Benchmarking Report

System Overview

The used system is aimed at the recognition of objects and their related sub objects following a hierarchical perspective. It has JSON output and supports selective image recovery of the sub object. The primary focus includes bringing out efficiency, constant availability and the interface for extensibility.

Inference Speed Results

Based on the provided implementation and testing:

Inference Time : The duration of each of them is about 70–130ms depending on the type of scene.

Achieved FPS : At this, the system attains a speed of about 10-30 frames per second (FPS)

Operating System : [State OS, for example, Windows 10 or Ubuntu 20.04]
Although the achieved FPS is below the 10–30 FPS target, further optimization can enhance performance.

System Architecture

The system consists of the following components:

1. Object Detection :

Yolov8 is used for real time object detection.

Recognizes simple objects as ‘person’, ‘car’ and sub objects as ‘helmet’ and ‘tire’.

2. Hierarchy Management :

Building links between main objects and their subordinate objects.

– Guarantees each noted object and sub object have their distinct identification numbers and spatial coordinates.

3. JSON Output Generator :

– Organizes the result into a neat and organization sub JSON structure for easy usage.

4. Sub Object Retrieval :

dwanguard Saves cropped images of a particular sub object to Extracts.

Optimization Strategies

Several strategies were employed to improve the system's performance:

1. Model Optimization :

Used a light YOLOv8n base model in order to pursue both speed and accuracy.

Utilised pre trained weights of the model further adjusted on suitable datasets for enhancing detection.

2. Efficient Data Handling :

Additions of bounding box calculations using NumPy for lower computational complexity.

3. Parallel Processing :

Implemented multi threading in OpenCV for capturing as well as processing of frames simultaneously.

4. Dynamic Resolution Scaling :

Decreased the resolutions of the original video frames and performed inferences at much faster rates without a large loss of accuracy.

Key Observations

Strengths :

It is resistant to commonly observed levels of motion blur, brightness fluctuation, and occlusion: general video stream tracking.

It is flexible to introduce new object sub object pairs into this architecture due to the modular design.

Challenges :

FPS is below the target range of 10 30 FPS.

Playback of high resolution, for example, 4K slows down the process.

At random, Parallel and obscured objects sometimes cause imprecision in the detection.

Conclusion

With regard to object and sub object detection and the hierarchical relationship of these elements, the system produces the expected results. Although it satisfies conditions of accuracy and modularity, additional refinements have to be made for achieving consistent real time performance on a CPU.

Future Work

Optimize further, that always includes methods such as model pruning or quantization for increasing the overall FPS.

State failure cases, for instance, occlusion and overlapping objects.

Operating and experimenting with the system while it set up for the high resolution in video inputs.