TED UNIVERSITY

CMPE 491

O.W.L. Border Security

High-Level Design Report

15/05/2026

Team Members:

Emre Kaan Arslan, 13021093920, Computer Engineering

Mehmet Yiğit Açdoyuran, 10130437010, Computer Engineering

Ege Yavuz, 14872032366, Computer Engineering

Supervisor:

Prof. Dr. Tansel Dökeroğlu

Jury Members:

Dr. Mehmet Evren Coşkun

Dr. Eren Ulu

1. Introduction

1.1 Purpose of the System

The OWL Border Security System is designed to improve border surveillance and security operations through the integration of artificial intelligence, computer vision, and behavioral analysis technologies. The primary purpose of the system is not only to detect humans, vehicles, or objects in border regions, but also to analyze behavioral patterns and infer the intent behind detected activities. By combining real-time video processing with multi-sensor data integration, the system aims to provide more accurate, reliable, and context-aware security monitoring.

The OWL system is intended to support preventive security strategies by identifying suspicious activities such as reconnaissance, preparation, smuggling attempts, and unauthorized border crossings before they escalate into critical threats. The system also aims to reduce false alarms, improve situational awareness, and assist security personnel in making faster and more informed decisions. In addition, the system provides secure event logging, real-time geospatial alerting, and role-based dashboard access to ensure effective monitoring, accountability, and long-term operational reliability.

1.2 Design Goals

The OWL Border Security System is developed according to several high-level design goals that guide the overall software and system architecture.

Real-Time Processing

The system must analyze video and sensor data with minimal latency in order to provide immediate alerts and support rapid response operations.

Behavior-Based Intelligence

Unlike traditional surveillance systems, OWL aims to focus on behavioral and intent analysis rather than isolated object detection.

High Reliability

The system should minimize false positives and false negatives through multi-sensor verification and intelligent data analysis.

Scalability

The architecture should support expansion across large border regions without requiring major redesigns.

Modularity

The system must be decomposed into independent subsystems so that components can be updated, maintained, or replaced easily.

Security and Privacy

Sensitive surveillance data must be protected using secure access control, encryption, and role-based authorization mechanisms.

Maintainability

The system should support long-term maintainability through modular design, documentation, and upgradable AI components.

Usability

The user interface should provide clear and understandable alerts, reports, and visualizations for security personnel.

Robustness

The system should continue functioning reliably under difficult environmental conditions such as fog, low visibility, or harsh weather.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
AI	Artificial Intelligence
CV	Computer Vision
GPS	Global Positioning System
UML	Unified Modeling Language
API	Application Programming Interface
Real-Time Processing	Processing and responding to data with minimal delay
Behavioral Analysis	Analysis of movement patterns and interactions
Intent Classification	Determining the purpose behind detected activities
Multi-Sensor Integration	Combining multiple sensor inputs for higher accuracy
Alert Manager	Subsystem responsible for generating and distributing alerts

1.4 Overview

This report presents the high-level design of the OWL Border Security System. The document explains the architectural decisions, subsystem decomposition, software strategies, hardware/software mapping, security mechanisms, and overall control structure of the system.

The report focuses on transforming the analysis model into a scalable and modular software architecture suitable for real-time border surveillance and behavioral threat analysis.

The following sections describe:

- the current architecture limitations,

- proposed software architecture,
- subsystem responsibilities,
- database and data management strategies,
- security mechanisms,
- global control flow,
- and subsystem services of the OWL system.

2. Current Software Architecture

The system does not currently have a specified software architecture. As part of the next steps to guarantee conformity with project goals and requirements, the software components' structure and design have not yet been determined.

3. Proposed Software Architecture

The proposed software architecture of the OWL Border Security System is designed as a modular, scalable, and real-time AI-based surveillance architecture. The system combines computer vision, behavioral analysis, multi-sensor integration, and intelligent alert generation within a unified software structure. The architecture is designed according to separation-of-concerns principles, where each subsystem is responsible for a specific task such as data acquisition, behavioral analysis, intent classification, alert management, or user interaction. This modular decomposition improves maintainability, scalability, and system reliability.

The OWL architecture follows a layered processing pipeline:

1. **Data Acquisition Layer**
   Responsible for collecting video streams and sensor inputs from external surveillance devices.
2. **Processing and Analysis Layer**
   Performs object detection, movement tracking, behavioral analysis, and intent classification using AI models.

3. **Decision and Alert Layer**
   Evaluates detected activities and generates alerts according to threat severity and behavioral context.
4. **Presentation Layer**
   Provides dashboards, maps, reports, and real-time monitoring interfaces for authorized users.
5. **Data Management Layer**
   Stores event logs, alert history, behavioral records, and system outputs securely.

The architecture is designed to support:

- real-time operation,
- distributed deployment,
- subsystem independence,
- future AI model upgrades,
- and integration with additional hardware or sensor technologies.

The system also emphasizes fault tolerance and robustness by enabling continuous operation even when some sensors or modules experience temporary failures.

3.1 Overview

The OWL Border Security System operates through continuous real-time monitoring and analysis workflow.

The overall workflow of the system is summarized as follows:

1. External cameras and sensors continuously provide environmental data to the system.
2. The Sensor Integration subsystem combines incoming video streams and sensor information into a unified data flow.
3. The Object Detection and Tracking modules identify humans, vehicles, animals, and suspicious objects while tracking their movements across frames.

4. The Behavioral Analysis subsystem evaluates movement sequences, group coordination, directional changes, and interaction patterns.
5. The Intent Classification subsystem determines the probable intent of the detected activity, such as:
    - reconnaissance,
    - preparation,
    - unauthorized crossing,
    - or smuggling activity.
6. The Alert Management subsystem evaluates the threat level and generates alerts according to system policies.
7. The alerts generated are:
    - displayed on the user dashboard,
    - mapped with geospatial information,
    - and stored securely in the database.
8. Security Officers interact with the system through the web interface to:
    - monitor alerts,
    - review reports,
    - verify incidents,
    - and manage responses.

The proposed architecture enables the OWL system to move beyond traditional surveillance approaches by incorporating intelligent behavioral understanding and context-aware threat assessment into border security operations.

3.2 Subsystem Decomposition

The OWL Border Security System is divided into multiple independent but interconnected subsystems. Each subsystem is responsible for specific functionality within the overall architecture. This modular decomposition improves scalability, maintainability, reliability, and ease of development. The major subsystems of the OWL system are described below.

3.2.1 Data Acquisition Subsystem

The Data Acquisition Subsystem is responsible for collecting raw data from external surveillance devices.

Responsibilities

- Receiving real-time video streams from cameras
- Receiving input from external sensors
- Managing incoming environmental data
- Forwarding collected data to processing modules

Inputs

- Camera feeds
- Motion sensor data
- Environmental sensor outputs
- Thermal sensor data

Outputs

- Unified raw monitoring data

3.2.2 Sensor Integration Subsystem

The Sensor Integration Subsystem combines data from multiple sources into a synchronized and reliable data stream.

Responsibilities

- Merging video and sensor information
- Synchronizing incoming data
- Cross-validating detections
- Reducing false positives

Inputs

- Video streams
- Sensor outputs

Outputs

- Verified and integrated monitoring data

3.2.3 Object Detection and Tracking Subsystem

This subsystem performs real-time object detection and movement tracking using AI-based computer vision algorithms.

Responsibilities

- Detecting humans, vehicles, animals, and suspicious objects
- Tracking movement trajectories
- Monitoring object interactions
- Providing positional data for analysis

Inputs

- Integrated sensor data

Outputs

- Detected object information
- Movement trajectories
- Tracking metadata

3.2.4 Behavioral Analysis Subsystem

The Behavioral Analysis Subsystem evaluates detected movement patterns and interaction sequences to identify suspicious activities.

Responsibilities

- Analyzing movement behavior
- Detecting suspicious patterns
- Evaluating group coordination
- Identifying abnormal activities

Inputs

- Tracking and positional data

Outputs

- Behavioral pattern classifications

3.2.5 Intent Classification Subsystem

This subsystem determines the probable intent behind detected activities based on behavioral analysis results.

Responsibilities

- Classifying activities into risk categories
- Determining intent type
- Calculating confidence scores
- Supporting threat assessment

Possible Intent Categories

- Reconnaissance
- Preparation
- Unauthorized crossing
- Smuggling activity

Outputs

- Intent classification results
- Threat severity level

3.2.6 Alert Management Subsystem

The Alert Management Subsystem generates and distributes alerts according to detected threat levels.

Responsibilities

- Creating alerts
- Assigning severity levels
- Escalating critical events
- Logging incidents
- Sending notifications to users

Outputs

- Real-time alerts
- Escalation notifications
- Event logs

3.2.7 User Interface Subsystem

The User Interface Subsystem provides interaction between users and the OWL system.

Responsibilities

- Displaying alerts and reports
- Providing real-time monitoring dashboards
- Showing geospatial event maps
- Supporting administrative operations

Users

- Security Officers
- System Administrators

3.2.8 Database and Logging Subsystem

This subsystem manages persistent storage of system data and operational records.

Responsibilities

- Storing alerts and incidents
- Maintaining behavioral analysis records
- Managing user activity logs
- Supporting report generation

Stored Data

- Event logs
- Alert history
- User actions
- Threat analysis results

3.2.9 Authentication and Security Subsystem

The Authentication and Security Subsystem ensures secure system access and protects sensitive information.

Responsibilities

- User authentication
- Role-based access control
- Secure communication
- Data encryption
- Audit logging

Security Features

- Multi-factor authentication
- Restricted access policies

3.3 Hardware/Software Mapping

The OWL Border Security System is designed as a distributed and modular AI-based surveillance architecture that integrates both hardware and software components to support real-time border monitoring, behavioral analysis, and intelligent threat detection.

The system architecture separates data acquisition, AI processing, alert management, data storage, and user interaction into different logical layers to improve scalability, maintainability, and operational reliability.

The hardware/software mapping of the OWL system is represented through a UML Deployment Diagram.

Hardware Components

Surveillance Cameras

Surveillance cameras are deployed in border regions to continuously capture real-time video streams. These cameras act as the primary visual input source of the system.

External Sensors

Additional external sensors such as:

- motion sensors,
- thermal sensors,
- and environmental sensors

are integrated into the system to improve detection reliability and reduce false alarms.

Processing Server

A centralized processing server executes the AI-based analysis operations of the system.

The processing server hosts:

- Object Detection Module
- Behavior Analysis Module
- Intent Classification Module
- Alert Management Module

The server is recommended to include:

- multi-core CPU,
- minimum 8 GB RAM,
- and GPU acceleration support for deep learning operations.

Database Server

A dedicated database server stores:

- event logs,
- alert records,
- behavioral analysis results,
- and user activity logs.

The database layer ensures secure and persistent data management.

Client Devices

Security Officers and System Administrators access the system through client devices such as:

- desktop computers,
- laptops,
- or tablets

using a secure web-based dashboard interface.

Software Components

AI Processing Modules

The software layer includes:

- object detection,
- movement tracking,
- behavioral analysis,
- and intent classification modules.

These modules are implemented using:

- Python,
- TensorFlow,
- PyTorch,
- and OpenCV.

Web Interface

The user interface provides:

- real-time monitoring,
- alert visualization,
- geospatial activity maps,
- and report management capabilities.

Database Management Layer

The database management subsystem handles:

- secure storage,
- retrieval,

- backup,
- and logging operations.

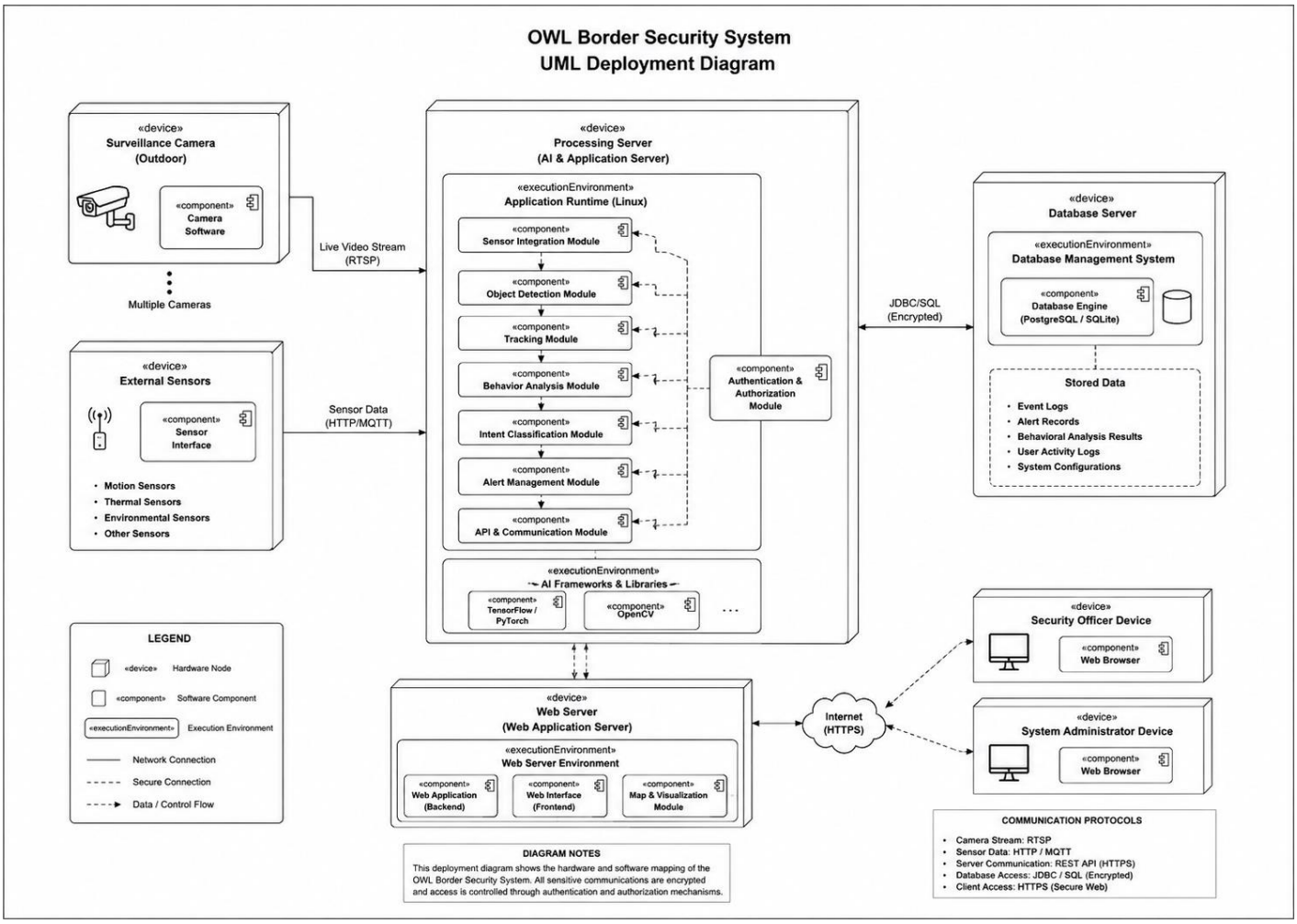
Communication Infrastructure

Subsystem communication is implemented using:

- REST APIs,
- modular service communication,
- and secure data transfer protocols.

This architecture allows independent subsystem operation and easier future expansion.

Deployment Diagram



The UML Deployment Diagram should illustrate:

- hardware nodes,
- deployed software components,
- and communication relationships between system elements.

The diagram should include:

- Surveillance Cameras
- External Sensors
- Processing Server
- Database Server
- Web Interface
- Security Officer Device
- System Administrator Device

The diagram should also show how AI modules, databases, and interfaces are mapped onto physical hardware components.

3.4 Persistent Data Management

The OWL Border Security System requires secure and efficient persistent data management to store system events, alerts, behavioral analysis results, and operational logs.

The data management strategy is designed according to:

- reliability,
- scalability,
- security,
- and maintainability principles.

Stored Data Types

The system stores several categories of data:

Event Logs

Detected activities and system-generated events are recorded with:

- timestamps,
- event type,
- threat level,
- and location information.

Alert Records

Generated alerts are stored for:

- incident tracking,
- historical analysis,
- and audit purposes.

Behavioral Analysis Results

Behavior classification outputs and intent analysis results are archived for future evaluation and system improvement.

User Activity Logs

Authentication attempts, report access, and administrative actions are logged to support security auditing.

Database Strategy

The OWL system initially uses lightweight and modular data storage solutions such as:

- JSON,
- CSV,
- and SQLite.

For future scalability, the architecture supports migration to larger database systems such as:

- PostgreSQL,
- MongoDB,
- or cloud-based storage solutions.

Data Security

All sensitive data is protected using:

- encryption,
- role-based access control,
- and secure authentication mechanisms.

Only authorized users may access restricted system information.

Data Retention Policy

The system prioritizes efficient storage usage by:

- storing only important incidents and alerts permanently,
- while temporary real-time processing data may be discarded after analysis.

This strategy reduces storage overhead while preserving critical evidence and operational records.

Backup and Reliability

Periodic backups are planned to ensure:

- data integrity,
- system recovery,
- and operational continuity in case of failures.

The architecture is designed to support reliable long-term data preservation for security and legal accountability purposes.

3.5 Access Control and Security

The OWL Border Security System handles sensitive surveillance data and security-related operations; therefore, strong access control and security mechanisms are essential components of the system architecture.

The system is designed according to the principles of:

- confidentiality,
- integrity,
- availability,
- and accountability.

Role-Based Access Control

The OWL system implements role-based access control (RBAC) to ensure that users can only access information and functionalities authorized for their roles.

User Roles

Security Officer

Security Officers are authorized to:

- monitor real-time alerts,
- view activity maps,
- review incident reports,
- and analyze suspicious activities.

They are not allowed to modify critical system configurations.

System Administrator

System Administrators are responsible for:

- managing system settings,
- configuring detection parameters,
- managing user accounts,
- and maintaining system operations.

Administrators have higher-level permissions compared to standard users.

Authentication Mechanisms

The system uses secure authentication methods to prevent unauthorized access.

Security Features

- Username and password authentication
- Multi-factor authentication (MFA)
- Session management
- Secure login verification

Authentication logs are stored for auditing and monitoring purposes.

Data Security

All sensitive system data is protected using secure communication and encryption techniques.

Protected Data

- Surveillance event records
- Alert logs
- User credentials
- Behavioral analysis results
- System configuration data

Security Measures

- Encrypted data transmission
- Encrypted database storage
- Secure API communication
- Restricted access permissions

Privacy Protection

Since the system processes surveillance data, privacy-aware mechanisms are included in the architecture.

Privacy Measures

- Limiting unnecessary personal data exposure
- Controlled access to surveillance records
- Optional anonymization techniques
- Secure evidence storage

The system is designed to support ethical and responsible use of AI-based surveillance technologies.

Audit and Logging

The OWL system maintains detailed audit logs for:

- user authentication attempts,
- alert access,
- configuration changes,
- and administrative operations.

These logs support:

- accountability,
- incident investigation,

- and security auditing.

3.6 Global Software Control

The OWL Border Security System follows an event-driven and real-time control architecture designed for continuous surveillance and rapid response operations.

The system operates through coordinated interactions between independent subsystems that communicate through controlled data flow and service-based processing.

Overall Control Structure

The overall control flow of the system follows the sequence below:

1. Data acquisition
2. Sensor integration
3. Object detection and tracking
4. Behavioral analysis
5. Intent classification
6. Threat evaluation
7. Alert generation
8. Data storage and visualization

This workflow enables continuous real-time monitoring and intelligent decision-making.

Event-Driven Architecture

The system primarily operates using an event-driven model.

Event Examples

- Motion detection
- Object appearance
- Suspicious movement pattern
- Unauthorized crossing attempt

- Sensor inconsistency

When an event occurs:

- the related subsystem processes the event,
- generates outputs,
- and forwards results to the next subsystem.

This approach improves:

- responsiveness,
- modularity,
- and scalability.

Real-Time Processing Control

The OWL system is designed for low-latency processing.

The software architecture ensures that:

- video streams are processed continuously,
- AI analysis is performed immediately,
- and alerts are generated with minimal delay.

Priority is given to high-risk activities to support preventive intervention.

Subsystem Coordination

Subsystems communicate through controlled interfaces and modular service interactions.

Coordination Principles

- Independent subsystem operation
- Controlled data exchange
- Minimal subsystem dependency

- Centralized alert management

This architecture simplifies:

- maintenance,
- testing,
- and future expansion.

Alert Control Logic

The Alert Management subsystem acts as the central decision and notification mechanism.

Alert Processing Workflow

- Low-risk events → logged only
- Medium-risk events → monitored and displayed
- High-risk events → immediate alert and escalation

Repeated suspicious behaviors may increase alert severity dynamically.

Failure and Exception Handling

The global control structure includes mechanisms for handling:

- sensor failures,
- communication interruptions,
- inconsistent data,
- and subsystem errors.

If a subsystem temporarily fails:

- the remaining subsystems continue operating,
- and the issue is logged for maintenance review.

This improves overall system robustness and operational continuity.

3.7 Boundary Conditions

The OWL Border Security System is designed to operate in challenging and dynamic border environments. Therefore, the system architecture must handle various boundary conditions and exceptional scenarios to ensure continuous, reliable, and secure operation.

The following boundary conditions are considered in the system design.

Environmental Conditions

The system must continue operating under harsh environmental conditions commonly encountered in border regions.

Examples

- Low visibility (night, fog, smoke)
- Rain, snow, and extreme weather
- Dust and difficult terrain
- Variable lighting conditions

AI models and detection algorithms are optimized to maintain acceptable performance under these conditions.

Sensor and Camera Failures

Hardware failures may occur due to:

- network interruptions,
- damaged sensors,
- camera malfunction,
- or power issues.

In such cases:

- the system logs the failure,

- continues operation using remaining active devices,
- and notifies administrators when necessary.

This prevents complete system shutdown caused by single-point failures.

Network and Communication Interruptions

Temporary communication problems between subsystems may occur during operation.

The architecture handles these situations by:

- retrying failed communications,
- buffering temporary data,
- and maintaining local processing when possible.

Critical alerts are prioritized once communication is restored.

High Data Load Conditions

Large border regions and multiple simultaneous events may create high processing load.

The system addresses this by:

- prioritizing high-risk events,
- processing events asynchronously,
- and distributing workload across modular services.

This helps maintain real-time responsiveness during peak activity.

False Positives and False Negatives

AI-based systems may occasionally produce:

- false alarms,
- or missed detections.

To reduce these risks, the OWL system uses:

- multi-sensor verification,
- behavioral analysis,
- and confidence-based classification mechanisms.

Security personnel may also manually review suspicious events through the interface.

Unauthorized Access Attempts

The system must protect against unauthorized access and malicious usage attempts.

Security mechanisms include:

- role-based access control,
- authentication procedures,
- encrypted communication,
- and audit logging.

Suspicious authentication attempts are logged and may trigger security alerts.

Database or Server Failure

If the database or central processing server becomes temporarily unavailable:

- critical system events are buffered locally,
- temporary logs are maintained,
- and synchronization occurs after recovery.

This minimizes data loss and operational interruption.

Scalability Boundary Conditions

The architecture is designed to support future expansion, including:

- additional cameras,

- new sensor types,
- increased user count,
- and larger monitoring regions.

The modular subsystem design allows scalable deployment without requiring major architectural redesign.

4. Subsystem Services

The OWL Border Security System consists of multiple subsystems that cooperate to provide real-time surveillance, behavioral analysis, and intelligent threat detection services. Each subsystem offers specific services to other parts of the system through controlled interfaces and modular communication mechanisms.

The services provided by each subsystem are summarized below.

4.1 Data Acquisition Subsystem Services

Services

- Collecting real-time video streams from surveillance cameras
- Receiving data from external sensors
- Managing continuous environmental monitoring
- Forwarding raw monitoring data to processing modules

Outputs

- Video stream data
- Sensor data packets

4.2 Sensor Integration Subsystem Services

Services

- Merging sensor and video data

- Synchronizing incoming information
- Validating detections across multiple sources
- Reducing false alarms using cross-verification

Outputs

- Integrated monitoring data
- Verified detection inputs

4.3 Object Detection and Tracking Subsystem Services

Services

- Detecting humans, vehicles, animals, and suspicious objects
- Tracking object movements across frames
- Monitoring object trajectories
- Providing positional information for analysis

Outputs

- Detection results
- Tracking metadata
- Movement trajectories

4.4 Behavioral Analysis Subsystem Services

Services

- Analyzing movement sequences
- Detecting suspicious behavioral patterns
- Evaluating coordination between individuals
- Identifying abnormal activity trends

Outputs

- Behavioral classifications
- Pattern analysis results

4.5 Intent Classification Subsystem Services

Services

- Determining probable activity intent
- Assigning threat categories
- Calculating confidence scores
- Supporting threat assessment processes

Possible Outputs

- Reconnaissance
- Preparation
- Unauthorized crossing
- Smuggling activity

4.6 Alert Management Subsystem Services

Services

- Generating real-time alerts
- Assigning severity levels
- Escalating critical threats
- Logging incidents
- Sending notifications to authorized users

Outputs

- Alert records
- Escalation notifications

- Event logs

4.7 User Interface Subsystem Services

Services

- Displaying real-time alerts
- Providing monitoring dashboards
- Showing geospatial activity maps
- Generating reports
- Supporting administrative operations

Users

- Security Officers
- System Administrators

4.8 Database and Logging Subsystem Services

Services

- Storing event logs
- Archiving alert records
- Managing user activity logs
- Supporting report generation
- Maintaining persistent data storage

Outputs

- Historical records
- Operational reports
- Archived threat data

4.9 Authentication and Security Subsystem Services

Services

- User authentication
- Role-based access control
- Session management
- Encryption and secure communication
- Audit logging

Outputs

- Access permissions
- Security logs
- Authentication records

5. Glossary

- **AI (Artificial Intelligence):** Technologies that enable intelligent data analysis and decision-making.
- **Behavioral Analysis:** Analysis of movement patterns to identify suspicious activities.
- **CV (Computer Vision):** AI field focused on analyzing images and video streams.
- **Event Log:** Stored records of detected activities and system operations.
- **GPS (Global Positioning System):** Technology used for location tracking and mapping.
- **Intent Classification:** Determining the purpose behind detected activities.
- **Multi-Sensor Integration:** Combining multiple sensor inputs to improve detection accuracy.
- **Object Detection:** Detecting humans, vehicles, animals, or suspicious objects in video streams.
- **Real-Time Processing:** Processing and responding to data with minimal delay.
- **RBAC (Role-Based Access Control):** Restricting system access according to user roles.
- **Threat Level:** Severity category assigned to suspicious activities.
- **Tracking:** Monitoring object movement across video frames.

- **UML (Unified Modeling Language):** Standard modeling language used for software system diagrams.
- **User Interface:** Dashboard used by Security Officers and System Administrators.
- **YOLO:** Real-time object detection algorithm used in AI systems.
- **OpenCV:** Open-source computer vision library for image and video processing.

6. References

1. **Bruegge, B., & Dutoit, A. H. (2004). *Object-Oriented Software Engineering: Using UML, Patterns, and Java* (2nd ed.). Prentice Hall.**
2. **Association for Computing Machinery (ACM). (2018). *ACM Code of Ethics and Professional Conduct*. Retrieved from https://www.acm.org/code-of-ethics**
3. **Institute of Electrical and Electronics Engineers (IEEE). *IEEE Code of Ethics*. Retrieved from https://www.ieee.org/about/corporate/governance/p7-8.html**
4. **Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Retrieved from https://arxiv.org/abs/1506.01497**
5. **Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. *You Only Look Once: Unified, Real-Time Object Detection*. Retrieved from https://pjreddie.com/darknet/yolo/**
6. **OpenCV Documentation. *Open Source Computer Vision Library*. Retrieved from https://docs.opencv.org/**
7. **TensorFlow Documentation. *TensorFlow Machine Learning Framework*. Retrieved from https://www.tensorflow.org/**
8. **PyTorch Documentation. *PyTorch Deep Learning Framework*. Retrieved from https://pytorch.org/**
9. **RAND Corporation. *Border Security and Surveillance Technologies*. Retrieved from https://www.rand.org/topics/border-security.html**
10. **Stanford Encyclopedia of Philosophy. *Computer and Information Ethics*. Retrieved from https://plato.stanford.edu/entries/ethics-computer/**