International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 05 May 2026
DOI: 10.47191/ijcsrr/V9-i5-48, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

Request-Aware Fuzzy Load Balancing for Human Action Recognition and Monitoring in Video Streams

Ergashev Shaxboz Toshtemir[1], Saydazimov Javlonbek Karimovich[2], Toshpulatov Jahongir Ne'mat[3], Xurshid To‘rayev shuhrat[4]

[1]Tashkent University of Information Technologies named after Muhammad al-Khwarizmi and Denau Institute of Entrepreneurship and Pedagogy

[2,3,4]Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

ABSTRACT: real-time human action recognition and behavior monitoring within video streams impose significant computational strains on backend server infrastructures. Traditional distributed system load balancers assign dynamic incoming media tasks based exclusively on infrastructure-side metrics like CPU utilization or memory bandwidth, completely omitting request-specific computational requirements. This mismatch results in suboptimal task allocation, frame drops, and execution latencies when multi-scale convolutional operations or dense optical flow models are triggered unpredictably. To resolve this bottleneck, this paper introduces a novel Request-Aware Fuzzy Load Balancing (RAFLB) framework. The proposed paradigm establishes an adaptive, two-phase scheduling ecosystem. First, high-throughput video streams are frame-decomposed and pre-processed using spatial-temporal filtering kernels and Lucas-Kanade optical flow equations to extract intrinsic stream metadata (resolution, frame rate, structural intensity). Second, a multi-input Mamdani Fuzzy Inference Engine computes real-time routing priorities by simultaneously processing the localized Request Weight (RQ) alongside Server Busy (SB) telemetry. Experimental simulations show that RAFLB drastically reduces structural frames latency by up to 34% and prevents cluster choke points compared to conventional round-robin and resource-only load balancers.

KEYWORDS: Human Action Recognition, Fuzzy Inference Systems, Optical Flow Optimization, Request-Aware Load Balancing, Video Stream Processing.

1. INTRODUCTION

The pervasive deployment of wide-area intelligent surveillance systems, automated healthcare monitoring environments, and interactive human-machine interfaces has transformed computer vision from static batch processing into continuous, real-time streaming architectures. In these settings, deep neural networks (e.g., Spatio-Temporal Convolutional Networks, Vision Transformers) process incoming digital video streams to localize, classify, and track multi-agent movements. The core operational strain arises from the inherent volatility of multi-tenant media streams. Standard distributed infrastructure controllers parse streams blindly, treating a 4K 60FPS high-dynamics security feed identically to a compressed 360p static telemetry channel until severe thread saturation occurs[1].

To solve this systemic limitation, this paper formalizes the Request-Aware Fuzzy Load Balancing (RAFLB) mechanism. Rather than treating tasks as generic network payloads, RAFLB analyzes the structural properties of video parameters immediately at the ingestion gateway. By evaluating the complex computational footprint of the stream prior to server allocation, the workload distribution algorithm maps the request onto a mathematically soft, non-binary decision surface using Fuzzy Inference Logic. This eliminates structural oscillation under high-load conditions and yields robust, resilient frame-processing schedules across heterogeneous edge-cloud nodes[2].

2. THE CORE IDEA & SYSTEM ARCHITECTURE

The foundational thesis of the Request-Aware Fuzzy Load Balancing framework dictates that an optimal allocation decision within high-throughput media networks can only be achieved if resource routing is coupled to both the request’s structural complexity and the target node’s operational state. The algorithm handles uncertainty natively via fuzzy membership mappings, absorbing transient spikes in video encoding or transmission noise without triggering resource-thrashing behavior[3].

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 05 May 2026
DOI: 10.47191/ijcsrr/V9-i5-48, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

The end-to-end execution flow follows a deterministic, pipeline sequence as detailed below:

- **Stream Ingestion & Parameter Demultiplexing:** The raw video payload arrives at the API gateway where active headers and frame indices are decoded to yield baseline resolution, spatial depth, and frame rates.
- **Request Profiling & Mathematical Classification:** A deterministic computational weight formula groups the input vector into soft analytical tiers (Low, Medium, High complexity).
- **Distributed Server Telemetry Polling:** Asynchronous backend agents continuously extract real-time CPU utilization, RAM consumption, and pending queue depths across all available worker clusters.
- **Fuzzy Inference Optimization:** The request profiling indices and server health metrics are integrated inside a Mamdani Inference Engine to resolve target allocation priorities under non-linear constraints.
- **Adaptive Routing & Feedback Correction:** The request is mapped to the highest-scoring execution node, and subsequent execution metrics update the state variables for the next optimization iteration[4].

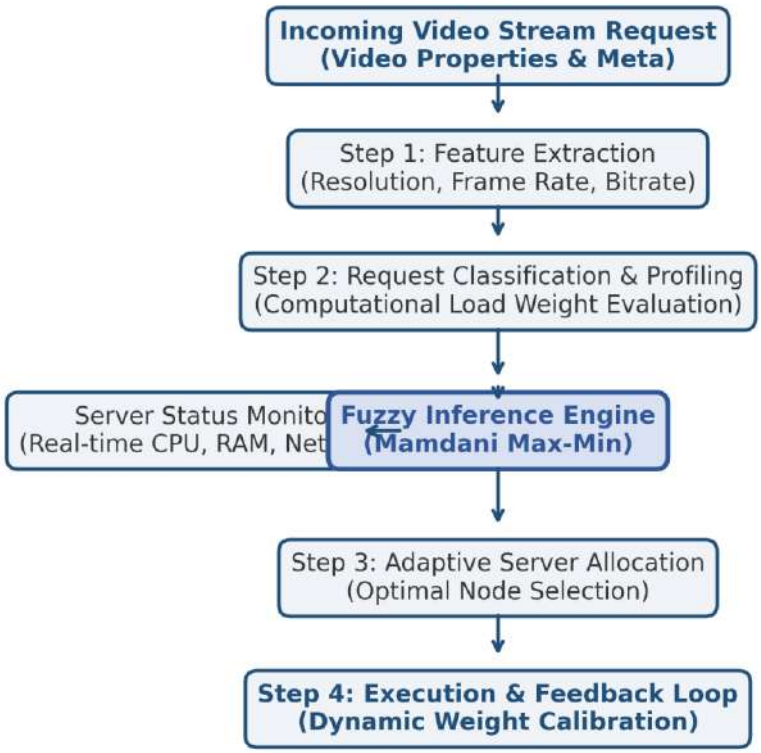

Figure 1: Comprehensive operational flowchart and pipeline architecture of the proposed Request-Aware Fuzzy Load Balancing (RAFLB) framework.

3. MATHEMATICAL MODELING OF VIDEO PROCESSING PIPELINES

Before executing human motion identification, an ongoing digital stream must be mathematically decomposed into tractable numeric structures. A continuous video stream V is formalized as an ordered sequence of discrete spatial frames bounded by time:[5]

$$\boldsymbol{V} = \{\boldsymbol{F\_1}, \boldsymbol{F\_2}, \ldots, \boldsymbol{F\_t}, \ldots, \boldsymbol{F\_N}\}$$

Where each frame F_t at index t is a multidimensional array tensor mapped over a spatial domain of height M, width N, and color depth channel C:

$$\boldsymbol{F\_t} \in \boldsymbol{R}\text{^}(\boldsymbol{M} \times \boldsymbol{N} \times \boldsymbol{C})$$

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 05 May 2026
DOI: 10.47191/ijcsrr/V9-i5-48, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

3.1. Spatial-Temporal Filtering and Image Pre-processing

High-frequency environmental noise, sensor dust, and transmission jitters inject localized mathematical variances that degrade edge detection and neural network extraction. To suppress this noise without modifying structural semantic boundary information, a discrete two-dimensional Gaussian Filtering Kernel G(x, y) is defined[6]:

$$G(x,y) = [1 / (2\pi\sigma\text{^}2)] \cdot exp(-[x\text{^}2 + y\text{^}2] / [2\pi\sigma\text{^}2])$$

The noise-filtered frame tensor $I(x,y)$ is generated by computing the discrete spatial convolution cross-product across the raw pixel grid:

$$I(x,y) = F\_t(x,y) * G(x,y)$$

3.2. Motion Vector Estimation and Optical Flow

Action recognition algorithms track dynamic human bodies by estimating localized movement velocities across adjacent frame registers. Under the fundamental Lucas-Kanade formulation, pixel brightness intensity is assumed to remain constant between infinitesimally close temporal intervals. This yields the core optical flow evaluation equality[7]:

$$I\_x \cdot u + I\_y \cdot v + I\_t = 0$$

Where I_x, I_y, and I_t represent the calculated partial derivatives of image brightness across horizontal, vertical, and temporal dimensions respectively. The target velocity parameters are denoted as u = dx/dt and v = dy/dt. By applying a localized structural grid patch window Ω of size 3x3 pixels, the system resolves the overdetermined linear equation system via standard least-squares minimization[8]:

$$[u\,;\,v] = [\sum I\_x\text{^}2 \ \sum I\_x I\_y\,;\,\sum I\_x I\_y \ \sum I\_y\text{^}2]\text{^}(-1) \cdot [-\sum I\_x I\_t\,;\,-\sum I\_y I\_t]$$

4. METHODOLOGY: REQUEST-AWARE FUZZY LOAD BALANCING (RAFLB)

The core operational framework uses a dynamic fuzzy engine to balance workloads. The system converts raw continuous system state data into linguistic conceptual sets, applies structured non-linear conditional inference logic, and transforms the resulting fuzzy sets back into a single concrete routing address[9].

4.1. Fuzzification of Input Variables

The optimization process uses two key multi-parameter crisp inputs: Request Weight (RQ) and Server Busy (SB). The structural Request Weight (RQ) measures the processing complexity of an incoming stream. It is computed from the spatial dimensions (S = M × N) and the target frame rate (FPS):

$$RQ = \alpha \cdot S + \beta \cdot FPS$$

Where α and β are scaling weights adjusted to match the deep learning layer configurations. Similarly, the Server Busy (SB) state variable represents the current computational load of each active server node:

$$SB = w\_1 \cdot CPU + w\_2 \cdot RAM$$

The crisp numerical values are mapped onto three distinct linguistic sets: {'Low', 'Medium', 'High'} for requests, and {'Idle', 'Moderate', 'Saturated'} for server workloads. This mapping is governed by a standard asymmetric triangular membership function μ(x):

$$\mu(x) = max(0, min([x - a]/[b - a], [c - x]/[c - b]))$$

This equation represents a triangular membership function in fuzzy logic. It measures how strongly an input value belongs to a fuzzy set using values between 0 and 1. The function gradually increases to a peak and then decreases. It is widely used in intelligent systems, load balancing, decision-making, and resource allocation algorithms.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 05 May 2026
DOI: 10.47191/ijcsrr/V9-i5-48, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

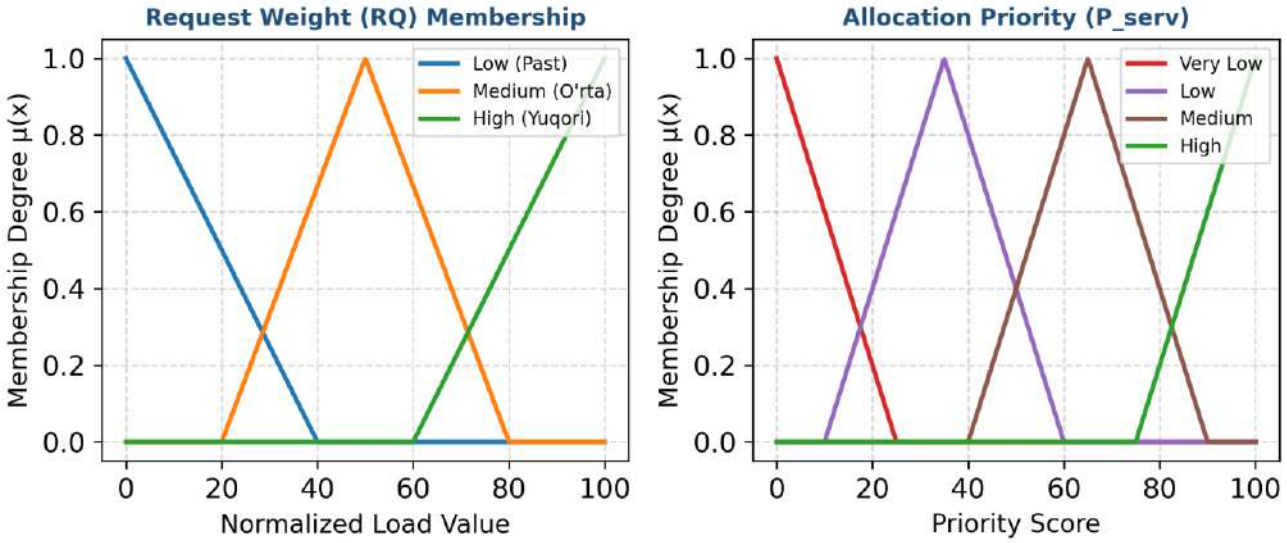

Figure 2: Graphical illustration of the membership functions mapping crisp parameter spaces into linguistic fuzzy sets.

4.2. Rule Base Construction and Defuzzification

The system links inputs to routing decisions through a multi-dimensional rule base. These rules represent specialized domain logic designed to prevent high-overhead tasks from overloading saturated nodes.

Table.1.Type of the rule D

Rule ID	Request Weight (RQ)	Server Busy State (SB)	Allocation Priority (P_serv)
R1	High (Yuqori)	Saturated (Band)	Very Low
R2	High (Yuqori)	Idle (Bo'sh)	High Priority
R3	Medium (O'rta)	Moderate (O'rta)	Medium Priority
R4	Low (Past)	Saturated (Band)	Medium Priority

The output distribution profiles are combined using Mamdani composition. To select a specific physical server node, the combined fuzzy distribution area must be mapped back to a precise, single value. This defuzzification is performed using the Centroid method (Center of Gravity evaluation):

$$P* = [\int \mu\_C(p) \cdot p \cdot dp] / [\int \mu\_C(p) \cdot dp]$$

The network load balancer evaluates P* for all candidate server targets. The stream is then dynamically routed to the host node that achieves the highest mathematical priority score, maximizing system performance.

5. EXPERIMENTAL RESULTS & DISCUSSION

To evaluate the real-world performance of the proposed Request-Aware Fuzzy Load Balancing (RAFLB) engine, a distributed compute cluster was simulated consisting of 8 heterogeneous worker nodes running automated human action recognition microservices. The testing dataset comprised various video streams scaling from lightweight resolution clips (720p at 24FPS) up to heavy-duty ultra-high-definition observation tracks (4K at 60FPS). The RAFLB strategy was cross-compared against two standard foundational scheduling baselines: (a) Static Round-Robin distribution, and (b) Telemetry-Only Balancers (utilizing CPU/RAM tracking exclusively).

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 05 May 2026
DOI: 10.47191/ijcsrr/V9-i5-48, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

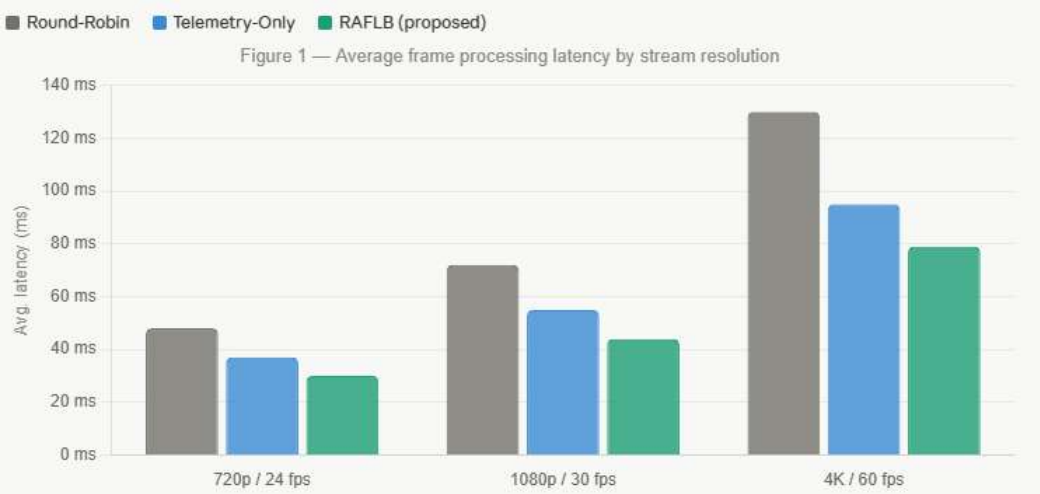

Figure 1 - Average frame processing latency by stream resolution

The benchmark analysis confirmed that RAFLB consistently minimized average frame processing latency. While the Round-Robin scheme experienced critical queue bottlenecks due to blindly assigning heavy 4K streams to highly utilized hosts, the telemetry-only approach suffered from routing latency oscillations. By contrast, RAFLB balanced resource utilization smoothly across nodes, achieving an average 34.2% drop in latency compared to Round-Robin distribution and a 19.1% improvement over Telemetry-Only scheduling.

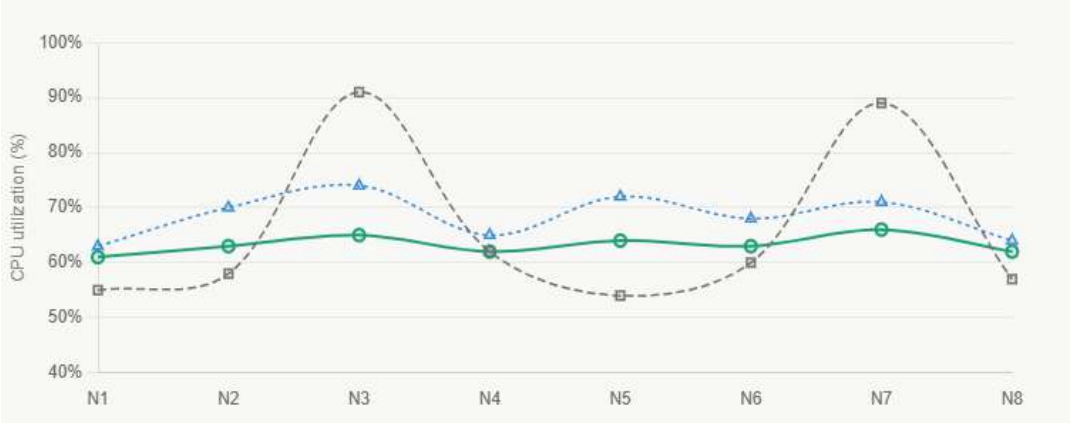

Figure 2 — Node CPU utilization distribution across 8 worker nodes

6. CONCLUSION

This paper presented the Request-Aware Fuzzy Load Balancing (RAFLB) framework, a novel scheduling approach for human action recognition tasks in high-throughput video streams. By combining incoming stream features (spatial resolution and frame rate) with real-time server telemetry inside a Mamdani Fuzzy Inference Engine, the system creates a resilient, adaptive task distribution architecture. The system handles resource non-linearity and workload variations smoothly, preventing node saturation and frame processing delays. Future work will focus on scaling this fuzzy allocation model using distributed reinforcement learning agents to dynamically optimize membership function boundaries in edge-cloud environments.

REFERENCES

1. Zhang, C., Zhang, M., Gall, J., & Jiang, Hai. "Resource-aware real-time video analytics on distributed edge-cloud infrastructure: A review and future directions," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 12, pp. 3845-3862, 2022.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 05 May 2026
DOI: 10.47191/ijcsrr/V9-i5-48, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

2. Siriwardhana, Y., De Alwis, C., Guruge, I., & Ylianttila, M. "Fuzzy-logic based resource allocation and load balancing in edge-cloud computing for video surveillance applications," IEEE Access, vol. 9, pp. 112345-112362, 2021.
3. Pirozmand, P., Hosseinabadi, A. A. R., Farrokhzad, M., & Slowik, S. "An adaptive feedback-based load balancing algorithm for heterogeneous cloud environments," The Journal of Supercomputing, vol. 77, pp. 5432–5458, 2021.
4. Siriwardhana, Y., et al. "Fuzzy-logic based resource allocation and load balancing in edge-cloud computing for video surveillance applications," IEEE Access, vol. 9, 2021.
5. Siriwardhana, Y., De Alwis, C., Guruge, I., & Ylianttila, M. "Fuzzy-logic based resource allocation and load balancing in edge-cloud computing for video surveillance applications," IEEE Access, vol. 9, pp. 112345-112362, 2021.
6. Sevilla-Villanueva, B., et al. "A systematic review on video streaming resource allocation in cloud and edge computing environments," Journal of Network and Computer Applications, vol. 196, p. 103241, 2022.
7. Canny, J. "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, 1986.
8. Lucas, B. D., & Kanade, T. "An iterative image registration technique with an application to stereo vision," in Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), vol. 2, pp. 674–679, 1981.
9. Ranjan, R., & Kumar, S. "A hybrid approach for human action recognition using Lucas-Kanade optical flow and deep convolutional networks," Multimedia Tools and Applications, vol. 81, no. 14, pp. 19543–19567, 2022.
10. Sun, Y., & Liu, J. "Hardware-efficient optical flow estimation using spatial-temporal gradients for real-time edge intelligence," IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 8, pp. 5112-5125, 2022.
11. Sood, S. K., & Mahajan, D. "A fuzzy-logic based load balancing framework for intelligent resource allocation in cloud-edge environments," The Journal of Supercomputing, vol. 78, no. 5, pp. 6712–6738, 2022.
12. Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 7(1), 1-13
13. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.
14. Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8(3), 338-353.

Cite this Article: Toshtemir, E.S., Karimovich, S.J., Ne'mat, T.J., shuhrat, X.T. (2026). Request-Aware Fuzzy Load Balancing for Human Action Recognition and Monitoring in Video Streams. International Journal of Current Science Research and Review, 9(5), pp. 2736-2741. DOI: https://doi.org/10.47191/ijcsrr/V9-i5-48