OpenCV C++ Performance Cheat Sheet

1. General Rules

- Build OpenCV with CUDA, TBB, cuDNN, FFmpeg/GStreamer.
- Reuse buffers, avoid .clone(), and preallocate Mats/GpuMats.
- Keep color formats stable across the pipeline.
- Profile with TickMeter (CPU) and CUDA events/streams (GPU).

2. Multithreading

- Pipeline parallelism: Capture → GPU preprocess → CPU work → Display. Use bounded queues to control latency.
- Data parallelism: Split frames into tiles/batches, use cv::parallel_for_ or TBB.

3. CUDA Acceleration

- Use cv::cuda::Stream to overlap upload, kernels, and download.
- Decode directly to GPU with cudacodec::VideoReader.
- Chain preprocessing ops (resize, cvtColor, normalize) on GPU before download.
- For DNNs, set backend/target to CUDA and preprocess on GPU.

4. Zero-Copy & Fast Transfers

- Pinned memory (HostMem::PAGE_LOCKED): Faster async transfers.
- Zero-copy (HostMem::SHARED): CPU & GPU share memory. Great on Jetson/iGPUs, benchmark on dGPUs.
- Avoid host-device ping-pong: keep data GPU-resident.

5. End-to-End Pipeline

- Capture thread writes to pinned buffer.
- Uploader thread transfers to GPU or uses zero-copy header.
- GPU preprocessing runs asynchronously.
- Only download if CPU needs the result.

6. Common Pitfalls

- Unbounded queues = high latency/memory use.
- Too many threads = worse cache behavior.
- Accidental deep copies (clone, implicit conversions).
- Frequent host-device transfers.
- Overusing waitForCompletion(). Sync only when needed.

7. Profiling

- Use TickMeter for CPU stages.
- CUDA events and Nsight for GPU.

- Add NVTX markers for Nsight Systems profiling.

8. Platform Tips

- NVIDIA dGPU: pinned transfers usually faster than zero-copy.
- Jetson: use GStreamer NVMM + cudacodec for zero-copy pipelines.
- Intel iGPU: shared HostMem can improve speed.
- DNN models: export with GPU-friendly preprocessing.

Short Version

- Keep data on GPU as long as possible.
- Overlap transfers with compute.
- Use bounded queues for pipelines.
- Only sync/copy when absolutely necessary.
- Profile everything.