YOLO Assignment

1. What is the fundamental idea behind the YOLO (You Only Look Once) object detection framework?
2. Explain the difference between YOLO V1 and traditional sliding window approaches for object detection.
3. In YOLO V1, how does the model predict both the bounding box coordinates and the class probabilities for each object in an image?
4. What are the advantages of using anchor boxes in YOLO V2, and how do they improve object detection accuracy?
5. How does YOLO V3 address the issue of detecting objects at different scales within an image?
6. Describe the Darknet-53 architecture used in YOLO V3 and its role in feature extraction.
7. In YOLO V4, what techniques are employed to enhance object detection accuracy, particularly in detecting small objects?
8. Explain the concept of PANet (Path Aggregation Network) and its role in YOLO V4's architecture.
9. What are some of the strategies used in YOLO V5 to optimise the model's speed and efficiency?
10. How does YOLO V5 handle real-time object detection, and what trade-offs are made to achieve faster inference times?
11. Discuss the role of CSPDarknet53 in YOLO V5 and how it contributes to improved performance.
12. What are the key differences between YOLO V1 and YOLO V5 in terms of model architecture and performance?
13. Explain the concept of multi-scale prediction in YOLO V3 and how it helps in detecting objects of various sizes.
14. In YOLO V4, what is the role of the CIOU (Complete Intersection over Union) loss function, and how does it impact object detection accuracy?
15. How does YOLO V2's architecture differ from YOLO V3, and what improvements were introduced in YOLO V3 compared to its predecessor?
16. What is the fundamental concept behind YOLOv5's object detection approach, and how does it differ from earlier versions of YOLO?
17. Explain the anchor boxes in YOLOv5. How do they affect the algorithm's ability to detect objects of different sizes and aspect ratios?
18. Describe the architecture of YOLOv5, including the number of layers and their purposes in the network.
19. YOLOv5 introduces the concept of "CSPDarknet53." What is CSPDarknet53, and how does it contribute to the model's performance?
20. YOLOv5 is known for its speed and accuracy. Explain how YOLOv5 achieves a balance between these two factors in object detection tasks.
21. What is the role of data augmentation in YOLOv5? How does it help improve the model's robustness and generalization?
22. Discuss the importance of anchor box clustering in YOLOv5. How is it used to adapt to specific datasets and object distributions?
23. Explain how YOLOv5 handles multi-scale detection and how this feature enhances its object detection capabilities.
24. YOLOv5 has different variants, such as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. What are the differences between these variants in terms of architecture and performance trade-offs?
25. What are some potential applications of YOLOv5 in computer vision and real-world scenarios, and how does its performance compare to other object detection algorithms?
26. What are the key motivations and objectives behind the development of YOLOv7, and how does it aim to improve upon its predecessors, such as YOLOv5?
27. Describe the architectural advancements in YOLOv7 compared to earlier YOLO versions. How has the model's architecture evolved to enhance object detection accuracy and speed?
28. YOLOv5 introduced various backbone architectures like CSPDarknet53. What new backbone or feature extraction architecture does YOLOv7 employ, and how does it impact model performance?
29. Explain any novel training techniques or loss functions that YOLOv7 incorporates to improve object detection accuracy and robustness.

Submission Guidelines:

- Answer all the questions in a single Jupyter Notebook file (.ipynb).
- Include necessary code, comments, and explanations to support your answers and implementation.
- Ensure the notebook runs without errors and is well-organized.
- Create a GitHub repository to host your assignment files.
- Rename the Jupyter Notebook file using the format "date_month_topic.ipynb" (e.g., "21st_September_YOLO.ipynb").
- Place the Jupyter Notebook file in the repository.
- Commit and push any additional files or resources required to run your code (if applicable) to the repository.
- Ensure the repository is publicly accessible.
- Submit the link to your GitHub repository as the assignment submission.

Grading Criteria:

- Understanding and completeness of answers: 40%
- Clarity and depth of explanations: 25%
- Correct implementation and evaluation of optimizer techniques: 15%
- Analysis and comparison of different optimizers: 10%
- Proper code implementation and organization: 10%

Note: Create your assignment in Jupyter notebook and upload it to GitHub & share that uploaded assignment file link through your dashboard. Make sure the repository is public.