AI-Based CCTV Analytics Solution

Proposal for APSCSCL Warehouse Hackathon 2025

Team Vizag AI

Smart CCTV Analytics
Tailor-made for APSCSCL warehouses

Intelligent Surveillance System for Warehouse CCTV cameras

Visit us at www.cctvai.org

Submitted by: **Team Vizag AI**
Email: info@vizag-ai.com
Phone: +91-8904473119
Project Website: www.cctvai.org
Team Website: www.vizag-ai.com
Point of Contact: Asapanna Rakesh (+91-8904473119)
Submission Date: 10th May, 2025

Contents

1 Team Credentials (In no particular order)

Meet our incredible team! Each member brings a unique set of skills and experiences that contribute to our collective success. Below, you'll find a glimpse into the individuals who drive our innovation and achievements.

Krishna Praveen from IIT Kharagpur 2014 is a Data Scientist working with Enterprise AI systems in supply chain, warehousing and distribution space. Has extensive experience in AI/ML and Gen AI. Cofounded AI company previously.

Parameswar Kumavath from IIT Kanpur 2015 and LJMU has extensive academic ML experience. Formerly an Audio Engineer, he now develops an LMS platform at his own startup. He has full-stack experience and builds & deploys ML models.

Asapanna Rakesh, Director of Technology at his own startup. Previously Sr. Research Engineer at India's largest geospatial company. He loves modelling intelligent systems, solving problems and building products that matter.

Sritej Reddy, a ML Engineer with diverse skillset such as full-stack application development, Python, Java, C++, TensorFlow, NumPy, and Pandas. He also has experience in RPA using Blue Prism, embedded systems, and IoT.

Ram Jaswanth, a Member of Technical Staff at Alphanome.AI where he builds innovative solutions in the AI and technology space. Works with Generative AI, LLMs and engineering at scale to build AI driven ventures.

Teja Saraswathula, a Senior Software Developer and Engineer at Tech Mahindra in automotive industry space. Has extensive experience in building production ready and battle-tested scalable software systems. Skilled programmer.

2 Executive Summary

This proposal presents a comprehensive AI-powered CCTV analytics solution addressing **all four use cases** identified by the Andhra Pradesh State Civil Supplies Corporation Limited (APSCSCL) in their hackathon challenge. Our solution leverages state-of-the-art computer vision, artificial intelligence, image processing techniques, and large scale infrastructure design to transform existing CCTV infrastructure into an intelligent monitoring system capable of:

- Real-time gunny bag counting and volumetric analysis
- AI-powered vehicle recognition and number plate authentication
- AI-driven facial recognition for personnel tracking
- Contextual intelligence for real-time analysis and query capabilities on video data

Our approach integrates cutting-edge open-source models with a scalable streaming architecture designed to operate efficiently across APSCSCL's warehouse network. The solution is built on principles of reliability, cost-effectiveness, and seamless integration with existing infrastructure while adhering to data privacy regulations.
This document outlines our technical approach, implementation methodology, project timeline, and expected outcomes, demonstrating our capabilities to deliver a production-ready solution that meets APSCSCL's short, mid, and long-term expectations.

3 Understanding of the Problem

APSCSCL manages a complex network of warehouses crucial to the Public Distribution System (PDS) in Andhra Pradesh. Current manual monitoring processes face challenges in accuracy, efficiency, and security. Key challenges identified include:

- Manual counting and tracking of gunny bags is labor-intensive and error-prone
- Vehicle entry/exit monitoring lacks automated authentication mechanisms
- Personnel access control relies on human vigilance, creating security vulnerabilities
- Limited ability to search through video footage for specific events or activities
- Underutilized CCTV infrastructure that could provide valuable operational insights

Our solution transforms these challenges into opportunities by leveraging AI to extract actionable intelligence from existing CCTV feeds, creating a smart surveillance system that enhances operational efficiency, security, and transparency across the supply chain.

4 Technical Approach and Architecture

Our approach addresses all four use cases through a unified architecture with specialized AI modules for each function. The system is designed as a layered architecture with the following components:

4.1 System Architecture

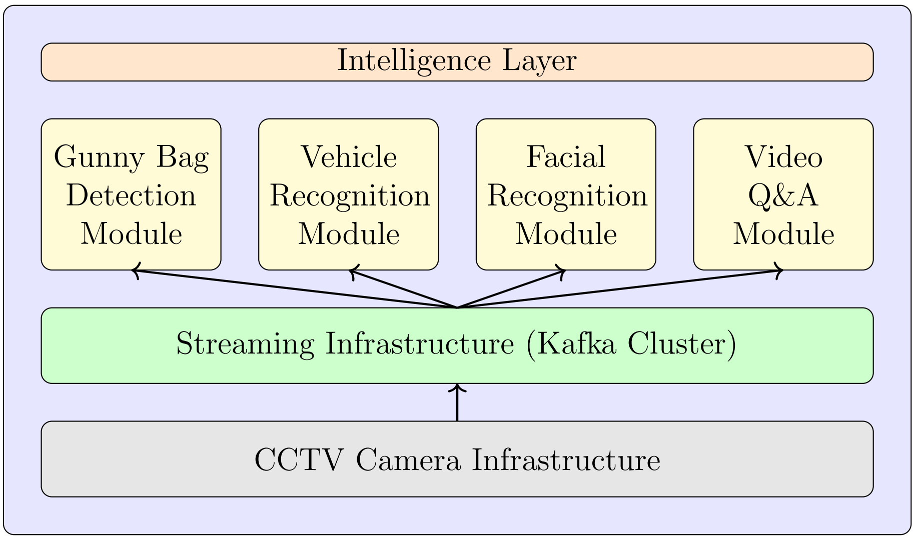
Intelligence Layer
Gunny Bag Detection Module
Vehicle Recognition Module
Facial Recognition Module
Video Q&A Module
Streaming Infrastructure (Kafka Cluster)
CCTV Camera Infrastructure

Figure 1: System Architecture (with Video Q&A Module)

4.2 Core Components

4.2.1 Streaming Infrastructure for Real-time Warehouse Operations

The backbone of our real-time warehouse operations is a robust streaming infrastructure designed for high-volume data ingestion, processing, and distribution. This infrastructure is crucial for enabling timely decision-making and optimizing various warehouse processes.

- **Apache Kafka for Core Message Queuing:** At the heart of our streaming platform lies Apache Kafka, a distributed streaming platform renowned for its high-throughput, fault-tolerance, and persistent message queuing capabilities.
  - *High-Throughput:* Kafka is engineered to handle millions of messages per second, making it suitable for the high-velocity data generated by network of CCTV cameras within a modern warehouse(s).
  - *Persistence and Durability:* Messages are durably stored on disk and replicated across the Kafka cluster, preventing data loss in case of node failures and ensuring reliable data delivery.
  - *Decoupling of Producers and Consumers:* Kafka acts as an intermediary buffer, decoupling data producers (CCTV cameras) from data consumers (e.g., analytics dashboards, alerting systems, and our business logic

applications and modules). This allows for independent scaling and evolution of different components.

- **Data Ingestion and Organization by Warehouse Locations:**
  - Input data streams, originating from various sources are logically grouped and partitioned by specific warehouse locations or zones.
  - This geographical or zonal partitioning allows for targeted processing and localized analytics, improving the relevance and speed of insights for location-specific operations. It also aids in managing data sovereignty or compliance requirements if different locations have different data handling rules.
- **Dedicated Business Logic Streams for Each Use Case:** To maintain modularity and clarity, distinct Kafka topics and stream processing applications are established for each specific business use case.
  - Our business use cases are face recognition, number plated detection, gunny bag counting, video analysis etc. But **they are scalable to other business use cases** like streams for real-time inventory tracking, order fulfillment monitoring, equipment health prediction, labor efficiency analysis, and safety alert generation.
  - This separation ensures that the logic for one use case does not interfere with another, simplifies development and maintenance, and allows for independent scaling of processing resources based on the demands of each use case.
  - Stream processing technologies (e.g., Kafka Streams, Apache Flink, Apache Spark Streaming) are often employed here to perform transformations, aggregations, and complex event processing on these dedicated streams.
- **Ensuring Scalability and Resilience Across the Warehouse Network:** The architecture is designed with scalability and resilience as primary considerations to support a growing network of warehouses and increasing data volumes.
  - *Horizontal Scalability:* Kafka clusters can be easily scaled out by adding more brokers to handle increased load. Similarly, stream processing applications can be scaled by deploying more instances.
  - *Fault Tolerance and High Availability:* Kafka's distributed nature, with data replication and leader election, ensures that the system remains operational even if some brokers or servers fail. Consumer groups in Kafka also allow for reprocessing of messages in case of consumer failures.
  - *Geographical Distribution (Optional but relevant for network):* For larger warehouse networks, Kafka clusters can be strategically deployed across different regions or availability zones, potentially using features like MirrorMaker for inter-cluster data replication, to enhance disaster recovery capabilities and reduce latency for geographically dispersed operations.

- *Monitoring and Management:* Robust monitoring tools (e.g., Prometheus, Grafana, Confluent Control Center) are typically integrated to provide visibility into the health and performance of the Kafka cluster and streaming applications, enabling proactive issue resolution.

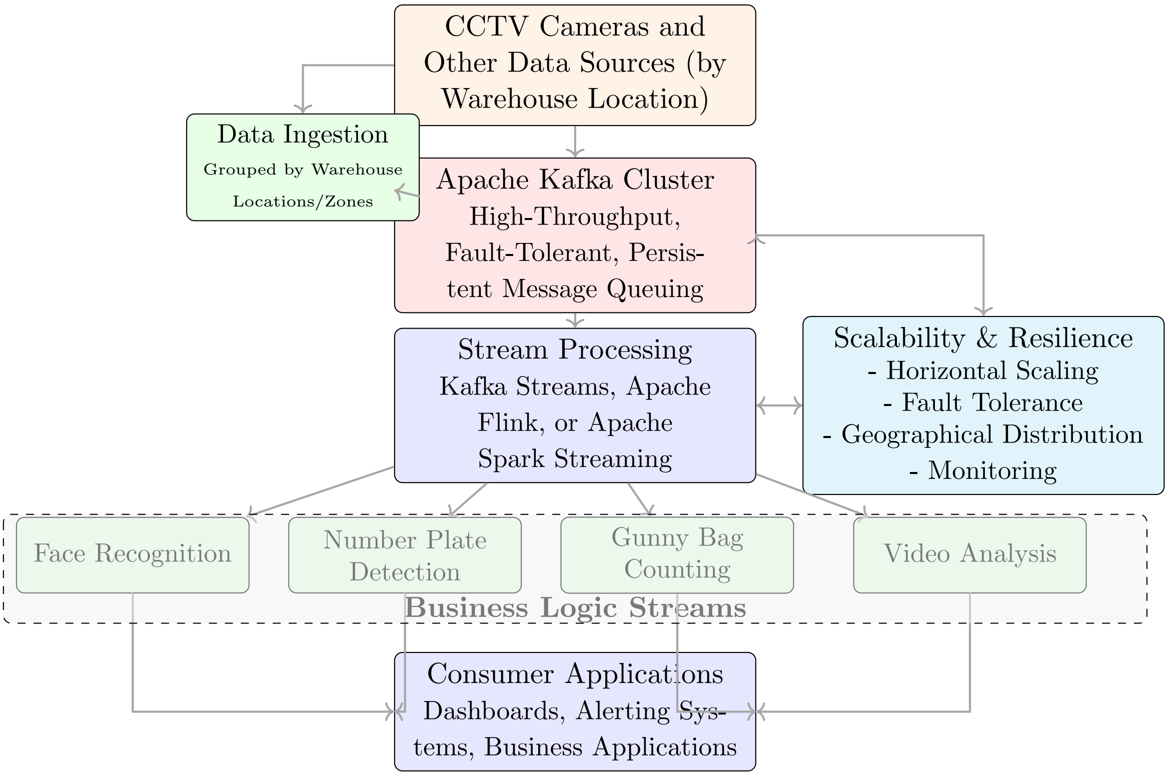
CCTV Cameras and Other Data Sources (by Warehouse Location)
Data Ingestion
Grouped by Warehouse Locations/Zones
Apache Kafka Cluster
High-Throughput, Fault-Tolerant, Persistent Message Queuing
Stream Processing
Kafka Streams, Apache Flink, or Apache Spark Streaming
Scalability & Resilience
- Horizontal Scaling
- Fault Tolerance
- Geographical Distribution
- Monitoring
Face Recognition
Number Plate Detection
Gunny Bag Counting
Video Analysis
Business Logic Streams
Consumer Applications
Dashboards, Alerting Systems, Business Applications

Figure 2: Warehouse CCTV Streaming Infrastructure

4.2.2 Data Processing Pipeline for AI-Driven Warehouse Operations

The efficacy of AI-powered solutions for warehouse management, including gunny bag tracking, vehicle recognition, facial authentication, and contextual event analysis, hinges on a **robust and meticulously designed data processing pipeline**. This pipeline ensures that raw data from sources like CCTV footage is transformed into actionable insights in real-time or near real-time.

- **Frame Extraction and Preprocessing:** This foundational stage is critical for preparing video data from CCTV for subsequent AI model processing, with specific considerations for each use case:
  - *For Gunny Bag Tracking & Volumetric Analysis:* Frames are extracted from continuous CCTV feeds monitoring loading, unloading, and stacking areas. Preprocessing is vital to handle challenging warehouse environments. This includes adaptive contrast enhancement for poor lighting conditions, noise reduction to clarify bag shapes, and resizing/normalization tailored for object detection models like YOLO, SAM, or DETR (decided after experimentation). Geometric corrections might also be applied if camera angles distort the perceived volume of bags.
  - *For AI-Powered Vehicle Recognition:* Frames capturing vehicle entry/exit points are extracted. Preprocessing focuses on optimizing these frames for Optical Character Recognition (OCR) and AI-powered image processing. This involves techniques to handle variable lighting conditions (e.g., day/night, shadows, glare), license plate region isolation, image sharpening, binarization for OCR clarity, and perspective correction if plates are viewed at an angle.
  - *For AI-Driven Facial Recognition:* Frames are extracted from cameras at access points or restricted zones. Preprocessing aims to enhance facial features for reliable identification. This includes handling different lighting conditions, face alignment, scale normalization, and potentially image enhancement techniques to improve the clarity of facial characteristics, even with partial occlusions or varying poses.
  - *For Contextual Intelligence and Query:* Frames are extracted from various CCTV feeds across the warehouse. Preprocessing needs to ensure consistency and quality across diverse sources. This involves handling varying conditions such as different lighting or camera angles, potentially image stabilization if cameras are subject to vibrations, and color normalization to aid in consistent event and anomaly detection.
- **Diversity Identification through Automated Clustering:** To ensure the AI models are **robust and generalize well**, it's crucial to train them on diverse data. **Automated clustering helps identify varied scenarios** within the collected and preprocessed frames:
  - *For Gunny Bag Tracking:* Clustering can identify variations in bag appearance (e.g., different colors, sizes, markings if any), stacking patterns (neatly stacked, haphazard piles), levels of occlusion (partially hidden bags), various lighting conditions (day, night, artificial light), and different camera

perspectives. This ensures models are trained to count and analyze volume accurately across these diverse situations.

  - *For Vehicle Recognition:* Clustering helps identify diverse license plate types (e.g., varying languages/fonts, colors, designs from different regions), different vehicle categories (trucks, tempos, cars for vehicle categorization), various states of plate cleanliness (muddy, clean), and a wide range of lighting and weather conditions affecting plate visibility.
  - *For Facial Recognition:* Clustering can group faces based on variations in lighting, pose, expression, accessories (e.g., glasses, headwear if permitted), and different distances from the camera. This helps in creating a balanced dataset representing warehouse managers, hamalis (labour), and a diverse set of potential unknown individuals.
  - *For Contextual Intelligence:* Clustering helps discover patterns representing normal activities versus contextual anomalies (e.g., unusual movements, loitering in restricted zones) and significant events (e.g., start/end of loading, safety incidents). It can also group data by different warehouse zones, times of day, or specific operational contexts to understand baseline activities.

- **Representative Sampling to Ensure Balanced Model Training:** Based on the diversity identified, sampling strategies are employed to create training, validation, and test datasets that prevent model bias and enhance generalization for each specific application:

  - *For Gunny Bag Tracking:* Ensures the training data for models like YOLO, SAM, or DETR contains a balanced representation of bags under all identified conditions (lighting, stacking, occlusion, camera views). This is critical for achieving high accuracy in both counting and volumetric analysis.
  - *For Vehicle Recognition:* Guarantees that the OCR and vehicle detection models are trained on a balanced set of license plates representing all supported languages/fonts, vehicle types, and encountered lighting/weather conditions. This is essential for reliable plate reading and alerting on unauthorized entries.
  - *For Facial Recognition:* Involves creating a training set with balanced representation of all authorized personnel across different appearances (lighting, pose) and a diverse set of unknown faces to minimize false positives/negatives in authenticating individuals and detecting unauthorized intrusions. Special attention is paid to avoid demographic biases.
  - *For Contextual Intelligence:* Focuses on ensuring that the training data includes sufficient examples of various anomalies and events alongside normal operational footage, across different lighting or camera angles. This balance is key for the system to reliably identify contextual anomalies and significant events and to support accurate search, retrieval, and querying of activities.

- **Real-time Inference with Error Correction Capabilities:** This stage focuses on deploying the trained models and ensuring their continuous, accurate operation in the dynamic warehouse environment:

- *For Gunny Bag Tracking:*
  - *Real-time Performance:* Optimized computer vision models like Yolo, SAM, Detr for object detection and motion tracking are deployed. Inference is handled via real-time streaming using Kafka streams to dynamically track, count, and perform volumetric analysis.
  - *Error Correction:* Motion tracking helps maintain counts even if bags are temporarily occluded. For volumetric analysis, algorithms may average estimations over several frames or use shape completion techniques for partially visible stacks. Cross-verification of counts during loading versus unloading events can trigger alerts for discrepancies. Temporal consistency checks ensure individual bags are not double-counted or missed.
- *For AI-Powered Vehicle Recognition:*
  - *Real-time Performance:* Fast execution of OCR and detection models is crucial for reading and logging vehicle license plates for authentication and tracking and generating immediate alerts to unauthorized entries.
  - *Error Correction:* OCR outputs are assigned confidence scores; low-confidence reads might trigger re-analysis of subsequent frames or flag for manual review. Algorithms can use character-level confidence or known plate syntax rules (if applicable for specific languages/regions) for correction. Timestamp logging and movement tracking provide contextual data to resolve ambiguities (e.g., confirming if a partially read plate matches an expected vehicle).
- *For AI-Driven Facial Recognition:*
  - *Real-time Performance:* Highly optimized facial recognition models are used to identify and verify individuals in CCTV footage and trigger near real-time alerts for unauthorized access.
  - *Error Correction:* Confidence scores from the recognition model are used; low scores might lead to re-assessment using multiple frames or flagging for security personnel review. The system must distinguish authorized personnel from unknown individuals reliably; thus, a robust mechanism to handle ambiguous cases or new authorized personnel is needed. The history log allows for auditing and correcting past misidentifications, which can feed back into model retraining.
- *For Contextual Intelligence and Query:*
  - *Near Real-time Performance:* Efficient models for complex event recognition and anomaly detection are deployed to process CCTV or video feeds in near real-time, identifying contextual cues.
  - *Error Correction and Refinement:* Anomalies often require contextual validation (e.g., an ”unusual movement” might be normal maintenance). The system allows users to search and retrieve video segments based on specific activities or events and query for detailed insights. User feedback on the relevance of retrieved events or the accuracy of anomaly detection can be used to refine models. The **searchable log of analyzed events** enables review and identification of systemic errors or

areas for model improvement, ensuring reliable operation under varying conditions.

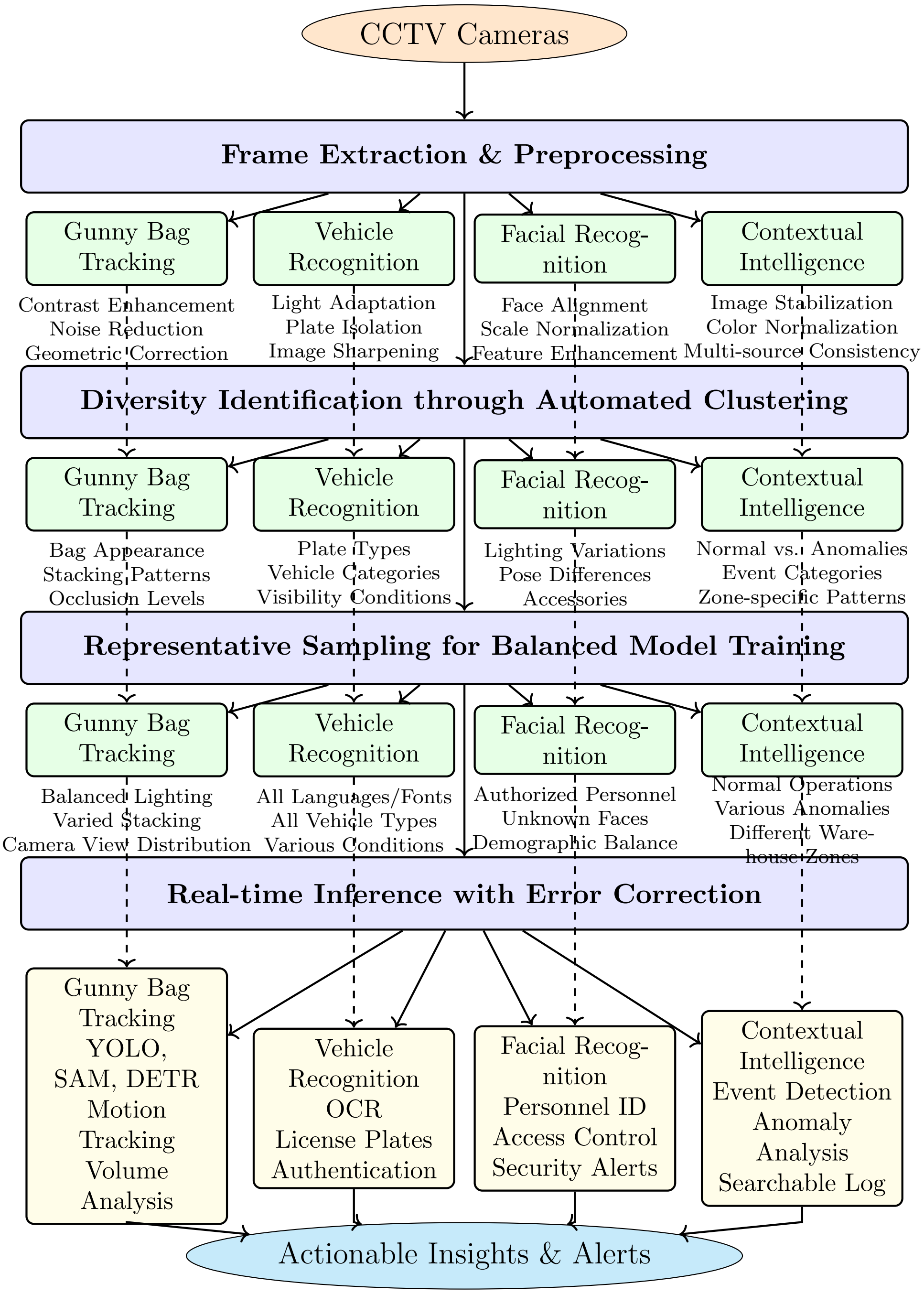

Figure 3: Data Processing Pipeline for AI-Driven Warehouse Operations

4.2.3 AI Models and Techniques: Detailed Application by Use Case

The proposed solution leverages a suite of advanced AI models and techniques, tailored to meet the specific requirements of each use case. For the initial three use cases focusing on detection, tracking, and recognition, supervised machine learning methodologies are central. For the fourth use case, which demands contextual understanding and querying of video data, a combination of vision-language models and retrieval-augmented generation is employed.

Use Case 1: Real-time Gunny Bag Counting and Volumetric Analysis To address the need for dynamic tracking, counting, and volumetric analysis of gunny bags, the following AI-driven video analytics components are utilized:

- **Object Detection for Gunny Bags**: We employ **YOLOv11 (You Only Look Once version 11)** models, specifically optimized for high-speed and accurate real-time object detection. These models are trained on extensive datasets of gunny bags in various warehouse environments, lighting conditions, and states (e.g., filled, partially filled, stacked, in motion). The model processes CCTV footage frame-by-frame to identify and locate each gunny bag. Adaptive contrast enhancement techniques are applied as a pre-processing step to the input frames from CCTV to improve visibility in poor lighting conditions before they are fed to the YOLOv11 model.

- **Real-time Object Tracking**: Once gunny bags are detected, the **DeepSORT (Deep Simple Online and Realtime Tracking)** algorithm is implemented. DeepSORT assigns a unique ID to each detected gunny bag and tracks its trajectory across consecutive frames. This is crucial for accurately counting bags during loading, unloading, and stacking operations, minimizing errors from occlusions or rapid movements. **Custom adaptations to DeepSORT** ensure robustness in cluttered warehouse scenes.

- **Counting and Volumetric Estimation**: The system maintains a real-time count of tracked gunny bags entering or leaving defined zones (e.g., truck beds, stacking areas). For volumetric analysis, while direct 3D measurement from 2D CCTV is challenging, the system can estimate volume based on the 2D bounding box dimensions of detected bags, calibrated with typical gunny bag sizes. Further precision could be achieved by integrating monocular depth estimation techniques like MiDaS, though the current AI model focus is on 2D video analytics. The number of bags and their estimated collective volume are automatically measured and verified.

- **Real-time Streaming Inference**: To ensure real-time processing, the inference pipeline is optimized to utilize Kafka streams. This allows for efficient communication between the CCTV cameras, the AI processing server, and any monitoring dashboards, enabling immediate tracking and counting information.

Use Case 2: AI-Powered Vehicle Recognition This system focuses on reading and logging vehicle license plates for authentication and tracking at warehouse premises using AI-powered image processing from CCTV footage:

- **Vehicle and License Plate Detection**: Initially, **YOLOv11** models, trained for vehicle detection, are used to identify vehicles within the CCTV camera's field of view. Subsequently, another specialized YOLOv11 model or a region proposal network is used to precisely localize the license plate area on the detected vehicle.

- **Optical Character Recognition (OCR)**: For reading the alphanumeric characters on the localized license plates, we use **PaddleOCR**. This OCR engine is fine-tuned using a custom dataset of license plates, encompassing various fonts, languages (including those relevant to the operational region), and plate conditions (e.g., dirty, damaged). This fine-tuning enhances accuracy under variable lighting conditions and diverse plate designs.

- **Authentication and Alerting**: The recognized license plate number is compared against a database of authorized vehicles. If a plate is not found or belongs to an unauthorized vehicle, the system generates an alert.

- **Logging and Tracking Features**: The system logs each recognized license plate with a timestamp, camera ID, and the recognized plate number. Movement tracking of vehicles within the premises is achieved by correlating sightings from multiple cameras or over time from a single camera. Vehicle categorization (e.g., truck, car, motorcycle) can be an additional output from the initial YOLOv11 vehicle detection model.

Use Case 3: AI-Driven Facial Recognition To authenticate authorized personnel and detect unauthorized intrusions in real-time, this system employs AI-based facial recognition:

- **Face Detection**: Similar to vehicle detection, a robust face detection model (often a component of or precursor to recognition systems, e.g., based on YOLO architecture or MTCNN) is used to locate faces in the CCTV footage in real-time.

- **Facial Feature Extraction and Recognition**: For each detected face, the **ArcFace** algorithm is utilized. ArcFace is a state-of-the-art deep learning model known for its high accuracy in face recognition due to its additive angular margin loss function, which enhances discriminative power. It generates a unique feature vector (embedding) for each face. Custom adaptations ensure optimal performance in warehouse environments, considering variations in lighting, pose, and occlusions (e.g., by hats or masks, to a certain extent).

- **Authentication and Alerting**: The generated facial embedding is compared against a pre-enrolled database of embeddings of authorized personnel (warehouse managers, hamalis). If the similarity score surpasses a defined threshold, the individual is authenticated. If a face does not match any authorized personnel or is an unknown individual, an alert is triggered in near real-time for potential unauthorized access.

- **History Logging and Robustness**: The system maintains a **history log** of all recognized (or attempted recognition) events, including timestamps, camera locations, and identities (if known). The chosen models and pre-processing techniques aim for efficient functioning in diverse lighting conditions.

Use Case 4: Contextual Intelligence for Near Real-time Analysis and Query
This system is designed to analyze video feeds for anomalies and significant events, allowing users to search, retrieve, and query video content contextually:

- **Video Processing and Temporal Analysis**: Live or recorded CCTV video is processed by **video chunking**, where frames are sampled at regular intervals (e.g., 1 frame every 5-10 seconds) or keyframes are extracted based on scene changes. This reduces computational load while retaining essential information.

- **Frame Sequence Analysis with Vision-Language Models (VLMs)**: Sequences of these frames or video chunks are then processed by advanced **vision-language models (VLMs)**. These models (e.g., based on architectures like VideoBERT, CoCa, or similar multimodal transformers) are capable of understanding the visual content and its temporal evolution, identifying objects, actions, interactions, and overall scene context. They can identify contextual anomalies (e.g., a person in a restricted area after hours, unusual loitering, sudden congregation of people) and significant events (e.g., loading/unloading operations, safety incidents).

- **Context-Aware Video Description with RAG**: To enhance the understanding and enable rich querying, a **Retrieval-Augmented Generation (RAG)** approach is implemented. When analyzing a video segment or responding to a query, the VLM's output can be augmented with information retrieved from a contextual knowledge base. This knowledge base might include warehouse SOPs, historical event data, or typical activity patterns. This allows the system to generate more accurate and contextually grounded descriptions and analyses of video scenes.

- **Natural Language Querying**: The system supports natural language querying capabilities in English (and Telugu in future). Users can ask questions like "Show me all instances of trucks arriving at Gate A yesterday" or "Were there any people near stacked bags in Warehouse B between 2 AM and 4 AM?". The system leverages the VLM's understanding and the RAG framework to retrieve relevant video segments and provide textual summaries or direct answers.

- **Reliability and Logging**: The system is designed to operate reliably under varying conditions, such as different lighting or camera angles, through robust VLM architectures and adaptive video processing. A searchable log of analyzed events, anomalies, and generated descriptions is maintained for efficient monitoring, review, and auditing.

5 Detailed Solution for Each Use Case

5.1 Common Components Across Use Cases

Our solution leverages a unified processing pipeline with shared core components across the first three use cases (Gunny Bag Analysis, Vehicle Recognition, and Facial Recognition). This architecture promotes modularity, reusability, and efficient resource management. The differentiation primarily occurs in the specialized models loaded for object detection and the subsequent post-processing stages tailored to each specific use case.

- **Video Stream Acquisition**: This foundational component is responsible for interfacing with the existing CCTV camera infrastructure. It will support common streaming protocols like RTSP (Real-Time Streaming Protocol) to ingest live video feeds. The system will be designed to handle multiple concurrent streams from various camera sources within the warehouse environment. Robust error handling and reconnection mechanisms will be implemented to ensure continuous operation even with intermittent network fluctuations. Adaptive contrast enhancement techniques will be applied at this stage or in pre-processing to improve video quality in poor lighting conditions, crucial for warehouses with variable illumination.

- **Frame Extraction**: From the continuous video streams, individual frames are extracted for analysis. Instead of processing every single frame (which is computationally expensive and often redundant), a standardized approach will be used to sample frames at appropriate intervals. This sampling rate can be dynamic, potentially increasing with detected motion or specific events, or set to a fixed rate (e.g., 5-10 frames per second) depending on the specific requirements of the use case and the activity level in the scene.

- **Object Detection**: We will primarily utilize the YOLOv11 (You Only Look Once version 11, representing an advanced iteration of the YOLO family) detection framework. YOLO models are renowned for their speed and accuracy, making them suitable for real-time applications. For each use case, a YOLOv11 model will be custom-trained or fine-tuned with specialized weights to accurately detect the target objects: gunny bags (Use Case 1), vehicles and license plates (Use Case 2), and human faces (Use Case 3). While YOLOv11 is the primary choice, the architecture will be flexible to incorporate or experiment with other advanced models like SAM (Segment Anything Model) for precise segmentation if needed for volumetric analysis, or DETR (Detection Transformer) for scenarios requiring a different architectural approach.

- **Object Tracking**: To maintain a consistent identity of detected objects across consecutive frames, we will implement the DeepSORT (Deep Simple Online and Realtime Tracking) algorithm. DeepSORT combines appearance information (deep learning-based features) with motion information (Kalman filtering) to effectively track objects even through occlusions or when they momentarily leave and re-enter the frame. This is critical for preventing issues like double-counting gunny bags or losing track of a vehicle or person. Real-time streaming inference using Kafka streams will be established to serve these tracking results for real-time monitoring dashboards or downstream applications.

5.1.1 Unified Processing Pipeline for Use Cases 1-3

Use cases 1 (Gunny Bag Analysis), 2 (Vehicle Recognition), and 3 (Facial Recognition) share a nearly identical processing workflow, enhancing system coherence and maintainability:

1. **Input Processing**: CCTV video streams are ingested, preferably through a scalable and resilient message queuing system like Apache Kafka. Kafka will act as a buffer, decoupling the video acquisition from the processing modules and allowing for robust handling of high-throughput video data.

2. **Frame Extraction & Pre-processing**: Frames are intelligently sampled from the video streams. Motion detection algorithms can be employed to optimize sampling – processing more frames when activity is high and fewer during idle periods. Pre-processing steps include resizing, normalization, and adaptive contrast enhancement, especially crucial for footage from warehouses with challenging and dynamic lighting conditions.

3. **Detection**: The pre-processed frames are fed into the YOLOv11 object detection model. The specific pre-trained weights loaded into the YOLOv11 framework will vary depending on the active use case:

   - For gunny bags: Weights trained to identify various sizes, colors, and stacking patterns of gunny bags.
   - For vehicles: Weights trained to detect different vehicle types (trucks, tempos, etc.) and a secondary model for localizing license plates.
   - For faces: Weights trained for robust face detection under diverse conditions (angles, lighting, partial occlusions).

4. **Tracking**: The detected objects (gunny bags, vehicles, faces) along with their bounding box coordinates are passed to the DeepSORT algorithm. DeepSORT assigns a unique ID to each detected object and tracks its trajectory across frames, enabling reliable counting, movement analysis, and persistent identification.

5. **Post-Processing**: This stage is highly case-specific, applying unique logic to the tracked object data:

   - **Use Case 1 (Gunny Bags)**:
     - *Counting*: Incrementing counts as new, tracked bags cross a defined virtual line or enter/exit a designated zone (e.g., truck loading bay). Logic to avoid double counting based on track IDs is critical.
     - *Volumetric Analysis*: Estimating the volume of individual bags (if dimensions are known and identifiable) or stacks of bags. This can be done using pixel-based measurements calibrated with known reference objects in the scene or by integrating with monocular depth estimation like intel MiDAS. The system will aim to automatically measure and verify their numbers while being loaded, unloaded, and stacked.

  - *Inventory Reconciliation*: Comparing the automatically counted numbers with expected inventory figures or manually entered data to flag discrepancies.

- **Use Case 2 (Vehicle Recognition)**:
  - *OCR Processing*: Once a license plate region is localized, Optical Character Recognition (OCR) techniques (e.g., a fine-tuned PaddleOCR) are applied to extract the alphanumeric characters from the plate. The system will support multiple languages/fonts if necessary for diverse plate formats.
  - *Vehicle Authentication & Logging*: The recognized license plate number is checked against a database of authorized vehicles. Timestamped logs of all vehicle entries and exits are maintained. Alerts are triggered for unauthorized entries. Vehicle categorization (e.g., truck, staff car) will also be logged.
- **Use Case 3 (Facial Recognition)**:
  - *Facial Feature Extraction*: For each detected and tracked face, a deep learning model (e.g., ArcFace, FaceNet) extracts a unique numerical vector (embedding) representing the facial features.
  - *Personnel Identification & Verification*: The extracted embedding is compared against a pre-enrolled database of embeddings of authorized personnel (warehouse managers, hamalis/labour). If a match is found above a certain confidence threshold, the individual is identified and verified. Alerts are triggered for unknown individuals or unauthorized access attempts. A history log is maintained.

6. **Result Storage & Alerting**: Standardized storage mechanisms (e.g., SQL or NoSQL databases - preferably NoSQL) will be used for metadata (timestamps, camera ID, object IDs), analysis results (counts, license plates, identities), and event logs. A real-time alerting module will push notifications (e.g., via SMS, email, or a dashboard update) for critical events like count discrepancies, unauthorized vehicle entry, or unauthorized personnel detection. Real-time streaming inference results for object tracking can be exposed via Kafka streams for integration with other systems or live dashboards.

This unified approach ensures efficient resource utilization by sharing computational load for common tasks, simplifies maintenance due to a common codebase for the core pipeline, and guarantees consistent performance and reliability across these three critical operational use cases.

5.1.2 Differentiated Approach for Use Case 4

Unlike the first three use cases which focus on real-time object detection, tracking, and immediate action based on predefined rules, Use Case 4 (Contextual Intelligence for near Real-time analysis and Query) necessitates a fundamentally different architectural and technological approach. This use case is about understanding broader scene context, identifying anomalies, and enabling semantic search over video content.

- **Data Processing Strategy**: Instead of continuous real-time object detection on every frame, this use case will typically involve periodic sampling of video frames (e.g., one frame every few seconds or keyframe extraction based on scene changes). The focus is on capturing representative snapshots of activities and environments over time to build a searchable index. Analysis is performed in near real-time, meaning there might be a slight delay (seconds to minutes) as opposed to the millisecond-level real-time processing of the other use cases.

- **Technology Stack**: This system will heavily leverage advanced vision-language models (VLMs) like CLIP (Contrastive Language-Image Pre-training) or similar multimodal architectures. These models can understand and associate textual descriptions with visual content, enabling powerful semantic search capabilities. This contrasts with the pure computer vision models (like YOLO) used for object detection in the other use cases.

- **Storage Requirements**: A key component will be a vector database. Video frames (or features extracted from them by VLMs) will be converted into dense vector embeddings. These embeddings capture the semantic meaning of the visual content. The vector database allows for efficient similarity searches, finding frames or video segments that are semantically similar to a natural language query or an example image.

- **Query Interface and Interaction**: The system will implement a Retrieval-Augmented Generation (RAG) architecture for user interaction. When a user poses a natural language query (e.g., ”Show me instances where gunny bags were left unattended in Zone A yesterday”), the RAG system first retrieves relevant video segments/frames from the vector database based on the semantic meaning of the query. Then, a large language model (LLM) uses this retrieved context to generate a coherent and informative answer, potentially including links to the video segments, timestamps, and a summary of the events.

- **Multilingual Support**: The system will be designed to process queries and generate descriptions in both English and Telugu (Telugu for later versioins), catering to the linguistic diversity of the users in the warehouse environment. This involves using VLMs and LLMs that have strong multilingual capabilities or employing translation services within the pipeline.

The contextual intelligence module, therefore, functions more like a sophisticated, multimodal search engine operating over the historical and near real-time video content from the warehouse. It enables users to ask complex questions and gain insights into activities, anomalies (e.g., unusual movements, presence of individuals in restricted areas at odd times), and events (e.g., specific operations, safety incidents) that might not be easily captured by rule-based systems. This capability complements the real-time monitoring and alerting functions of the first three use cases, creating a comprehensive and intelligent warehouse video analytics and management system. The system will operate reliably under varying conditions, such as different lighting or camera angles, and maintain a searchable log of analyzed events for efficient monitoring and review.

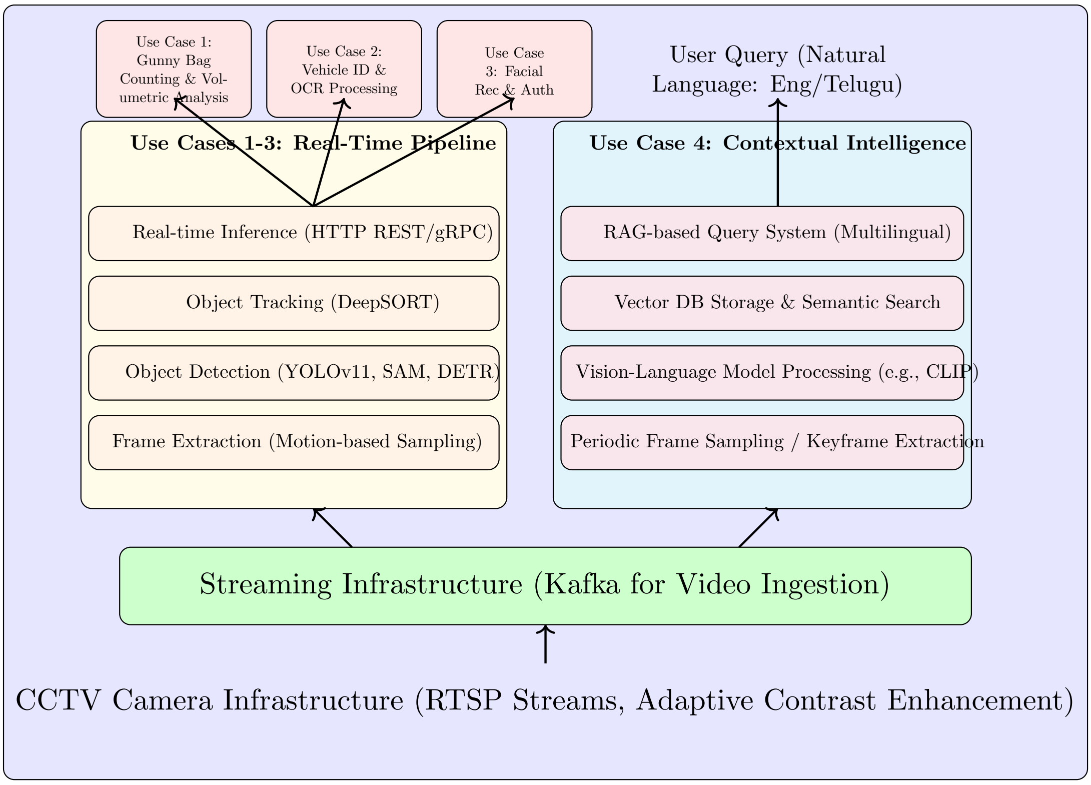
Use Case 1: Gunny Bag Counting & Volumetric Analysis
Use Case 2: Vehicle ID & OCR Processing
Use Case 3: Facial Rec & Auth
User Query (Natural Language: Eng/Telugu)
Use Cases 1-3: Real-Time Pipeline
Use Case 4: Contextual Intelligence
Real-time Inference (HTTP REST/gRPC)
Object Tracking (DeepSORT)
Object Detection (YOLOv11, SAM, DETR)
Frame Extraction (Motion-based Sampling)
RAG-based Query System (Multilingual)
Vector DB Storage & Semantic Search
Vision-Language Model Processing (e.g., CLIP)
Periodic Frame Sampling / Keyframe Extraction
Streaming Infrastructure (Kafka for Video Ingestion)
CCTV Camera Infrastructure (RTSP Streams, Adaptive Contrast Enhancement)

Figure 4: Unified and Differentiated Processing Pipelines with Case-Specific Components and Technologies. The diagram illustrates shared infrastructure for video ingestion and streaming, a common real-time pipeline for Use Cases 1-3 involving frame extraction, object detection (e.g., YOLOv11), tracking (DeepSORT), and real-time inference APIs, followed by specialized post-processing. Use Case 4 employs a distinct pipeline with periodic sampling, vision-language models, vector databases, and a RAG-based query system for contextual analysis and natural language search.

5.2 Use Case 1: Real-time Gunny Bags Counting and Volumetric Analysis

This use case is central to inventory management and operational efficiency in warehouses, focusing on the dynamic tracking, counting, and volumetric analysis of gunny bags during loading, unloading, and stacking, as highlighted by the reference. The solution uses AI-driven video analytics from CCTV footage.

5.2.1 Technical Approach

Our solution employs a robust three-stage real-time pipeline specifically designed for gunny bags:

1. **Detection**: A custom-trained YOLOv11 model forms the core of this stage. This model will be trained on a diverse dataset of gunny bags, enabling it to accurately identify individual gunny bags even when they are in motion, partially occluded, or tightly packed on trucks. The model will learn features like texture, shape, and common visual cues of gunny bags used in APSCSCL warehouses. Adaptive contrast enhancement applied prior to detection will ensure reliability

even in poor lighting conditions commonly found in warehouse interiors or during night operations. Alternative models like SAM could be explored if pixel-perfect segmentation is needed for highly accurate volumetric analysis of irregular stacks or we can employ our own segmentation models using detectron2.

2. **Tracking**: The DeepSORT algorithm will take the bounding boxes of detected gunny bags from YOLOv11 and assign a unique tracking ID to each bag. As bags move across frames (e.g., being moved by hamalis, on a conveyor, or during vehicle loading/unloading), DeepSORT maintains their identity. This is crucial to prevent double-counting a single bag multiple times as it moves through the camera's field of view and to accurately count bags crossing a virtual line or entering/exiting a defined zone. Motion tracking capabilities are inherent in this stage.

3. **Counting and Volumetric Analysis**:

   - **Real-time Counting**: Virtual lines or zones will be defined in the CCTV camera's view (e.g., across the entrance of a truck, or entry/exit points of a storage area). When a tracked gunny bag crosses this line in a specified direction, its count is registered. Logic will be implemented to handle complex scenarios, such as bags being temporarily placed down and then picked up again within the counting zone.
   - **Volumetric Analysis**: For volumetric analysis, an initial calibration step is required. This involves providing the system with known dimensions of standard gunny bag types (e.g., 25kg, 50kg bags) and a reference object of known size in the camera's view.
     - *Individual Bags*: The system can estimate the volume of an individual detected bag based on its pixel dimensions in the frame, correlated with its known type (if distinguishable by the model or other inputs).
     - *Stacks/Piles*: For stacks of bags, the system can estimate the overall volume by analyzing the dimensions of the detected stack. If individual bags within the stack can be segmented (potentially using SAM), a more accurate count and volume can be derived.
     - *Vehicle Load Estimation*: By analyzing the space occupied by gunny bags within a detected truck or loading area, an estimation of the total volume of bags loaded or unloaded can be made. This helps in automatically measuring and verifying their numbers.

   Real-time streaming inference using Kafka streams will provide these counts and volumetric estimations to a central dashboard or inventory system.

5.2.2 Training Data Requirements

High-quality, diverse training data is paramount for the accuracy of the AI models:

- **Labeled Dataset of Gunny Bags**: Thousands (for PoC maybe hundreds) of images and video frames containing gunny bags under various conditions, with precise bounding boxes annotating each bag. This includes bags of different sizes, colors, materials, and states (e.g., full, partially full, empty if relevant).

- **Classification by Type (e.g., 25kg, 50kg, etc.)**: If different types of bags need to be counted or analyzed separately, the training data must include labels for these types. The model will then learn to distinguish them based on visual cues like size, shape, or markings.
- **Data from Loading/Unloading/Stacking Scenarios**: Specific footage capturing the dynamic movement of bags during these operations is essential. This includes bags carried by workers, moved on forklifts or conveyors, and stacked in various configurations.
- **Vehicle Detection with Volume Attributes**: For analyzing loads on vehicles, images of trucks and other transport vehicles, both empty and loaded with gunny bags, are needed. Annotations should define vehicle boundaries and potentially the region where bags are loaded.
- **Diverse Lighting and Occlusion Conditions**: Crucially, the dataset must include examples from all expected operational conditions: bright daylight, dim artificial light, shadows, night-time (if applicable), partial occlusions (e.g., bags hidden by workers or equipment), and different camera angles and distances. This ensures the system is robust and functions effectively with adaptive contrast enhancement.

5.2.3 Model Training, Deployment, and Operational Details

- **Training Time and GPU Hours (Estimates)**:
  - *YOLOv11 (Gunny Bag Detection Model)*:
    - ∗ Initial full training on a comprehensive dataset (e.g., 2,000-20,000 annotated images): Estimated 40-80 GPU hours. This typically involves training for hundreds of epochs on 1-4 high-end GPUs.
    - ∗ Fine-tuning or retraining with new data (e.g., a few thousand new images targeting specific failure cases): Estimated 20-60 GPU hours per cycle.
  - *DeepSORT (Re-identification Feature Extractor - if custom trained)*:
    - ∗ If a custom feature extractor is trained for better re-identification of gunny bags: Estimated 40-120 GPU hours. Often, pre-trained models are fine-tuned, reducing this time.
- **GPU Machine Configuration (Indicative)**:
  - *Training Setup*:
    - ∗ GPUs: 1 to 4 units of NVIDIA A100 (40GB/80GB VRAM), H100 (80GB VRAM), or high-end consumer GPUs like RTX 3090/4090 (24GB VRAM).
    - ∗ CPU: Modern multi-core processor (e.g., Intel Xeon, AMD EPYC).
    - ∗ RAM: 64GB to 256GB+, depending on dataset size and batch processing.
    - ∗ Storage: Fast NVMe SSDs for datasets and model checkpoints.

  - *Inference Setup (per processing node, handling multiple streams)*:
    - GPUs: NVIDIA T4 (16GB VRAM), RTX A4000/A5000/A6000, or Jetson AGX Orin/Xavier for edge deployments. The number of GPUs and their power will scale with the number of concurrent video streams and the desired processing framerate (e.g., 5-10 FPS per stream).
    - CPU/RAM: Adequate to support the GPU workload and I/O operations.

- **Feedback Loop and Reiterations**:
  - *Continuous Monitoring*: Post-deployment, the system's performance (counting accuracy, detection robustness, tracking consistency) will be continuously monitored using automated metrics and human review of flagged events or samples.
  - *Data Collection*: A mechanism will be in place to collect challenging or misclassified frames/video segments. This includes scenarios where bags are missed, incorrectly counted, or misidentified.
  - *Re-annotation and Retraining*: Collected data will be annotated and incorporated into the training set. Models will be periodically retrained (e.g., bi-weekly or monthly initially, then quarterly or as needed) to improve accuracy and adapt to changing conditions or bag types.
  - *Iterative Refinement*: Each retraining cycle constitutes an iteration. We anticipate 3-5 major retraining iterations during the first year to achieve high robustness, followed by less frequent updates.
  - *A/B Testing*: New model versions will be tested in a staging environment or on a subset of cameras before full rollout to ensure improvements and prevent regressions.

5.2.4 Expected Outcomes

- **Real-time Accurate Count**: Continuous and accurate counting of gunny bags during loading, unloading, and stacking operations, delivered in real time to relevant stakeholders.

- **Volumetric Estimation with Defined Accuracy**: Volumetric estimation of individual bags and stacks, aiming for an accuracy of ±5% or better, subject to calibration quality and scene complexity. This allows for verification of loaded/unloaded quantities against dispatch notes.

- **Automatic Reconciliation with Expected Inventory**: For later versions, the system will provide data that can be automatically compared against existing inventory records, waybills, or stock ledgers, flagging discrepancies immediately (Long term plan)

- **Alert Mechanism for Discrepancies and Anomalies**: An automated alert system will notify supervisors or managers in real-time (e.g., via dashboard, SMS, email) in case of significant count mismatches, unusual delays in loading/unloading, or other defined anomalies.

- **Improved Operational Efficiency**: Reduction in manual effort for counting, faster turnaround times for vehicles, and minimized human error.

5.3 Use Case 2: AI-Powered Vehicle Recognition

This system is designed to enhance security and streamline gate operations at warehouse premises by automatically reading and logging vehicle license plates for authentication and tracking.

5.3.1 Technical Approach

Our AI-based vehicle recognition system will implement a multi-stage workflow:

1. **Vehicle Detection and Classification**: A YOLOv11 model, trained on a comprehensive dataset of vehicles typically visiting warehouses (trucks, lorries, tempos, staff cars, etc.), will first detect the presence of a vehicle in the CCTV feed from entry/exit points. The model will also classify the type of vehicle.

2. **License Plate Localization (LPL)**: Once a vehicle is detected, a secondary, specialized detection model (which could also be a fine-tuned YOLOv11 or another dedicated LPL model) will be employed to precisely locate the license plate region on the vehicle. This model is trained to identify license plates of various sizes, formats, and in different positions on vehicles.

3. **OCR Processing for Number Plate Reading**: The localized license plate image is then passed to an Optical Character Recognition (OCR) engine. We propose using PaddleOCR due to its strong performance on diverse text, including Indian license plates, and will apply custom enhancements. These enhancements may include:

   - Specific pre-processing for plate images: binarization, noise removal, de-skewing, and super-resolution if needed, especially for low-quality footage or challenging lighting.
   - Fine-tuning the OCR model on a dataset of Indian license plates, encompassing various fonts, character types (including support for multiple languages/fonts if present on plates, although standard Indian plates have a defined format), and plate conditions (e.g., dirty, damaged).
   - Character segmentation and validation rules based on standard license plate formats to improve accuracy.

4. **Authentication, Logging, and Alerting**: The extracted license plate number is then processed:

   - **Authentication**: Verified against a database of registered/authorized vehicles. This database would contain plate numbers, associated transporter details, approved cargo types, etc.
   - **Timestamp Logging**: All vehicle entries and exits are logged with timestamps, camera ID, vehicle type, and the recognized license plate number. This creates an auditable trail of vehicle movements.

- **Movement Tracking**: By correlating data from multiple cameras, the system can track vehicle movement within the premises if required.
- **Alerts for Unauthorized Entries**: If a detected license plate is not found in the authorized database or is flagged for any reason, the system will trigger an immediate alert to security personnel.

The system will be designed to function effectively in variable lighting conditions, day and night, through robust image processing and model training.

5.3.2 Training Data Requirements

- **Vehicle Images with Bounding Boxes and Type Classification**: Large dataset of vehicles of various types (motor bikes, trucks, tankers, utility vehicles, cars etc.) captured from angles typical of CCTV cameras at entry/exit gates. Each vehicle should be annotated with a bounding box and its type.
- **License Plates with Bounding Boxes and Type Classification**: Extensive collection of images specifically containing license plates. Annotations should include bounding boxes for the plates. If different plate styles (e.g., commercial vs. private, different state formats within India) need special handling, these should be classified.
- **Synthetic and Real OCR Dataset for Character Recognition**:
  - *Synthetic Data*: Generation of a large volume of synthetic license plate images with diverse fonts, characters, backgrounds, and augmentations (blur, noise, lighting variations). This helps in bootstrapping the OCR model.
  - *Real Data*: A substantial collection of real license plate images, cropped and annotated with the correct characters, is crucial for fine-tuning the OCR model to handle real-world complexities.
- **Various Lighting Conditions and Camera Angles**: Data must cover a wide spectrum of environmental conditions: direct sunlight, shadows, overcast weather, rain (which can cause reflections), and night-time (requiring IR illumination or good low-light camera performance). Different camera heights, distances, and angles must also be represented.

5.3.3 Model Training, Deployment, and Operational Details

- **Training Time and GPU Hours (Estimates)**:
  - *YOLOv11 (Vehicle Detection & LPL Models)*:
    - Initial training for vehicle detection: 20-200 GPU hours (dataset size dependent).
    - Initial training for license plate localization (LPL): 30-150 GPU hours (dataset of cropped vehicle images or specific LPL datasets).
    - Fine-tuning: 15-40 GPU hours per cycle for each model.
  - *PaddleOCR (or similar OCR model fine-tuning)*:

    - Fine-tuning on custom Indian license plate dataset (real and synthetic): 30-100 GPU hours. Training from scratch is usually not required for OCR models.

- **GPU Machine Configuration (Indicative)**:
  - *Training Setup*:
    - GPUs: Similar to Use Case 1; 1-4x NVIDIA A100/H100 or RTX 3090/4090.
    - CPU/RAM/Storage: Similar robust configuration as Use Case 1.
  - *Inference Setup (per gate or processing node)*:
    - GPUs: NVIDIA T4, RTX A2000/A4000, or Jetson AGX Orin (especially if processing is at the gate). The LPL and OCR stages are sequential and typically less demanding than continuous object tracking across many streams, but real-time performance is key.
    - CPU/RAM: Moderate, sufficient for I/O and model orchestration.

- **Feedback Loop and Reiterations**:
  - *Performance Tracking*: OCR accuracy (character error rate, plate recognition rate), vehicle detection rate, and LPL success will be key metrics.
  - *Data Augmentation*: Collection of images where plates are misread, vehicles are missed, or LPL fails (e.g., due to mud, damage, challenging angles, new plate designs).
  - *Targeted Retraining*: Models (YOLO for detection/LPL, OCR model) will be retrained with new data, focusing on improving performance on failure cases. We anticipate 2-4 major retraining iterations in the first year for ANPR.
  - *Rules Engine Refinement*: Post-OCR logic (e.g., format validation, character correction rules) may also be updated based on observed error patterns.

5.3.4 Expected Outcomes

- **High Accuracy License Plate Recognition**: Aiming for 95%+ accuracy in reading license plate characters under normal operating conditions (clear plates, adequate lighting).

- **Automated Vehicle Entry/Exit Logging**: Comprehensive and automated logs of all vehicle movements with timestamps, plate numbers, and vehicle types, reducing manual record-keeping.

- **Real-time Unauthorized Vehicle Alerts**: Immediate notification to security personnel upon detection of unauthorized or blacklisted vehicles attempting entry.

- **Enhanced Security and Access Control**: Improved control over vehicles entering and exiting the warehouse premises.

- **Integration Capability with VAHAN Portal**: The system will be designed with the capability to integrate with the VAHAN portal (India's national vehicle registry) via APIs, if available and permissible, to fetch vehicle owner details for further verification or information. This would require handling API request/response and data parsing.
- **Vehicle Categorization**: Automatic classification of vehicles (e.g., truck, tempo, car) for better traffic analysis and management within the premises.

5.4 Use Case 3: AI-Driven Facial Recognition

This system aims to enhance security and personnel management within warehouses by authenticating managers and hamalis (labour), and detecting unauthorized intrusions in real time using AI-based facial recognition from CCTV footage.

5.4.1 Technical Approach

Our facial recognition system will employ a sophisticated pipeline optimized for accuracy and real-time performance:

1. **Face Detection**: The initial step involves accurately detecting human faces in the video frames. We will utilize a robust face detector like RetinaFace, which is known for its high performance even with faces of varying scales, poses, and partial occlusions (e.g., people wearing caps or partially turned away). The model will be optimized to function efficiently in different lighting conditions typically found in warehouses.
2. **Facial Landmark Detection and Alignment (Implicit/Explicit)**: Once a face is detected, key facial landmarks (e.g., eyes, nose, mouth corners) are often identified. These landmarks are used to align the face to a canonical pose, which improves the consistency and accuracy of the subsequent feature extraction step. Many modern feature extractors perform this implicitly.
3. **Feature Extraction (Embedding Generation)**: For each detected and aligned face, a deep convolutional neural network (CNN) model, such as ArcFace, CosFace, or SphereFace, will be used to extract a compact and discriminative numerical representation (embedding). ArcFace, for example, is known for its excellent performance in learning highly discriminative features by imposing an additive angular margin penalty during training. This embedding vector uniquely represents the identity of the face.
4. **Identity Matching and Verification**: The extracted facial embedding is then compared against a database of pre-enrolled embeddings of authorized personnel (warehouse managers, hamalis, and other staff). This comparison involves calculating a similarity score (e.g., cosine similarity) between the live face embedding and the stored embeddings.
   - If the similarity score with an enrolled individual surpasses a predefined confidence threshold, the system identifies and verifies the person.

- If the score does not match any enrolled personnel with sufficient confidence, the individual may be flagged as unknown or potentially unauthorized.
- We will test other similarity scores as well.

5. **Access Verification, Logging, and Alerting**:
    - **Role-Based Authentication**: The system can be configured for role-based access control. For instance, certain areas might be restricted, and alerts can be triggered if a person (even if authorized in general) tries to access an area not permitted for their role.
    - **Logging**: All recognition events (successful verifications, failed attempts, detection of unknown individuals) are logged with timestamps, camera location, and the identity (if known). This history log is crucial for monitoring and security audits.
    - **Near Real-time Alerts**: In cases of unauthorized access attempts or detection of unknown individuals in restricted zones, the system will trigger near real-time alerts to security staff or management.

The solution will be engineered to distinguish authorized personnel from unknown individuals reliably.

5.4.2 Training Data Requirements

- **Face Images of Authorized Personnel**: A curated dataset of high-quality face images for all individuals to be enrolled in the system (managers, workers/hamalis, etc.). Ideally, multiple images per person, captured under varying conditions:
    - Different angles (frontal, profiles, tilted).
    - Various lighting conditions (indoor, outdoor, dim, bright).
    - Diverse facial expressions.
    - With and without accessories like glasses (if commonly worn), but not masks that heavily occlude features critical for the chosen model.
- **Classification by Role and Access Privileges**: Metadata associated with each enrolled individual, specifying their role (e.g., manager, hamali, visitor) and corresponding access privileges within the warehouse. This is used for role-based authentication.
- **Data for Handling Variations**: To ensure robustness, the underlying face detection and feature extraction models should be pre-trained on large-scale, diverse public datasets. Fine-tuning might be considered if specific challenging conditions in the warehouse are not well-covered. This includes images with partial obstructions (e.g., hand near face, part of face in shadow).

5.4.3 Model Training, Deployment, and Operational Details

- **Training Time and GPU Hours (Estimates)**:
  - *Face Detection Model (e.g., RetinaFace, or YOLOv11 fine-tuned for faces)*:
    - Typically, pre-trained models are used. Fine-tuning on warehouse-specific imagery (if needed for very challenging conditions): 20-60 GPU hours.
  - *Facial Feature Extraction Model (e.g., ArcFace)*:
    - Pre-trained models on large public datasets (e.g., MS-Celeb-1M, VGGFace2) are standard. Training such models from scratch can take thousands of GPU hours.
    - Fine-tuning on a domain-specific dataset (e.g., if a specific demographic or common occlusions need better handling): 30-150 GPU hours. For most applications, high-quality pre-trained models suffice.
- **GPU Machine Configuration (Indicative)**:
  - *Training/Fine-tuning Setup*:
    - GPUs: NVIDIA A100/H100 or RTX 3090/4090, especially if fine-tuning large feature extraction models.
    - CPU/RAM/Storage: Similar robust configuration.
  - *Inference Setup (per processing node or for multiple cameras)*:
    - GPUs: NVIDIA T4, RTX A2000/A4000, Jetson AGX Orin. Face detection and embedding extraction need to be fast for real-time identification.
    - CPU/RAM: Moderate, for database lookups and managing identities.
- **Feedback Loop and Reiterations**:
  - *Enrollment Quality*: The primary ”training” for identification is the enrollment process. Ensuring high-quality, diverse enrollment images per individual is critical.
  - *Performance Metrics*: False Acceptance Rate (FAR), False Rejection Rate (FRR), and True Positive Identification Rate will be monitored.
  - *Data Collection for Model Improvement*: Instances of failed detections, misidentifications, or issues with specific lighting/angles will be collected.
  - *Model Updates*: Pre-trained models for detection/feature extraction may be updated to newer, more robust versions. Fine-tuning cycles (1-2 per year if necessary) based on collected data to address specific performance gaps.
  - *Threshold Tuning*: The similarity threshold for matching will be periodically reviewed and adjusted based on FAR/FRR performance to optimize security versus convenience.

5.4.4 Expected Outcomes

- **High Accuracy in Personnel Recognition**: Aiming for 90%+ accuracy in correctly identifying enrolled personnel under typical warehouse conditions. The exact accuracy will depend on the quality of CCTV cameras, lighting, and adherence to enrollment best practices.
- **Real-time Unauthorized Access Alerts**: Prompt alerts (e.g., notifications on a security dashboard, SMS, or mobile app) when unauthorized individuals are detected in restricted areas or when access is attempted by unrecognised persons.
- **Personnel Attendance Tracking (Potential Application)**: The system can contribute to automated attendance tracking by logging the first Sighting of authorized personnel in designated areas or at specific times.
- **Historical Access Logs for Security Audits**: Comprehensive logs of all facial recognition events, providing an auditable trail for security reviews, incident investigations, and compliance purposes. This history log allows for better monitoring.
- **Enhanced Security Posture**: Deterrence of unauthorized entry and improved overall security for sensitive areas within the warehouse.
- **Efficient Functioning in Different Lighting**: The system will be designed to maintain performance across varied lighting scenarios through robust algorithms and appropriate camera configurations (e.g., WDR - Wide Dynamic Range cameras).

5.5 Use Case 4: Contextual Intelligence for Near Real-time Analysis and Query

This use case aims to develop an AI-powered system capable of analyzing videos in near real-time, grounded in context to capture anomalies and significant events. It will enable users to search, retrieve, and query activities and occurrences within video scenes using natural language.

5.5.1 Technical Approach

Our contextual intelligence system moves beyond simple object detection and implements a deeper understanding of video content:

1. **Video Chunking and Pre-processing**: Continuous CCTV video streams are first segmented into manageable, shorter clips or ”chunks” (e.g., 1-5 minutes long). This makes the subsequent processing and indexing more efficient.
2. **Frame Sampling/Keyframe Extraction**: Instead of processing every frame in a chunk, keyframes are extracted. This can be done by sampling at a fixed interval (e.g., 1 frame every 5-10 seconds) or by using algorithms that detect significant visual changes or represent the core content of a short video segment.

3. **Scene Understanding with Vision-Language Models (VLMs)**: The extracted keyframes or short video segments are then processed by powerful VLMs (e.g., models based on CLIP, VideoCLIP, or similar architectures that can process sequences). These models generate rich embeddings (vector representations) that capture the semantic content of the frames/clips, including objects present, actions occurring, and the overall scene context. They can also generate textual descriptions of the scenes. This allows the system to identify contextual anomalies (e.g., unusual crowding of people) and significant events (e.g., start of loading activity, a safety incident like fire).

4. **Vector Database for Semantic Indexing**: The generated embeddings (and potentially textual descriptions) are stored in a specialized vector database (e.g., FAISS, Milvus, Pinecone). This database is optimized for efficient similarity searches. When a user queries the system, their query is also converted into an embedding, and the database retrieves video chunks/frames whose embeddings are closest (most similar) to the query embedding.

5. **Natural Language Processing (NLP) and RAG for Query Interface**:

    - Users interact with the system using natural language queries (e.g., ”Find all instances of trucks arriving at Dock 5 between 2 PM and 4 PM last Tuesday,” or ”Show me any unusual activity near the high-value storage area last night”).
    - A Retrieval-Augmented Generation (RAG) system is employed. The NLP component first understands the user’s query. The retrieval part then queries the vector database to find the most relevant video segments. Finally, a Large Language Model (LLM) takes these retrieved segments (the context) and the original query to generate a comprehensive, human-readable answer. This answer can include direct links to video segments, timestamps, locations, individuals involved (if cross-referenced with Use Case 3), and a summary of the findings.

    This AI-driven video analysis will operate reliably under varying conditions, such as different lighting or camera angles, due to the robustness of modern VLMs.

5.5.2 Implementation Strategy

- **Few-Shot/Zero-Shot Learning Approach**: Many modern VLMs exhibit strong few-shot or even zero-shot learning capabilities. This means they can often identify and understand new objects, actions, or scenarios with very few or no specific training examples from the warehouse itself, significantly reducing the initial labeled data requirement for basic contextual understanding. However, fine-tuning on domain-specific data can further improve performance for very specific warehouse events or anomalies.

- **Bilingual Capability (English and Telugu)**: The system will be designed to support descriptions and queries in both English and Telugu. This will involve using VLMs and LLMs that are either inherently multilingual or can be effectively used with translation layers. User interface elements will also support both languages.

- **Metadata Tagging and Event Logging**: Video chunks/frames will be tagged with rich metadata, including timestamps, camera location, detected objects (from Use Cases 1-3 if integrated), generated textual descriptions, and identified events or anomalies. A searchable log of all analyzed events will be maintained for efficient monitoring and review.
- **RAG System for Context-Aware Responses**: The RAG architecture ensures that the LLM's responses are grounded in the actual video content retrieved, minimizing hallucination and providing accurate, context-aware answers to user queries.
- **User-Configurable Event Definitions**: While the system can detect common anomalies, users will be able to define specific events or scenarios they want to monitor (e.g., ”alert if more than 5 people gather near Gate B after 10 PM,” or ”log all instances of forklifts operating in pedestrian areas”). This could be achieved through a rule-engine interface that leverages the outputs of the VLM.

5.5.3 Model Training, Deployment, and Operational Details

- **Training Time and GPU Hours (Estimates)**:
  - *Vision-Language Models (VLMs, e.g., CLIP-based)*:
    - Primarily leverage large pre-trained foundation models. Training these from scratch requires massive datasets and thousands to tens of thousands of GPU hours (typically done by research labs/large corporations).
    - Fine-tuning on domain-specific data (e.g., several thousand warehouse-specific image-text pairs or video clips with descriptions): Estimated 50-200 GPU hours on 1-4 A100/H100 GPUs.
  - *Large Language Models (LLMs for RAG)*:
    - Primarily leverage pre-trained models (e.g., Llama 2/3, Mistral, or proprietary models via API).
    - Fine-tuning for specific query understanding, response style, or domain terminology on a curated dataset (e.g., few thousands to tens of thousands of query-context-answer examples): Estimated 20-100 GPU hours for smaller open-source models (7B-13B parameters) on 1-8 A100/H100 GPUs. Larger models require more.
- **GPU Machine Configuration (Indicative)**:
  - *Training/Fine-tuning Setup*:
    - GPUs: Multi-GPU servers with NVIDIA A100 (80GB) or H100 (80GB) are essential for fine-tuning VLMs and LLMs due to large model sizes and memory requirements.
    - CPU: High-performance multi-core CPUs.
    - RAM: 256GB to 1TB+ to handle large datasets and models.
    - Storage: Terabytes of fast NVMe SSD storage.

- *Inference Setup*:
    - VLM Embedding Generation: Can be batched. NVIDIA A10G, L4, T4, or A100/H100 for higher throughput. Several GPUs might be needed depending on the volume of video data processed.
    - LLM for RAG System: Dependent on LLM size. Smaller models (e.g., 7B) might run on high-end CPUs or single GPUs like A10G/L4/T4. Larger models (70B+) will require A100/H100 GPUs for acceptable latency.
    - Vector Database: Primarily CPU and RAM intensive for search, though some (e.g., Milvus with GPU-enabled FAISS) can leverage GPUs to accelerate similarity search.

- **Feedback Loop and Reiterations**:
    - *Query Performance Monitoring*: Track relevance of search results, user satisfaction (e.g., through implicit feedback like click-through rates on results, or explicit feedback mechanisms), and query failure rates.
    - *Data Curation for Fine-tuning*: Collect examples of poor searches, misidentified contexts, or new types of events/anomalies not well understood by the VLM/LLM. Create curated datasets for fine-tuning.
    - *Iterative Fine-tuning*: Periodically fine-tune VLM and LLM components (e.g., quarterly or semi-annually) to improve understanding of warehouse-specific context, query nuances, and multilingual capabilities. Anticipate 2-3 significant fine-tuning iterations in the first 1-2 years.
    - *Prompt Engineering*: Continuously refine prompts used in the RAG system to improve the quality and relevance of LLM-generated answers.
    - *Embedding Updates*: As new video data is processed, its embeddings are continuously added to the vector database. The underlying VLM used for embedding generation might be updated, potentially requiring re-embedding of historical data if significant model improvements occur.

5.5.4 Expected Outcomes

- **Searchable Video Database with Natural Language Interface**: A powerful ”Google-like” or ”YouTube-style” search capability for the warehouse’s video footage. Users can ask questions in English or Telugu to quickly find relevant video segments without manually scrubbing through hours of recordings.

- **Near Real-time Event Detection and Alerting for Predefined and Anomalous Scenarios**: The system will automatically identify and flag:
    - User-defined significant events (e.g., completion of a loading process, arrival of a specific type of material).
    - Contextual anomalies (e.g., a person in a restricted zone, a vehicle left for too long in a loading bay, unusual congregation of people, sudden fast movement in a normally quiet area).

  Alerts for critical events can be sent in near real-time.

- **Efficient Monitoring and Review**: Drastically reduces the time and effort required for security personnel or managers to review footage for investigations or monitoring compliance. Detailed insights like timestamps, locations, or individuals involved can be easily queried.
- **Improved Situational Awareness and Incident Response**: Faster identification of incidents or emerging situations allows for quicker response.
- **Data-Driven Insights**: Over time, the logged data can be analyzed to identify patterns, optimize warehouse layouts or processes, and improve overall safety and security.

5.6 Warehouse Video Analytics System: A UML Class Diagram Overview

The diagram below introduces the architecture of a comprehensive Warehouse Video Analytics System through a UML class diagram. The system is designed to process and analyze video feeds from CCTV cameras to provide actionable insights for various warehouse operations. It leverages a modular design, incorporating components for video input and streaming, core processing tasks like object detection and tracking, and specific modules for different use cases such as gunny bag analysis, vehicle recognition, facial recognition, and contextual intelligence.

UML Class Diagram for the Warehouse Video Analytics System. This diagram details the software architecture, showcasing distinct packages for input/streaming infrastructure (e.g., `CCTVCameraInfrastructure`, `VideoStreamAcquisition`, `KafkaService`), core processing modules (e.g., `FrameExtractionModule`, `ObjectDetectionModel` with implementations like `YOLOv11_Model`, `ObjectTrackingModule`), a unified real-time pipeline orchestrator, use-case specific processors (for gunny bag analysis, vehicle recognition, and facial recognition), a contextual intelligence system leveraging vision-language models (e.g., `CLIP_Model`) and RAG querying, and shared services like databases (`SQL_DB`, `NoSQL_DB`, `VectorDBService`) and alerting. Relationships highlight the data flow from video ingestion through various analytical stages to output generation and storage.

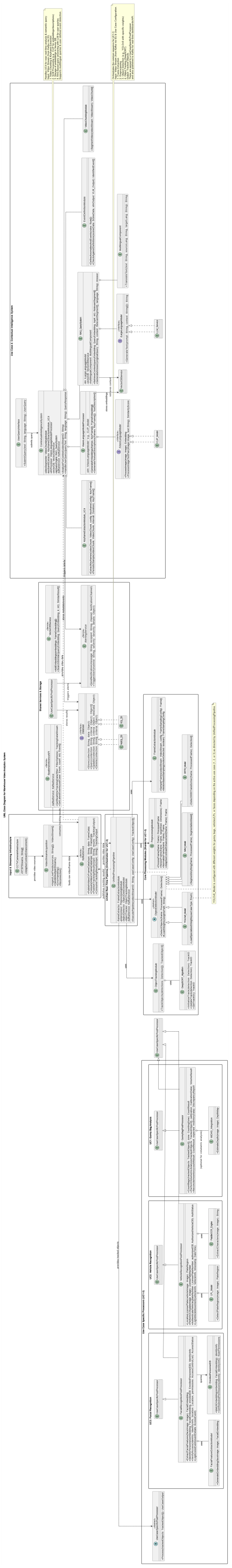

Figure 5: Full resolution image is available at: https://gist.github.com/INF800/2b9e83f7461794c58c269020b5d3105f

6 Implementation Methodology and Timeline

Our implementation follows a structured approach, meticulously aligned with APSCSCL's hackathon phases. We are committed to delivering a solution that excels in technical feasibility, scalability, cost-effectiveness, impact, and seamless integration, addressing all four outlined use cases. The timeline below details our plan from initial development through to deployment readiness.

6.1 Phase 1: Development and Initial Testing (Current - Shortlisted period)

This crucial initial phase, spanning from the present date through to the shortlisting announcement, focuses on building a robust foundation for all use cases. Key activities include:

- **Architecture design and component selection:** Defining a scalable system architecture that leverages existing CCTV infrastructure. This involves selecting appropriate AI models (e.g., for object detection for gunny bags, OCR for vehicle plates, facial recognition, and complex event analysis) and technologies to build an efficient software layer.
- **Data preprocessing strategy development:** Create data strategy based on specifications above.
- **Initial model selection and adaptation:** Selecting robust baseline AI models tailored for each specific use case:
  - Real-time gunny bag counting and volumetric analysis.
  - AI-Powered Vehicle Recognition.
  - AI-Driven Facial Recognition.
  - Contextual Intelligence for Real-time analysis and Query.

  We will begin adapting these models to the specific nuances of the warehouse environment.
- **Proof-of-concept (PoC) development for all use cases:** Developing initial working PoCs demonstrating the core functionalities and technical feasibility of each solution. This includes ensuring the PoCs for real-time gunny bag counting address accuracy in motion and that OCR for vehicle recognition shows promise across various conditions. This phase culminates with the application submission.

6.2 Phase 2: On-Ground Testing (Two weeks after shortlised period)

Aligning directly with the "On-ground testing" period (Two weeks), this phase is dedicated to refining the solutions in real-world warehouse conditions:

- **Access and integration with CCTV feeds:** Collaborating with APSCSCL to establish secure access to data dump or integration with existing CCTV camera feeds from designated warehouses.

- **Data diversity analysis and sampling:** Collecting and meticulously analyzing diverse video datasets that represent a wide array of operational scenarios. This includes capturing data under challenging conditions such as low light, varying angles for bag detection, partial occlusions for faces (e.g., masks), and different number plate conditions for vehicles.
- **Model fine-tuning with warehouse-specific data:** Iteratively fine-tuning the AI models using the collected on-ground data to significantly enhance performance, accuracy, and robustness for:
  - *Gunny Bags:* Improving real-time detection accuracy of bags in motion and volumetric analysis, while minimizing latency.
  - *Vehicle Recognition:* Enhancing OCR accuracy for lighting, angles, and plate formats.
  - *Facial Recognition:* Boosting face detection and recognition accuracy, especially with obstructions like masks, low illumination, and difficult angles, for staff and intruder verification.
  - *Contextual Intelligence:* Refining the ability under diverse warehouse conditions.
- **Performance optimization for real-time processing:** Optimizing algorithms and deployment strategies to ensure minimal latency and efficient real-time processing, leveraging the existing CCTV infrastructure as a software-only solution where possible to maintain cost-effectiveness.
- **Edge case identification and handling:** Proactively identifying potential edge cases and challenging scenarios specific to APSCSCL warehouse operations and developing robust strategies to address them, ensuring high reliability.

6.3 Phase 3: Solution Showcase (Two to three days on-ground testing)

Coinciding with the ”Presentation round”, this phase will demonstrate the capabilities, performance, and potential impact of our integrated solution, in anticipation of the evaluation and scoring and award announcement.

- **Technical demonstration of all four use cases:** Delivering compelling live demonstrations of each use case:
  - Automated real-time gunny bag counting improves transparency and reduces pilferage.
  - AI-Powered Vehicle Recognition, showcasing high OCR accuracy and its role in securing in-bound/out-bound logistics and preventing unauthorized access.
  - AI-Driven Facial Recognition, highlighting accurate staff verification (even with masks), intruder detection, and attendance validation capabilities to increase warehouse security.

  - Contextual Intelligence, demonstrating the system's ability for near real-time analysis and query of video feeds to enable high recall of event detection.

- **Performance metrics and accuracy reports:** Presenting comprehensive reports detailing the achieved performance levels, including quantitative metrics on detection accuracy, OCR precision, facial recognition rates under varied conditions, and event detection recall, along with latency benchmarks.

- **Scalability and integration roadmap:** Actionable roadmap for scaling the solution across warehouses and entry/exit points, detailing integration with APSCSCL and VAHAN portals.

- **full-scale deployment plan:** Full-scale deployment plan with timelines, resources, and cost-effectiveness, emphasizing software reuse with existing CCTV. Reiterates benefits to automation, security, and efficiency.

6.4 Phase 4: Deployment Readiness (Post-Hackathon)

Following a successful hackathon outcome and aligning with the "Procurement" phase, we will transition to full deployment readiness:

- **MLOps infrastructure setup:** Establishing a robust Machine Learning Operations (MLOps) pipeline. This will facilitate continuous monitoring, retraining, and redeployment of AI models, ensuring sustained high performance and adaptability to evolving conditions.

- **Full integration with APSCSCL portal (and VAHAN):** Finalizing and deploying the seamless integration of all four AI solutions with APSCSCL's designated portal. This includes completing the VAHAN portal integration for the AI-Powered Vehicle Recognition system.

- **Monitoring and maintenance protocols:** Defining and implementing comprehensive protocols for ongoing system health monitoring, performance tracking, and scheduled maintenance. This ensures long-term operational reliability, minimal downtime, and includes clear update paths for models.

Task	**W1**	**W2**	**W3**	**W4**
System Architecture	●			
Data Processing	●			
Model Training	●	●		
On-Ground Testing		●	●	
Solution Showcase			●	
Deployment Setup				●

Table 1: Implementation Timeline (4 Weeks) - Illustrative high-level plan for core development and testing sprints within the overall hackathon schedule.

7 Deployment and Scalability Strategy

Our deployment and scalability strategy is designed to ensure a phased and robust rollout of the AI-powered CCTV analytics solution, aligning with APSCSCL's evolving needs from initial Proofs of Concept (PoCs) to full-scale, integrated systems across all warehouses. This strategy considers the specific requirements of each use case, from real-time gunny bag counting to contextual intelligence.

7.1 Inference Infrastructure

The inference infrastructure will be built to support the increasing demands of the AI models as they mature from PoC to full deployment.

- **RESTful API Architecture for Model Serving:** This will be fundamental from the short-term PoCs for all use cases (gunny bag counting, vehicle recognition, facial recognition, and contextual intelligence) to allow initial integration and display of data (e.g., real-time count data, vehicle license plates, facial recognition status, verbose event descriptions). In the long term, these APIs will serve fully automated systems integrated with the AP Civil Supplies portal and enable complex queries.

- **Containerized Deployment using Docker and Kubernetes:** Essential for managing deployments across multiple warehouse sites (mid-term for gunny bag counting and vehicle recognition) and scaling to all warehouses (long-term for all use cases). This ensures consistency and simplifies updates.

- **Load Balancing for High-Availability Service:** Crucial as the system scales and real-time alerts and monitoring become more critical (mid-term for facial recognition alerts, long-term for centralized vehicle tracking and real-time stock monitoring).

- **GPU Acceleration for Compute-Intensive Models:** Necessary from the outset for real-time processing of CCTV footage for all use cases. As accuracy requirements and model complexity increase (e.g., optimizing for different lighting conditions, partial obstructions for facial recognition, broad event detection in mid-term), GPU acceleration will be vital for maintaining real-time performance.

7.2 Data Privacy and Security

Data privacy and security are paramount, especially when handling vehicle and personnel data, and will be ingrained in all deployment phases.

- **Compliance with Digital Personal Data Protection (DPDP) Act, 2023:** A foundational requirement for all use cases, especially for AI-Powered Vehicle Recognition and AI-Driven Facial Recognition, from PoC through to full deployment.

- **On-Premise Processing to Minimize Data Exposure:** CCTV footage will be processed on-premise at AP Civil Supplies warehouses for all use cases in the short term. As the system scales, edge computing (mentioned under Scalability

Approach) will further support this by processing data closer to the source, minimizing data transfer to a central location. Otherwise, we will collaborate with local cloud providers to make sure data stays with restricted area(s).

- **End-to-End Encryption for Data in Transit:** While on-premise processing reduces exposure, any data that needs to be transmitted (e.g., to a centralized portal in the long-term for stock monitoring or vehicle tracking) will be encrypted. This includes data for integration with the VAHAN portal (mid-term for vehicle recognition) and the AP Civil Supplies portal (long-term for gunny bags and vehicle recognition).

- **Access Control Mechanisms for Sensitive Information:** Essential for all use cases, particularly for facial recognition data and vehicle authentication information. This will involve role-based access to ensure that only authorized personnel (e.g., managers for attendance records in mid-term facial recognition) can access specific data. Real-time alerts for unauthorized vehicles (long-term vehicle recognition) or unauthorized access (mid-term facial recognition) will be part of this.

7.3 Scalability Approach

The solution is designed to scale from initial PoCs to a comprehensive, multi-warehouse system.

- **Horizontal Scaling of Processing Nodes by Warehouse:** This approach directly supports the phased deployment:
  - **Short-term:** Deployment in selected AP Civil Supplies warehouses for PoCs of all four use cases.
  - **Mid-term:** Deployment at multiple warehouse sites for enhanced gunny bag counting and key locations for vehicle recognition. Facial recognition will also see expanded deployment for real-time alerts.
  - **Long-term:** Implementation of fully automated systems across all warehouses for all use cases.

- **Edge Computing Architecture to Reduce Central Processing Load:** As the system expands to multiple warehouses (mid-term and long-term), edge computing will be employed (Alternatively, cloud is also possible) to process CCTV footage locally. This is crucial for real-time analysis (gunny bags, vehicle recognition, facial recognition, contextual intelligence) and reduces bandwidth requirements for transmitting raw footage to a central server.

- **Adaptive Resource Allocation Based on Workload:** Kubernetes will enable dynamic scaling of resources based on the processing demands at different sites and times, ensuring efficient use of compute power as more cameras and more complex analyses are added (e.g., predictive analytics for stock monitoring in long-term gunny bag counting, or predictive threat detection in long-term facial recognition).

- **Optimization for Different Hardware Configurations:** Recognizing that existing CCTV hardware will be reused, the software will be optimized to perform efficiently across various camera types and specifications encountered in AP Civil Supplies warehouses. This is relevant from the PoC stage onwards.

7.4 Model Maintenance and Improvement

Continuous improvement is key to long-term success and accuracy.

- **Data Drift Monitoring with Automated Alerts:** As warehouse conditions change (e.g., different types of gunny bags, new vehicle plate designs, varied lighting), models may experience data drift. Automated alerts will trigger reviews. This is important for enhancing accuracy in the mid-term for all use cases (e.g., different lighting conditions for gunny bags and vehicle plates, partial obstructions for faces).

- **Concept Drift Detection Through Performance Metrics:** The system will monitor the accuracy of gunny bag counts, vehicle recognition, facial recognition, and the relevance of contextual insights. Degradation in performance metrics will indicate concept drift (e.g., if a new ”event” type becomes common but isn't recognized). Mid-term goals like enhancing accuracy and robustness necessitate this.

- **Online Learning Capabilities for Continuous Improvement:** Where feasible, models will incorporate online learning to adapt to minor changes in real-time. For more significant adaptations, periodic retraining will be used. This supports the goal of progressively enhancing accuracy and optimizing models throughout the mid-term and long-term phases. For instance, allowing end-users to reconfigure an “event” or “incident” (mid-term for Contextual Intelligence) will feed into this improvement cycle.

- **Periodic Retraining Schedule with Validation Protocols:** Regular retraining will be scheduled to incorporate new data and maintain high accuracy, especially as the system scales and integrates more deeply (e.g., integration with AP Civil Supplies portal and VAHAN). Validation protocols will ensure that retrained models perform better than their predecessors before deployment. This applies to all use cases as they move from moderate accuracy in PoCs to enhanced accuracy in mid-term and robust performance in the long term.

8 Alignment with Evaluation Criteria

Our proposed AI-Based CCTV Analytics Solution has been meticulously designed to excel across all specified evaluation criteria: Technical Feasibility, Scalability, Cost-effectiveness, Impact, and Integration. We are confident in our ability to deliver a high-scoring solution that meets and exceeds APSCSCL's expectations for all four use cases.

8.1 Technical Feasibility

The technical feasibility of our solution is robustly established through a combination of proven technologies, advanced AI models, and a detailed data processing methodology.

- **For Real-time Gunny Bag Counting and Volumetric Analysis:** We employ optimized YOLOv11 models for high-speed, accurate bag detection in motion, coupled with DeepSORT for reliable tracking, minimizing latency. Adaptive contrast enhancement and planned volumetric calibration ensure functionality even in challenging warehouse conditions. The system is designed for real-time streaming inference via Kafka.
- **For AI-Powered Vehicle Recognition:** Our approach leverages YOLOv11 for vehicle and plate localization, followed by a fine-tuned PaddleOCR engine to achieve high OCR accuracy across diverse lighting conditions, angles, and Indian license plate formats. Pre-processing techniques further enhance recognition reliability.
- **For AI-Driven Facial Recognition:** We utilize advanced models like ArcFace, known for high accuracy even with partial occlusions (e.g., masks, to a certain extent), varying angles, and low-light conditions. This ensures reliable staff verification and intruder detection.
- **For Contextual Intelligence for Real-time Analysis and Query:** The use of Vision-Language Models (VLMs) and Retrieval-Augmented Generation (RAG) enables sophisticated analysis of video feeds to capture significant events and anomalies under diverse conditions, supporting natural language queries in both English and Telugu.

Our phased implementation plan, including on-ground testing and model fine-tuning with warehouse-specific data, further de-risks the technical aspects and ensures the development of a practical, high-performing solution. The detailed data processing pipeline, from frame extraction to real-time inference with error correction, underscores the system's capability to deliver accurate and timely results.

8.2 Scalability

Our solution is architected for scalability from the ground up, ensuring it can be deployed effectively across a few pilot warehouses and eventually span the entire APSCSCL network.

- **Across Multiple Warehouses and Entry/Exit Points:** The core streaming infrastructure, built on Apache Kafka, is inherently designed for high-throughput and horizontal scaling. This allows the system to handle increasing data volumes from multiple CCTV cameras across numerous warehouses and all entry/exit points.
- **Modular Architecture:** The system's modular design, with distinct components for data ingestion, processing, and AI model serving (as illustrated in our system architecture and UML diagrams), facilitates independent scaling of different parts of the solution based on demand.
- **Deployment Strategy:** We propose a containerized deployment using Docker and Kubernetes, enabling adaptive resource allocation and simplified management of processing nodes. An edge computing architecture is planned to reduce central processing load and bandwidth requirements as the system expands, ensuring real-time performance across all sites for all use cases, including the application of contextual intelligence across diverse events and incidents.
- **Phased Rollout:** The implementation methodology supports a phased rollout, allowing for gradual scaling and refinement as the system is deployed to more locations.

8.3 Cost-Effectiveness

A key design principle of our solution is to maximize utility while minimizing additional expenditure, primarily by leveraging existing infrastructure.

- **Software Layer Over Existing Infrastructure:** For all four use cases, our solution is predominantly a software layer that intelligently utilizes APSCSCL's existing CCTV camera network. This significantly reduces the need for new hardware investments.
- **Reuse of CCTV Hardware:** The proposal explicitly states the reuse of existing CCTV cameras and infrastructure. Our system is designed to be compatible with standard streaming protocols (e.g., RTSP) and will be optimized for various hardware configurations found in APSCSCL warehouses.
- **Open-Source Technologies:** We leverage powerful open-source technologies such as YOLOv11, PaddleOCR, ArcFace, Apache Kafka, and frameworks for VLMs. This minimizes licensing costs associated with proprietary software.
- **Reduced Inference Costs:** For contextual intelligence, strategies like video chunking and keyframe extraction are employed to optimize processing, thereby reducing the computational cost of inference from continuous video streams. The low per-site cost after the base models are developed and trained is a significant advantage.

By focusing on software enhancements and efficient resource utilization, our solution offers a highly cost-effective pathway to advanced warehouse analytics.

8.4 Impact

The proposed system is poised to deliver substantial positive impacts on APSCSCL's operations, directly addressing the outlined benefits for each use case.

- **Automation and Efficiency (Use Case 1):** Real-time gunny bag counting and volumetric analysis will automate a labor-intensive manual task, significantly reducing errors, improving operational efficiency, enhancing transparency in stock handling, and actively reducing pilferage.
- **Enhanced Security (Use Cases 2 & 3):** AI-Powered Vehicle Recognition will secure inbound/outbound logistics by automating vehicle authentication and preventing unauthorized access. AI-Driven Facial Recognition will increase warehouse security through reliable staff verification, intruder detection, and supporting attendance validation.
- **Improved Transparency and Accountability:** Across all use cases, the system provides auditable logs and real-time data, improving transparency and accountability in warehouse operations.
- **Actionable Insights and Proactive Management (Use Case 4):** Contextual intelligence capabilities enable high recall of event detection across a wide range of target events. The ability to search and query video data in near real-time allows for proactive incident response, optimized resource allocation, and data-driven decision-making.

Overall, the solution transforms raw CCTV footage into actionable intelligence, leading to a more secure, efficient, and transparent supply chain.

8.5 Integration

Seamless integration with APSCSCL's existing and future digital ecosystem is a core component of our technical strategy.

- **APSCSCL Portal Sync:** The solution is designed for seamless synchronization and data exchange with APSCSCL's designated portal for all four use cases. This includes providing real-time counts, vehicle logs, personnel access records, and contextual event data.
- **VAHAN Portal Integration (Use Case 2):** For AI-Powered Vehicle Recognition, we have planned for integration with the VAHAN portal to facilitate comprehensive vehicle verification, enhancing the security of inbound/outbound logistics.
- **API-Driven Architecture:** The system will expose well-defined RESTful APIs for model serving and data access. This allows for flexible and robust integration with various internal dashboards, reporting tools, and other enterprise systems used by APSCSCL.
- **Streaming Data for Real-Time Dashboards:** Real-time inference results, particularly from object tracking and event detection, can be streamed via Kafka to live dashboards, providing immediate operational visibility.

- **Phased Integration:** Our implementation plan includes dedicated tasks in Phase 4 (Deployment Readiness) for finalizing and deploying these integrations, ensuring a smooth transition and operational utility from day one of full deployment.

This comprehensive integration strategy ensures that our AI analytics solution becomes an integral and valuable part of APSCSCL's broader Digital Stack initiative.

9 Conclusion and Expected Benefits

Our comprehensive AI-based CCTV analytics solution offers APSCSCL a transformative approach to warehouse monitoring and management. By implementing this system, APSCSCL can expect:

- Enhanced operational efficiency through automated counting and verification
- Improved security with real-time personnel and vehicle authentication
- Greater transparency across the supply chain
- Reduced manual errors and labor costs
- Valuable operational insights through searchable video analytics
- Scalable infrastructure that grows with organizational needs

The solution leverages existing CCTV infrastructure while adding intelligent capabilities that align with APSCSCL's digital transformation goals. Our team brings expertise in computer vision, AI, and large-scale systems deployment to ensure successful implementation across the warehouse network.

We look forward to demonstrating our solution's capabilities during the hackathon and potentially collaborating with APSCSCL on full-scale deployment as part of your Digital Stack initiative.

For more information about our project and detailed technical documentation, please visit project Website: www.cctvai.org