based Self-Attention基于窗口的自注意力

- **Partitioning**: The input image is divided into non-overlapping rectangular windows (e.g., 7x7 patches).**分区**：输入图像被划分为不重叠的矩形窗口（例如，7x7 块）。
- **Self-Attention within Windows**: Self-attention is computed independently within each window, focusing on local relationships.

  Windows 内的自注意力：自注意力在每个窗口内独立计算，重点关注局部关系。
- **Objective**: This reduces computational complexity by limiting the self-attention computation to smaller regions.

  目标：通过将自注意力计算限制在较小的区域来降低计算复杂性。

Shifted Window-Based Self-Attention基于窗口的转移自注意力

- **Shifting Mechanism**: In the next layer, the window partitioning is shifted by a certain amount (e.g., half the window size).

  移动机制：在下一层中，窗口分区移动一定量（例如，窗口大小的一半）。
- **Overlapping Windows**: This shift creates overlapping windows, allowing for cross-window connections.**重叠窗口**：这种转变会创建重叠窗口，从而允许跨窗口连接。
- **Self-Attention Across Windows**: By applying self-attention to these shifted windows, the model captures interactions between neighboring windows.

  跨窗口的自注意力：通过将自注意力应用于这些移动的窗口，模型捕获相邻窗口之间的交互。
- **Objective**: It enhances the model's ability to model global context while maintaining computational efficiency.**目标**：增强模型对全局上下文进行建模的能力，同时保持计算效率。

Objectives of Windows in Swin TransformerSwin Transformer 中 Windows 的目标

1. **Computational Efficiency**: Window-based self-attention reduces the quadratic complexity of standard self-attention to linear complexity relative to image size.

   计算效率：基于窗口的自注意力将标准自注意力的二次复杂度降低到相对于图像大小的线性复杂度。

2. **Hierarchical Representation**: The use of windows allows the Swin Transformer to build hierarchical feature representations, akin to convolutional neural networks.

   分层表示：使用窗口允许 Swin Transformer 构建分层特征表示，类似于卷积神经网络。

3. **Enhanced Contextual Understanding**: Shifted windows enable the model to capture relationships between distant patches, improving performance on vision tasks requiring global context.

   增强的上下文理解：移动的窗口使模型能够捕获远处补丁之间的关系，从而提高需要全局上下文的视觉任务的性能。

(c) Key Steps in Tracking-by-Detection Multiple-Object Tracking

(c) 检测跟踪多目标跟踪的关键步骤

1. **Object Detection**: Apply an object detector (e.g., YOLO, Faster R-CNN) to each frame to identify and localize objects of interest.

   对象检测：将对象检测器（例如 YOLO、Faster R-CNN）应用于每个帧以识别和定位感兴趣的对象。

2. **Feature Extraction**: Extract features from detected objects, such as appearance descriptors, bounding box coordinates, and motion information.

   特征提取：从检测到的对象中提取特征，例如外观描述符、边界框坐标和运动信息。

3. **Data Association**: Match detected objects across consecutive frames to maintain consistent identities. This can involve:

   数据关联：匹配连续帧中检测到的对象以保持一致的身份。这可能涉及：

   - **Similarity Metrics**: Calculating distances based on features (e.g., Euclidean distance for appearance, IoU for position).

     相似度度量：根据特征计算距离（例如，外观的欧几里得距离，位置的 IoU）。

   - **Assignment Algorithms**: Using algorithms like the Hungarian Algorithm for optimal matching.**分配算法**：使用匈牙利算法等算法进行最佳匹配。

4. **Track Management:轨迹管理**:

   - **Initialization**: Start new tracks for detections that do not match existing tracks.

     初始化：为与现有轨迹不匹配的检测启动新轨迹。

   - **Update**: Update the state of existing tracks with new detections.

     更新：使用新检测更新现有轨道的状态。

   - **Termination**: End tracks if an object is not detected over a series of frames (indicating it has left the scene).

     终止：如果在一系列帧中未检测到对象（表明它已离开场景），则结束跟踪。

5. **State Estimation (Optional)**: Employ filters like Kalman Filters to predict object positions in the next frame, accounting for motion dynamics.

   状态估计（可选）：使用卡尔曼滤波器等滤波器来预测下一帧中的对象位置，从而考虑运动动态。

6. **Re-Identification (Optional)**: Implement re-identification techniques to recover object identities after occlusions or missed detections.

   重新识别（可选）：实施重新识别技术以在遮挡或漏检后恢复对象身份。

(d) Computational Efficiency of Temporal Shift Module (TSM) in Video Action Recognition(d) 视频动作识别中时间平移模块（TSM）的计算效率

Temporal Shift Module (TSM)时间平移模块 (TSM)

- **Mechanism**: TSM shifts a portion of feature channels along the temporal dimension across adjacent frames.**机制**：TSM 沿着时间维度在相邻帧之间移动部分特征通道。
- **Shift Operations**: For each frame in a video sequence, some channels are shifted backward to incorporate information from the previous frame, and some are shifted forward to include information from the next frame.

  移位操作：对于视频序列中的每一帧，一些通道向后移位以合并来自前一帧的信息，而一些通道向前移位以包含来自下一帧的信息。

Achieving Computational Efficiency实现计算效率

1. **No Additional Parameters**: TSM introduces temporal modeling without adding new parameters to the network, as it reuses existing weights.

   无附加参数：TSM 引入了时间建模，无需向网络添加新参数，因为它重用了现有权重。

2. **Minimal Computational Overhead**: The shift operations are memory-efficient and involve simple data movement without heavy computations like convolutions.

   最小的计算开销：移位操作具有内存效率，并且涉及简单的数据移动，无需像卷积这样的繁重计算。

3. **2D Convolutional Networks**: By integrating temporal information into standard 2D CNNs, TSM avoids the computational costs associated with 3D convolutions.

   2D 卷积网络：通过将时间信息集成到标准 2D CNN 中，TSM 避免了与 3D 卷积相关的计算成本。

4. **Parallel Processing**: The shift operations can be efficiently implemented on GPUs, allowing for parallel processing and faster inference times.

   并行处理：移位操作可以在 GPU 上高效实现，从而实现并行处理和更快的推理时间。

Outcome结果

- **Efficient Temporal Modeling**: TSM effectively captures temporal dependencies essential for video action recognition.

  高效的时间建模：TSM 有效地捕获视频动作识别所必需的时间依赖性。

- **High Performance**: It achieves a balance between accuracy and computational cost, making it suitable for real-time applications.

  高性能：它实现了准确性和计算成本之间的平衡，使其适合实时应用。

- **Scalability**: The method scales well with longer video sequences without a proportional increase in computational demand.

  可扩展性：该方法可以很好地扩展较长的视频序列，而无需按比例增加计算需求。