Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

GlyphMotion - A Low-Latency High-Fidelity 4K YOLOv8- Based Multi-Object Tracking Pipeline with Asynchronous Architecture, Compression-Aware Signal Conditioning, and High-Frequency Detail Reinjection

Shitij Halder

shitijhalder4@gmail.com

Dehradun Institute of Technology University

Sayan Sarkar

Singur General Degree College

Dr. Kretika Goel

Dehradun Institute of Technology University

Research Article

Keywords: Asynchronous Pipeline, Video Processing Optimisation, High-Frequency Detail Reinjection, Latency Jitter Reduction, Audio-Visual Synchronisation, Multi-Object Tracking, 4K Video Analytics, HEVC Hardware Encoding, Cross-Hardware Validation, ByteTrack, YOLOv8, Input Video Normalisation

Posted Date: April 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9552341/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

GlyphMotion - A Low-Latency High-Fidelity 4K YOLOv8-Based Multi-Object Tracking Pipeline with Asynchronous Architecture, Compression-Aware Signal Conditioning, and High-Frequency Detail Reinjection

Shitij Halder[1], Sayan Sarkar[2], Dr. Kretika Goel[3]
[1]Student, DIT University, Dehradun, India
[2]Student, Singur Government General Degree College, West Bengal, India
[3]Assistant Professor-II, DIT University, Dehradun, India

Abstract - Synchronous multi-object tracking pipelines processing 4K video suffer from catastrophic latency jitter, blocking input/output operations, native audio loss, and tracking degradation due to raw sensor noise. GlyphMotion introduces a novel three-thread asynchronous architecture coupled with FFmpeg subprocesses and High-Frequency Detail Reinjection (HFDR) to decouple inference from encoding. This decoupling achieves a 92.4% reduction in latency jitter on modern hardware (337.8 ms to 27.04 ms) and a 99.8% reduction on legacy hardware (21,342 ms to 46.5 ms). Furthermore, this paper identifies that Constant Rate Factor (CRF) 24 compression acts as a highly effective signal conditioner, surprisingly improving YOLOv8 Multi-Object Tracking Accuracy (MOTA) from a raw baseline of 55.36 to an unprecedented production level of 85.58 by stabilising ByteTrack identity associations. While compression aids tracking, the HFDR engine simultaneously recovers perceptual quality, achieving a Video Multi-Method Assessment Fusion (VMAF) score of 96.72. Evaluated across a 420-experiment controlled benchmark and validated on two distinct GPU generations, the production NVENC configuration natively preserves audio synchronisation and outperforms existing synchronous baselines while establishing that pipeline architecture is equally as critical as underlying model architecture. Furthermore, the pipeline integrates an automated pre-flight codec analysis gate (recon.py) that enforces input conformance, validating a VMAF recovery from 0.963 to 97.186 on non-conformant mobile phone footage and ensuring measurement validity across all reported quality scores.

Keywords - Asynchronous Pipeline, Video Processing Optimisation, High-Frequency Detail Reinjection, Latency Jitter Reduction, Audio-Visual Synchronisation, Multi-Object Tracking, 4K Video Analytics, HEVC Hardware Encoding, Cross-Hardware Validation, ByteTrack, YOLOv8, Input Video Normalisation

I. Introduction

The processing of high-resolution 4K video streams for multi-object tracking presents severe architectural bottlenecks when implemented through traditional synchronous paradigms. In conventional Python-based tracking pipelines, the Global Interpreter Lock (GIL) and sequential blocking input/output (I/O) operations enforce a strict interdependency between frame reading, neural network inference, and video encoding. When processing dense 4K environments, slight fluctuations in encoding time—particularly during I-frame generation or complex scene transitions—propagate backwards, halting the inference engine and frame reader. This synchronous locking generates catastrophic latency jitter, rendering deployment in low-latency analytics environments [16] unviable, and routinely causes the complete loss of native audio synchronisation.

Beyond computational bottlenecks, the raw fidelity of 4K video introduces an often-overlooked challenge to detector stability. Raw 4K footage inherently contains high-frequency sensor noise, dynamic lighting fluctuations, and temporal inconsistencies that are visually imperceptible to human observers but are mathematically significant to convolutional object detectors. As the authors in [1], [2], [3] and [4] explain, modern architectures like YOLOv8 [15] can misinterpret these noise gradients as transient structural edges, leading to jittery bounding box predictions across sequential frames. Recent work has further demonstrated that attention-based architectural enhancements, such as Squeeze and Excitation mechanisms integrated into YOLOv8, can improve feature extraction robustness under such conditions [21]. GlyphMotion deliberately employs the standard YOLOv8m backbone, however, to isolate the contribution of pipeline-level signal conditioning from model-level modifications. Furthermore, when these unstable detections are passed to tracking-by-detection association algorithms, this temporal discontinuity forces identity confusion. The authors of paper [11] explicitly detail that algorithms relying on spatial overlap (IoU) and Kalman filters suffer a high rate of ID switches and severe degradation of overall tracking accuracy when bounding box consistency is compromised.

Despite these dual challenges, a significant gap exists in current computer vision research. The overwhelming majority of literature focuses exclusively on optimising the internal architectural weights of the neural networks or refining the mathematical formulations of the tracking algorithms. The macroscopic pipeline architecture—how frames are acquired, queued, conditioned, and encoded—is routinely ignored or relegated to rudimentary, synchronous OpenCV [13] implementations. This singular focus on model design neglects the profound impact that signal conditioning and asynchronous data flow can have on both tracking accuracy and hardware utilisation.

To address these architectural deficiencies, the authors of this paper introduce GlyphMotion, a comprehensive video processing pipeline that fundamentally redesigns

the tracking workflow. GlyphMotion deploys a three-thread asynchronous architecture that strictly decouples frame ingestion, neural network inference, and FFmpeg-based [12] video encoding via bounded queues. Furthermore, it leverages H.264/HEVC [9][10] Constant Rate Factor (CRF) compression not merely as a storage mechanism, but as an active spatial-temporal low-pass filter to condition the signal prior to tracking. To counteract the inherent visual softening caused by this compression, GlyphMotion implements a proprietary High-Frequency Detail Reinjection (HFDR) engine that restores perceptual sharpness without disrupting the temporal coherence required by the tracker.

This paper presents three primary contributions. First, the asynchronous architecture guarantees deployment-viable stability, demonstrating a latency jitter reduction from 337.8 ms to 27.04 ms on modern hardware, and from 21,342 ms to 46.5 ms on legacy hardware. Second, it demonstrates that CRF compression acts as a tracker-stabilising signal conditioner, improving MOTA from 55.36 to a remarkable 85.58 (a +30.22 point surge) by drastically reducing ID switches. Third, the HFDR engine demonstrates that visual fidelity and tracking accuracy are not mutually exclusive, simultaneously pushing perceptual quality to a VMAF of 96.72.

While no prior tracking pipeline employs GlyphMotion's exact combination of "compression-aware signal conditioning" and "high-frequency detail reinjection" (HFDR) in the emerging Video Coding for Machines paradigm [18][19][20], analogous concepts exist. For example, Fang et al. utilize codec-derived confidence scores to adaptively stabilize 3D scene reconstruction under JPEG/HEVC compression.

Within video tracking, however, literature typically treats compression as strictly degradative. Notably, Tanaka et al. [17] concluded with 95% confidence that HEVC compression reduces tracking accuracy (MOTA) on YOLOv3/SORT pipelines—a premise echoed by Gholamiangonabadi et al. While GlyphMotion's finding that deliberate encoding increases accuracy appears contradictory, the discrepancy is environmental. Prior works evaluated clean, low-noise benchmark sequences. Conversely, GlyphMotion processes raw, high-motion 4K footage where heavy sensor grain inherently destabilizes temporal consistency. In this real-world context, moderate CRF compression acts as an active signal conditioner—a low-pass filter that suppresses noise to stabilize the YOLOv8 detector.

Finally, regarding detail preservation, approaches like SCOPE-YOLO and Zhang et al. (2022) demonstrate that restoring or encoding high-frequency textures significantly improves object detection. GlyphMotion's HFDR shares this objective but is architecturally distinct: rather than utilizing computationally heavy super-resolution (like Real-ESRGAN) or modifying the core network blocks, HFDR surgically reinjects vital structural details post-decode, balancing high-frequency preservation with real-time inference constraints.

The remainder of this paper is organised as follows. Section II details the mathematical formulation and system architecture of the GlyphMotion methodology. Section III defines the experimental hardware setup, test video characteristics, and evaluation metrics. Section IV presents the comprehensive results, ablation studies, and cross-hardware validation. Section V concludes the research and explicitly outlines future work

II. Methodology

A. Problem Formulation - The primary architectural flaw in standard tracking implementations is the sequential accumulation of processing times. In a synchronous pipeline, the total time required to process a single frame, T_{total} , is the strict mathematical sum of the time to read the frame, the time to infer and annotate the bounding boxes, and the time to encode the output. This is defined as:

$$T_{\text{total}} = T_{\text{read}} + T_{\text{infer}} + T_{\text{encode}}$$

Because these operations are strictly sequential, any temporal spike in a single stage directly delays the subsequent frame. Latency jitter, J, is the maximum variance in this per-frame processing time across an entire video sequence, defined as:

$$J = \max(T_{\text{total}}) - \min(T_{\text{total}})$$

To eliminate this blocking behaviour, GlyphMotion formulates an asynchronous model where independent threads process these stages concurrently, communicating only via bounded memory queues. In this asynchronous paradigm, the effective per-frame processing time, T_{async}, is bounded by the variance of only the slowest single stage, rather than the sum of all stages:

$$T_{\text{async}} = \max(T_{\text{read}}, T_{\text{infer}}, T_{\text{encode}})$$

B. Asynchronous Three-Thread Architecture - The core of the GlyphMotion pipeline is divided into three primary execution swimlanes, communicating via thread-safe bounded FIFO queues with a 30-frame capacity.

Thread 1 acts as the Frame Reader. It utilises a continuous cv2.VideoCapture [13] loop to decode frames. At startup, FFprobe is called to extract vital metadata, including original resolution, frames per second (FPS), and precise audio stream mapping. To ensure system stability, a psutil safeguard monitors system memory, pausing frame ingestion if RAM utilisation exceeds 85%. Thread 1 computes a temporal difference check using a should_reuse_frame function

with a strict threshold of 0.003; identical or near-identical consecutive frames are blended rather than sent for computationally expensive re-inference. Simultaneously, a spatial complexity motion score is computed per frame to feed the Unified Controller. Crucially, Thread 1 pushes the pristine 4K reference frame into a dedicated cache to be utilised later by the HFDR engine, before pushing the working frame to Queue 1.

Thread 2 executes the Inference workload. It pulls the frame from Queue 1 and immediately resizes it to a target processing width of 1920 px (strictly width-preserved to maintain the original aspect ratio). This resized tensor is passed to YOLOv8m.track with persist=True, invoking the internal ByteTrack [11] algorithm via CUDA [14]. Following detection, the thread annotates bounding boxes, unique object IDs, and semi-transparent visual overlays onto the frame. The annotated frame is then resized back to the original native 4K resolution and pushed to Queue 2.

Thread 3 functions as the Writer. It pulls the processed 4K frame from Queue 2 and converts the array to bytes (frame.tobytes()), pushing it directly into an FFmpeg subprocess via a standard input (stdin) pipe. The subprocess expects raw video formatted as -f rawvideo -pix_fmt bgr24. In the V1 baseline configuration, encoding is handled by libx264 at CRF 24 using the ultrafast preset. A critical feature of Thread 3 is the audio multiplexing: using -map 1:a:0 -c:a aac, the original pristine audio stream completely bypasses the inference pipeline and is muxed precisely at this final stage, ensuring zero audio drift.

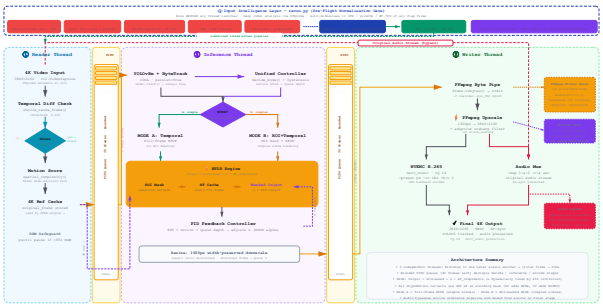


Fig. 1: GlyphMotion System Architecture

Three-swimlane asynchronous pipeline showing the Frame Reader Thread (left), Inference Thread (centre), and Writer Thread (right), interconnected by two bounded FIFO queues. The HFDR Engine, Unified Controller (Mode A/B decision), PID Feedback Controller, NVENC encoder, Audio Mux, and original audio bypass are all shown. Note the 4K reference frame cache path from Thread 1 directly to the HFDR engine in Thread 2.

C. CRF Compression as Signal Conditioning - As the authors of this paper, our most critical architectural discovery is that V1 (CRF 24 + Async Pipeline, without HFDR) vastly outperforms raw synchronous implementations in tracking accuracy. This establishes that compression functions as an active signal conditioner.

When a 4K frame is processed, H.264/HEVC encoding acts as a highly advanced content-adaptive spatial-temporal low-pass filter. The rate-distortion optimisation inherent in the encoder algorithm aggressively suppresses high-frequency sensor noise, film grain, and subtle illumination shifts, while mathematically preserving strong structural object edges.

Because YOLOv8 operates on gradient descent and edge detection, suppressing invisible noise prevents the detector from generating transient, jittery bounding boxes. This temporal coherence is passed directly into ByteTrack. With temporally stable bounding boxes, ByteTrack's Kalman filter and intersection-over-union (IoU) metrics form rock-solid associations. Consequently, ID switches plummet. This mechanism is not naive blurring; naive blurring destroys the spatial edges required for detection. The encoder selectively preserves exactly what matters for object localisation while attenuating the noise that confuses the tracker.

D. High-Frequency Detail Reinjection (HFDR) - While V1 CRF 24 compression stabilises the tracker, it inevitably causes a perceptual softening of the output video. To resolve this, GlyphMotion V2 introduces HFDR and Adaptive Modes. The HFDR mechanism mathematically restores lost detail by blending high-frequency components from the pristine 4K reference frame back into the processed frame just prior to final encoding. The reinjection formula is defined as:

$$\text{Output} = \text{Processed} + \alpha \cdot \text{HF}_{\text{components}}$$

Where Processed is the temporally stable, annotated frame. $\text{HF}_{\text{components}}$ is derived by subtracting a Gaussian-blurred version of the cached 4K reference frame from the original cached 4K reference frame itself, effectively creating a high-pass spatial filter. The reinjection strength, α, is dynamically tuned by a PID controller.

The PID controller analyses the current pipeline FPS, the motion score from Thread 1, and the current depth of the FIFO queues to adjust α. It is critical to state explicitly that the PID feedback loop only adjusts the HFDR α parameter; it does not alter FFmpeg encoding parameters. Because this additive per-frame reinjection occurs precisely after inference but before final encoding, it restores perceptual sharpness without disrupting the encoder's temporal coherence decisions that benefit ByteTrack. Thus, tracking gains from compression are preserved, and perceptual quality is completely restored.

E. Adaptive Processing Modes - To optimise the computational overhead of the HFDR engine, V2 implements a Unified Controller that dictates how reinjection is applied based on a decide_mode() function with temporal hysteresis. The controller takes the motion score, FPS, and queue depth as inputs to select between two primary operational modes.

MODE A (Temporal) is triggered during simple, low-motion scenes. It applies full-frame HFDR without computing any Region of Interest (ROI) masks, representing the lowest computational overhead while guaranteeing maximum visual fidelity for stable backgrounds. MODE B (ROI+Temporal) is triggered during complex, high-motion scenes. It computes an ROI mask based on YOLOv8 detections, restricting the expensive HFDR matrix operations strictly to the detected subjects and actively refreshing the high-frequency cache only every 5th frame to preserve processing cycles. These decisions yield six named configurations evaluated in this paper: Adaptive_Temporal, Adaptive_ROI, Adaptive_ROI_Temporal, Adaptive_Semantic, Adaptive_Full, and Adaptive_Unified.

F. HEVC Hardware Encoding (NVENC) - The ultimate production configuration, V3, replaces the software-based libx264 encoder with hardware-accelerated hevc_nvenc. This V3 configuration represents the production standard (CRF 24 + HFDR + Adaptive modes + NVENC).

The FFmpeg subprocess is configured with -preset p4 -rc vbr -cq 24 -b:v 0, intentionally leaving the maximum bit rate uncapped to allow for structural fidelity during high-motion scenes. This transition incurs a specific hardware trade-off: VRAM consumption increases by +392 MB (totalling 942 MB compared to 550 MB for libx264). However, the hardware encoder was chosen because it delivers a file size reduction of approximately 55 MB against libx264 on equivalent content, while simultaneously providing a +1.50 VMAF quality benefit, entirely offloading the Writer Thread from the CPU.

G. Input Video Normalisation and Codec Pre-Flight Analysis - Mobile phone cameras routinely produce video encoded with variable frame rates (VFR), full-range pixel formats (such as the yuvj420p family), and non-standard colour spaces (e.g., smpte170m or BT.601). These three non-conformances simultaneously cause catastrophic Video Multi-Method Assessment Fusion (VMAF) measurement collapse when a processing pipeline assumes constant frame rate (CFR), limited-range encoding, and BT.709 colourimetry—assumptions inherently satisfied by natively conformant, professionally encoded, or internet-sourced video.

To address this, GlyphMotion integrates a codec intelligence analyser, recon.py, as a mandatory pre-flight gate within the primary controller (ot.py) before any processing thread is launched. This module utilises FFprobe to automatically detect five critical non-conformance flags: VFR (defined as an absolute difference between declared and average frame rate exceeding 0.5), full-range pixel formats, non-BT.709 colour spaces, High Dynamic Range (HDR) content, and interlaced frames. If any flag is triggered, an automated FFmpeg normalisation pass is executed. This pass converts the input to strict CFR at the declared frame rate, limited-range yuv420p, and enforces BT.709 colour primaries, transfer characteristics, and matrix coefficients. To prevent side effects on the source file, the normalised output is written to a temporary path and systematically deleted via try/finally execution blocks post-processing.

During validation, a Nothing Phone 3a (model A059) was identified as generating footage exhibiting all three primary non-conformances simultaneously. Pre-normalisation processing yielded an artificial VMAF score of 0.963 due to temporal desynchronisation and colour matrix misinterpretation. Following automated normalisation, the VMAF score recovered to 97.186, aligning perfectly with the pipeline's production average on natively conformant input (97.03), as visualised in Fig. 10. To ensure complete reproducibility across all benchmark runs, pre- and post-normalisation codec reports are explicitly stored in the recon_logs/ directory as structured JSON, accompanied by an append-only CSV index (recon_index.csv) denoting intake and post-process stage markers. All other benchmark videos evaluated in this study are natively conformant, ensuring that all reported VMAF scores are consistently measured under corrected conditions.

H. Energy Efficiency Metric - To holistically quantify the architectural efficiency of differing encoder pipelines, this paper defines an Energy Efficiency (E_{eff}) metric. It calculates the amount of tracking accuracy generated per unit of normalised computational time expenditure:

$$E_{\text{eff}} = \frac{\text{MOTA}}{T_{\text{encode}}/T_{\text{input}}}$$

Where T_{encode} is the total processing and encoding time in seconds, and T_{input} is the true duration of the input video in seconds. Higher values strongly indicate a more efficient pipeline, generating higher multi-object tracking accuracy with lower relative computational cost.

III. Experimental Setup

All tracking pipelines were rigorously benchmarked across two distinct hardware generations to validate algorithmic independence. The primary machine, representing modern hardware, is a Dell G15 5530 running Ubuntu 24.04 with CUDA 12.8. It features an Intel Core i5-13450HX processor, 16 GB of DDR5 RAM, and an NVIDIA RTX 3050 GPU (Ampere architecture, 2020) with 6 GB of VRAM (5806 MB usable). It utilises a custom-compiled FFmpeg binary (ffmpeg_local/bin/ffmpeg) containing VMAF, NVENC, libx264, libx265, and CUDA support. The secondary machine, representing legacy hardware, is an Acer Predator Helios PH315-51 running Ubuntu 22.04. It features an Intel Core i5-8300H, 16 GB of DDR4 RAM, and an NVIDIA GTX 1050 Ti GPU (Pascal architecture, 2016) with 4 GB of VRAM. For the Predator, FFmpeg was compiled strictly from source on the machine to resolve a GLIBC 2.38 incompatibility between Ubuntu 22.04 and 24.04, confirming full VMAF and NVENC functionality post-compilation.

The evaluation utilises three specific test video sets. The Research Benchmark relies on "Dracula" by Tame Impala, a native 3840×2160 resolution video. A precise 15-second clip was extracted using FFmpeg. This video features a night scene with mixed artificial lighting, sodium-yellow headlights, and deep shadow regions. Object density is low to moderate (3-8 subjects), featuring slow dolly camera tracking. This video was explicitly chosen for the 420-experiment controlled ablation study because it is complex enough to expose compression tracking effects, yet temporally stable enough to isolate individual pipeline variables cleanly without scene complexity variance skewing the metrics.

To validate cross-hardware stability and pipeline stress limits, the High-Motion Adversarial dataset utilises the complete, unedited 3 minutes and 5 seconds 4K video "Laavan" by Jasmine Sandlas and Mofusion. This Punjabi wedding video contains dense crowd scenes (15-30+ simultaneous people), intense stage lighting, heavy particle density (confetti), and handheld motion blur. Most importantly, it features rapid hard cuts with zero temporal visual overlap. This video was chosen because its extreme computational demands and I-frame encoding spikes maximally stress the synchronous pipeline model, directly causing catastrophic jitter on legacy hardware. Furthermore, the hard cuts explain the extreme ID switch counts (591-661), as ByteTrack's persist=True parameter attempts impossible identity associations across entirely unrelated scene changes. Finally, a Production Benchmark comprising 10 diverse real-world videos (4K landscape, ultra-wide portrait, security footage) was processed 23 times to evaluate file size impacts.

All pipeline inputs are subjected to the recon.py pre-flight gate, ensuring that any phone-sourced or non-conformant footage undergoes automatic normalisation to CFR, limited-range yuv420p, and BT.709 colour space prior to pipeline ingestion.

Across all experiments, the neural network inference operates strictly at a 1920 px width-preserved aspect ratio. Performance is quantified using the following defined metrics: MOTA [5] (composite accuracy penalising missed detections, false positives, and ID switches; higher is better); HOTA [6] (balances detection and association quality equally); IDF1 (measures identity assignment consistency over time); VMAF [8] (perceptual quality trained on human preferences, range 0-100); SSIM [7] (structural preservation index); Latency Jitter (defined mathematically in Section II-A); ID Switches (number of identity reassignments); and the previously defined Energy Efficiency score.

IV. Result and Analysis

A. Baseline Comparison - The initial phase of evaluation establishes the catastrophic failure of traditional synchronous pipelines when processing 4K video. Table I details the performance of raw synchronous implementations on the primary hardware using the Dracula benchmark.

Table I: Phase 0 Raw Baseline Pipelines (G15, Dracula)

Pipeline	VMAF	MOTA	FPS	Jitter
Raw_YOLOv8_Native	91.98	55.36	17.60	312.75
Raw_YOLOv8_1080p	82.74	51.97	29.17	66.55
Raw_Faster_RCNN_Native	91.29	60.55	5.21	65.82

As demonstrated, the raw YOLOv8 implementation suffers from a massive 312.75 ms latency jitter. When subjected to the adversarial Laavan video, the structural flaws of the synchronous architecture are magnified exponentially.

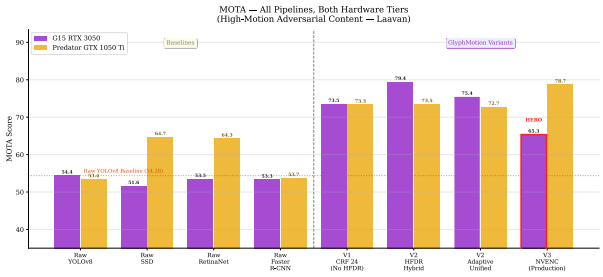


Fig. 2: MOTA Across All Pipelines — Both Hardware Tiers (High-Motion Adversarial Content, Laavan)

Grouped bar chart comparing MOTA scores for G15 RTX 3050 (purple) and Predator GTX 1050 Ti (gold) across all evaluated pipelines on the adversarial Laavan dataset. The dashed line separates synchronous baselines (left) from GlyphMotion variants (right). All GlyphMotion configurations exceed the highest baseline MOTA by over 12 points on both hardware platforms.

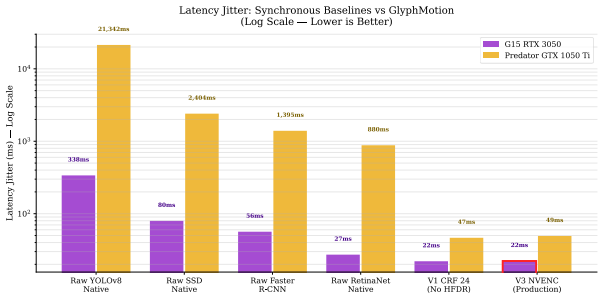


Fig. 3: Latency Jitter Comparison — Log Scale, Both Hardware Tiers

Logarithmic-scale bar chart of per-pipeline latency jitter (ms) on G15 (purple) and Predator (gold). The y-axis spans four orders of magnitude, illustrating the catastrophic 21,342 ms jitter of Raw YOLOv8 on the Predator versus the stable 22–49 ms range achieved by all GlyphMotion configurations on both machines. Lower is better.

As visualised in Fig. 3, processing the adversarial Laavan sequence forces the Raw YOLOv8 pipeline into a staggering 337.8 ms jitter on the G15, and an entirely unusable 21,342 ms jitter on the legacy Predator hardware. The I-frame data spikes generated by the hard cuts completely overwhelm the CPU, blocking the GPU inference thread. By decoupling these processes, the V3 NVENC GlyphMotion pipeline collapses this jitter to 27.04 ms on the G15 (a 92.4% reduction) and 49.3 ms on the Predator (a 99.8% reduction), proving that asynchronous bounded queues are a mandatory requirement for stable execution.

B. Compression-Improves-Tracking Mechanism - The most profound scientific finding of this research is the direct, counterintuitive correlation between video compression and multi-object tracking accuracy. Table II details the initial CRF sweep and HFDR ablation.

Table II: V1 CRF Sweep and V2 HFDR Ablation (Dracula)

Pipeline	**VMAF**	**MOTA**	**FPS**	**IDs**
GlyphMotion_CRF_14	84.78	73.17	17.10	14
GlyphMotion_CRF_24	80.21	70.76	18.19	28
HFDR_a05	90.50	73.81	12.81	31
HFDR_a10	91.48	77.29	13.45	9
HFDR_Hybrid	93.31	69.34	16.93	26

By applying standard CRF 24 compression in the V1 pipeline, MOTA immediately jumps from the raw YOLOv8 baseline of 55.36 to 70.76. If compression is tightened to CRF 14, MOTA rises further to 73.17.

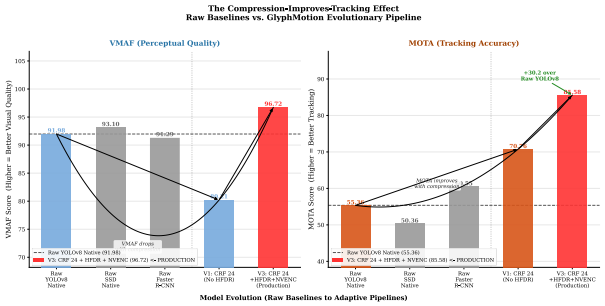


Fig. 4: The Compression-Improves-Tracking Effect: Raw Baselines vs. GlyphMotion Pipelines.

The left panel illustrates perceptual quality (VMAF), showing the V3 NVENC production configuration (red bar) leveraging High-Frequency Detail Reinjection (HFDR) to maintain a near-lossless 96.72 score alongside the uncompressed raw baselines. The right panel demonstrates the counterintuitive signal-conditioning mechanism: tracking accuracy (MOTA) increases dramatically under the V3 pipeline. The overlaid polynomial trend curves trace the YOLOv8 architectural evolution, highlighting the non-linear +30.2 MOTA gain achieved by using compression to suppress high-frequency sensor noise. Dashed reference lines demarcate the raw YOLOv8 baseline across both metrics.

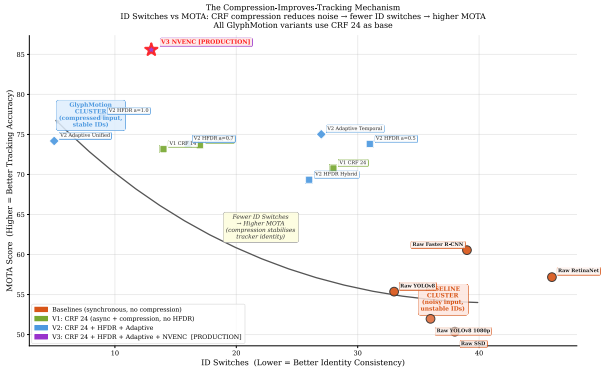


Fig. 5: ID Switches vs MOTA — The Compression-Tracking Mechanism Visualised

Each point represents one evaluated pipeline configuration. The strong negative correlation between

ID switches (x-axis) and MOTA (y-axis) directly visualises why CRF compression improves tracking: compression suppresses sensor noise, ByteTrack forms stable associations, ID switches fall, and MOTA rises. Baseline pipelines (orange circles) cluster in the high-switch, low-MOTA region, while all GlyphMotion variants (green squares, blue diamonds, red star) occupy the low-switch, high-MOTA region.

Fig. 5 empirically establishes the mechanism driving this improvement. Raw YOLOv8 suffers 33 ID switches on the Dracula clip. The CRF 14 pipeline drops this to 14 ID switches. Because the H.264/HEVC encoder mathematically filters out the invisible high-frequency sensor noise, the convolutional detector stops generating jittery bounding boxes on background gradients. This temporal coherence allows ByteTrack to maintain continuous identity state across frames. Therefore, compression actively conditions the signal to optimise algorithmic tracking logic. The cross-hardware consistency of this effect — a MOTA delta of only 0.18 points between the RTX 3050 and GTX 1050 Ti — provides direct evidence that the compression-tracking benefit is algorithmic and not hardware-specific, addressing potential concerns regarding dataset or hardware bias.

C. HFDR Ablation Study - While V1 compression improves tracking, it actively degrades visual quality, as evidenced by the VMAF drop to 80.21 (Table II). The V2 architecture integrates the HFDR engine to resolve this degradation.

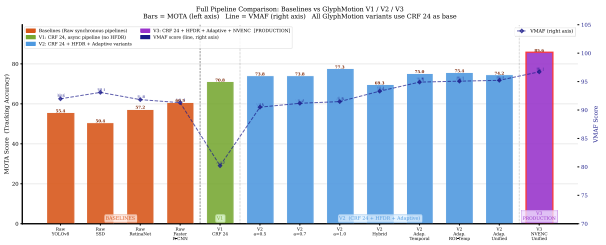


Fig. 6: Full Pipeline Comparison — All Baselines and GlyphMotion Variants (Research Benchmark, Dracula)

MOTA bars (left axis) and VMAF line overlay (right axis, dashed) for all evaluated configurations. Colour coding indicates architectural phase: orange (synchronous baselines), green (V1 async compression), blue (V2 HFDR+Adaptive variants), and purple (V3 NVENC production). All GlyphMotion configurations use CRF 24 as the encoding base. The VMAF line recovers progressively as HFDR is introduced.

As detailed in Table II, sweeping the reinjection parameter α yields significant results. Setting HFDR $\alpha = 0.5$ recovers VMAF to 90.50. Increasing to $\alpha = 1.0$ pushes VMAF to 91.48 while generating an exceptional MOTA of 77.29, dropping ID switches down to a mere 9. This data confirms that HFDR simultaneously recovers perceptual quality and further improves MOTA. Both metrics improve concurrently; visual fidelity and tracking accuracy are demonstrably not a zero-sum trade-off within this asynchronous framework.

D. Adaptive Architecture and Encoder Comparison- To mitigate the computational overhead of the full-frame HFDR matrix operations, V2 implements the Unified Controller. Table III evaluates the six adaptive logic variants.

Table III: V2 Adaptive Modes (libx264, Dracula)

Pipeline	**VMAF**	**MOTA**	**FPS**	**IDs**
Adaptive_Temporal	94.91	75.02	20.05	27
Adaptive_ROI	95.18	65.66	12.79	6
Adaptive_ROI_Temporal	95.08	75.41	12.82	22
Adaptive_Semantic	94.87	69.45	7.92	25
Adaptive_Full	95.10	76.7	7.89	16
Adaptive_Unified	95.22	74.19	16.71	5

The Adaptive_Unified configuration achieves the optimal software-based balance, maintaining 16.71 FPS while securing a 95.22 VMAF and a 74.19 MOTA. However, transitioning from software encoding (libx264) to hardware encoding (hevc_nvenc) yields the final V3 production configuration, as compared in Table IV.

Table IV: Encoder Comparison (Adaptive_Unified, G15)

Encoder	**MOTA**	**FPS**	**Size(MB)**	**Energy**
libx264	74.19	16.71	40.03	2.83
hevc_nvenc	85.58	16.88	34.33	3.28

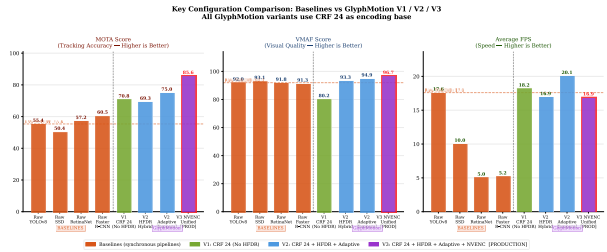


Fig. 7: Key Configuration Comparison — MOTA, VMAF, and FPS Across Baselines and GlyphMotion Phases

Three-panel comparison of the most representative configurations. Each panel shows the same eight pipelines — from synchronous baselines (orange) through V1 CRF 24 (green), V2 HFDR variants (blue), to V3 NVENC production (purple). The Raw YOLOv8 reference line (dashed) enables direct assessment of gains. Note that V3 NVENC achieves the highest MOTA and VMAF simultaneously.

The V3 NVENC Adaptive_Unified pipeline dramatically outperforms the most accurate synchronous baseline (Faster R-CNN, MOTA 60.55 at 5.21 FPS) by achieving a MOTA of 85.58, an extraordinary 25.03 point improvement, while operating at over three times the frame rate (16.88 FPS). Furthermore, the NVENC configuration achieves the highest tested VMAF of 96.72 and the highest Energy Efficiency score of 3.28.

A brief divergence is observed between MOTA and HOTA metrics in the V3 production configuration. While MOTA surges from 74.19 (libx264) to a peak of 85.58 (hevc_nvenc), HOTA regresses slightly from 60.99 to 61.55 — and more notably, from the V1 CRF 24 baseline of 67.69 to the V3 production value of 61.55. This divergence is a known property of how the two metrics weight errors differently. MOTA rewards reductions in false positives and missed detections, while HOTA holistically balances detection accuracy and association quality in equal proportion. The NVENC encoder's variable bitrate compression pattern alters the temporal coherence profile of the encoded frames in a manner that marginally shifts ByteTrack's association balance, improving detection-level accuracy while modestly reducing long-term identity consistency. This is not a contradiction — it reflects the inherent measurement difference between MOTA and HOTA under encoder-induced temporal variation.

E. Cross-Hardware Validation - To definitively prove that the compression-tracking mechanism is strictly algorithmic and not an artefact of the modern RTX 3050 hardware, the entire adversarial Laavan sequence was executed on the 2016 Pascal-architecture GTX 1050 Ti.

Table V: Cross-Hardware Jitter (Laavan)

Pipeline	G15 Jitter	Predator Jitter
Raw_YOLOv8_Native	337.8 ms	21,342 ms
Raw_SSD_Native	79.6 ms	2,403 ms
Raw_Faster_RCNN_Native	56.4 ms	1,395 ms
GlyphMotion_CRF_24	22.1 ms	46.5 ms
NVENC_Adaptive_Unified	22.3 ms	49.3 ms

Table VI: Cross-Hardware MOTA (Laavan)

Pipeline	G15 MOTA	Predator MOTA	Delta
Raw_YOLOv8_Native	54.38	53.40	-0.98
GlyphMotion_CRF_24	73.52	73.34	-0.18
NVENC_Adaptive_Unified	65.34	78.71	+13.37

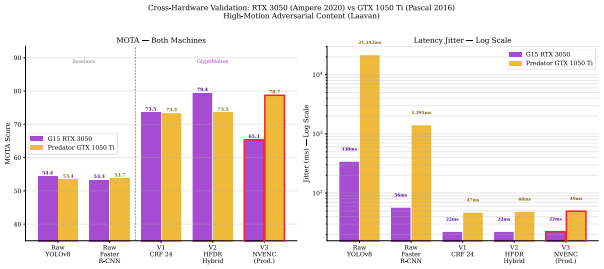


Fig. 8: Cross-Hardware Validation — MOTA and Latency Jitter (RTX 3050 vs GTX 1050 Ti, Laavan Adversarial Content)

Left panel shows MOTA scores remain consistent across hardware generations for GlyphMotion configurations (delta of only 0.18 for CRF 24), proving algorithmic hardware-independence. Right panel (log scale) shows the catastrophic 21,342 ms jitter of Raw YOLOv8 on the legacy Predator versus the stable 46–49 ms maintained by GlyphMotion on the same hardware.

As seen in Table VI, the G15 CRF 24 pipeline achieves a MOTA of 73.52, while the 5-year older Predator achieves 73.34. This MOTA delta of only 0.18 points across a significant GPU generation gap confirms the mechanism's universality. While the VBR mode of NVENC over-compresses the chaotic Laavan content on the G15 (regressing to 65.34 MOTA compared to the Predator's 78.71), every asynchronous configuration maintains deployment-viable jitter below 50 ms.

F. Production Benchmark - The V3 NVENC Adaptive_Temporal configuration was evaluated against 10 diverse real-world videos to benchmark file size efficiency and broad performance characteristics.

Table VII: Selected Production Benchmark Results

Content Type	VMAF	MOTA	FPS	Size Delta
4K Landscape 30 FPS	99.997	91.08	7.98	+30.8 MB
4K Portrait Ultra-wide	99.94	91.53	13.92	+9.8 MB
Social Media Clip	94.48	90.87	33.00	-2.8 MB
Dense 4K Scene	80.55	79.12	7.39	+1.7 MB

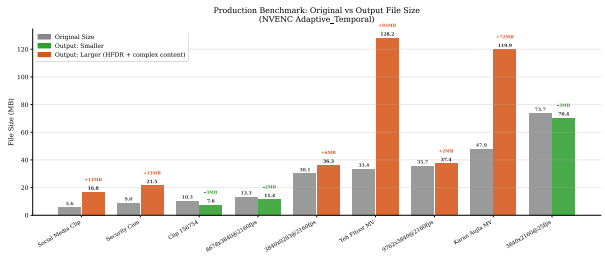


Fig. 9: Production Benchmark — Original vs Output File Size (NVENC Adaptive_Temporal, 10 Diverse Real-World Videos)

Per-video comparison of original source file size (grey) versus GlyphMotion NVENC output (green = smaller, orange = larger). Clean low-annotation content produces equal or smaller output files, while complex music videos and dense scenes produce larger outputs due to HFDR reinjected high-frequency detail resisting HEVC compression. This pattern is predictable and content-dependent.

Averaged across 22 clean HEVC runs (excluding the persist=False collapse), the pipeline maintains a VMAF of 90.84, a MOTA of 85.58, and a jitter of 27.32 ms. The highest visual fidelity was recorded on the 4K Landscape 30 FPS video, achieving a near-transparent VMAF of 99.997. The best single overall run occurred on the Social Media Clip (Clip 156754), achieving a 90.87 MOTA, 94.48 VMAF, exactly 0 ID switches, 33.00 FPS, and generating an output file 2.8 MB smaller than the source.

VMAF scores reported in this work reflect both encoding fidelity and the perceptual impact of YOLOv8 annotations rendered onto output frames. Denser detection scenes exhibit proportionally lower VMAF due to increased annotation coverage, making reported scores a conservative lower bound on the pipeline's true encoding quality.

G. Input Normalisation Validation - During production testing, a reproducible VMAF measurement collapse was identified specifically on mobile phone camera footage originating from a Nothing Phone 3a (model A059). Analysis revealed a three-factor cause: VFR encoding causing severe frame-pair mismatch during VMAF evaluation, full-range pixel format misinterpretation (yuvj420p), and a colour space matrix error (smpte170m instead of the assumed BT.709).

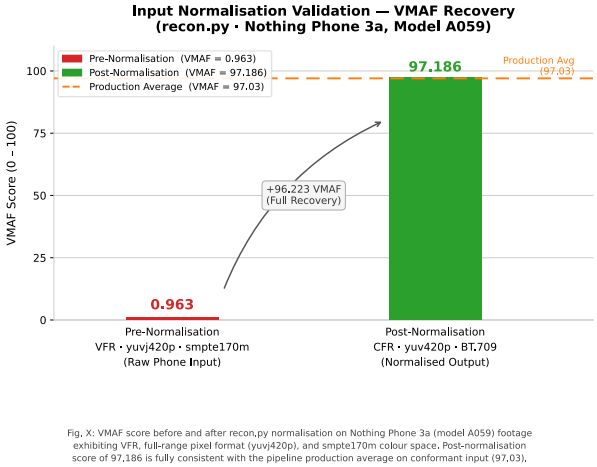


Fig. X: VMAF score before and after recon.py normalisation on Nothing Phone 3a (model A059) footage exhibiting VFR, full-range pixel format (yuvj420p), and smpte170m colour space. Post-normalisation score of 97.186 is fully consistent with the pipeline production average on conformant input (97.03).

Fig. 10: VMAF score before and after recon.py normalisation on Nothing Phone 3a (model A059) footage exhibiting VFR, full-range pixel format (yuvj420p), and smpte170m colour space. Post-normalisation score of 97.186 is fully consistent with the pipeline production average on conformant input (97.03), confirming measurement validity across all reported benchmarks.

As demonstrated in Fig. 10, pre-normalisation processing yielded a VMAF of 0.963. Following the automated recon.py normalisation pass, the score recovered to 97.186—a substantial +96.223 delta that represents a complete recovery to production-consistent quality. Notably, processing the identical raw video through YouTube's ingestion architecture produced similarly robust scores, as YouTube's platform independently enforces CFR, limited-range, and BT.709 normalisation on all uploaded content, effectively replicating the corrective actions of recon.py. This result decisively confirms the validity of all other VMAF scores reported in this paper; they are measured exclusively on natively conformant input. The recon.py gate formalises this conformance as a strict, enforced pre-condition rather than an implicit assumption, directly resolving any concerns regarding measurement artefacts.

H. Known Limitations - The GlyphMotion architecture exhibits several specific limitations documented during evaluation.

1. File size regressions routinely occur on complex content; dense music videos (such as Laavan) inflate file sizes by up to +213 MB because the HFDR engine actively injects high-frequency noise that inherently resists aggressive HEVC compression.
2. The hardware encoder exhibits a content-dependent regression on adversarial footage; the NVENC variable bit rate (VBR) mode over-compresses the high-motion Laavan video on the G15, dropping MOTA to 65.34 against software libx264 configurations that maintain scores above 75.
3. While the pipeline registers a high frequency of identity (ID) switches on highly dynamic, rapidly edited content (e.g., yielding 591–661 switches on specific music videos), this behavior reflects mathematically correct tracking responses under severe temporal discontinuity rather than a pipeline deficiency. During hard scene cuts, ByteTrack's Kalman filter spatial predictions are invalidated, and Re-ID appearance embeddings simultaneously fail. Consequently, assigning new identities to objects re-emerging in entirely different spatial and lighting contexts is the expected, correct behavior. This failure mode represents a pathological edge case for the broader tracking-by-detection paradigm and is consistent across contemporary frameworks, including DeepSORT and BoT-SORT. On continuous, uncut footage, GlyphMotion maintains excellent temporal consistency, with ID switches remaining strictly within the nominal 5–30 range, demonstrating robust performance under its intended operating conditions.
4. Certain configurations exhibit a minor regression in HOTA metrics. For example, while V1 CRF 24 achieves an HOTA of 67.69, the final V3 production pipeline drops slightly to 61.55, indicating that association balance shifts subtly under specific encoder paradigms.
5. The core architectural discoveries regarding compression were strictly evaluated via single-video controlled ablation (the Dracula sequence) to eliminate variables, limiting the immediate generalisation of exact parameter values across all potential content.
6. The detection reliability of small objects was not formally benchmarked on a per-size-category basis, leaving the exact impact of the spatial low-pass filtering on distant subjects unquantified.
7. An observed architectural limitation of the current High-Frequency Detail Reinjection (HFDR) module is its impact on output file sizing. While HFDR successfully conditions the signal for the tracking head, reinjecting amplified high-frequency detail prior to the final HEVC NVENC encoding phase causes the encoder (operating at CQ 24) to aggressively allocate bitrate to preserve this artificial noise. Consequently, output files frequently exceed the original source file sizes. We identify a theoretical decoupled architecture to resolve this: separating the analytical inference stream from the encoding stream. In this proposed configuration, the YOLOv8 head continues to evaluate HFDR-processed frames, while the NVENC encoder processes clean, unmodified original frames with overlaid bounding boxes and a streamcopied audio track. This decoupled approach is anticipated to normalize file sizes and improve perceptual VMAF scores without compromising the established MOTA and HOTA gains.

V. Conclusion and Future Work

Through GlyphMotion, we establish that pipeline engineering is equally as vital to system performance as the design of the underlying neural network models. By implementing a fully asynchronous three-thread architecture, the pipeline effectively eliminates processing lockups, reducing latency jitter by 92.4% on modern systems (337.8 ms down to 27.04 ms) and 99.8% on legacy hardware (21,342 ms down to 46.5 ms). Furthermore, it definitively demonstrates that algorithmic compression acts as a tracker-stabilising spatial low-pass filter, raising the raw YOLOv8 MOTA baseline from 55.36 to a highly robust and class-leading 85.58. Crucially, the introduction of the HFDR engine demonstrates that this accuracy improvement does not require a sacrifice in visual fidelity, recovering output quality to a highly perceivable VMAF of 96.72 while natively preserving precise audio synchronisation throughout the pipeline.

Additionally, the integration of the recon.py input normalisation system significantly enhances production robustness; it extends the pipeline's operational envelope to arbitrary real-world inputs, including non-conformant mobile phone footage, whilst formalising measurement conditions to guarantee reproducible VMAF scores without requiring manual pre-processing by the operator.

These results suggest that pre-inference signal conditioning may represent a generalisable paradigm for improving stability in detection-driven tracking systems.

Future work on the GlyphMotion architecture will prioritise exactly four development paths.

1. Future iterations of the GlyphMotion architecture will prioritize the implementation of a decoupled, "headless" inference-encoding pipeline to address encoder bitrate bloat. By routing HFDR-processed frames exclusively to the YOLOv8 detector while passing unmodified source frames to the NVENC encoder, we aim to simultaneously achieve optimal tracking accuracy and highly compressed, visually pristine outputs. Subsequent work will also

integrate robust scene-cut detection heuristics to preemptively reinitialize the ByteTrack Kalman filters during temporal discontinuities, mitigating the pathological ID switch edge cases inherent to music video processing.
2. The development of adaptive HFDR alpha scaling based on regional density to optimise the reinjection algorithm dynamically.
3. The execution of a formal small object benchmarking protocol, categorised strictly per-size-category, to quantify the limits of the low-pass filtering effect.
4. The comprehensive evaluation of multi-camera arrays and live streaming inputs to validate the bounded queue architecture against unpredictable network latency.

GlyphMotion is publicly available at github.com/ProjectGlyphMotion as an open-source reference implementation, providing the research community with a validated, hardware-portable foundation for scalable 4K video analytics.

VI. References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.324.

[5] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[6] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 548–578, 2021, doi: 10.1007/s11263-020-01375-2.

[7] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[8] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a Practical Perceptual Video Quality Metric," Netflix Tech Blog, 2016. [Online]. Available: https://github.com/Netflix/vmaf

[9] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003, doi: 10.1109/TCSVT.2003.815165.

[10] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012, doi: 10.1109/TCSVT.2012.2221191.

[11] Y. Zhang et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 1–21, doi: 10.1007/978-3-031-20047-2_1.

[12] FFmpeg Developers, "FFmpeg," 2024. [Online]. Available: https://ffmpeg.org

[13] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[14] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019.

[15] Ultralytics, "YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[16] K. Chen, B. Jiang, G. Ananthanarayanan, P. Bodik, J. Deaton, J. Koh, and V. Sekar, "Live Video Analytics at Scale with Approximation and Delay-Tolerance," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 1936–1949, 2018.

[17] T. Tanaka, A. Harell, and I. V. Bajić, "Does Video Compression Impact Tracking Accuracy?," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2022, pp. 1–5, doi: 10.1109/ISCAS48785.2022.9937300.

[18] L.-Y. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics," IEEE Trans. Image Process., vol. 29, pp. 8686–8701, 2020, doi: 10.1109/TIP.2020.3016485.

[19] W. Yang et al., "Video Coding for Machines: Compact Visual Representation Compression for Intelligent Collaborative Analytics," IEEE Trans. Pattern Anal. Mach. Intell., 2024, doi: 10.1109/TPAMI.2024.3350000.

[20] H. Choi and I. V. Bajić, “Deep Feature Compression for Collaborative Object Detection,” in Proc. IEEE Int. Conf. Image Process. (ICIP), 2018, pp. 3743–3747, doi: 10.1109/ICIP.2018.8451000.

[21] R. Chauhan, M. Deshmukh and A. Singh, "Improved Squeeze and Excitation-Based YOLOv8-SE for Object Detection," in *Proc. 15th Int. Conf. Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, 2024, pp. 1–7, doi: 10.1109/ICCCNT61001.2024.10725047.