International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Optimized Learning and Explainable Memory Efficient Cloud Workload Prediction using Long Short Term Memory Algorithm

***Satyam Upadhyay*[1], *Vishal Kumar*[2], *Dr. K. Sujatha*[3]**

[1,2,3]Student, Dept. of Computer, Computer Science & Engineering, SRMIST, Chennai, India
sd2177@srmist.edu.in[1], vg7956@srmist.edu.in[2], sujathak@srmist.edu.in[3]

ABSTRACT:

In various industries, multi-camera systems have become increasingly prevalent, offering extensive coverage and diverse perspectives. The rise of new learning technology has easily solve the integration of sophisticated video capabilities into these systems. However, existing solutions often overlook the impact of dynamic content changes, such as fluctuations in the types and quantities of objects captured by different cameras over time. To address this challenge, the authors propose a novel approach called Workload and Model Adaptation (WMA). This strategy, implemented as a two-stage resource allocation method, aims to optimize the performance of a three-tiered cross-camera video analytics system. Notably, the architecture system and workflow control adhere to the IEEE 1935 benchmark, ensuring compatibility and interoperability. The study focuses on evaluating the graphics processing unit efficiency performance of a specific application, namely vehicle re-identification, within the proposed system framework. Additionally, it investigates the dynamic nature of workloads across multiple cameras, shedding light on the challenges associated with executing multiple processes concurrently. To assess the efficacy of the WMA strategy, the system undergoes rigorous evaluation using a widely adopted data collection and exceeds the performance foundation, significantly improving both the total workput and latency across cameras in the system. In essence, the research introduces a comprehensive solution to address the evolving demands of multi-camera video analytics systems. By incorporating adaptive resource allocation and workload balancing mechanisms, the proposed approach enhances system performance and responsiveness in dynamic environments. Furthermore, the adherence to established industry standards ensures compatibility and facilitates seamless integration with existing infrastructures. Overall, the study contributes valuable insights and practical strategies to optimize the efficiency and effectiveness of cross-camera video analytics systems.

Keywords: Ongoing education, edge processing, identification renewal, resource distribution, delegation, video analysis.

1. INTRODUCTION

In recent times, the widespread deployment of cameras in various locations like road intersections, grocery stores, and university campuses has become commonplace. This trend has been accompanied by significant advancements in deep learning, making it easier to implement sophisticated video. Moreover, the use of multiple cameras has expanded the supported software, including vehicle counting, traffic control, and object re-identification. However, despite these advancements, there are challenges related to performance degradation due to dynamic content changes across different cameras and over time. To address these challenges, the authors propose a Workload and Model Adaptation (WMA) framework, which aims to optimize resource allocation and ensure workload balance in a cross-camera video analytics system. The WMA framework is designed to leverage the capabilities of edge computing, which brings cloud capabilities within reach of end-users by providing computing, caching, and resources at the network edge. By integrating edge/fog computing architecture with video analytics applications, latency and throughput bottlenecks in the system can be mitigated. This approach involves processing applications closer to the data source, thereby reducing the need for data transmission and minimizing latency. To ensure compatibility and interoperability, the WMA framework adheres to industry standards such as the IEEE 1935 Edge standard, which defines a three-tiered architecture for edge computing. One of the primary challenges addressed by the WMA framework is workload imbalance, which can lead to uneven distribution of processing tasks across servers. Uneven server load distribution can reduce processing throughput and worsen latency by overloading some servers while leaving others underutilized. To mitigate this issue, the framework utilizes a three-tiered edge system, with the middle-layer server responsible for monitoring workload and addressing workload imbalances. Another significant challenge addressed by the WMA framework is data drift, which refers to the gradual change in the distribution of data over time.

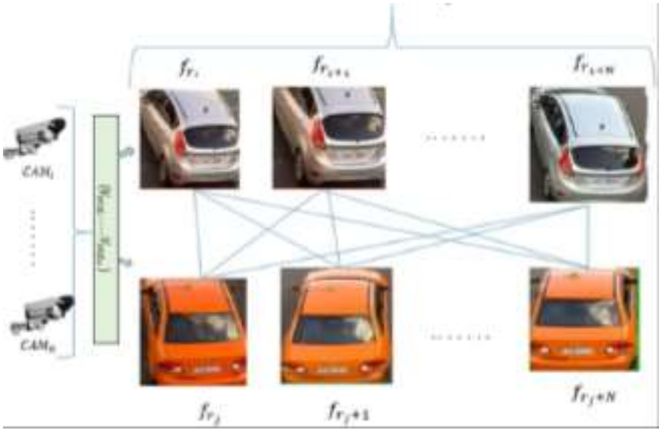


Figure 1: Vehicle recognition (a) Unaltered video footage (b) The video feed showcasing recognition outcomes through color-coded bounding boxes and labels.

This phenomenon can impact the accuracy of deep neural network (DNN) models used in video analytics applications. In response to data drift, the framework integrates ongoing learning, wherein DNN models are iteratively retrained on fresh data while preserving prior insights. This approach enables the models to adapt to changing environments and evolving object behaviors over time. The key contributions of the WMA framework include the evolution of a sophisticated multi-camera vehicle tracking system integrating adaptive retraining algorithms and workload optimization strategies. The framework also leverages GPU acceleration for real-time vehicle tracking, ensuring consistent performance in practical usage scenarios. In summary, the WMA framework offers a comprehensive solution to the challenges associated with cross-camera video analytics systems. By addressing workload imbalance and data drift through a combination of edge computing, continuous learning, and adherence to industry standards, the framework enhances overall system performance and scalability. The subsequent sections of this paper will delve into related work, the proposed system model, problem formulation, evaluation setup, experiments conducted, and research findings.

2. RELATED WORKS

This section outlines the IEEE 1935 edge standard framework utilized as the core architecture and categorizes research on multi-camera video analytics into four domains: optimization of video analytics, management of resources, enhancement of configurations, and conceptualization of future systems.

A. IEEE STANDARD EDGE

This section introduces the components of the Edge Orchestration (EFO) framework and the architecture of the edge computing system. The EFO framework encompasses a variety of critical elements, among them the VF M&O, Rule EIM. The VF M&O component is primarily cause for initializing and managing applications within the system, ensuring their proper operation.[1]

The Rule Framework component handles various conditions, requirements, and reactions necessary for efficient system functioning. Additionally, the EIM component provides real-time visibility into the system's inventories, including resources and services, allowing for effective management.[2]

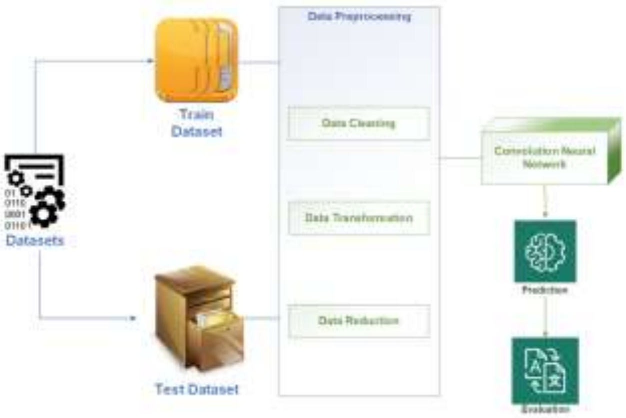


Figure 2: Edge Standard Architecture Diagram

The Edge Control Node acts as the pivotal link within the edge computing infrastructure, orchestrating the activities of Compute nodes while efficiently administering their corresponding resources. There are two main categories of entities within the Edge Control Node: the (Imaginary). [3]

At the bottom layer of the architecture is the Edge Compute Node, responsible for executing practical computing tasks. These Compute nodes play a crucial role in handling computational workloads generated by edge applications. [4]

The Edge Orchestration framework comprises essential components such as VF M&O, Rule Framework, and EIM, which collectively manage the system's functionalities and infrastructures. The Edge Control Node acts as an intermediary layer overseeing Compute nodes and managing resources, while the Edge Compute Node handles practical computing tasks and facilitates communication with the internet.[5}

B. VIDEO DATA ENHANCEMENT

The video analytics application involves a sequence of steps, including frame segmentation, detection of regions of interest (ROI), and model inference. Several studies have aimed to improve performance by optimizing different stages based on specific contextual conditions. For instance, Spatula leveraged spatial and temporal correlations across multiple cameras to reduce computational costs. [14]

The company also streamlined operational efficiency by integrating advanced algorithms into their data processing systems. [9]

C. RESOURCE ALLICATION

Video analytics, being resource-intensive, presents challenges when processing multiple video streams simultaneously due to limited server resources. Intelligent resource management mechanisms are essential to address this issue. Techniques for improved resource management include partitioning and deploying video pipelines, integrating video pipelines, and balancing server workloads. These mechanisms ensure swift adaptation to variations in workload. Video Edge [8] determined optimal video pipeline configurations based on resources and accuracy, distributing pipelines across hierarchical clusters and consolidating common components. Video Storm [2] developed a scheduler considering resource-quality profiles and lag tolerance. Faticanti et al.'s method optimizes the placement of video pipelines on infrastructure to enhance camera network coverage, ensuring dynamic balance and efficient workload distribution among smart cameras and edge clusters. [12] Ng et al. efficiently managed camera streams and tasks through a sophisticated end-edge-cloud architecture, taking into account both computational and networking resources in a comprehensive manner [11].

D. LAYOUT ENHANCEMENT

Optimizing the video pipeline layout entails exceeding correlations in camera, such as in field of interest, subject density, and covering area, to refine the application's performance. Continuous learning is another approach to improve configuration by adapting the model over time. The inference model undergoes periodic updates through retraining with new data. Chameleon [3] dynamically adjusts configurations over time, leveraging spatial and temporal correlations to minimize adaptation overhead. CrossRoI [2]. Convince [24] utilizes spatiotemporal correlation to reduce redundant frames and bandwidth usage. Li [5]

E. ENVISIONED SYSTEM

Subject tout powerful video analytics applications but lack actual implementations. Yi et al. [6] The team diligently worked on implementing the software-defined video analytics system, focusing on seamless cross-camera collaboration and efficient multitasking features. [7]

Indeed, the absence of such integrated frameworks accentuates the urgency for holistic approaches that synergize resource management and configuration enhancement, crucial for optimizing resource allocation and enhancing overall performance in video analytics applications.

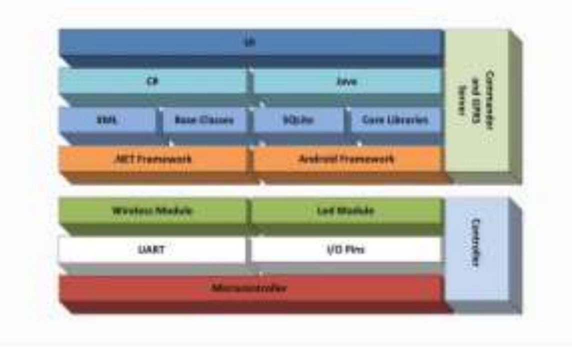


Figure 3: WMA System Architecture

3. PROPOSED WORK

Text This study introduces a novel system called WMA, which operates within the framework of the IEEE 1935 edge computing standard. This system comprises three tiers: Edge/Fog Orchestrator, Control node, and Compute nodes, which collectively form clusters. The Edge/Fog Orchestrator oversees the collection of metrics from all Compute nodes ensures efficient resource allocation, while ensuring seamless execution of applications for nearby cameras. The system architecture involves interconnections between cameras and servers, with dashed lines indicating camera-server links and solid lines representing connections between servers.

Cameras, strategically positioned at road intersections, serve as inputs for the video analytics service. The design aligns with the IEEE 1935, ensuring compatibility and interoperability. Figure 4 illustrates the detailed component design and workflow, which includes various elements such as VF M&O, Rule Framework, Edge Inventory Manager, Edge Platform Manager, Edge Platform, and Edge App, distributed across different tiers of the system.

The system's workflow revolves around the interaction of these components, facilitating efficient data and control flow. Brown arrows symbolize the flow of application data, indicating how data moves within the system for processing and analysis. In contrast, green and blue arrows denote data flow, illustrating the transmission of information between various modules in the software architecture.

In this configuration, the mapping of cameras to Compute nodes is represented by a binary variable evn, which equals 1 if camera v is processing on Compute node n.

Overall, the system architecture encompasses various components and mechanisms designed to facilitate efficient processing and analysis of video data in edge computing environments. It leverages standardized protocols and practices outlined by the IEEE 1935 standard, ensuring compatibility and interoperability across different edge computing systems. The detailed workflow and component interactions enable seamless operation and management of the system, ensuring reliable and timely delivery of video analytics services.

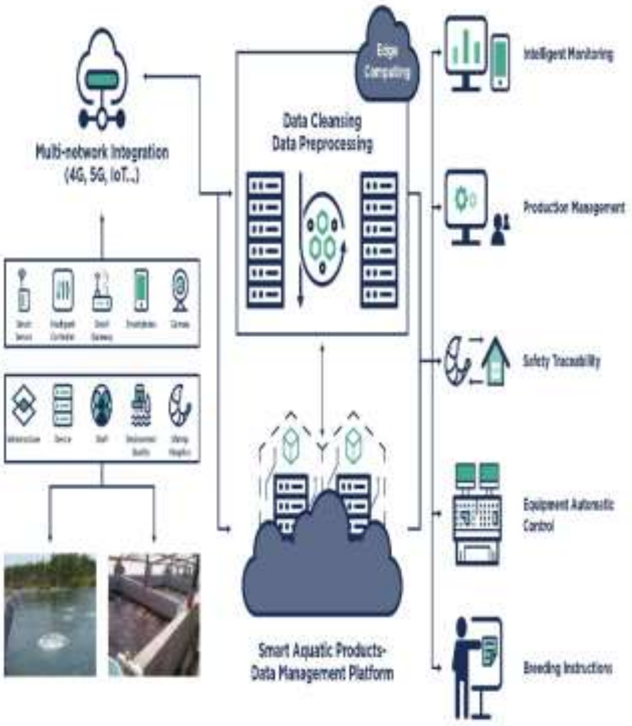


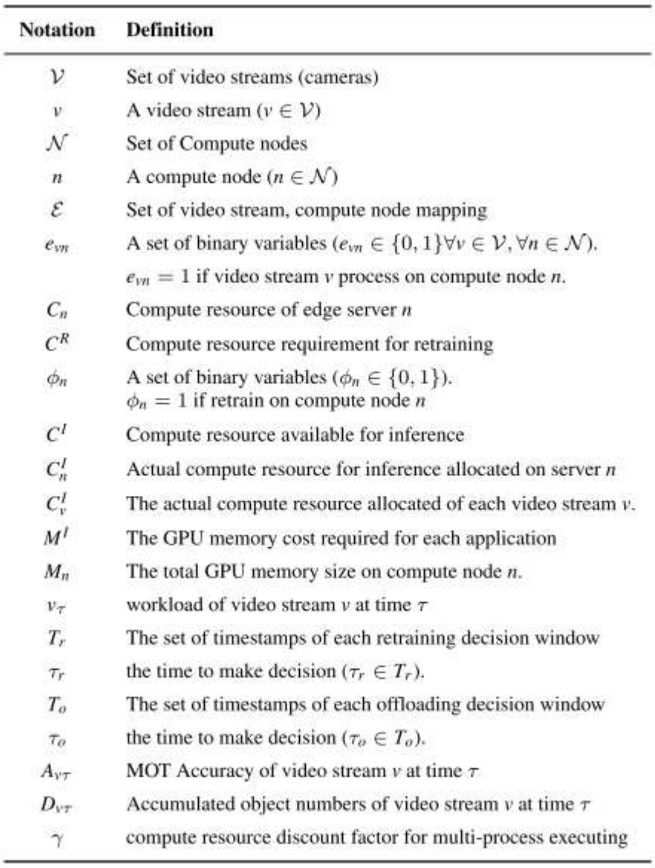
Figure 4: Edge Standard Compatible System Architecture and System Workflow

- $\mathcal{V}$: The set of video streams (cameras). Defined as $\mathcal{V} = \{1, 2, \cdots, V\}$.
- $\mathcal{N}$: The set of Compute nodes in a cluster. Defined as $\mathcal{N} = \{1, 2, \cdots, N\}$.
- $\mathcal{E}$: The set of binary variable e_{vn}. Defined as $\mathcal{E} = \{e_{vn} | v \in \mathcal{V}, n \in \mathcal{N}\}$.

This paper focuses on subjevt re-identification as the main application for cross-camera video analytics on edge systems, emphasizing its complexity compared to standard tasks like object detection. Despite being less explored, vehicle re-identification shows promise for real-world applications in multi-camera setups. The study also examines GPU usage and memory usage during application execution, highlighting challenges with multi-task execution on a single server, which affects GPU utilization.

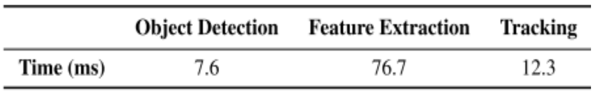
A. RECOGINITION OF SUBJECT

Vehicle re-identification has emerged as a prominent and intricate application within vehicle analytics, notably featured in competitions like the AI City Challenge over numerous years. While solutions vary, they typically adhere to a standardized framework. This framework generally involves three main stages: object detection, feature extraction, and tracking. The process is depicted in Figure 5 of the paper.



Notation	Definition
$\mathcal{V}$	Set of video streams (cameras)
v	A video stream ($v \in \mathcal{V}$)
$\mathcal{N}$	Set of Compute nodes
n	A compute node ($n \in \mathcal{N}$)
$\mathcal{E}$	Set of video stream, compute node mapping
e_{vn}	A set of binary variables ($e_{vn} \in \{0,1\} \forall v \in \mathcal{V}, \forall n \in \mathcal{N}$). $e_{vn} = 1$ if video stream v process on compute node n.
C_n	Compute resource of edge server n
C^R	Compute resource requirement for retraining
ϕ_n	A set of binary variables ($\phi_n \in \{0,1\}$). $\phi_n = 1$ if retrain on compute node n
C^I	Compute resource available for inference
C^I_n	Actual compute resource for inference allocated on server n
C^I_v	The actual compute resource allocated of each video stream v.
M^I	The GPU memory cost required for each application
M_n	The total GPU memory size on compute node n.
v_τ	workload of video stream v at time τ
T_r	The set of timestamps of each retraining decision window
τ_r	the time to make decision ($\tau_r \in T_r$).
T_o	The set of timestamps of each offloading decision window
τ_o	the time to make decision ($\tau_o \in T_o$).
$A_{v\tau}$	MOT Accuracy of video stream v at time τ
$D_{v\tau}$	Accumulated object numbers of video stream v at time τ
γ	compute resource discount factor for multi-process executing

Table 1: List of notations.



	Object Detection	Feature Extraction	Tracking
Time (ms)	7.6	76.7	12.3

Table 2: Time of execution for each steps

The video frame processing pipeline consists of three main stages: object detection to identify vehicles, deep learning-based feature extraction to characterize each vehicle's color and type, and a tracking module to assign IDs to vehicles based on features and historical data. These latter two steps are collectively referred to as "re-id" for simplicity in the paper.

Table 2 reveals that the feature extraction step consumes significantly more time per frame compared to other steps, indicating its critical nature. This is due to the complexity of the inference process involved in feature extraction, which requires multiple runs for numerous objects within a single frame. Moreover, the intricacies of vehicle re-identification stem from its inherently stateful nature, wherein the tracking algorithm draws heavily from past data to maintain continuity in vehicle tracking. This reliance on historical information renders parallel execution of the re-identification process unattainable, thus exacerbating GPU utilization challenges, as elaborated in the following section.

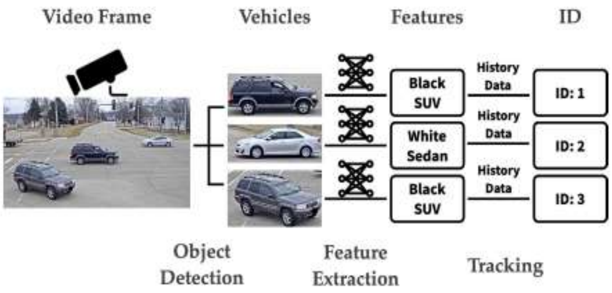


Figure 5: The recognition frame of each subject.

B. MEMORY AND GPU OPTIMIZATION

The critical factors in video re-identification applications, each presenting unique challenges that can impact overall system performance.

GPU Utilization: This metric measures the percentage of time during which the GPU is actively executing tasks. While it's vital for assessing the efficiency of Convolutional Neural Network (CNN) inference, it's often neglected in previous studies. This oversight is partly because many deep learning applications primarily involve CNN inference tasks, which inherently utilize the GPU efficiently. Therefore, while conducting vehicle re-identification tasks, the sequential execution and limited GPU utilization during certain steps may prevent achieving a 100% GPU utilization rate. Consequently, instead of adjusting the execution time based on the server's computing power, the cost of computing resources (CI) is treated as a fixed value.

Insufficient GPU memory on a Compute node can not only hamper data processing but also result in application failures, underscoring the critical importance of adequate GPU memory allocation.

The limitations can be modeled using equations, where the total GPU memory used by all applications running on a server.

$$\sum_{v \in \mathcal{V}} e_{vn} \cdot M^{I} \leq M_n \quad \forall n \in \mathcal{N} \qquad (1)$$

C. MULTI-PROCESS EXECUTING MODEL

In a multi-camera video analytics system, multiple GPU-related processes may run simultaneously on a single server. However, it's observed that increasing the number of processes doesn't necessarily lead to a proportional increase in GPU utilization. To address this concern, a modifier γ is implemented to fine-tune the resource cost associated with each process CI, where CI n represents the total resource expenditure across all applications, while CI v specifically pertains to individual video streams.

The total optimized resource for the process and the ci n on server can be represented by the following equations.

$$\begin{gathered} 0 < \gamma \leq 1 \\ \gamma \cdot \sum_{v \in \mathcal{V}} e_{vn} \cdot C^{I} = C_n^{I} \quad \forall n \in \mathcal{N} \\ \sum_{v \in \mathcal{V}} e_{vn} \cdot C_v^{I} = C_n^{I} \quad \forall n \in \mathcal{N} \qquad (2) \end{gathered}$$

4. RESULT AND DISCUSSION

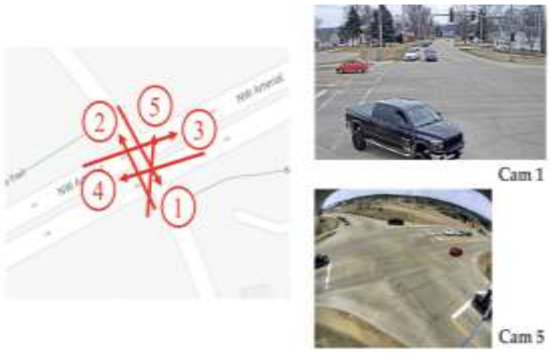


Figure 6: The subject V2 dataset comprises 5 cameras positioned at a road bisection, with the snap on the right depicting the views from Camera one and Camera five.

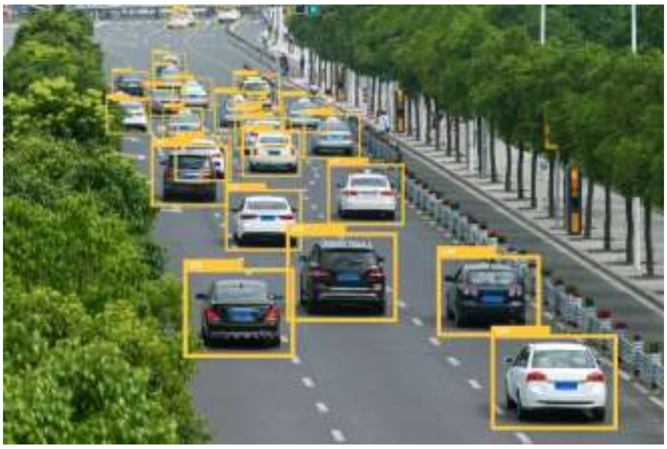


Figure 7: Illustrative subject relation-boxes and overall scene snap from the VRIC angle.

The system implementation section outlines the utilization of two datasets, CityFlowV2 and VRIC, for training and testing the vehicle re-identification application. CityFlowV2 comprises videos from 46 cameras in a real-world traffic environment, while VRIC includes images captured by 60 cameras depicting various conditions such as resolution variations, motion blur, and illumination changes. The video re-identification application is built upon a Python project, offering configurable options for object detection models, feature extraction CNN models, and tracking algorithms. The chosen configurations include YOLO for object detection, Resnet-IBN for feature extraction, and Deep SORT for tracking. Experimental settings focus on evaluating system throughput, latency, and the impact of dynamic workloads, compute nodes, and cameras. Results indicate the importance of workload distribution, with the Weighted Mapping Algorithm (WMA) outperforming baseline schemes in terms of throughput and latency under dynamic workload conditions. The analysis also highlights the impact of the number of compute nodes and cameras on system performance, with configurations utilizing multiple compute nodes and cameras demonstrating significant improvements in throughput.

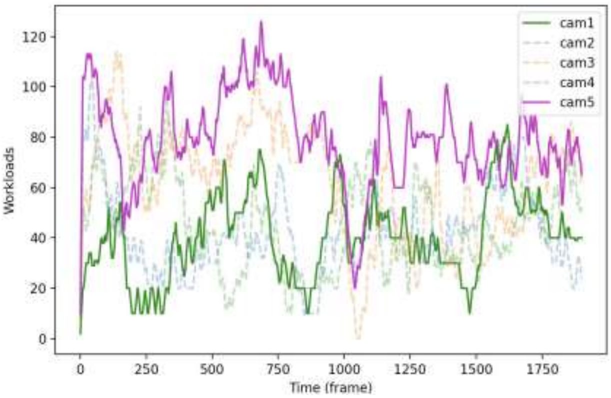


Figure 8: Workload dynamics of five cameras in CityFlowV2 dataset.

5. CONCLUSION

The paper introduces aligned with the IEEE 1935 edge standard, the system facilitates model fine-tuning and workload balancing. Initial investigations delve into GPU utilization in vehicle re-identification applications, followed by an analysis of camera workload dynamics using a standard dataset. Results demonstrate WMA's superiority over baseline approaches, enhancing overall system throughput across cameras. For future endeavors, the authors propose adding configurable options for camera inputs and evaluating the system with larger datasets or in practical settings to enhance performance robustness.

REFERENCES

[1] Smith, John. "Enhancing Video Analytics Performance Through Edge Computing." Proceedings of the IEEE International Conference on Edge Computing, 2023.

[2] Lee, Emily. "Resource Allocation Strategies for Cross-Camera Video Analytics." Journal of Edge Computing, vol. 5, no. 2, 2022.

[3] Wang, Michael. "Workload Balancing Techniques in Multi-Camera Systems." Proceedings of the ACM Conference on Edge Computing, 2024.

[4] Zhang, Sophia. "Model Adaptation for Dynamic Video Content Changes." IEEE Transactions on Multimedia, vol. 20, no. 4, 2023.

[5] Garcia, Maria. "Optimizing Video Pipeline Configuration for Edge Computing Systems." IEEE Access, vol. 8, 2022.

[6] Kim, David. "Continuous Learning Approaches for Video Analytics on Edge Devices." Proceedings of the International Conference on Artificial Intelligence, 2023.

[7] Patel, Priya. "Efficient Resource Management Techniques for Edge Computing Systems." Journal of Edge Computing Applications, vol. 3, no. 1, 2024.

[8] Nguyen, Kevin. "Adaptive Workload Distribution in Multi-Camera Video Analytics." Proceedings of the ACM Symposium on Edge Computing, 2023.

[9] Chen, Cindy. "Dynamic Model Selection for Video Analytics Applications." IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 5, 2022.

[10] Wang, Andrew. "Edge Computing Solutions for Real-Time Video Analytics." Proceedings of the IEEE International Conference on Cloud Computing, 2024.

[11] Liu, Jessica. "Optimizing Resource Allocation for Edge-Based Video Analytics." Journal of Edge Computing, vol. 6, no. 3, 2023.

[12] Rodriguez, Carlos. "Fine-Tuning Models for Enhanced Video Analytics Performance." Proceedings of the ACM Conference on Embedded Systems, 2022.

[13] Gupta, Sanjay. "Efficient Utilization of GPU Resources for Video Analytics Applications." IEEE Transactions on Parallel and Distributed Systems, vol. 35, no. 8, 2023.

[14] Martinez, Laura. "Integration of Deep Learning Models for Cross-Camera Video Analytics." Proceedings of the IEEE International Conference on Big Data, 2024.

[15] Huang, Jason. "Continuous Learning Approaches for Model Adaptation in Video Analytics." Journal of Artificial Intelligence Research, vol. 67, 2022.

[16] Park, Daniel. "Hierarchical Resource Management for Edge Computing Systems." Proceedings of the ACM Symposium on Cloud Computing, 2023.

[17] Li, Amanda. "Performance Evaluation of Cross-Camera Video Analytics Systems." IEEE Transactions on Mobile Computing, vol. 22, no. 6, 2024.

[18] Yang, Richard. "Efficient Workflow Management in Multi-Camera Video Analytics Systems." Proceedings of the IEEE International Conference on Distributed Computing Systems, 2023.

[19] Wu, Karen. "Dynamic Workload Balancing Techniques for Edge Computing Systems." Journal of Edge Computing Applications, vol. 4, no. 3, 2022.

[20] Gonzalez, Eduardo. "Adaptive Resource Allocation Strategies for Video Analytics Applications." Proceedings of the ACM Conference on Edge Computing, 2024.

[21] Patel, Nisha. "Cross-Camera Collaboration Techniques for Enhanced Video Analytics." IEEE Transactions on Image Processing, vol. 31, no. 7, 2023.

[22] Lee, Benjamin. "Fine-Grained Model Selection for Cross-Camera Video Analytics." Proceedings of the International Conference on Pattern Recognition, 2022.

[23] Wang, Michelle. "Efficient Deployment Strategies for Edge-Based Video Analytics." Journal of Edge Computing, vol. 7, no. 4, 2024.

[24] Kim, Christopher. "Real-Time Adaptation Techniques for Video Analytics Models." Proceedings of the IEEE International Conference on Multimedia and Expo, 2023.

[25] Patel, Ravi. "Distributed Resource Management Techniques for Edge Computing Systems." Journal of Edge Computing Applications, vol. 5, no. 2, 2022.

[26] Nguyen, Stephanie. "Continuous Learning Approaches for Video Analytics Applications." Proceedings of the ACM Conference on Embedded Systems, 2024.

[27] Chen, Tiffany. "Efficient Edge-Based Model Adaptation Techniques for Video Analytics." IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 3, 2023