Hardware-Driven Limitations: Vision-Based OCR Deployment Constraints on Low-Resource Systems

Hardware-driven limitations severely restrict deploying vision-based OCR models like Qwen-VL, InternVL, and Mini-CPM-Vision on Systems A (8GB RAM, Iris Xe iGPU) and B (16GB RAM, RTX 3050 4GB VRAM). These constraints prioritize lightweight, CPU-friendly alternatives for reliable OCR pipelines.

Hardware Bottlenecks

- RAM vs. model needs: System A (8GB) and B (16GB) fall short for VLMs; Qwen2-VL-2B inference demands ~16GB+ VRAM recommended, plus system RAM for preprocessing, exceeding available memory.
- VRAM requirements: Vision OCR/VLMs like InternVL2-8B need 20GB+ for inference; RTX 3050 Laptop GPU offers only 4GB GDDR6, Iris Xe shares system RAM (dynamic, <4GB effective).
- CPU/GPU imbalance: i5-1235U (1.3GHz) and i5-11400H (2.7GHz) handle CPU fallback poorly for vision models; integrated Iris Xe lacks CUDA cores, forcing slow CPU paths.
- iGPU vs. dGPU impact: Iris Xe (shared memory, 80 EUs) throttles under ML loads; RTX 3050 (4GB) limits batch=1 for small VLMs, vs. 24GB+ GPUs needed for viability.

Infeasibility Reasons

- Model size/parameters: Qwen2-VL-2B (~2B params), InternVL2-8B (~8B), Mini-InternVL-2B (~2B) generate 1K+ visual tokens/image, bloating memory.
- GPU pressure/latency: 4GB VRAM caps resolution/tokens; dynamic patching (e.g., InternVL 448px tiles) spikes to 6-12GB, causing 10x+ latency on low VRAM.
- Batch limitations: Batch>1 infeasible; even batch=1 OOMs on complex docs due to KV cache growth (e.g., +0.5GB at 32K context).
- Throttling/stability: Laptops throttle CPUs/GPUs under sustained loads (e.g., RPi4B analog shows 90% throughput loss without cooling).

Operational Risks

- OOM crashes: Frequent on image resizing/tokenization; e.g., Qwen-VL utils push >4GB easily.
- Unstable times: Variance from swapping/throttling; 4GB VRAM forces quantization, degrading OCR accuracy.
- Scalability issues: No production throughput; single-image inference >10s vs. ms for classical OCR.
- Iteration slowdown: Fine-tuning impossible; crashes halt debugging in Jupyter/Kaggle workflows.

Trade-off Evaluation

- Accuracy vs. deployability: VLMs excel in layout reasoning (e.g., OCRBench 794 for Qwen2-VL-2B) but crash; Tesseract trades ~10-20% accuracy for stability.
- Multimodal vs. deterministic: VLMs handle docs/charts but nondeterministic; classical OCR ensures repeatable pipelines.
- Research vs. production: VLMs for prototyping (e.g., Mini-InternVL 90% SOTA at 5% size) unfit ops; prioritize uptime over bleeding-edge.

Model Selection Strategy

- CPU-friendly preference: Tesseract/EasyOCR use <300MB RAM, pure CPU, multilingual; ideal for constrained setups.
- Viability thresholds: RTX 40-series 8GB+ VRAM, 32GB+ RAM for small VLMs (e.g., Qwen2-VL-2B quantized); desktop H100/A100 for full.
- Hybrid approaches: Layout detection (YOLOv8 nano, CPU-ok) + Tesseract OCR; boosts complex docs without VLM overhead.

Conclusion

- Hardware limits (4-16GB memory, laptop thermals) exclude vision-based OCR, not model immaturity; e.g., 4GB VRAM vs. 16GB+ needs.
- Engineering demands favor deployable systems like Tesseract hybrids for production OCR pipelines.
