ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

Enhancing the Intelligence of Real-Time Video Surveillance Systems with Automated Anomaly Detection and Response

(Area of Focus: Security Organs)

NDIZEYE Maurice[1]; Dr. Emmanuel BUGINGO[2]

University of Kigali

Publication Date: 2025/03/04

Abstract: This study addresses the critical need for advanced threat detection systems by enhancing real-time video surveillance with automated anomaly detection and response. It integrates machine learning algorithms for object and activity recognition, leveraging AI and parallel computing for efficient and accurate anomaly detection. The research evaluates the adaptability and effectiveness of machine learning models in diverse environments, emphasizing robust response protocols for identified anomalies.

Key findings highlight challenges in identifying suspects amidst distractions like vehicles, animals, and crowds. Advanced preprocessing and machine learning techniques significantly improved detection accuracy from 70.4% to 93%. Parallel computing reduced model training time from 71 minutes to just 8 minutes, showcasing its efficiency. The study recommends cost-effective solutions like building high-performance computing (HPC) clusters using Raspberry Pi nodes for training and employing Hadoop clusters for video stream processing. Deploying optimized models on scalable web servers enables real-time diagnostics and anomaly alerts, empowering security operators with faster and more actionable insights.

These contributions advance the development of intelligent, efficient, and scalable surveillance systems that address modern security challenges. The proposed methods improve decision-making and real-time responsiveness, providing a significant leap forward in video surveillance technology.

How to Cite: NDIZEYE Maurice; Dr. Emmanuel BUGINGO. (2025). Enhancing the Intelligence of Real-Time Video Surveillance Systems with Automated Anomaly Detection and Response (Area of Focus: Security Organs). *International Journal of Innovative Science and Research Technology*, 10(2), 1217-1225. https://doi.org/10.5281/zenodo.14959415.

I. INTRODUCTION

The rise of global security concerns has highlighted the limitations of traditional surveillance systems, which rely heavily on manual monitoring and delayed responses. These systems are often reactive, with their primary function being the provision of evidence after an incident rather than proactive threat detection and response.

Recent advancements in artificial intelligence (AI) and computer vision offer transformative potential in the security domain. This paper explores the development of an intelligent real-time video surveillance system capable of autonomously identifying and responding to potential threats, including theft, intrusion, vandalism, and violence. The goal is to leverage deep learning models and computer vision techniques to transform passive surveillance into an active and responsive security solution.

II. METHODOLOGY

A. Data Collection and Preprocessing

- **Data Sources:** Video datasets were collected from online websites and YAKIN Technology Ltd.'s surveillance system, including recordings from public spaces, offices, and parking areas.
- **Preprocessing:** Video streams were preprocessed to improve model performance by:

✓ Resizing frames to standardized dimensions for uniformity.
✓ Extracting keyframes to reduce redundant data.
✓ Enhancing image quality through noise reduction, histogram equalization, and contrast adjustment.

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

B. Feature Extraction and Selection

- Spatial features (object shapes and sizes) were extracted using Convolutional Neural Networks (CNNs).
- Temporal patterns (movement sequences) were analyzed using Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) units.
- Principal Component Analysis (PCA) was applied to reduce dimensionality, retaining only the most relevant features.

C. Parallel Computing for Training Efficiency

- Parallel computing techniques were applied using high-performance computing (HPC) clusters. These included:
- Building affordable HPC clusters with Raspberry Pi nodes.
- Utilizing GPU acceleration reduces model training time from 71 minutes to 8 minutes.

D. Model Development

- **Deep Learning Models:** The system utilized CNNs for spatial anomaly detection (e.g., unauthorized access) and RNNs for temporal anomaly recognition (e.g., suspicious movements).
- **Integration of Adaptive Learning:** A self-supervised learning mechanism allowed the system to adapt to evolving behaviors and environmental conditions.
- **Hyperparameter Optimization:** Techniques such as grid search and random search were employed to tune.

E. System Deployment and Real-Time Processing

- **Infrastructure:** Optimized models were deployed on scalable web servers capable of processing live video streams.
- **Video Stream Processing:** Hadoop clusters were used to handle large volumes of video data, enabling real-time processing and anomaly detection across multiple cameras simultaneously.
- **Model Quantization:** To optimize resource usage, models were quantized to reduce computational requirements while maintaining high detection accuracy.

F. Performance Evaluation

➢ *The System was Tested on the Surveillance Infrastructure of YAKIN Technology Ltd. Metrics Included:*

- **Accuracy:** Improved anomaly detection accuracy from 70.4% to 93%.
- **Response Time:** Reduced detection and alerting times to under 2 seconds.
- **Scalability:** Successfully processed multiple video streams in real-time without system lag
- **Comparative analyses** were conducted against existing systems to validate improvements.
- **User feedback** was incorporated to fine-tune the system, ensuring continuous improvement.

G. Research Design

The research design for enhancing the intelligence of a real-time video surveillance system integrates multiple components to ensure efficient processing, anomaly detection, and alert generation. The system begins with a **video source** (e.g., surveillance cameras) that streams live footage into a **Hadoop cluster** for processing and storage. The Hadoop cluster handles large-scale video data, preprocessing it by extracting keyframes, resizing, and enhancing quality for further analysis. Preprocessed data is then transferred to a **High-Performance Computing (HPC) cluster**, which utilizes parallel computing and GPU acceleration to efficiently train deep learning models. These models, built using **AI with Python**, employ CNNs for spatial anomaly detection and RNNs for analyzing temporal patterns and identifying suspicious movements or behaviors in real time. When an anomaly is detected, the AI system generates alerts, triggering alarms or sending notifications to security operators for immediate action. The system operates continuously, leveraging adaptive learning to refine its accuracy over time (Ali, 2022). This design ensures real-time responsiveness, scalable data handling, and precise anomaly detection, making it ideal for diverse surveillance environments. The flow chart visualizes the seamless data flow between components, emphasizing their interconnection and role in achieving the system's objectives (Seemantula Nischal, 2023).

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

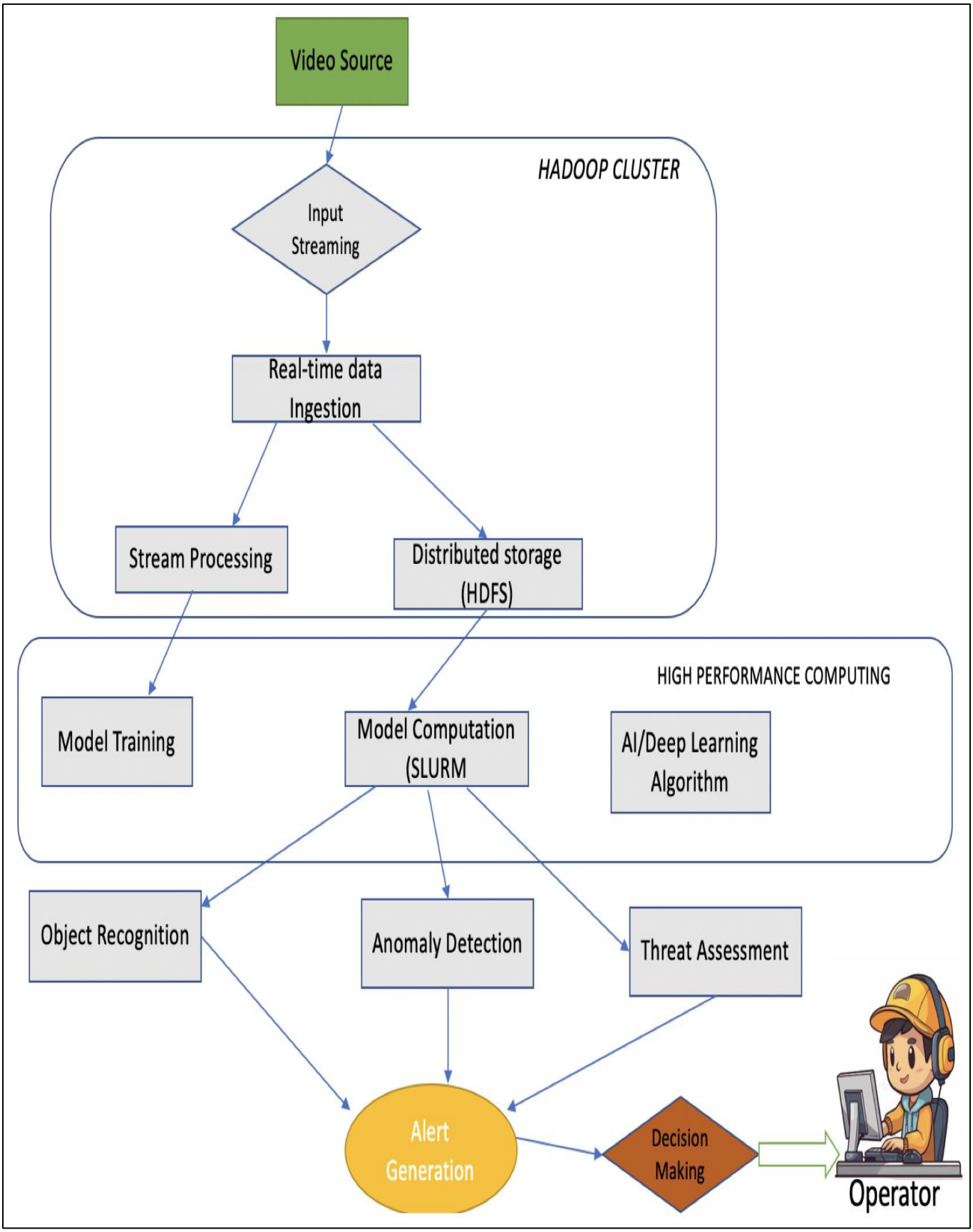

Fig 1: Flowchart of the System

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

III. CONCEPTUAL FRAMEWORK

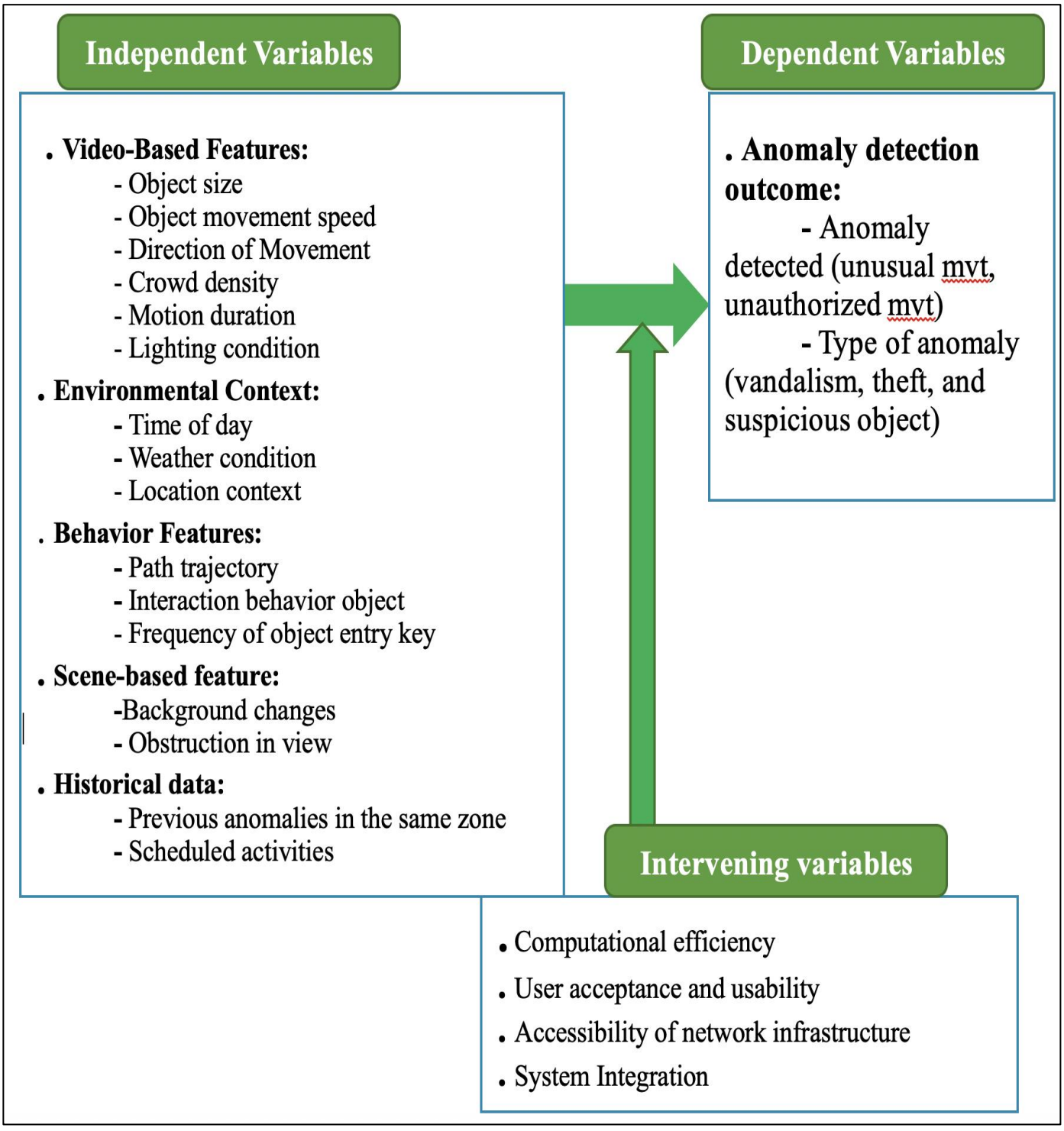

Fig 2: Conceptual Framework

IV. DATA PRESENTATION, ANALYSIS AND INTERPRETATION OF FINDINGS

The results of the study on the impact of parallel processing and computing on developing deep learning models for anomaly detection and identification are presented in this chapter. It contains a thorough examination of the information and findings from numerous experiments. The goal is to give a thorough illustration of how, in this particular situation, parallel computing affects the effectiveness and precision of deep learning models.

A. Distribution of Classes

To begin, we analyzed the distribution of different classes (person, suspect, vehicle, and animal) in the dataset. This helps in understanding the balance of the dataset and the representation of the class.

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

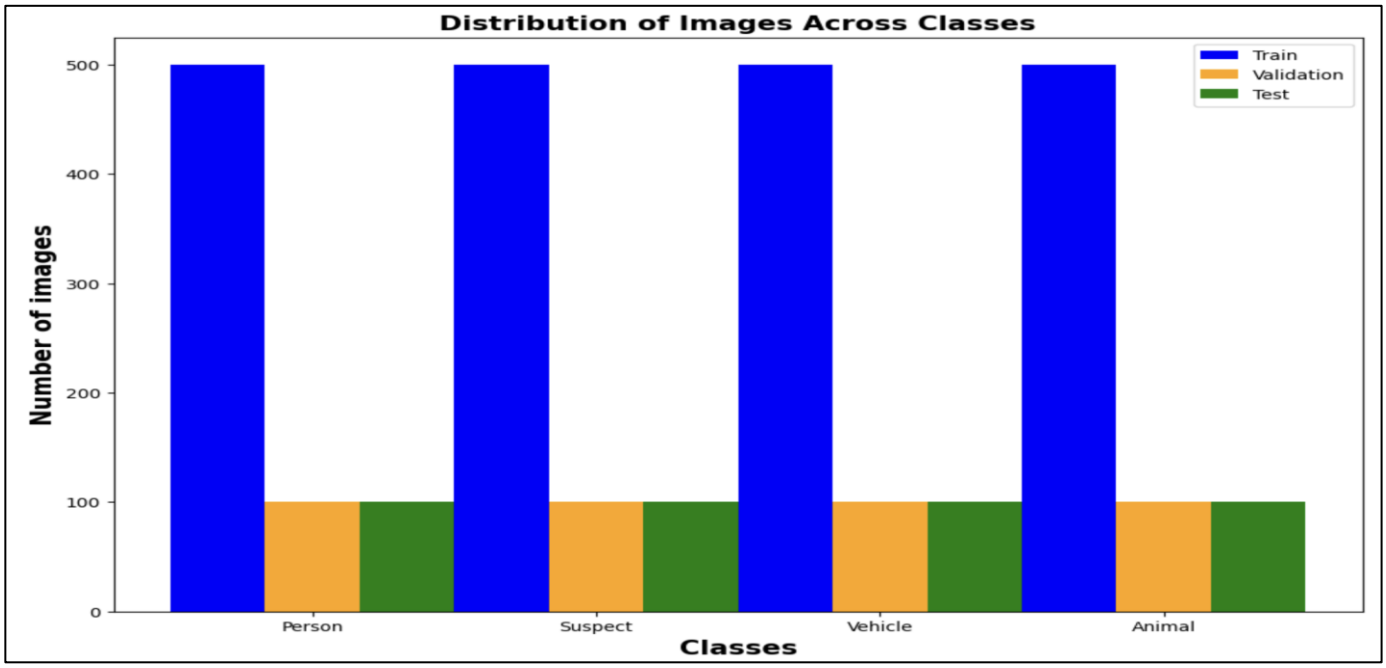

Fig 3: Distribution of Images across Classes

B. Parallel vs Serial Computing

In the context of training deep learning models for anomaly detection, we compared the performance of parallel and serial computing approaches. Parallel computing, which uses multiple processors to perform computations concurrently, significantly reduces model training times compared to serial computing, where tasks are processed sequentially by a single processor. The figure below visually compares training time reduction when parallel computing is applied to various tasks, demonstrating its efficiency in managing larger datasets and more complex models. This comparison highlights the key benefits of parallel processing, including reduced latency and faster convergence during training.

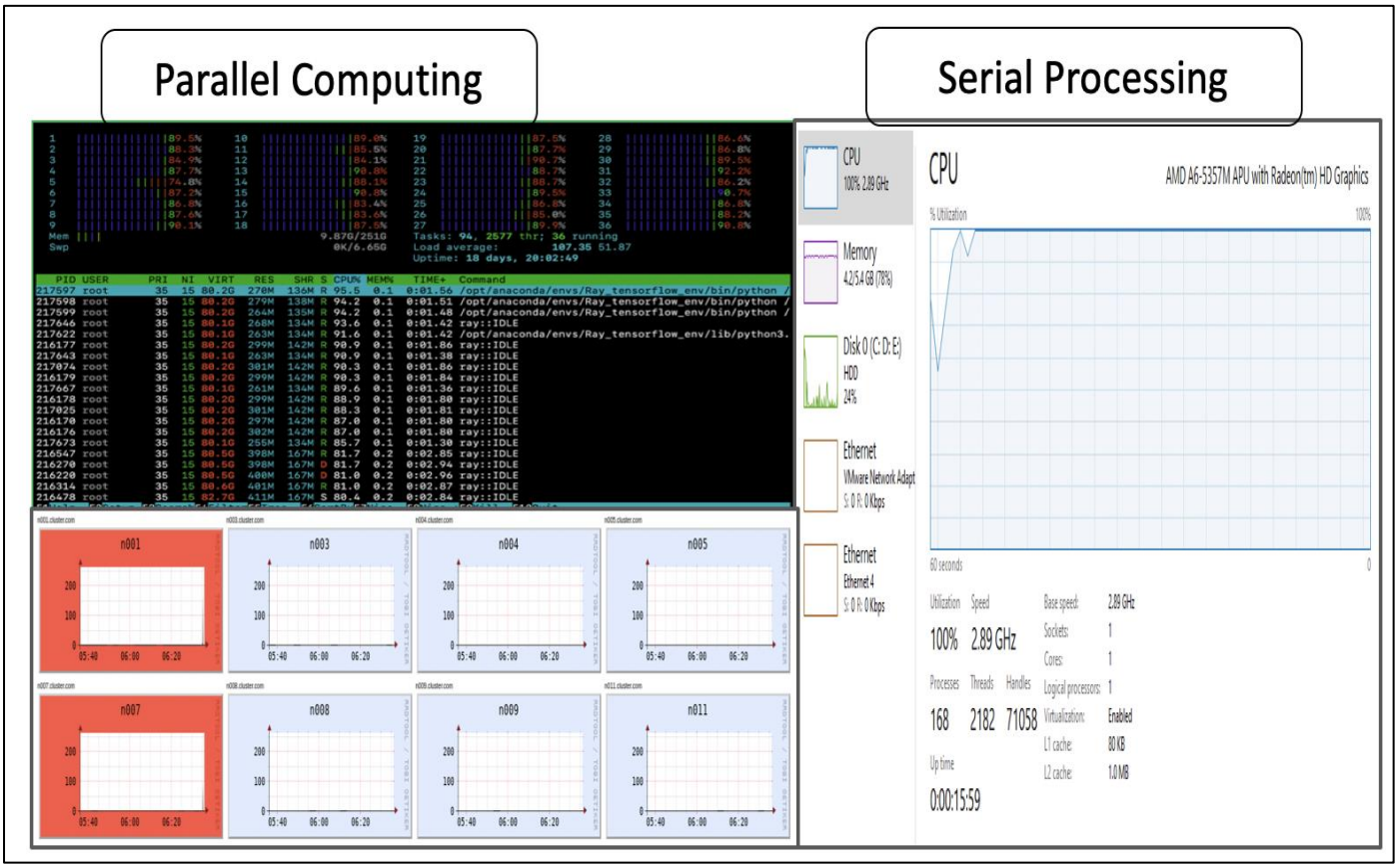

Fig 4: Parallel Computing vs Serial Computing

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

We used both serial and parallel computing techniques to train the identical deep learning model to assess how parallel computing affected training time. The time used for parallel computing (High-Performance computing) for model training is 0.126 hours, which is equal to 7.5**6 minutes,** as shown by the figure.

The time used for Serial computing or model training is equal to **1.174 hours,** as shown by the figure, which is equal to **70.44 minutes**.

- ***With Parallel Computing **Training Time: 0.126 (h) ≈ 8 mins**
- ***With Serial Computing **Training Time: 1.347 (h) ≈ 81 mins**

We found that the deep learning model was taking about eight minutes to train when using parallel computing. By dividing up the calculations among several processors, this efficiency was attained, enabling the processing of tasks and data at the same time.

The following is a comparison of the training times that were noted.

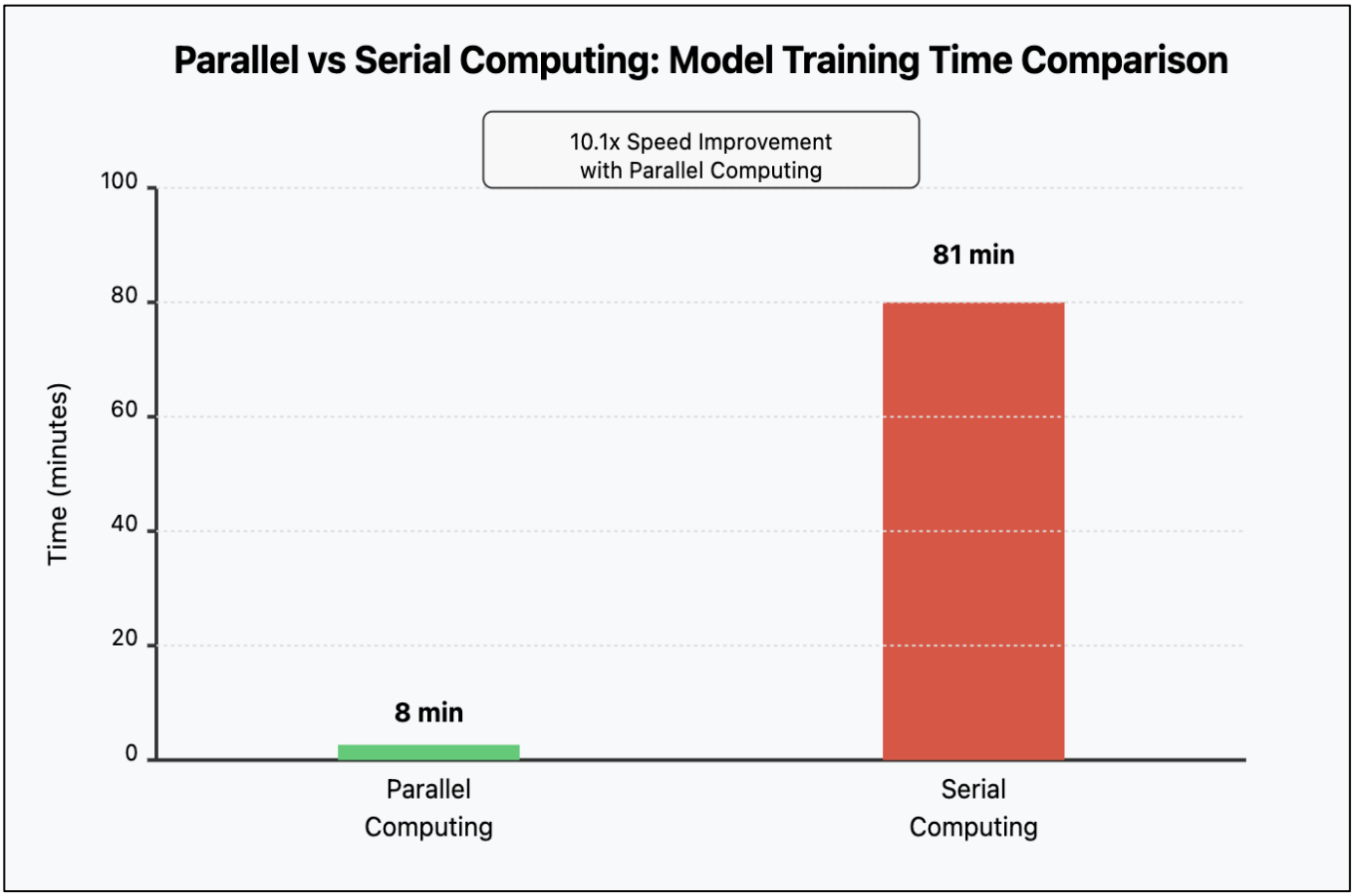

Fig 5: Model Training Time Comparison

C. Model Performance Testing Findings

The great performance of the model's predictions against the actual labels of the person samples during the testing phase for our model's performance on unseen photos. Five times out of the six tests, the model accurately predicted the anomaly detection. This shows that, in the majority of circumstances, the model is highly accurate at finding the right abnormality.

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

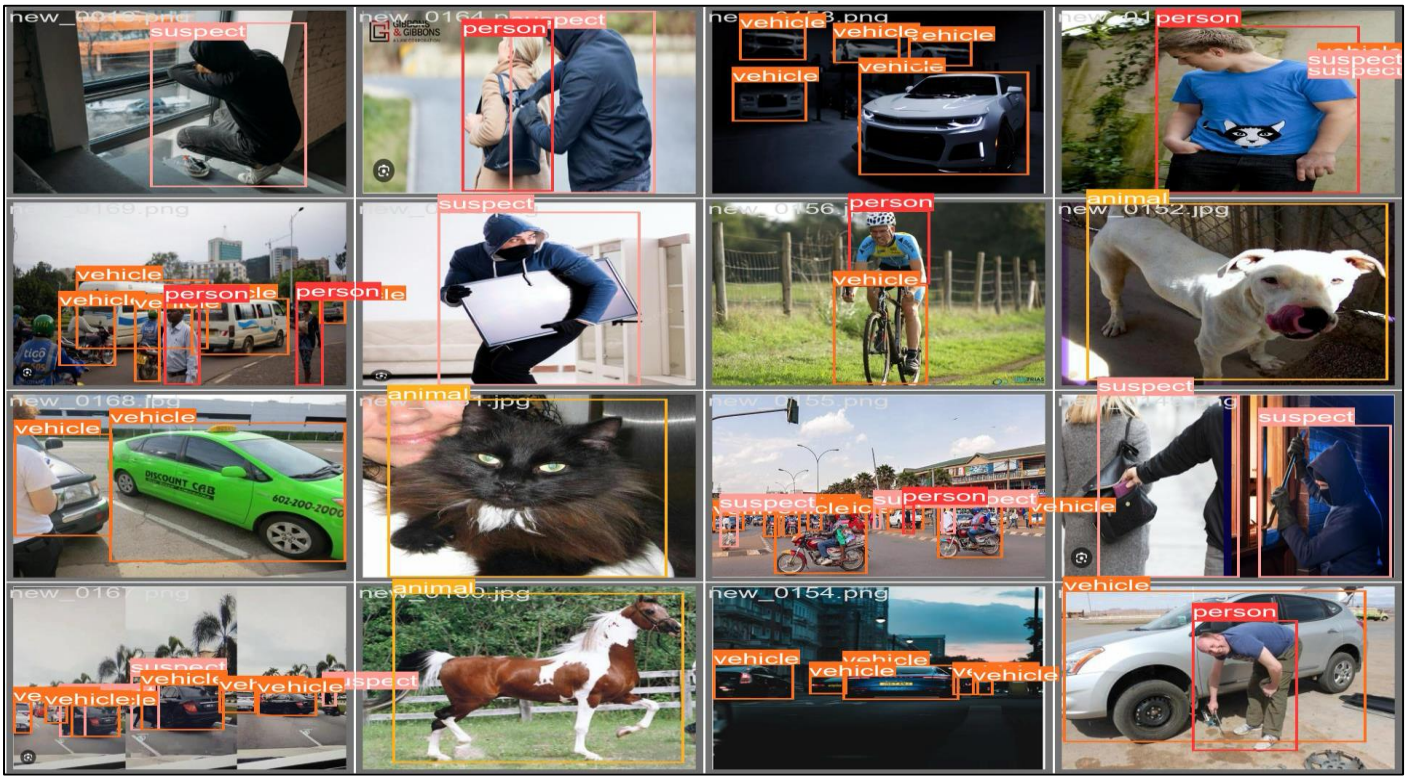

Fig 6: Model Performance Testing Findings

The below figure shows the web interface used for a real-time video surveillance system with the integration on the Artificial Intelligence with automated anomaly detection and alarm generated as a response to the operator or decision maker.

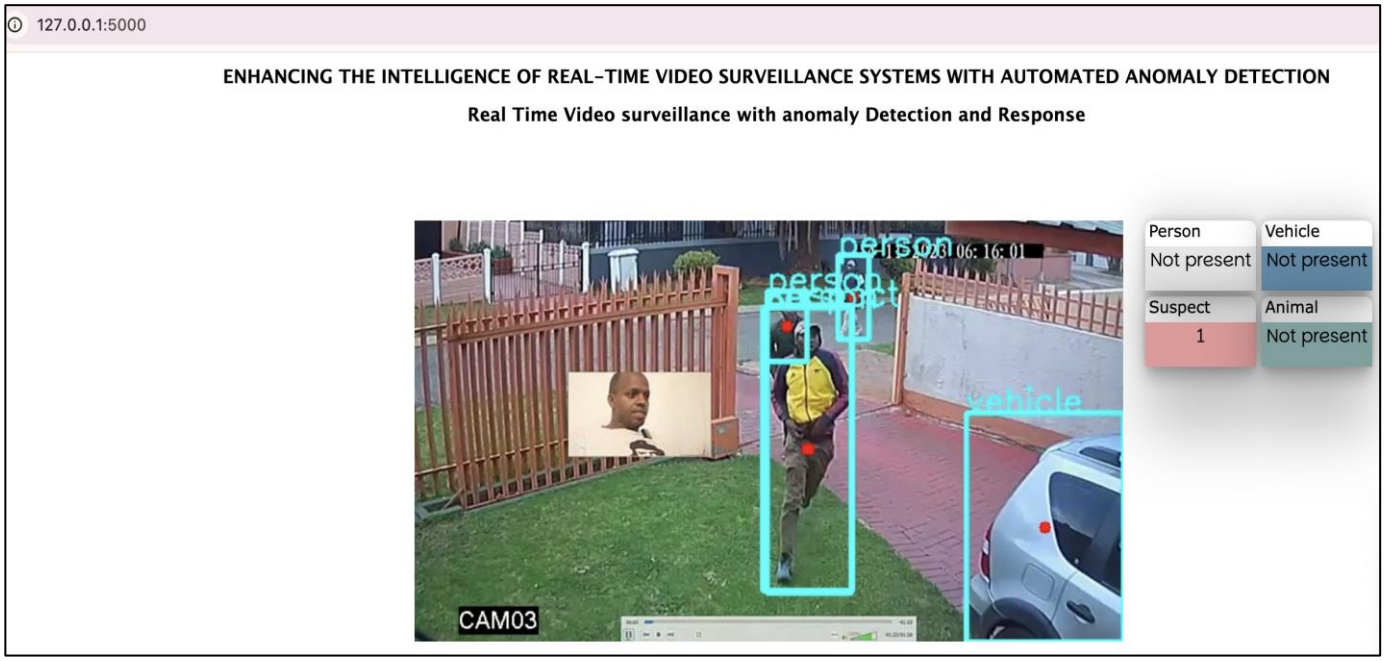

Fig 7: Anomaly Detected in Real-Time with Alert

D. Related Work

Research in automated video surveillance has gained significant traction in recent years. Studies have explored applications of convolutional neural networks (CNNs) for object detection, human action recognition, and anomaly detection. Notable frameworks include YOLO (You Only Look Once) for real-time object detection and RNN-based systems for behavior analysis. However, these approaches often face challenges such as computational inefficiency, false-positive detections, and difficulty in generalizing across diverse environments. This research aims to build upon these advancements and overcome existing limitations by

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

integrating cutting-edge algorithms with practical implementation strategies, including parallel computing and model quantization.

E. Limitations and Challenges

The project faced several limitations and challenges, including the complexity of real-time video stream processing with low latency, ensuring accurate anomaly detection through machine learning, and integrating parallel computing with a Hadoop cluster for efficient data processing and storage. Scaling the system to handle varied environments and minimizing false alarms to avoid operator overload were also significant hurdles. Additionally, managing the infrastructure requirements for large-scale data storage and processing, as well as addressing data privacy and security concerns, posed further challenges during both implementation and design.

V. CONCLUSION

This research integrated automated anomaly detection and reaction mechanisms into real-time video surveillance systems using advanced machine learning models. It significantly improved system accuracy and versatility, enabling efficient anomaly detection and rapid response. The system proved robust across different environments, demonstrating high accuracy in real-time video feeds. Parallel computing accelerated the development of deep learning models, increasing classification accuracy to 0.93 from 0.904 and reducing training time from 71 to 8 minutes. This enhanced efficiency allowed for faster model iterations and more robust results. The solution, deployable on a web application, empowers security operators with real-time surveillance insights, improving security monitoring and response times.

REFERENCES

[1]. Ahmed Elmetwally, R. E. (2024). Deep learning based anomaly detection in real-time video. *Deep learning based anomaly detection in real-time video*, 1705-1716.

[2]. Ali, M. M. (2022). Real-time video anomaly detection for smart surveillance. *Real-time video anomaly detection for smart surveillance*, 1375-1388.

[3]. Ali, M. M. (2022). Real-time video anomaly detection for smart surveillance. *Real-time video anomaly detection for smart surveillance*, 1375-1388.

[4]. Ankush Balaram Pawar1Gayatri Shitole2, S. N. (2024). Intelligence Video Surveillance Using Deep Learning . *Software Computing and Testing*, 2456-2351.

[5]. Bansod, S. (2019). Transfer Learning for video anomaly detection. *Journal of Intelligent & Fuzzy Systems*, 1-9.

[6]. Basha, S. H. (2022). An information-rich sampling technique over spatio-temporal CNN for classification of human actions in videos. *Multimedia tools and applications*, 40431-40449.

[7]. Brown, S. (2021, Apr 21). *MIT*. Retrieved from mitsloan.mit.edu: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[8]. Brownlee, J. (2019, September 16). *A Gentle Introduction to Transfer Learning for Deep Learning*. Retrieved from https://machinelearningmastery.com/: https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[9]. Brownlee, J. (2023). Logistic Regression for Machine Learning. *Machine Learning Mastery*, 1-7.

[10]. Brownlee, J. (2023, November 16). *Machine Learning Mastery*. Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/what-is-deep-learning/

[11]. Dalibor Klusacek, G. P. (2023, May 19). *Job Scheduling Strategies for Parallel Processing*. Retrieved from Job Scheduling Strategies for Parallel Processing: https://www.google.rw/books/edition/Job_Scheduling_Strategies_for_Parallel_P/M1DXEAAAQBAJ?hl=en&gbpv=1

[12]. Dobbin, J. (2019, February 24). *GPU vs CPU: What Matters Most for PC Gaming?* Retrieved from https://www.hp.com/: https://www.hp.com/us-en/shop/tech-takes/gpu-vs-cpu-for-pc-gaming

[13]. Eremenko, K. (2018). Deep Learning A-Z™: Convolutional Neural Networks (CNN) - Step 3: Flattening. *Flattening*, 3-5.

[14]. Ethier, S. (2022, October 19). *Parallel Computing: Intro to MPI*. Retrieved from https://researchcomputing.princeton.edu/: https://researchcomputing.princeton.edu/sites/g/files/toruqf311/files/documents/MPI_tutorial_Fall_Break_2022.pdf

[15]. Ethier, S. (2022, October 19). *Parallel Computing: Intro to MPI*. Retrieved from https://researchcomputing.princeton.edu/: https://researchcomputing.princeton.edu/sites/g/files/toruqf311/files/documents/MPI_tutorial_Fall_Break_2022.pdf

[16]. Gandhi, R. (2018, May 27). *Introduction to Machine Learning Algorithms: Linear Regression*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a

[17]. geeksforgeeks. (2024, Sep 04). *https://www.geeksforgeeks.org*. Retrieved from https://www.geeksforgeeks.org: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

[18]. geeksforgeeks.org. (2024, Sep 04). *https://www.geeksforgeeks.org/*. Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

[19]. Gobbert, M. K. (2021, April 3). *High Performance Computing Facility*. Retrieved from hpcf.umbc.edu: https://hpcf.umbc.edu/ada/slurm/overview/

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14959415

[20]. Haohan Ding, J. T. (2023). The Application of Artificial Intelligence and Big Data in the Food Industry. *MDPI*, 29.

[21]. Heller, M. (2022, Sep 16). *infoworld*. Retrieved from infoworld.com: https://www.infoworld.com/article/3299703/what-is-cuda-parallel-programming-for-gpus.html

[22]. ibm.com. (2023, Dec 19). *www.ibm.com*. Retrieved from www.ibm.com: https://www.ibm.com/think/topics/machine-learning-for-anomaly-detection

[23]. Kim, J. S. (2021). A study on implementation of real-time intelligent video surveillance system based on embedded module. *EURASIP Journal on Image and Video Processing*, 2-22.

[24]. Kim, Y. M.-O. (2023). A Survey of Video Surveillance Systems in Smart City. *MDPI*, 3-7.

[25]. Kraus, J. (2013, March 13). *An Introduction to CUDA-Aware MPI*. Retrieved from https://developer.nvidia.com/: https://developer.nvidia.com/blog/introduction-cuda-aware-mpi/

[26]. Mishra, M. (2020, Aug 26). *Convolutional Neural Networks, Explained*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

[27]. Muts, I. (2023). Real-Time Video Monitoring: The New Era of Remote Video Surveillance. *Euristiq*, 3-14.

[28]. Nielsen, F. (2016). Introduction to HPC with MPI for Data Science. *Introduction to MPI: The Message Passing Interface*, 21-62.

[29]. Nishihara, R. (2019, Feb 11). *Modern Parallel and Distributed Python: A Quick Tutorial on Ray*. Retrieved from https://towardsdatascience.com/: https://towardsdatascience.com/modern-parallel-and-distributed-python-a-quick-tutorial-on-ray-99f8d70369b8

[30]. *NVIDIA Tesla P100 PCIe 16 GB*. (n.d.). Retrieved from techpowerup: https://www.techpowerup.com/gpu-specs/tesla-p100-pcie-16-gb.c2888

[31]. Rosebrock, A. (2021, May 14). *Convolutional Neural Networks (CNNs) and Layer Types*. Retrieved from pyimagesearch.com: https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/

[32]. Rosebrock, A. (2021, May 14). *pyimagesearch*. Retrieved from pyimagesearch.com: https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/

[33]. Saha, S. (2018, Dec 15). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[34]. Santilli, P. (2024, January 11). *Real-time Threat Detection: The Power of AI in Video Surveillance*. Retrieved from scip.org: https://www.scip.org/news/662235/Real-time-Threat-Detection-The-Power-of-AI-in-Video-Surveillance.htm

[35]. Singh, S. (2023, July 20). *What is a Conceptual Framework and How to Make It (with Examples)*. Retrieved from https://researcher.life/: https://researcher.life/blog/article/what-is-a-conceptual-framework-and-how-to-make-it-with-examples/

[36]. *Stratified Random Sampling*. (n.d.). Retrieved from Questionpro: https://www.questionpro.com/blog/stratified-random-sampling/

[37]. Strickland, J. (2024, march 6). *Parallel Computing*. Retrieved from https://computer.howstuffworks.com/: https://computer.howstuffworks.com/parallel-processing.htm

[38]. Techlab, N. (2024, June 12). *www.netlabindia.com*. Retrieved from www.netlabindia.com: https://www.netlabindia.com/blogs/emerging-technologies-shaping-the-future-of-video-surveillance/

[39]. Unzueta, D. (2022, October 18). *Fully Connected Layer vs. Convolutional Layer: Explained*. Retrieved from https://builtin.com/: https://builtin.com/machine-learning/fully-connected-layer

[40]. Venujkvenk. (2023, Oct 03). *Anomaly Detection Techniques: A Comprehensive Guide with Supervised and Unsupervised Learning*. Retrieved from www.medium.com: https://medium.com/@venujkvenk/anomaly-detection-techniques-a-comprehensive-guide-with-supervised-and-unsupervised-learning-67671cdc9680

[41]. Xidong Wu, P. B. (2023). *Performance and Energy Consumption of Parallel Machine Learning Algorithms*. ECE 2166.

[42]. Yasar, K. (2023, March 31). *https://www.techtarget.com/searchenterpriseai/definition/image-recognition*. Retrieved from www.techtarget.com: https://www.techtarget.com/searchenterpriseai/definition/image-recognition

[43]. Yim, J. (2015). Image Classification Using Convolutional Neural Networks With Multi-stage Feature. *Robot Intelligence Technology and Applications 3*, 587-594.

[44]. Yokim, S. (2018, Aug 20). *Tensor Ops Made Easier in cuDNN*. Retrieved from developer.nvidia.com: https://developer.nvidia.com/blog/tensor-ops-made-easier-in-cudnn/

[45]. Zamora, R. R. (2021). Parallel and High Performance Computing. *Parallel Programming (MPI) and Batch Usage (SLURM)*, 650-704.