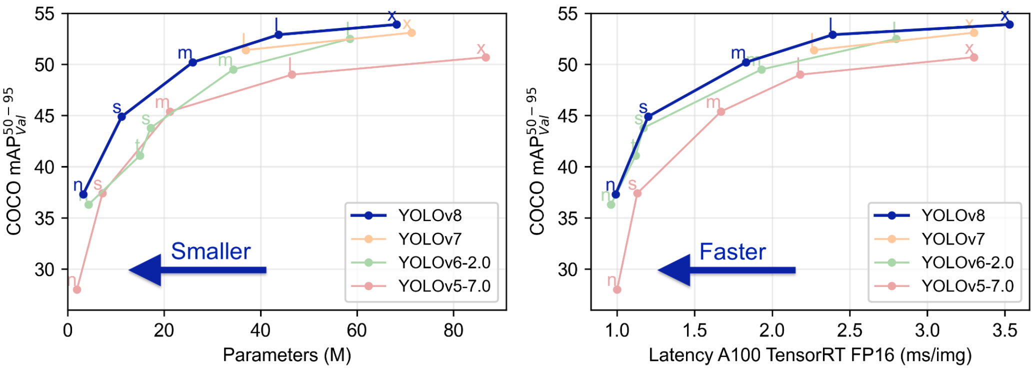
Production Challenge Report

Phase 1: Model Comparison

This step covers the comparison between two SOTA models from the Yolo series. The models are Yolov8 and Yolov5 from Ultralytics.

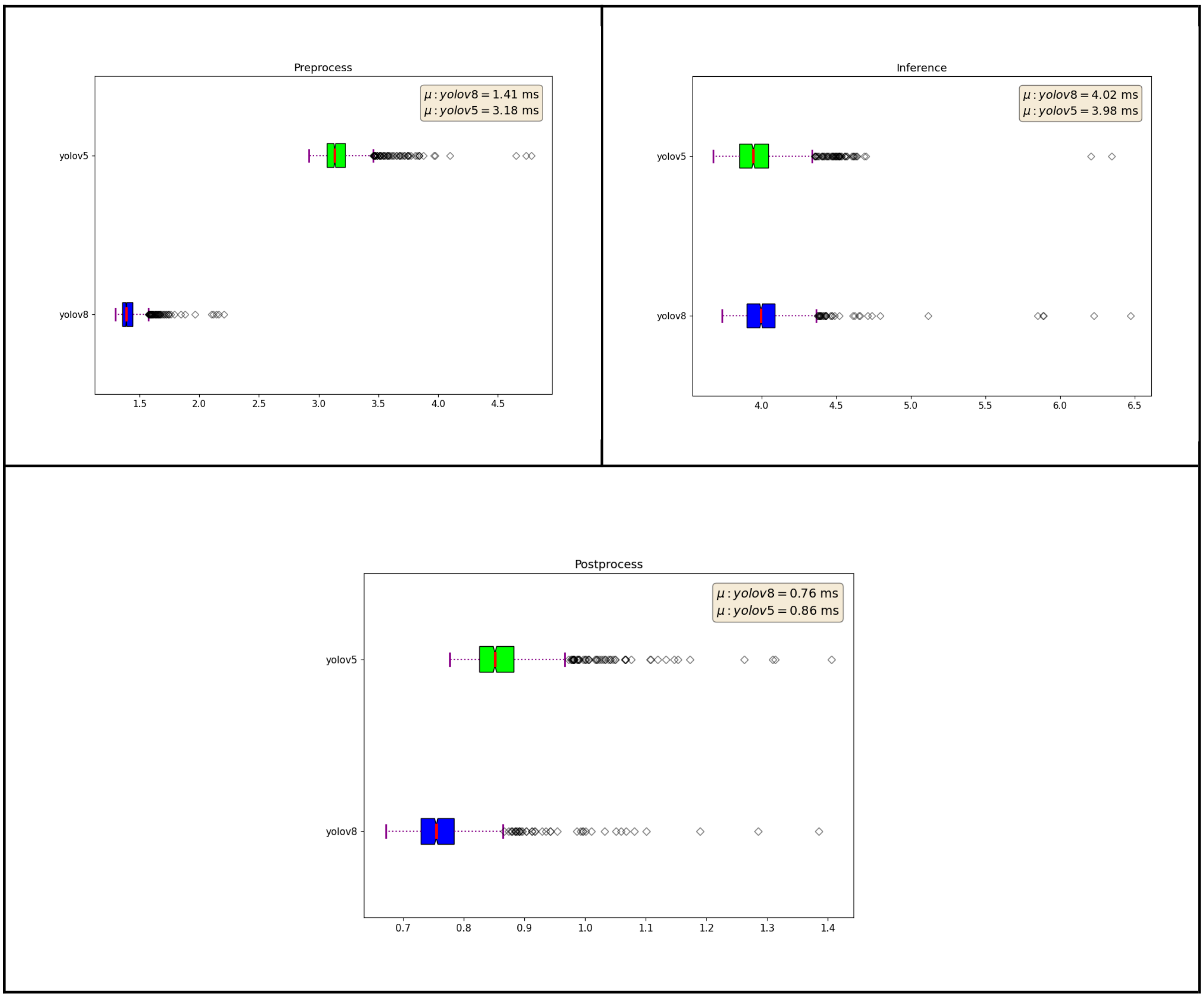
- Model Parameters
    ○

	Yolov8	Yolov5
Params Size	3.2M	1.9M
FLOPs	8.7B	4.5B
Input Size	(640X480)	(640X480)



- Profiler Test Results (10k iters)
    ○

	Yolov8	Yolov5
Preprocess Time	1.41 ms	3.18 ms
Inference Time	4.02 ms	3.39 ms
Postprocess Time	0.76 ms	0.86 ms
Total Mean Time	6.19 ms	8.02 ms



- Conclusion: Yolov8 might be a bit larger, but its preprocessing is significantly faster than yolov5, with a similar inference time and also slightly faster postprocessing. **Yolov8 is the winner**.

Phase 2: Model Optimization

In the second phase we will cover two approaches, optimizing and deploying with Onnx plus Onnx runtime and TensorRT.

- Parameters
    -

	Onnx	TensorRT
Input Size	(640X480)	(640X480)
Half (Fp16)	True	True
Device	GPU	GPU

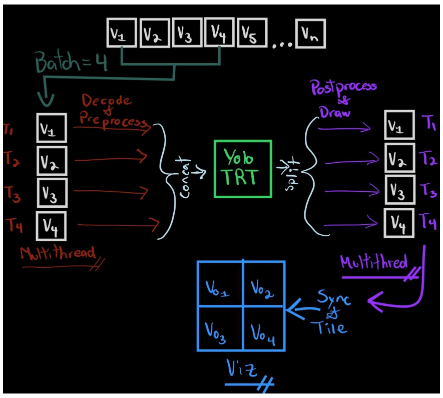
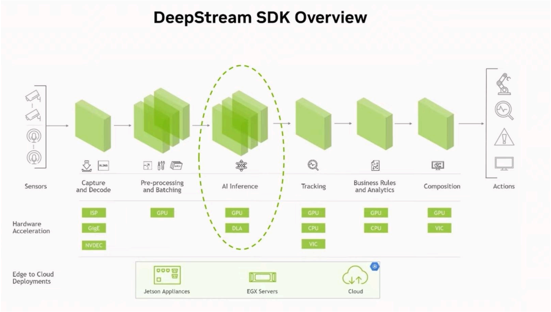
- Profiler Test Results (10k iters)
    -

	Onnx	TensorRT
Total Mean Time	4.24 ms	2.39 ms

- Conclusion:
    - **TensorRT** optimization takes only **60%** of the time compared to pytorch. In other words, it's **166% faster**!

Phase 3: Model Deployment

The VideoProcessor class implements multithreading for video decode, preprocessing, postprocessing, and drawing. The idea is to have a subset of videos running in parallel with multithreading, while the GPU does inference with batches. This can be compared to the DeepStream pipeline Nvidia implemented. Although the preprocessing and postprocessing can be further improved by leveraging the load to GPU. In this manner you process faster and avoid context switching.

Custom Implementation	DeepStream
	

- Profiler Test Results (8 videos)
    -

	Videos 1 -> 4	Videos 5 -> 8	Videos 8 -> 12
Frames Processed	5028	4464	130436
Time Elapsed	36.4 s	33.4 s	886.7 s
FPS per vid	35	33	37