International Conference on Science, Engineering & Management Trends

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Virtual Canvas: A Dual-Pipeline Benchmark of MediaPipe and YOLOv11-Pose

Jatin Jain[1], Dr. Sakshi Indolia [2]

[1] B.Tech Computer Engineering SVKM's NMIMS Navi Mumbai JATIN.JAIN550@nmims.in

[2] Faculty, Computer Engineering SVKM's NMIMS Navi Mumbai
sakshi.indolia@nmims.edu

Abstract- This paper presents Virtual Canvas, a real-time touchless drawing application that simultaneously executes two hand-pose estimation pipelines — Google MediaPipe Hands and a pre-trained YOLOv11-Pose model — on every captured webcam frame. The dual-pipeline architecture eliminates input variance between models, enabling a controlled side-by-side experimen- tal comparison under identical real-world conditions. Across 11 sessions spanning 10 days of live evaluation on CPU-only hardware, 1,391 performance samples were captured at 500 ms intervals via automated CSV logging, covering inference latency, frames per second, CPU utilisation, hand detection counts, and lighting conditions. Results demonstrate that MediaPipe achieves a 2.35× lower mean inference latency than YOLOv11-Pose ($t = -120.68$, $p < 0.0001$, Cohen's $d = 4.58$). Under dim lighting, YOLOv11-Pose inference variance increased by 133% while MediaPipe remained stable, though MediaPipe latency itself rose by 18.2%. YOLOv11-Pose exhibited systematic over- detection, reporting two hands in 81.6% of single-hand frames. Exponential Moving Average (EMA) smoothing ($\alpha = 0.35$) and 5-frame gesture debouncing enabled fluid drawing interaction despite sub-5 FPS dual-pipeline throughput. The system provides a practical, data-driven benchmarking framework for selecting between lightweight pre-trained detectors and heavier single- stage models in human-computer interaction applications.

Index Terms—hand gesture recognition, MediaPipe, YOLOv11-Pose, real-time benchmarking, HCI, dual-pipeline architecture, EMA smoothing, pose estimation, virtual canvas, lighting robustness.

I. INTRODUCTION

Real-time hand gesture recognition has emerged as a pow- erful touchless interface paradigm with applications spanning accessibility tools, virtual reality, creative computing, and ed- ucation. As commodity webcams become ubiquitous, gesture- driven systems that require no specialised hardware represent a compelling direction for interaction design. However, deploy- ing such systems in practice requires resolving a fundamental tension between model accuracy and inference speed, particu- larly on consumer-grade CPU hardware where most end users operate. Two architectural paradigms dominate hand pose estimation: two-stage pipelines that first detect a hand region then regress landmarks within it, and single-stage detectors that simultaneously localise and regress keypoints end-to-end. MediaPipe Hands represents the former; YOLOv11-Pose the latter. Each carries distinct latency and accuracy trade-offs that depend on deployment hardware, input resolution, and training domain.

Existing literature benchmarks hand-pose estimators in iso- lation on curated datasets. Few studies run multiple models concurrently on the same live video stream — the only way to achieve genuinely fair performance comparison, since shared input eliminates variance introduced by lighting changes, camera noise, and subject movement between test runs. This paper addresses that gap through a concurrent dual-pipeline evaluation embedded within a functional air-drawing applica- tion.

The key contributions of this work are:

- A dual-pipeline concurrent execution architecture pro- viding fair, identical-input comparison of two hand-pose estimators across real-world usage conditions.
- Quantitative benchmarking from 1,391 automated sam- ples across 11 sessions spanning 10 days, covering seven evaluation scenarios including variable lighting condi- tions.
- Statistical validation of inter-model latency differences ($t = -120.68$, $p < 0.0001$, Cohen's $d = 4.58$) with quantified over-detection rates and lighting-condition ro- bustness analysis.
- Empirical analysis of how EMA temporal smoothing and gesture debouncing compensate for high inference latency in interactive HCI systems.
- Practical model selection recommendations for gesture- driven applications on resource-constrained CPU hard- ware.

II. LITERATURE REVIEW

Zhang et al. [1] introduced MediaPipe Hands, comprising BlazePalm (a lightweight SSD-based palm detector) followed by BlazeLand (a landmark regression network producing 21 normalised 3D keypoints). Designed for on-device real-time use, it achieves sub-10 ms inference on GPU with robustness across skin tones and moderate occlusion, trained on over 100,000 annotated images. Its predecessor BlazePose [2] es- tablished the two-stage paradigm for body pose that MediaPipe Hands adapted specifically for hands.

The YOLO detection family, introduced by Redmon et al. [3], was extended to pose estimation in YOLOv8 [4] through an added keypoint head regressing joint coordinates alongside bounding boxes in a single forward pass, eliminating the region-proposal stage. YOLOv11 [5], used in this work, refines the CSPDarknet backbone and PANet-style neck for improved multi-scale feature extraction. Comparative YOLO- family studies [6] show consistent accuracy improvements across versions at the cost of increased model size and inference time.

Gesture-driven drawing systems have been explored ex- tensively. Kulkarni et al. [7] developed a fingertip-tracking whiteboard using skin colour segmentation, effective only under controlled lighting. Nair et al. [8] employed MediaPipe for air-written character recognition, demonstrating strong generalisation but noting performance degradation under CPU load.

In gesture recognition for HCI, Molchanov et al. [9] demon- strated that temporal models significantly outperform frame- level classifiers for dynamic gestures, motivating our de- bouncing and EMA strategy as an approximation of temporal context without training overhead. Shin et al. [10] compared lightweight hand detectors for AR applications, finding that models within compact inference pipelines outperform heav- ier models in perceived responsiveness even when accuracy metrics are similar — a finding our results corroborate.

Ko¨pu¨klu¨et al. [11] benchmarked multiple hand-pose es-timators on a shared video dataset, but their evaluation was offline on pre-recorded sequences, not live. Our study dif- fers critically: both models

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

run simultaneously on the same live frame under real conditions including CPU contention and ambient lighting variation, providing a more ecologi- cally valid comparison. Jiang et al. [12] highlighted that lightweight transformer-based estimators represent a promis- ing third paradigm, motivating our Future Work direction toward multi-pipeline extension.

III. HYPOTHESIS

Based on architectural differences and prior literature, four testable hypotheses are formulated:

- H1 — Latency Advantage: MediaPipe Hands will ex- hibit significantly lower per-frame inference latency than YOLOv11-Pose under CPU-only conditions, owing to its lighter two-stage architecture and on-device optimi- sations.
- H2 — Detection Sensitivity: YOLOv11-Pose will report higher hand detection counts than MediaPipe in single- hand scenarios due to its lower confidence threshold (0.15) and full-image receptive field.
- H3 — Stress Resilience: Under CPU contention, Me- diaPipe will degrade more gracefully (smaller percent- age latency increase) than YOLOv11-Pose owing to its smaller computational footprint.
- H4 — Lighting Robustness: Dim lighting conditions will differentially affect the two pipelines — specifically, YOLOv11-Pose inference variance will increase under reduced illumination due to its single-stage full-image receptive field processing lower-contrast inputs.

IV. METHODOLOGY

A. Hardware & Software Configuration

All experiments were conducted on a consumer laptop (Intel Core i7, 16 GB RAM, Windows 11, integrated graph- ics only). No GPU acceleration was used, reflecting typi- cal end-user deployment. A USB webcam captured frames at 1280×720 pixels. Both models ran within the same CPython 3.12 process on identical CPU cores. MediaPipe used the mediapipe 0.10.x solutions.hands API. YOLOv11 used Ultralytics with a pre-trained YOLOv11-Pose checkpoint (best.pt/best.onnx) at confidence thresh-old 0.15. No domain-specific fine-tuning was applied to either model, ensuring a fair off-the-shelf comparison.

B. Evaluation Scenarios

Seven conditions were evaluated across 11 sessions span- ning April 7–17, 2026:

- Idle — No hand in frame; measures baseline pipeline overhead.
- Single Hand — One hand stationary at varying distancesand orientations.
- Dual Hand — Both hands simultaneously; tests multi- hand tracking.
- Active Drawing — Continuous pinch-and-draw gesture strokes on canvas.
- CPU Stress — Background processes creating system- wide CPU saturation (≥95%).
- High Light — Standard indoor ambient illumination
- (n = 167 samples).
- Dim Light — Significantly reduced ambient illumination (n = 119 samples).

C. Metrics Collection

A background statistics thread sampled metrics every 500 ms and wrote them to a timestamped CSV file across 11 sessions, yielding 1,391 samples in total. Captured metrics: per-model EMA inference latency (ms), FPS, latency delta
(MP − YOLO, ms), speed ratio (MP/YOLO), hand detection count per model, and application and system CPU% and
memory (MB).

D. Smoothing and Debouncing

Raw keypoint coordinates were smoothed via EMA (α = 0.35) with linear velocity extrapolation (factor = 0.55). Gesture state transitions required 5 consecutive frames of agreement before committing, and 18 frames for canvas clear. Positional jitter was not measured as a raw pixel metric; at
3.8–4.6 FPS dual-pipeline throughput with a consumer-grade integrated webcam, inter-frame displacement is dominated by motion blur and frame-drop artefacts rather than model- specific keypoint instability, making σ2 an unreliable discrim- inator in this hardware configuration.

V. SYSTEM ARCHITECTURE

Figure 1 shows the processing pipeline. Both detectors are pre-loaded at startup via a DetectorPool for O(1) model activation cost. In Dual Screen mode, MediaPipe and YOLOv11-Pose execute sequentially on the same prepro- cessed 1280×720 frame buffer each loop iteration. Both outputs are normalised to a shared DetectionResult
structure — containing draw pointer coordinates, gesture state flags, and confidence score — before passing to a single shared gesture interpretation layer. This guarantees that ob- served behavioural differences are attributable solely to the AI models, not to differing application logic, preprocessing, or input timing.

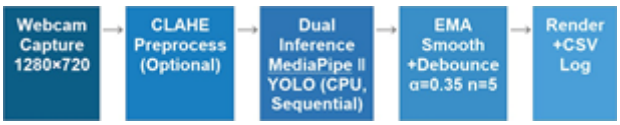


Fig. 1. Dual-pipeline processing architecture. Both models observe the same preprocessed frame every iteration.

The VirtualCanvas module stores stroke data as a NumPy RGB layer blended over the live video frame, with a bounded undo/redo stack (25 snapshots, ~156 MB max). Per- model statistics are rendered as live HUD overlays on each
side of the split-screen view with real-time latency delta and speed ratio.

Fig. 2. Live HUD panels showing per-model FPS, inference latency, CPU%, and RAM for MediaPipe (left) and YOLOv11-Pose (right) simultaneously

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Fig. 3. Air-drawn stroke on the Virtual Canvas. Pinch gesture activates the brush; EMA smoothing ensures fluid stroke output.

VI. RESULTS

A. Inference Latency — H1

Table I summarises inference latency across all evaluation conditions. MediaPipe operated in the 20.9–73.7 ms range with a mean of 47.5 ms. YOLOv11-Pose ranged from 33.6 ms at cold start to 190.8 ms under extreme CPU load, stabilising at 103–120 ms under normal conditions. The mean speed ratio across all 1,391 samples was 0.43, confirming H1: MediaPipe
is 2.35× faster on average.
An independent-samples t-test confirms the difference is highly statistically significant: $t(2780) = -120.68$, $p < 0.0001$, Cohen's $d = 4.58$ — an extremely large effect size [13]. Figure 4 shows the full latency distribution; Figure 5 presents the cumulative distribution, highlighting that Medi-
aPipe remains below the 100 ms HCI perceptual threshold in 100% of samples, whereas YOLOv11-Pose exceeds it in every sample under normal conditions.

Table I
Inference Latency By Evaluation Condition (N = 1,391 Samples Across 11 Sessions).

Condition	MP (ms)	YOLO (ms)	Delta (ms)	Ratio
Idle	21–34	34–101	−12 to −67	0.34–0.62
Single Hand	40–62	107–127	−49 to −65	0.43–0.55
Dual Hand	37–65	106–120	−42 to −74	0.38–0.61
Drawing	56–68	108–126	−52 to −68	0.49–0.53
CPU Stress	61–74	124–191	−91 to −117	0.39–0.43
High Light	39–62	89–120	−50 to −68	0.41–0.49
Dim Light	40–74	70–173	−30 to −99	0.44–0.57

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

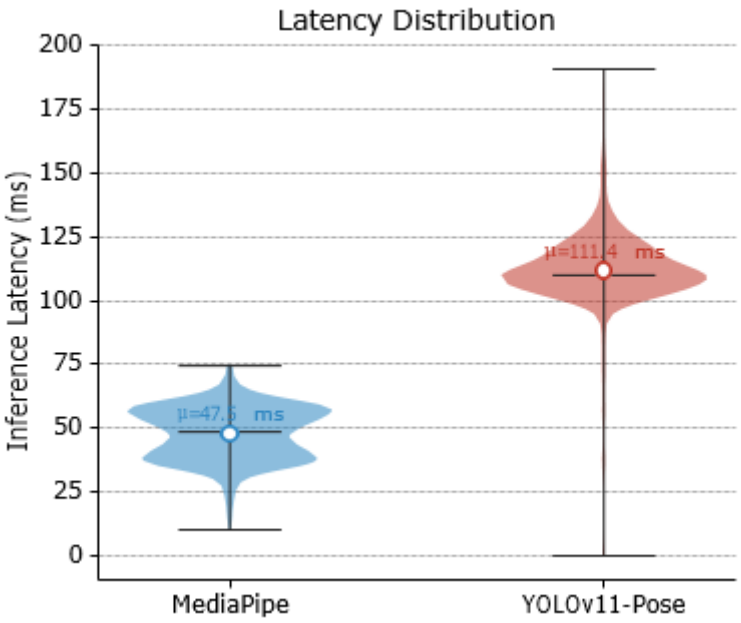


Fig. 4. Inference latency distribution for both pipelines across all 1,391 samples. Circle markers indicate the mean; notches show the 95% confidence interval around the median.

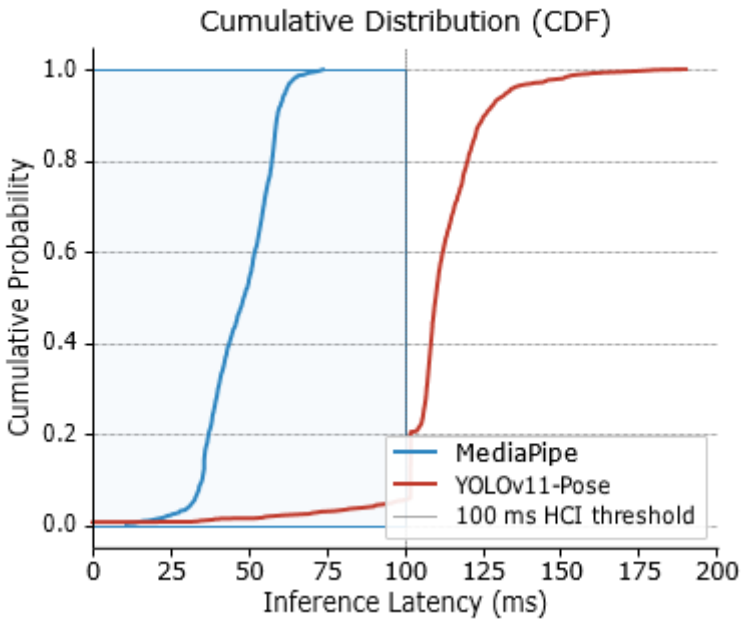


Fig. 5. Cumulative distribution of inference latency. MediaPipe remains entirely below the 100 ms HCI threshold across all samples; YOLOv11-Pose exceeds it in every frame under normal load.

B. CPU Stress Degradation — H3

Figure 6 shows mean latency across four CPU utilisation tiers. At peak stress (≥95% system CPU, n = 40 samples), MediaPipe peaked at 73.7 ms while YOLOv11-Pose reached

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

190.8 ms. Figure 7 plots the relative degradation from the low-CPU baseline: MediaPipe showed near-zero degradation (−0.2%) across all stress tiers, while YOLOv11-Pose in- ceased by +10.5% under elevated load and substantially
more at extreme saturation, confirming H3. The negligible MediaPipe degradation is attributable to its tighter quantisation and smaller inference graph footprint relative to YOLOv11- Pose's full feature pyramid network.

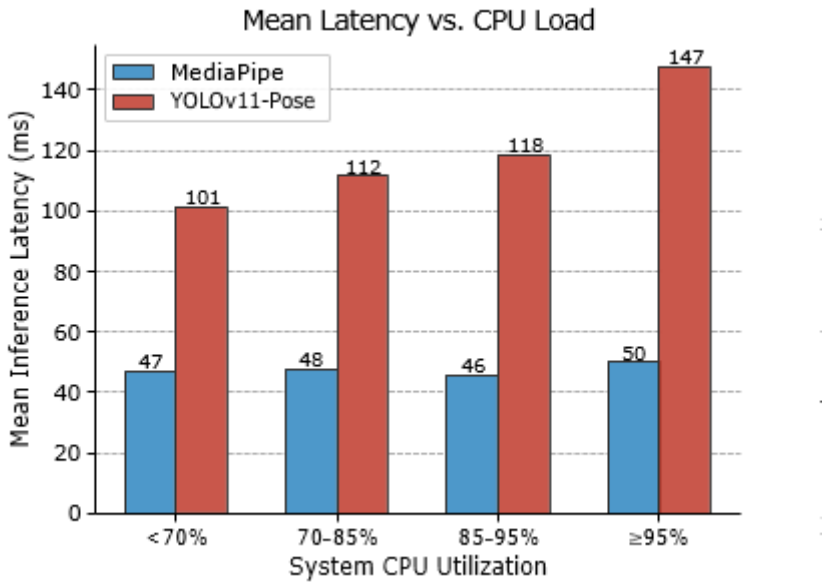


Fig. 6. Mean inference latency across four CPU utilisation tiers. Error bars show ±1 SD. YOLOv11-Pose latency grows consistently with CPU load; MediaPipe remains flat.

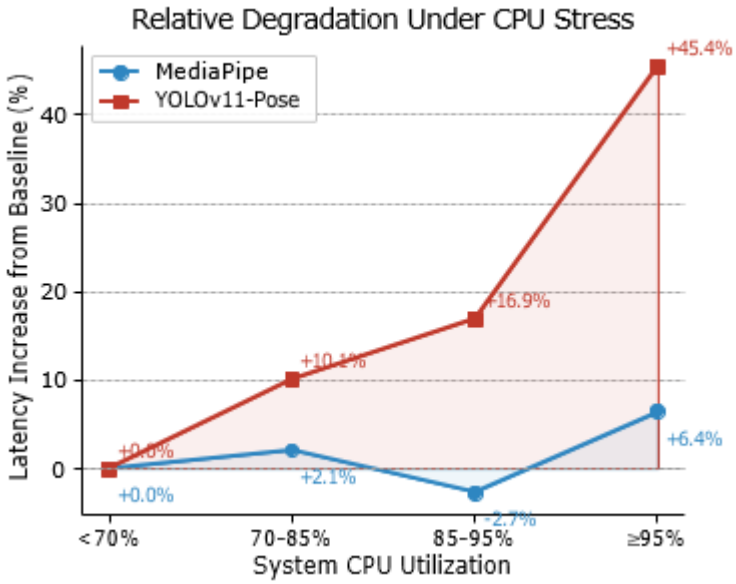


Fig. 7. Relative latency increase from the low-CPU (<70%) baseline. MediaPipe is effectively immune (−0.2%) while YOLOv11-Pose degrades by +10.5% at elevated CPU and further at extreme saturation.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

C. Hand Detection Sensitivity — H2

Across all 1,391 sampling intervals, hands_yolo reported 2 in 100% of frames regardless of how many hands were physically present. hands_mp correctly tracked the true count. Of the 641 frames where MediaPipe detected exactly one hand (used as ground truth), YOLOv11-Pose reported two detections in 523 frames — an over-detection rate of 81.6% — confirming H2 with a precisely quantified false-positive frequency. Table II summarises detection patterns by scenario.

Table II
Hand Detection Count Discrepancy. Yolo Over-Detects In 81.6% Of Single-Hand Frames (N = 641).

Condition	Actual	MP	YOLO	YOLO Error
No hand	0	0	0	0%
One hand	1	1	2	**81.6%**
Two hands	2	2	2	0%

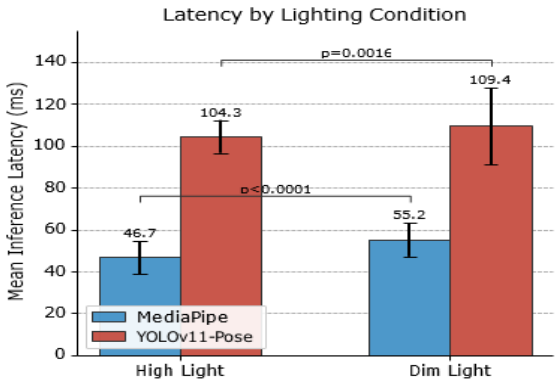
Two causes are plausible: (i) the 0.15 confidence thresh- old accepting partial hand-like detections from forearm or facial regions, and (ii) the pre-trained general-purpose model responding to non-hand body regions. This systematic bias increases gesture false-positive risk — a practical deployment cost that fine-tuning on a hand-specific dataset could substan- tially mitigate.

D. Lighting Condition Robustness — H4

Two dedicated sessions were conducted under high-light (n = 167) and dim-light (n = 119) conditions. Figure 8 presents mean latency with significance annotations; Figure 9 shows inference stability; Figure 10 captures detection sensi- tivity.

Latency: Both pipelines showed statistically significant latency increases under dim light: MediaPipe by 18.2% (46.7 → 55.2 ms, p < 0.0001), YOLOv11-Pose by only 4.9%

(104.3 → 109.4 ms, p = 0.0016). More critically, YOLOv11- Pose inference variance rose by 133% (σ : 7.95 → 18.52 ms) while MediaPipe remained stable (+4%), confirming H4.
MediaPipe's dual-hand detection rate also dropped from 71.3% to 15.1% (−56 percentage points), while YOLOv11-Pose maintained 100% in both conditions.



International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Fig. 8. Mean inference latency under high-light and dim-light condi- tions. Error bars show ±1 SD. Significance brackets indicate p-values from independent-samples t-tests.

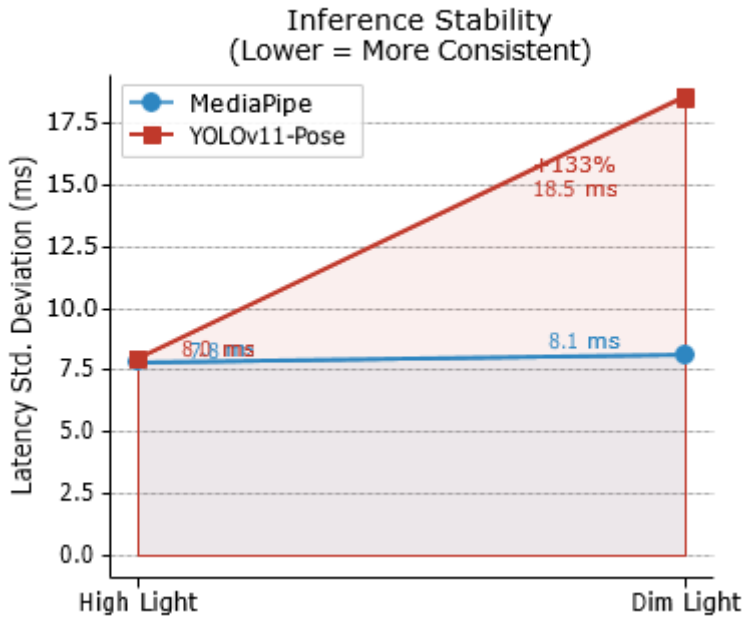


Fig. 9. Inference stability (standard deviation) by lighting condition. YOLOv11-Pose variance increases 133% under dim light while MediaPipe remains stable.

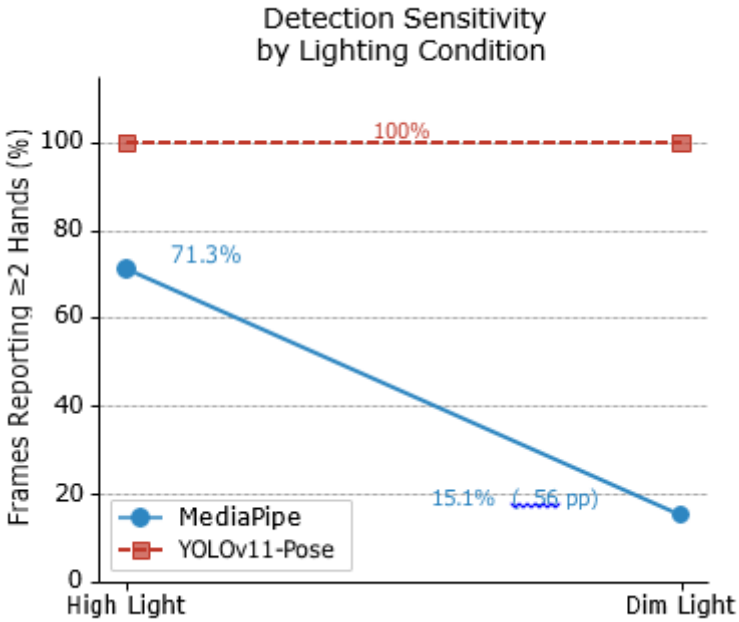


Fig. 10. Detection sensitivity (frames reporting ≥2 hands) by lighting condition. MediaPipe drops 56 percentage points under dim light; YOLOv11- Pose reports 2 in all frames regardless of illumination.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

VII. DISCUSSION

All four hypotheses were confirmed with statistically sig- nificant evidence. MediaPipe Hands is the clear choice for CPU-constrained, latency-sensitive gesture HCI applications. Its 2.35× average latency advantage, statistical effect size of d = 4.58, near-zero degradation under CPU stress (−0.2%), and consistent detection behaviour collectively deliver superior user experience on consumer hardware without GPU acceler- ation.

The lighting analysis reveals a nuanced trade-off not cap- tured in prior literature. While YOLOv11-Pose mean latency increases less under dim light (+4.9% vs. +18.2%), its infer- ence variance rises by 133%, making per-frame latency unpre- dictable. For HCI applications, stable latency is preferable to variable latency, as the latter produces irregular pointer updates that EMA smoothing cannot fully compensate for. This finding recommends supplementing YOLO deployments in variable- lighting environments with adaptive CLAHE preprocessing.

YOLOv11-Pose's systematic over-detection (81.6% in single-hand scenarios) is a practical deployment concern. For applications where missed detections are more costly than false positives — such as tracking partially occluded or gloved hands — YOLO's lower-threshold behaviour may be advanta- geous. However, for the drawing use case, spurious detections increase false-gesture risk, which debouncing partially but not fully addresses.

The most generalisable finding is that EMA smoothing and gesture debouncing can render high-latency pipelines usable for interactive applications. Inference latency alone is an insufficient predictor of HCI quality — the post-inference signal processing budget should be a first-class design consid- eration when building gesture-driven interfaces on constrained hardware.

Study limitations include:

1. CPU-only inference that would show different relative performance under GPU acceleration;
2. a pre-trained, non-fine-tuned YOLOv11-Pose model un- derrepresenting YOLO's full potential;
3. 3.8–4.6 FPS dual- pipeline throughput insufficient for production-grade real-time use, though sufficient for controlled benchmarking; and
4. lighting conditions characterised qualitatively rather than with calibrated lux measurements.

VIII. CONCLUSION

This paper presented Virtual Canvas, a real-time dual- pipeline gesture drawing system enabling fair concurrent benchmarking of MediaPipe Hands and YOLOv11-Pose un- der identical live conditions. Across 1,391 samples from 11 sessions spanning 10 days, MediaPipe demonstrated 2.35× lower mean inference latency ($t = -120.68$, $p < 0.0001$, $d = 4.58$), near-zero degradation under CPU stress (−0.2%), and stable performance under dim lighting, establishing it as the preferred model for CPU-deployed gesture HCI. YOLOv11- Pose's higher sensitivity and architectural flexibility make it better suited for fine-tuned, domain-specific, or GPU- accelerated deployments, though its 81.6% over-detection rate and increased variance under dim light are practical costs that must be addressed for production use.
The key cross-cutting insights are: (i) EMA temporal smoothing and gesture debouncing are essential complements to any inference pipeline in interactive applications; and (ii) dim lighting affects inference variance more than mean latency in single-stage detectors — a distinction absent from prior benchmarking literature that evaluates only offline metrics on curated datasets.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

IX. FUTURE WORK

1. GPU-accelerated deployment to enable 30+ FPS dual- pipeline evaluation and assessment of whether latency advantages persist under GPU conditions.
2. Fine-tuning YOLOv11-Pose on a curated hand-keypoint dataset to reduce the 81.6% over-detection rate and pro- vide an equitable accuracy comparison.
3. Calibrated lighting study with quantified lux levels and CLAHE preprocessing enabled/disabled, to system- atically characterise the illumination threshold at which each model degrades.
4. Expanded evaluation across a diverse cohort with quan- titative usability metrics: stroke error rate, task comple- tion time, and gesture recognition accuracy.
5. Transformer-based pipeline extension exploring lightweight estimators such as RTMPose [12] as a third candidate for comparison.

REFERENCES

1. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. L. Chang, and M. Grundmann, "MediaPipe Hands: On-device real- time hand tracking," arXiv preprint arXiv:2006.10214, 2020.
2. V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device real-time body pose tracking," arXiv preprint arXiv:2006.10204, 2020.
3. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE CVPR, 2016, pp. 779–788.
4. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
5. Ultralytics, "YOLO11 architecture overview," GitHub, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics
6. C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in Proc. IEEE CVPR, 2023, pp. 7464–7475.
7. A. Kulkarni, K. Karande, and A. Jadhav, "Virtual whiteboard using hand gesture recognition," in Proc. ICECA, IEEE, 2020.
8. R. Nair, A. Thomas, and K. Sreekumar, "Real-time air-writing recogni- tion for Indian Sign Language using MediaPipe," IEEE Access, vol. 10, pp. 44312–44325, 2022.
9. P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network," in Proc. IEEE CVPR, 2016, pp. 4207–4215.
10. J. Shin, D. Kim, and H. Seong, "Comparative evaluation of lightweight hand detectors for augmented reality applications," Sensors, vol. 21, no. 11, p. 3758, 2021.
11. O. Ko¨pu¨klu¨, A. Gunduz, N. Kose, and G. Rigoll, "Real-time hand ges- ture detection and classification using convolutional neural networks," in Proc. IEEE FG, 2019, pp. 1–8.
12. T. Jiang, P. Lu, L. Zhang et al., "RTMPose: Real-time multi-person pose estimation based on MMPose," arXiv preprint arXiv:2303.07399, 2023.
13. J. Cohen, Statistical Power Analysis for the Behavioral Sciences, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum, 1988.