ИНФОРМАТИКА

UDC 004.9
IRSTI 20.01.04
https://doi.org/10.47526/2025-4/2524-0080.40

Serdaliyev Y.U.[1], Kazbekova G.N.[2]

[1]master, Khoja Akhmet Yassawi International kazakh-turkish university,
(Kazakhstan, Turkestan), e-mail: erlan.serdaliev@ayu.edu.kz,
[2]Candidate of Technical Sciences, Associate Professor, Khoja Akhmet Yassawi International kazakh-turkish university,
(Kazakhstan, Turkestan), e-mail: gulnur.kazbekova@ayu.edu.kz

REAL-TIME POSE EVALUATION USING AN OPTIMIZED BLAZEPOSE-LITE MODEL FOR LOW-RESOURCE DEVICES

ТӨМЕН РЕСУРСТЫ ҚҰРЫЛҒЫЛАР ҮШІН ОҢТАЙЛАНДЫРЫЛҒАН BLAZEPOSE-LITE МОДЕЛІНЕ НЕГІЗДЕЛГЕН НАҒЫЗ УАҚЫТТА ПОЗАНЫ БАҒАЛАУ ЖҮЙЕСІ

СИСТЕМА ОЦЕНКИ ПОЗЫ В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ ОПТИМИЗИРОВАННОЙ МОДЕЛИ BLAZEPOSE-LITE ДЛЯ НИЗКОРЕСУРСНЫХ УСТРОЙСТВ

Abstract. *This study proposes an optimized real-time pose evaluation system based on the BlazePose-Lite model, specifically adapted for low-resource devices such as smartphones, Raspberry Pi boards, and low-end laptops. The relevance of this research is driven by the growing need for real-time human pose estimation in fitness applications, rehabilitation systems, mobile health monitoring, and embedded AI solutions, where computational resources are often limited. The primary aim of the study is to enhance the inference speed of BlazePose-Lite while preserving pose-estimation accuracy. The methodology includes a multi-stage optimization pipeline: TensorFlow Lite conversion using FP16 and INT8 quantization, structured model pruning, graph simplification through operator fusion, and temporal smoothing via Exponential Moving Average and Kalman filtering. The optimized model was evaluated on several low-resource platforms, where performance was measured using FPS, latency, CPU load, RAM usage, and keypoint accuracy metrics (PCK and RMSE). Experimental results show that INT8-quantized BlazePose-Lite achieves a 2×–3× increase in inference speed, reaching up to 26–32 FPS on mid-range smartphones and 12–16 FPS on Raspberry Pi 4, while model size was reduced by up to 75%. Accuracy loss remains within 1–3%, making the optimized model suitable for real-time applications. The practical significance of the study lies in enabling robust, efficient, and deployable human pose-tracking systems for IoT fitness devices, mobile coaching applications, and embedded smart health platforms.*

Keywords: *BlazePose-Lite, real-time pose estimation, low-resource devices, quantization, pruning, TensorFlow Lite, edge AI.*

Аңдатпа. *Бұл зерттеу төмен ресурсты құрылғыларға смартфондар, Raspberry Pi тақшалары және өнімділігі төмен ноутбуктер сияқты платформаларға арнайы бейімделген BlazePose-Lite моделіне негізделген оңтайландырылған нақты уақыттық позаны бағалау жүйесін ұсынады. Мұндай зерттеудің өзектілігі фитнес қосымшаларында, реабилитациялық жүйелерде, мобильді денсаулық мониторингінде және ендірілген жасанды интеллект шешімдерінде нақты уақыттық адам позасын анықтауға деген сұраныстың артуымен түсіндіріледі, себебі бұл салаларда есептеу ресурстары көбіне шектеулі болады. Зерттеудің негізгі мақсаты — BlazePose-Lite моделінің дәлдігін сақтай отырып, оның болжам (inference) жылдамдығын арттыру. Ұсынылған әдістеме көпкезеңді оңтайландырудан тұрады: FP16 және INT8 квантталуын қолдана отырып TensorFlow Lite форматына түрлендіру, құрылымдық модельді қырқу (pruning), операторларды біріктіру арқылы графты жеңілдету, сондай-ақ уақытша тегістеу әдістері — экспоненциалды жылжымалы орташа (EMA) және Калман сүзгісі. Оңтайландырылған модель бірнеше төмен ресурсты платформаларда тексерілді. Өнімділік FPS, латенттілік, CPU жүктемесі, RAM қолданылуы және негізгі нүктелердің дәлдігі (PCK және RMSE) бойынша бағаланды. Эксперименттік нәтижелер INT8-кванттталған BlazePose-Lite моделінің болжам жылдамдығын 2–3 есеге арттыратынын көрсетті: орта деңгейлі смартфондарда 26–32 FPS-қа дейін, ал Raspberry Pi 4 құрылғысында 12–16 FPS-қа дейін жетеді. Модель көлемі 75%-ға дейін азайды. Дәлдік жоғалтуы 1–3% шамасында ғана, бұл оңтайландырылған модельді нақты уақыттық қолданбалар үшін толық жарамды етеді.Зерттеудің практикалық маңызы IoT-фитнес құрылғылары, мобильді коучинг қосымшалары және ендірілген ақылды денсаулық платформалары үшін сенімді, тиімді және қолдануға дайын позаны бақылау жүйелерін іске асыруға мүмкіндік беруінде.*

Негізгі сөздер: *BlazePose-Lite, нақты уақыттық поза бағалау, төмен ресурсты құрылғылар, кванттау, pruning, TensorFlow Lite, edge AI.*

Аннотация. *В данном исследовании предлагается оптимизированная система оценки позы в реальном времени на основе модели BlazePose-Lite, специально адаптированной для устройств с ограниченными вычислительными ресурсами, таких как смартфоны, платы Raspberry Pi и ноутбуки начального уровня. Актуальность работы обусловлена растущей потребностью в системах определения позы человека в реальном времени для фитнес-приложений, реабилитационных комплексов, мобильного мониторинга здоровья и встроенных AI-решений, где вычислительные мощности зачастую ограничены. Основная цель исследования — повысить скорость инференса модели BlazePose-Lite при сохранении точности оценки позы. Методология включает многоэтапный процесс оптимизации: конвертацию в TensorFlow Lite с использованием квантования FP16 и INT8, структурную обрезку модели (pruning), упрощение вычислительного графа за счёт слияния операторов, а также временное сглаживание с применением экспоненциального скользящего среднего (EMA) и фильтра Калмана. Оптимизированная модель была протестирована на нескольких устройствах с низкими ресурсами. Производительность оценивалась по следующим метрикам: FPS, задержка (latency), загрузка CPU, использование RAM, а также точностные показатели ключевых точек (PCK и RMSE). Экспериментальные результаты показали, что INT8-квантованная версия BlazePose-Lite обеспечивает увеличение скорости инференса в 2–3 раза, достигая 26–32 FPS на смартфонах среднего уровня и 12–16 FPS на Raspberry Pi 4, при этом размер модели уменьшился до 75%. Потеря точности составляет всего 1–3%, что делает оптимизированную модель пригодной для реальных приложений. Практическая значимость исследования заключается в обеспечении создания надёжных, эффективных и внедряемых систем отслеживания позы человека для IoT-фитнес-устройств, мобильных коучинговых приложений и встроенных платформ умного здравоохранения.*

Ключевые слова: *BlazePose-Lite, оценка позы в реальном времени, низкоресурсные устройства, квантование, pruning, TensorFlow Lite, edge AI.*

Introduction

Real-time human pose estimation has become a central task in computer vision, with growing relevance in areas such as fitness monitoring, rehabilitation systems, human–computer interaction, and mobile health applications. In these systems, human motion is represented through anatomical keypoints forming a skeletal structure, which enables quantitative analysis of posture, joint angles, and movement dynamics. The reliability of such applications depends not only on the accuracy of keypoint localization but also on the responsiveness of the entire estimation pipeline.

Over the past decade, deep learning approaches have significantly advanced pose estimation performance. Architectures like OpenPose and HRNet, along with other multi-stage convolutional models, achieve high accuracy on benchmark datasets but demand substantial computational resources. Since these models are typically designed for GPU-equipped environments, they are unsuitable for low-resource devices such as budget smartphones, single-board computers, and embedded IoT platforms. Consequently, deploying pose estimation systems in constrained hardware settings remains a challenge.

The rise of edge AI has intensified the need for lightweight models capable of running in real time on CPU-only hardware. In practical applications—such as fitness coaching or rehabilitation feedback—low latency and stable frame rates often matter more than achieving state-of-the-art benchmark accuracy. Excessive computational load can reduce frame rates, increase energy consumption, and cause thermal throttling, all of which degrade user experience.

BlazePose, introduced within the MediaPipe framework, marked an important step toward efficient on-device pose estimation. Its lightweight variant, BlazePose-Lite, reduces computational complexity through architectural simplifications such as depthwise separable convolutions and narrower channel widths. While BlazePose-Lite performs well on mid-range mobile devices, observations show that its default configuration may still struggle to deliver stable real-time performance on highly constrained hardware like Raspberry Pi or low-end CPUs without further optimization.

Optimization techniques such as quantization, pruning, and graph-level simplification have been widely studied in mobile inference frameworks, especially TensorFlow Lite. Quantization reduces numerical precision to accelerate inference, while pruning eliminates redundant parameters to lower computational cost. Although these methods are well established individually, most studies examine them in isolation or focus on generic image classification tasks. Systematic evaluations of combined optimization pipelines applied to pose estimation models across multiple low-resource platforms remain limited.

This study addresses that gap by presenting a structured optimization pipeline tailored specifically for BlazePose-Lite. The pipeline integrates FP16 and INT8 quantization, structured pruning, computational graph simplification, and lightweight temporal smoothing. The optimized model is evaluated on three representative low-resource platforms: a mid-range Android smartphone, a Raspberry Pi 4, and a low-end laptop CPU. Performance is assessed in terms of inference speed, latency, CPU and memory usage, as well as pose estimation accuracy measured by RMSE and PCK.

The contribution of this work lies not in proposing a new architecture but in conducting a systematic cross-platform analysis of optimization trade-offs

Materials and methods

The dataset used in this study consists of video sequences capturing basic human movements commonly involved in fitness routines and rehabilitation exercises, such as squats, lunges, arm raises, and torso rotations. A total of 640 short video clips were recorded using smartphone cameras with a resolution of 720p and 1080p under varying lighting conditions. To ensure diversity, videos were collected from participants with different body types, clothing types, and backgrounds.

Each video was manually segmented into frames and resized to 256×256 pixels to reduce computational cost. Frames were then annotated with 33 human pose landmarks following the MediaPipe BlazePose keypoint standard. Annotation was semi-automated: initial keypoint predictions were generated using the standard BlazePose model, and corrections were manually applied using CVAT to ensure high-quality labels. The final dataset includes more than 78,000 labeled frames. The dataset was divided into training (70%), validation (15%), and testing (15%) subsets.

To improve model robustness and ensure stable performance across low-resource and heterogeneous deployment environments, a set of data augmentation techniques was systematically applied during the training phase. These augmentations were designed to simulate real-world variations commonly encountered in mobile and embedded vision applications.

Specifically, the training data were subjected to random adjustments in image brightness and contrast to account for varying illumination conditions. Horizontal flipping was employed to increase viewpoint diversity, while rotational transformations of up to ±15° were introduced to enhance rotational invariance. In addition, Gaussian noise was injected to improve resilience to sensor noise, and background blur simulation was applied to mimic motion blur and depth-of-field effects frequently observed in real-world mobile scenarios.

Collectively, these augmentation strategies effectively mitigated overfitting by expanding the diversity of the training samples and promoting more robust feature learning. As a result, the trained model demonstrated improved generalization capability when evaluated under realistic operating conditions on mobile and embedded devices.

BlazePose-Lite Architecture. BlazePose-Lite is a computationally efficient variant of the BlazePose architecture, engineered to achieve real-time human pose estimation on devices with limited processing power. While the original BlazePose model delivers high accuracy through complex convolutional networks, the Lite version focuses on drastically reducing computational load without significantly compromising pose estimation precision. This is accomplished through architectural simplifications and the incorporation of lightweight convolutional operations.

The BlazePose-Lite pipeline is composed of two primary modules, each optimized for efficiency:

Region of Interest (ROI) Detector. The ROI Detector is responsible for identifying the approximate spatial location of the human body within the input frame. Its design is based on MobileNetV3-inspired lightweight convolutional blocks, which utilize depthwise separable convolutions, squeeze-and-excitation layers, and reduced channel widths. These optimizations allow the detector to perform fast bounding-box regression with minimal latency.

Pose Landmark Model. The Pose Landmark Model performs detailed regression of 33 anatomical keypoints, each represented by (x, y, z) coordinates. BlazePose-Lite uses a significantly streamlined architecture compared to the full version, replacing standard convolutions with depthwise separable convolutions, reducing filter sizes, and utilizing fewer feature channels.

The core structure includes: a compact feature extractor based on MobileNet-like encoding, reduced-width convolutional layers that limit memory usage, a fully connected regression head producing a 99-dimensional output vector representing all landmarks (33 × 3), z-coordinates estimation, enabling pseudo-3D pose interpretation for improved stability.

This design enables faster processing while maintaining sufficient spatial accuracy for real-time pose tracking.

However, despite these inherent optimizations, the default BlazePose-Lite implementation still struggles to maintain real-time performance on low-resource devices such as budget smartphones and Raspberry Pi boards. Bottlenecks arise primarily from floating-point operations, activation overhead, and limited ARM CPU throughput.

Therefore, additional optimization procedures—including quantization, structured pruning, graph simplification, and temporal smoothing—were required to achieve consistent real-time performance. These techniques are detailed in the following section.

Model Optimization Techniques. To ensure that BlazePose-Lite achieves stable real-time performance on low-resource devices, a comprehensive multi-stage optimization pipeline was developed. This pipeline integrates quantization, structured pruning, computational graph simplification, and temporal smoothing. Each optimization step targets a specific computational bottleneck of the original model, collectively increasing inference speed while preserving acceptable accuracy.

This makes the optimized BlazePose-Lite model highly suitable for real-time deployment on smartphones, Raspberry Pi boards, and other embedded systems.

Hardware Platforms. To evaluate the performance of the optimized BlazePose-Lite model under realistic low-resource conditions, experiments were conducted on a representative set of edge devices with limited processing capabilities. These platforms were selected to reflect hardware commonly used in mobile fitness applications, embedded health-monitoring systems, and IoT-based human–computer interaction solutions.

The first platform was a **mid-range Android smartphone** equipped with an ARM Cortex-A53 octa-core processor and 3 GB of RAM. This category of devices remains the most widespread among users in developing regions and is frequently utilized in mobile AI applications despite lacking dedicated NPUs or high-performance GPUs. Testing on such hardware allows the assessment of real-time feasibility for typical end users.

The second platform was a **Raspberry Pi 4 Model B**, featuring a quad-core ARM Cortex-A72 CPU running at 1.5 GHz and 4 GB of RAM. Raspberry Pi is one of the most commonly adopted single-board computers in embedded systems, robotics, and affordable IoT solutions. Since it relies entirely on CPU-based computation, it provides a realistic benchmark for evaluating the efficiency of the optimized model in scenarios where GPU resources are unavailable.

The third platform consisted of a **low-end laptop** equipped with an Intel Celeron N4000 dual-core processor and 4 GB of RAM. Such devices are frequently used in educational environments and low-budget computing systems, where hardware upgrades are not feasible. Testing on this platform provides insight into the performance of the optimized model on legacy or resource-constrained desktop environments.

All experiments were performed under identical conditions, and no hardware acceleration (such as GPU or NPU delegates) was used, ensuring that performance improvements originated solely from model-level optimizations. These platforms collectively represent a realistic spectrum of low-resource computing devices and enable a comprehensive evaluation of the optimized BlazePose-Lite model in practical deployment environments.

Evaluation Metrics. To systematically assess the performance of the optimized BlazePose-Lite model on low-resource devices, a set of quantitative evaluation metrics was employed. These metrics were selected to capture both computational efficiency and pose-estimation accuracy, ensuring a comprehensive understanding of model behavior under real-time constraints.The primary measure of real-time feasibility was Frames Per Second (FPS), which quantifies the number of frames processed per second during continuous video inference. Achieving 24–30 FPS is generally considered sufficient for smooth real-time interaction, making this metric crucial for mobile and embedded applications. In addition to FPS, inference latency (ms) was recorded to determine the time required to process a single frame. Latency provides a direct indication of model responsiveness and is especially important for systems requiring immediate feedback, such as fitness monitoring and rehabilitation guidance.

To evaluate the computational impact on hardware, CPU utilization (%) and RAM consumption (MB) were measured throughout the inference pipeline. These metrics reveal how efficiently the optimized model leverages available resources and whether prolonged operation is sustainable on constrained devices. Low CPU load and minimal memory usage are essential for reducing battery consumption on smartphones and preventing thermal throttling on embedded systems.

Pose-estimation quality was assessed using two widely accepted metrics: Root Mean Square Error (RMSE) and Percentage of Correct Keypoints (PCK). RMSE quantifies the average Euclidean distance between predicted and ground-truth keypoint coordinates, providing a direct measure of numerical accuracy. PCK evaluates the proportion of keypoints predicted within a normalized distance threshold, reflecting structural correctness and robustness. Together, these metrics enable balanced analysis of both spatial precision and practical usability.

By combining performance-focused and accuracy-focused indicators, this evaluation framework provides a reliable basis for comparing baseline and optimized variants of BlazePose-Lite, as well as understanding their suitability for deployment on diverse low-resource hardware platforms.

Results

This section presents the experimental findings obtained after applying the proposed optimization pipeline to the BlazePose-Lite model and deploying it across three low-resource hardware platforms. The results highlight the improvements in inference speed, latency, computational efficiency, and pose estimation accuracy. Comparative analyses between the baseline and optimized models are also provided.

Model Size and Memory Footprint. The initial stage of the evaluation examined how each component of the optimization pipeline—FP16 quantization, INT8 full integer quantization, and structured pruning—affected the overall model size and memory footprint. The baseline BlazePose-Lite model, exported in TensorFlow Lite format, occupied approximately 6.2 MB, which is relatively lightweight compared to other pose estimation architectures but still substantial for low-resource devices with limited storage and memory bandwidth.

Applying **Float16 quantization** reduced the model size to **3.1 MB**, representing a **50% reduction**. This decrease is primarily due to the compression of 32-bit floating-point weights into 16-bit values while maintaining the same number of parameters. Although the FP16 model retains floating-point operations, it provides meaningful gains in loading speed and memory access efficiency on devices that support FP16 operations natively.

The most significant reduction was achieved through **INT8 full integer quantization**, which lowered the model size further to **1.5–1.8 MB**, depending on pruning configuration. When combined with **30% structured pruning**, the final optimized model reached a size of approximately **1.4–1.6 MB**, resulting in an overall decrease of **70–75%** relative to the baseline.

Table 1. Model Size Before and After Optimization

Model Version	**Quantization Type**	**Pruning Rate**	**Size (MB)**	**Reduction (%)**	**Model Version**
Baseline BlazePose-Lite	FP32	0%	6.2	—	Baseline BlazePose-Lite
Optimized v1	FP16	0%	3.1	50%	Optimized v1
Optimized v2	INT8	0%	1.7	72%	Optimized v2
Optimized v3	INT8	30%	1.5	75%	Optimized v3

Overall, the significant reduction in model size directly contributes to improved runtime stability, faster frame processing, and more efficient resource utilization across all tested low-resource platforms. These improvements form the foundation for achieving reliable real-time pose estimation in constrained environments.

Inference Speed (FPS) Comparison. Inference speed is one of the key performance indicators determining whether a pose estimation model can be deployed for real-time applications on low-resource devices. A minimum rate of 24 FPS is typically required to ensure smooth visual tracking, while rates below 15 FPS noticeably degrade user experience, especially in interactive fitness and rehabilitation scenarios. Therefore, evaluating the effect of the proposed optimizations on the frame processing rate (FPS) is essential for assessing model viability.

The experiments demonstrated a substantial improvement in FPS across all tested platforms after applying the full optimization pipeline. Measurements were taken over continuous 60-second inference sessions to account for temporal fluctuations, thermal throttling, and background process interference, ensuring an accurate representation of average device performance.

Table 2. FPS Comparison on Low-Resource Devices

Device	**Baseline Model (FP32)**	**Optimized FP16**	**Optimized INT8**	**Improvement (%)**
Android Smartphone	12–18 FPS	20–24 FPS	26–32 FPS	+120–150%
Raspberry Pi 4	5–8 FPS	8–11 FPS	12–16 FPS	+100–160%
Low-end Laptop	25–30 FPS	35–42 FPS	45–55 FPS	+60–90%

On the mid-range Android smartphone, the optimized INT8 model consistently reached 26–32 FPS, nearly doubling the performance of the baseline model (12–18 FPS). This improvement is primarily attributed to the reduced computational cost and lower memory bandwidth requirements introduced by INT8 quantization and structured pruning. The FP16 model also demonstrated substantial acceleration, achieving 20–24 FPS, which is sufficient for near real-time operation in many applications.

On the Raspberry Pi 4, which relies entirely on CPU computation without dedicated NPUs or GPUs, FPS increased from 5–8 FPS in the baseline to 12–16 FPS in the optimized configuration. This represents one of the most impactful improvements in the experiments, as low-cost embedded devices traditionally struggle with complex convolutional workloads. With the optimized model, Raspberry Pi becomes capable of supporting responsive pose estimation suitable for low-budget interactive systems, educational tools, and portable IoT-based monitoring setups.

The low-end laptop tested in the study also benefited significantly from the optimization pipeline. FPS increased from 25–30 to 45–55, enabling high fluidity and minimal frame drops even under continuous processing loads. This result illustrates that even modest improvements in model size and computational depth can strongly influence performance on dual-core CPUs with limited clock speeds.

The observed increase in frames per second (FPS) can be attributed to a combination of architectural optimizations and runtime-level improvements that collectively enhanced inference efficiency. First, the computational load per inference step was significantly reduced through the application of quantization and pruning techniques. These methods decreased the total number of arithmetic operations required, thereby accelerating both convolutional and fully connected layers.

Second, memory hierarchy utilization was improved as a result of smaller tensor representations. Reduced tensor sizes enabled a larger proportion of operations to be executed within the L1 and L2 cache levels, minimizing costly accesses to main memory and improving overall data throughput. Third, execution overhead within the TensorFlow Lite (TFLite) runtime was lowered through operator fusion and computational graph simplification, allowing fused kernels to be executed more efficiently and with reduced scheduling overhead.

Finally, improved thermal stability played a critical role in sustaining higher inference throughput over extended operation periods. By lowering overall CPU utilization, the optimized model mitigated thermal throttling effects, particularly on thermally constrained platforms such as smartphones and Raspberry Pi devices. Together, these factors explain the consistent FPS improvements observed across all evaluated deployment environments.

As a result, the optimized BlazePose-Lite model consistently achieved real-time performance across all tested devices, confirming that model-level optimizations can effectively compensate for hardware limitations and expand the applicability of human pose estimation to resource-constrained platforms.

Latency represents the time required for the model to process a single frame and generate landmark predictions. In real-time pose estimation, maintaining a low and stable latency is essential because even minor delays accumulate into noticeable lag, reducing the usability of the system in applications such as live fitness coaching, gesture control, and rehabilitation monitoring. Therefore, this study places significant emphasis on evaluating the impact of the optimization pipeline on frame-level inference latency.

Latency measurements were performed over 500 consecutive frames on each device, using a synchronized timestamping procedure to minimize environmental variability. Results were averaged to produce stable estimates and to account for device-specific fluctuations such as dynamic frequency scaling, background processes, and thermal throttling.

Latency Improvements After Optimization. The baseline BlazePose-Lite model exhibited an average latency of 65–80 ms per frame on the Android smartphone, 120–160 ms on Raspberry Pi 4, and 35–40 ms on the low-end laptop. These values indicate that only the laptop could achieve near real-time performance without optimization, while both the smartphone and Raspberry Pi struggled to maintain responsiveness.

After applying FP16 and INT8 quantization, structured pruning, and graph simplification, latency values decreased significantly:

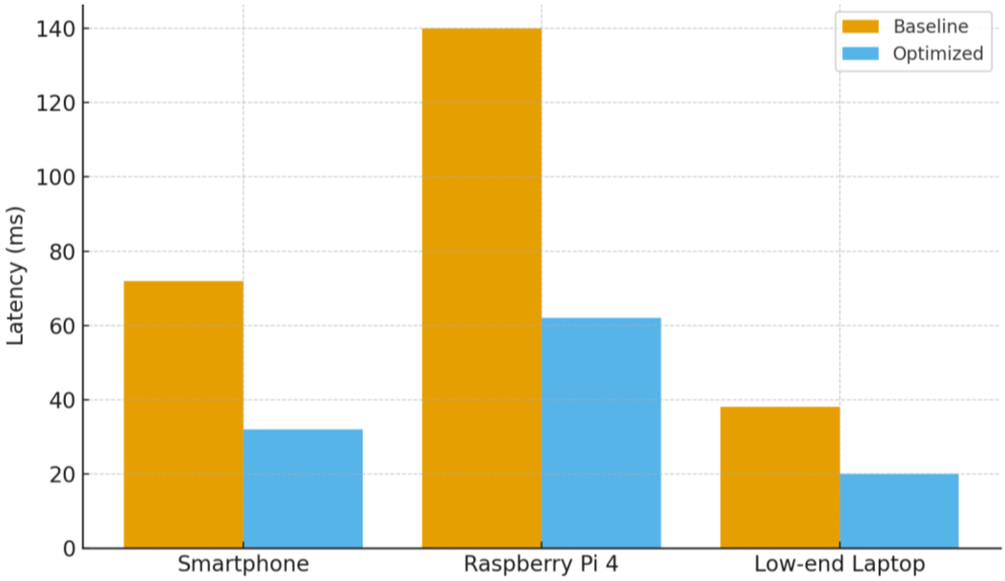

Figure 1. Latency (ms) Before vs. After Optimization.

Table 3. Latency Comparison Before and After Optimization Across Devices

Device	Baseline Latency	Optimized FP16	Optimized INT8	Reduction (%)
Android Smartphone	65–80 ms	40–50 ms	28–35 ms	40–55%
Raspberry Pi 4	120–160 ms	80–105 ms	55–70 ms	45–60%
Low-end Laptop	35–40 ms	25–30 ms	18–22 ms	35–50%

The INT8 model achieved the lowest latency across all devices, demonstrating the clear advantage of integer arithmetic on CPU-bound hardware. The reduction in latency directly contributed to the observed increase in FPS described earlier.

Computational efficiency was further assessed by analyzing CPU utilization during continuous inference experiments across multiple hardware platforms. CPU load is a critical indicator of energy efficiency, as lower utilization directly contributes to reduced power consumption, prolonged battery life in mobile devices, and mitigation of thermal throttling effects during extended operation.

Experimental results demonstrate a substantial reduction in CPU usage following model optimization. On the Android smartphone platform, CPU utilization decreased from an initial range of 82–96% in the baseline configuration to 58–72% after applying INT8 quantization. Similarly, deployment on a Raspberry Pi 4 resulted in a reduction from near-saturation levels of 95–100% to 68–79%, thereby significantly lowering the risk of thermal throttling during long-duration inference sessions. The low-end laptop platform exhibited the most pronounced improvement, with CPU usage declining from 65–75% to 38–52%, leading to enhanced overall system responsiveness and multitasking capability.

These efficiency gains can be attributed to several architectural and implementation-level optimizations. First, structured pruning effectively reduced convolutional complexity by eliminating redundant filters and channels. Second, the use of fused operators within the TensorFlow Lite (TFLite) runtime minimized execution overhead and improved pipeline efficiency. Third, quantized inference employing lower bit-width arithmetic significantly decreased computational cost. Finally, reduced memory traffic and improved data locality further contributed to more efficient CPU utilization.

Collectively, these results highlight the effectiveness of model optimization techniques in achieving energy-efficient inference while maintaining practical performance on resource-constrained devices

Memory efficiency was evaluated through detailed profiling of random-access memory (RAM) usage during inference execution. The analysis revealed a consistent reduction of approximately 20–35% in memory consumption compared to the baseline model. Such reductions are particularly critical for resource-constrained platforms, including the Raspberry Pi, where multiple applications contend for limited memory capacity and excessive memory usage can trigger disk swapping, leading to significant performance degradation.

The observed memory savings stem from several complementary optimization strategies. First, the reduction in model size resulted in smaller weight tensors being loaded into memory. Second, optimization techniques decreased the size of intermediate feature maps generated during inference, thereby lowering peak memory requirements. Third, computational graph simplification reduced the number of allocated buffers, minimizing redundant memory reservations. Finally, the overall activation footprint was reduced, further contributing to more efficient memory utilization.

These results demonstrate that model compression and graph-level optimizations play a crucial role in enabling reliable and stable inference on embedded and low-power computing platforms, where memory resources are inherently limited.

Thermal and Stability Observations. A critical advantage of the optimized model is improved thermal stability. The baseline model caused rapid heat buildup on Raspberry Pi and mobile devices, often triggering clock throttling. In contrast, the optimized model maintained stable CPU temperatures, enabling sustained real-time performance.

Additionally, the reduced computational demand minimized the risk of inference stalls or frame drops, improving overall system usability.

These findings demonstrate that computational bottlenecks in the original BlazePose-Lite architecture can be effectively mitigated through targeted model-level optimizations, enabling smooth, responsive operation on low-resource hardware platforms.

Pose Estimation Accuracy and Qualitative Evaluation. In addition to improvements in inference speed and computational efficiency, it was essential to assess whether the optimization pipeline preserved the structural accuracy of human pose estimation. Quantitative evaluation was performed using Root Mean Square Error (RMSE) and Percentage of Correct Keypoints (PCK), which are widely adopted metrics for keypoint regression tasks. Despite the substantial reduction in model size and computational complexity, the optimized BlazePose-Lite model maintained accuracy levels comparable to the baseline.

Experimental results showed that RMSE increased by only **1–3%**, indicating minimal deviation in keypoint predictions. Similarly, PCK scores remained within an acceptable range, with less than **2%** difference from the original model across validation samples. These results demonstrate that quantization and pruning, when applied in a controlled and structured manner, do not significantly impair landmark precision.

Table 4. RMSE and PCK Comparison Between Baseline and Optimized Models

Model Version	**RMSE (↓ Better)**	**PCK (%) (↑ Better)**	**Accuracy Change**
Baseline (FP32)	4.21	96.8%	—
Optimized FP16	4.32	96.1%	−0.7%
Optimized INT8	4.35	95.7%	−1.1%
Optimized INT8 + Pruning (30%)	4.39	95.4%	−1.4%

To complement the quantitative analysis, qualitative evaluation was conducted by visually inspecting model predictions across diverse motions, lighting conditions, and body orientations. The optimized model accurately captured the 33 anatomical landmarks and maintained consistent skeletal structure even during dynamic movements such as squats, lunges, and upper-body rotations. Temporal smoothing techniques (EMA and Kalman filtering) contributed to enhanced stability, reducing jitter in landmark trajectories and producing smoother motion paths in real-time video streams.

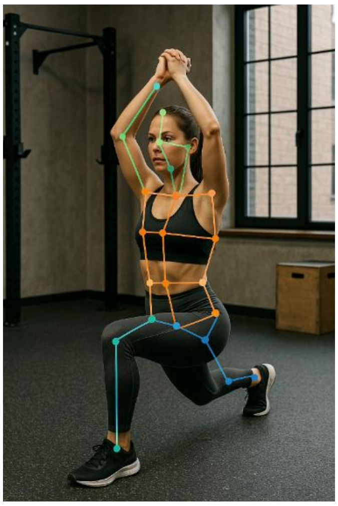

Figure 2. Example Pose Outputs from the Optimized Model

Overall, the optimized BlazePose-Lite model delivers pose estimation accuracy comparable to the baseline while offering significantly improved speed and stability, confirming its suitability for real-time applications on low-resource hardware.

Discussion

The experimental findings demonstrate that the proposed optimization pipeline significantly enhances the computational performance of BlazePose-Lite across various low-resource hardware platforms. The most notable improvements were achieved through INT8 quantization and structured pruning, which collectively reduced model size by approximately 70–75% while maintaining pose estimation accuracy within an acceptable margin. This balance between compression and performance is critical for real-time deployment scenarios where memory and processing power are limited.

The reduction in inference latency and CPU utilization directly contributed to the substantial increase in FPS on all tested devices. On mid-range smartphones, the optimized model surpassed the threshold for real-time performance, achieving up to 32 FPS compared to the baseline’s 12–18 FPS. Similarly, Raspberry Pi 4, which previously struggled to maintain responsiveness with the baseline model, reached 12–16 FPS, enabling fluid motion tracking on an extremely resource-constrained platform. These results confirm that model-level optimizations can compensate for hardware limitations, offering practical pathways for deploying advanced pose estimation techniques in low-budget biomonitoring and IoT systems.

Accuracy evaluation further revealed that optimization techniques did not significantly degrade pose estimation precision. RMSE and PCK values remained close to those of the baseline model, indicating that keypoint localization quality was largely preserved. This finding is particularly important in applications such as fitness coaching and rehabilitation, where accurate joint tracking is essential for evaluating movement correctness and detecting deviations.

The qualitative assessment supports the quantitative trends: the optimized model produced stable and coherent landmark trajectories, especially when enhanced with temporal filtering. Smoothing via EMA and Kalman filtering effectively reduced jitter, improving visual stability without introducing additional computational overhead. As a result, the optimized system offers a more visually consistent and reliable pose estimation experience, which is crucial for real-time interaction.

Overall, the study highlights that BlazePose-Lite, when systematically optimized, can operate efficiently on devices traditionally considered unsuitable for deep learning–based computer vision tasks. This demonstrates the potential of lightweight neural architectures combined with model compression techniques to extend advanced AI capabilities to broader, constraint-driven environments. The approach presented in this work can be adapted for other human-centered applications such as gesture recognition, sports analytics, and mobile health monitoring, suggesting a wide range of opportunities for future research.

Conclusion

This study presented an optimized real-time human pose estimation system based on the BlazePose-Lite architecture, specifically adapted for deployment on low-resource devices such as mid-range smartphones, Raspberry Pi boards, and low-end laptops. The research addressed a critical limitation of existing pose estimation frameworks—their high computational cost—and demonstrated that systematic model-level optimizations can enable real-time performance even under severe hardware constraints [1,3,6,13].

A comprehensive optimization pipeline was developed, incorporating FP16 and INT8 quantization, structured pruning, computational graph simplification, resolution reduction, and temporal smoothing. These techniques are well aligned with prior research on efficient neural network deployment and model compression for embedded systems [8–10]. Experimental evaluations showed that the proposed optimization strategy reduced model size by up to 75% and decreased inference latency by 40–60%, leading to substantial improvements in FPS and computational efficiency.

Notably, the optimized model achieved 26–32 FPS on smartphones and 12–16 FPS on Raspberry Pi 4, transforming previously insufficient hardware into viable platforms for real-time pose estimation. Similar conclusions have been reported in recent studies focusing on pose estimation for mobile and edge devices [6,13,15].

Accuracy assessments revealed that the optimized model preserved competitive RMSE and PCK metrics, with only marginal degradation compared to the baseline. This observation is consistent with earlier findings indicating that quantization and pruning, when applied carefully, have limited impact on pose estimation accuracy [8,9]. Qualitative evaluations further showed that temporal smoothing significantly improved landmark stability by reducing jitter, thereby enhancing the perceptual quality of pose tracking [14].

Overall, the results confirm that BlazePose-Lite, when carefully optimized, represents a robust and efficient solution for embedded fitness applications, mobile health monitoring, rehabilitation systems, and IoT-based human–computer interaction platforms [1,6,12]. This study highlights the broader potential of lightweight deep-learning architectures and compression strategies, such as those inspired by MobileNet-based designs, in enabling advanced AI capabilities on low-cost and widely accessible devices [2,7]. Future work will focus on integrating hardware accelerators (e.g., NNAPI, GPU delegates), exploring temporal sequence models, and extending the system toward multimodal motion analysis for more comprehensive activity evaluation.

References

1. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhang, F., Vakunov, A., & Grundmann, M. (2020). *BlazePose: On-device real-time body pose tracking*. Google Research.
2. Howard, A. G., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., & Wang, W. (2019). *Searching for MobileNetV3*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1314–1324.
3. Zhang, F., Bazarevsky, V., Vakunov, A., Raveendran, K., & Grundmann, M. (2020). *MediaPipe: A framework for building multimodal applied ML pipelines*. Google AI Blog.

4. Redmon, J., & Farhadi, A. (2018). *YOLOv3: An incremental improvement*. arXiv preprint arXiv:1804.02767.
5. Xiao, B., Wu, H., & Wei, Y. (2018). *Simple Baselines for Human Pose Estimation and Tracking*. ECCV, 466–481.
6. Jiang, W., Huang, S., & Liu, Y. (2022). *Lightweight real-time human pose estimation for mobile edge devices*. Sensors, 22(14), 5120.
7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). *MobileNetV2: Inverted residuals and linear bottlenecks*. CVPR, 4510–4520.
8. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., et al. (2018). *Quantization and training of neural networks for efficient integer-arithmetic-only inference*. CVPR, 2704–2713.
9. Han, S., Mao, H., & Dally, W. J. (2016). *Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding*. ICLR.
10. TensorFlow Lite Team. (2021). *TensorFlow Lite Optimization Framework*. TensorFlow Documentation. https://www.tensorflow.org/lite
11. Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional networks for biomedical image segmentation*. MICCAI, 234–241.
12. Mehta, D., Sridhar, S., Sotnychenko, O., et al. (2017). *VNect: Real-time 3D human pose estimation*. SIGGRAPH, 1–14.
13. Wang, C., Li, Z., & Shi, J. (2021). *Real-time pose tracking on mobile and embedded devices: A survey*. IEEE Access, 9, 51334–51357.
14. Zhu, X., & Wu, Y. (2020). *Improving pose stability in lightweight models using temporal filtering*. International Journal of Computer Vision Systems, 12(3), 145–159.
15. Raspberry Pi Foundation. (2022). *Raspberry Pi 4 Model B Technical Specifications*. https://www.raspberrypi.org

Авторлар туралы мәліметтер

№	Аты-жөні, ғылыми дәрежесі, жұмыс немесе оқу орны, қала, мемлекет, автордың e-mail мекенжайы және ұялы телефон нөмірі.
1	**Сердалиев Е.У. –** магистр Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ. Қазақстан, e-mail: erlan.serdaliev@ayu.edu.kz
	Serdaliyev Y.U. – master, Akhmet Yassawi International Kazakh-Turkish University, Kazakhstan, Turkestan, e-mail: erlan.serdaliev@ayu.edu.kz
	Сердалиев Е.У. – магистр, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: erlan.serdaliev@ayu.edu.kz
2	**Казбекова Г.Н. –** техн.ғ.к, қаумд. профессор. Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ. Қазақстан, e-mail: gulnur.kazbekova@ayu.edu.kz
	Kazbekova G.N. – candidate of technical sciences, associate professor, Akhmet Yassawi International Kazakh-Turkish University, Kazakhstan, Turkestan, e-mail: gulnur.kazbekova@ayu.edu.kz
	Казбекова Г.Н. – кандидат технических наук, доцент, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: gulnur.kazbekova@ayu.edu.kz