TED UNIVERSITY

CMPE491/SENG491
Computer Engineering Senior Project
Software Engineering Senior Project

PyroGuard Fire Detection System

High Level Design Report
07/01/2024

Team Members:

Ecem Sıla Gök (1118505616)
Ece Selin Adıgüzel (1988345282)
Zeynep Beyza Uçar (1004263330)
Doruk Aydoğan (1354737307)

Table of Contents

1. Introduction:

This document focuses on the system design of PyroGuard Fire Detection System. It serves as a guide to understand the design goals, software architecture, hardware/software mapping, data management, and access control as well as global software control of the system. In addition to those topics, boundary conditions and subsystems are also discussed and explained.

1.1 Purpose of the system:

There are currently hundreds of fire detection models and applications. The most important features and also our main purposes that distinguish our system PyroGuard from others are that it is an end-to-end system that makes a real-time, detailed detection. It also ensures that warnings and notifications are made to the relevant people after the detection. Based on this definition, the main purposes of the system are to achieve efficiency, conceptual integrity, accuracy, completeness, consistency, verifiability and user-friendliness.

1.2 Design goals:

The system basically consists of the fire detection system, location identification system, alarm system and notification system. Therefore our primary goal is to achieve seamless integration among these subsystems. Measure the compatibility and interoperability among these systems must ensure smooth data flow and communication. Secondly, the system should attain a high accuracy rate for fire detection to minimize false alarms and ensure prompt action when a fire is detected. We aim to achieve of a 80-90% accuracy score. In addition to those first phase goals, for second phase, we aim to make systems as fast as possible in terms of making the notification. We'll minimize the time between fire detection and the issuance of notifications to relevant authorities or occupants. As a basic need for any software system, we'll ensure system reliability in diverse environmental conditions and scenarios, minimizing errors or failures by conducting many test cases. Since our end product serves for immediate events, the user interface must be as clear and understandable as possible.

1.3 Definitions, acronyms, and abbreviations:

PyroGuard FDS: PyroGuard Fire Detection System, an extensive system of fire detection and safety.

RBAC: Role-Based Access Control, access control based on user roles. Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals. [1]

APIs: Application Programming Interfaces, a set of defined rules that enable different applications to communicate with each other. [2]

GUI: Graphical User Interface, a graphical user interface that offers controls, data visualization, and historical analytics for the PyroGuard system.

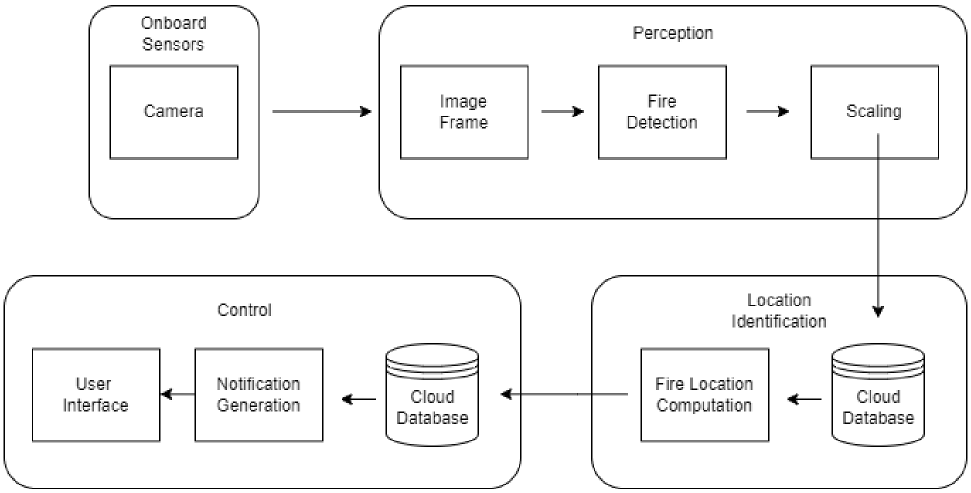
IoT: Internet of Things, a network of interrelated devices that connect and exchange data with other IoT devices and the cloud. [3]

1.4 Overview:

This document is a detailed guide that thoroughly explains how the PyroGuard FDS software and hardware interact and support each other. It dives into details about how data is managed, processed, and utilized within the system, outlining the steps and methods involved. It also covers the rules and procedures for users to access different parts of the system, ensuring a clear understanding of who can do what. Additionally, it describes the overall control mechanisms in place, detailing the ways the system is managed. The document also defines operational limits, explaining what the system can and cannot do. Furthermore, it breaks down the smaller components within the system, offering a comprehensive view of its sub parts and their roles within the PyroGuard FDS.

2. Current software architecture (if any):

Our software architecture involves four primary tasks. The first one is gathering data from sensors; this is done by a high resolution camera followed by interpreting this data to detect fires or anomalies. The third task involves calculating the location of the fire detection event, and finally, the system engages in control by appropriately alerting the user, controller or responsible individual and enabling telemetry for monitoring purposes.



3. Proposed software architecture:

3.1 Overview:

The software architecture of the PyroGuard Fire Detection System is intended to offer a reliable, scalable, and effective framework for fire detection and response. This design is set up to take advantage of cutting-edge techniques and technologies in order to guarantee excellent user accessibility, dependability, and performance.

***Key Components*:**

1. **Fire Detection and Analysis Engine:** The fire detection engine, which makes use of an advanced deep learning model, is the central component of the architecture. This engine accurately detects possible fire events by processing live video inputs from cameras.

2. **Location Identification Module:** The PyroGuard Fire Detection System's Location Identification Module uses image processing techniques to analyze images and determine the location of the fire. Effective fire localization and timely response coordination depend on this module.

3. **Data Processing and Management Layer:** This layer handles the ingestion, storage, and processing of video data. It is optimized for high-throughput data handling, ensuring minimal latency in fire detection.

4. **User Interface:** Users can interact with the system through a simple and easy-to-use interface. This part offers system controls, historical data and analytics access, and real-time data visualization.

5. **Notification Module:** This module is in charge of informing emergency response teams and pertinent personnel about fires and other important information. It guarantees prompt and efficient communication in the event of a fire.

6. **System Integration Framework:** This framework ensures compatibility and smooth operation by making it easier to integrate the system with existing software and hardware infrastructures (such as cameras and alarm systems).

7. **Security and Access Control:** Robust security protocols and access control mechanisms are integrated into the design. These are intended to guarantee data security, prevent unwanted access, and maintain system integrity.

8. **Updating and Maintenance System:** The system is kept up to date with the newest security patches and algorithm advancements thanks to a dedicated subsystem for routine updates and maintenance.

Design Principles:

- Modularity: The architecture's modular nature makes it simple to scale, maintain, and update.
- Resilience: System resilience is emphasized in order to guarantee uninterrupted operation under a variety of circumstances.
- User-Centric Design: The user interface and overall user experience are central to the design, ensuring ease of use and accessibility.

This overview of the suggested software architecture for the PyroGuard Fire Detection System shows a thorough and innovative approach that is made to satisfy the exacting standards of contemporary fire detection and safety regulations. The architecture is a flexible solution for a range of situations since it prioritizes not only technical efficiency but also user-friendliness and adaptability.

3.2 Subsystem decomposition:

1. **Fire Detection Algorithm Subsystem**
   a) Functionality: This subsystem is the brain of PyroGuard. It processes live video feeds from cameras, using advanced deep learning models to detect the presence of fire and smoke.
   b) Components: It includes the deep learning model (possibly based on YOLO), image processing tools, and data annotation software.
   c) Interaction: Receives input from the Camera Subsystem and sends alerts to the Alarm Activation Subsystem and Notification System.
2. **Camera Subsystem**
   a) Functionality: Comprises all the cameras installed in the monitored area. These cameras continuously capture and transmit video footage to the Fire Detection Algorithm Subsystem.
   b) Components: Includes various types of cameras (possibly including thermal cameras) and network components for data transmission.
   c) Interaction: Directly feeds live video to the Fire Detection Algorithm Subsystem.
3. **Alarm Activation Subsystem**
   a) Functionality: Activated when a potential fire is detected. This subsystem is responsible for triggering alarms to warn about the fire hazard.
   b) Components: Includes alarm hardware, manual override controls, and security PIN systems.
   c) Interaction: Receives fire detection alerts from the Fire Detection Algorithm and activates onsite alarms.
4. **Location Identification Subsystem**
   a) Functionality: Identifies the specific location of detected fire within the monitored area, using image analysis and camera perspective data.
   b) Components: Utilizes algorithms for image-based location mapping and data from the Camera Subsystem.
   c) Interaction: Works with the Fire Detection Algorithm to pinpoint fire locations, aiding in effective response.
5. **Fire-Extinguishing Stations Integration**
   a) Functionality: Coordinates with fire-extinguishing resources, providing information about the nearest fire extinguishing stations relative to the detected fire.
   b) Components: Includes a database of fire extinguishing stations and algorithms for distance calculation.
   c) Interaction: Collaborates with the Location Identification Subsystem to provide efficient firefighting solutions.

6. **Notification System**
   a) Functionality: Responsible for notifying the administrative and security personnel immediately after fire detection.
   b) Components: Includes communication modules, contact databases, and alert dissemination software.
   c) Interaction: Receives information about fire detection and location from other subsystems and sends notifications to relevant personnel.
7. **User Interface and Control Subsystem**
   a) Functionality: Offers a user-friendly interface for system monitoring, control, and configuration.
   b) Components: Comprises interface software, authentication systems, and configuration tools.
   c) Interaction: Provides users access to system status, allows configuration of settings, and displays alerts and notifications.
8. **Maintenance and Configuration Subsystem**
   a) Functionality: Ensures the system operates optimally through regular maintenance and updates.
   b) Components: Includes software for system diagnostics, update modules, and maintenance scheduling tools.
   c) Interaction: Interacts with all other subsystems for performance checks and updates.

Each of these subsystems plays a crucial role in the effective operation of PyroGuard Fire Detection System, working in harmony to detect, alert, and respond to fire incidents efficiently. Seamless interaction between these subsystems is what makes PyroGuard a robust and reliable fire detection solution.

3.3 Hardware/software mapping:

1. **Hardware Components**
   The PyroGuard Fire Detection System's hardware consists of the following:
   - High-Definition Cameras: These are positioned to provide the best possible coverage and are able to record the high-definition footage required for precise fire detection.
   - Network Infrastructure: This refers to the actual networking hardware, such as switches, routers, and other networking devices, that is required for data transfer.
   - Alarm Systems: Alarm systems that are physically installed and intended to sound when a fire is detected.
   - Physical security systems: These include access control systems to protect critical components.

2. **Software Components**
   The PyroGuard Fire Detection System's software architecture consists of:
   - Fire Detection Engine: Analyzes video streams for fire detection using cutting-edge deep learning algorithms.
   - Data Processing and Management Layer: Large amounts of video data are ingested, stored, and processed by the data processing and management layer.
   - User Interface: A software application that offers a thorough and user-friendly platform for system interaction and monitoring.
   - Notification and Communication Module: During a fire occurrence, the notification and communication module is in charge of informing the appropriate individuals and emergency response teams.
   - System Integration Framework: The framework for system integration guarantees smooth operation and integration with current software and hardware infrastructures.

- Security and Access Control Layer: Oversees data encryption, cybersecurity, and access permissions.
- Maintenance and Update System: Oversees system updates and maintenance to ensure continuous operation.

3. **Hardware/Software Mapping**
    - Cameras and Fire Detection Engine: Cameras capture real-time video, which the Fire Detection Engine analyzes using deep learning for fire detection.
    - Network Infrastructure and System Integration Framework: Supports data transmission and ensures smooth integration and communication among system components.
    - Alarm Systems and Notification Module: Alarm systems are activated by the software upon fire detection, while the module manages alerts and communication.
    - Physical Security Systems and Security Layer: Physical security measures are complemented by software-based cybersecurity and access control to protect the system and data.

In this mapping, each hardware component of the PyroGuard Fire Detection System is strategically paired with corresponding software components to create a harmonious and effective operation for fire detection and safety management.

3.4 Persistent data management:

Key Components and Functionality:

1. **Data Storage:**
    a) Type of Data Stored: This includes video footage from cameras, detection logs (time, location, type of detection), system operation logs, user activity logs, and configurations.

    b) Storage Solutions: The system utilizes a combination of cloud-based storage and local database solutions. Cloud storage is ideal for large video files and historical data, while local databases quickly access real-time detection logs and system configurations.

2. **Data Access and Retrieval:**
    a) Efficient Access: Data is structured to allow quick retrieval, especially for recent detection logs and live video feeds, crucial for real-time response.

    b) User Query Handling: Users can access historical data through the interface, where they can query past incidents, video archives, and system logs based on parameters like date, location, and type of incident.

3. **Data Security and Privacy:**
    a) Encryption: All data, especially video feeds and user information, is encrypted using industry-standard protocols to ensure privacy and security.

    b) Compliance: The system adheres to relevant privacy laws and regulations, like the Civil Defense Law and KVKK, to protect sensitive data and user privacy.

4. **Data Backup and Redundancy:**
   a) Regular Backups: The system performs regular backups of its data to prevent loss due to system failures or other unforeseen events.

   b) Redundancy Mechanisms: Redundant storage solutions are in place, ensuring that critical data is replicated in multiple locations for extra security.

5. **Data Archiving:**
   a) Long-Term Storage: Older data, which is not required for immediate access, is archived in a cost-effective and secure manner, ensuring that it can be retrieved if needed for analysis or legal purposes.

6. **Data Maintenance and Cleanup:**
   a) Routine Maintenance: Regular data maintenance routines are in place to ensure data integrity and optimal storage performance.
   b) Cleanup Processes: Automated data cleanup processes help in managing storage space efficiently, archiving or deleting data based on predefined rules and regulations.

In summary, the PyroGuard Fire Detection System's approach to persistent data management is comprehensive. It balances the need for immediate access to real-time data for fire detection with the requirements of long-term data storage, security, privacy, and compliance. This robust data management framework is essential for the system's reliability and effectiveness in preventing and responding to fire incidents.

3.5 Access control and security:

To prevent unwanted access and guarantee data security, the PyroGuard Fire Detection System's suggested software architecture incorporates basic yet powerful access control and security features.

Core Elements of Access Control and Security:

1. **User Authentication and Authorization**:
   Implements a secure login system with username and password authentication to ensure that only authorized users can access the system.
   Utilizes role-based access control (RBAC) to define different user roles, such as administrators, operators, and viewers, granting appropriate access levels and permissions based on the user's role.

2. **Data Security and Encryption:**
   Sensitive data, including video streams and user information, is encrypted during transmission to protect against unauthorized access.
   Ensures that data is stored securely and is protected when it's not in use with the right encryption.

3. **Network Security Measures:**
   Incorporates firewall protection to monitor and control incoming and outgoing network traffic, based on predetermined security rules.
   Frequent assessments of network security are carried out in order to find and fix vulnerabilities and lower the possibility of unauthorized access.

4. **Software Security and Updates:**
   All software components, including the fire detection algorithms and the user interface, are regularly updated to fix vulnerabilities and improve security features.
   Adopts best practices in software development to minimize the risk of security flaws.

5. **Audit and Activity Monitoring:**
   To provide efficient audit trails and monitoring, the system maintains logs of all user actions, system modifications, and access histories.
   Regular review of these logs helps in detecting any unusual activities or potential security breaches.

6. **Contingency Planning and Backup:**
   Implements contingency plans for data backup and system recovery to ensure business continuity in the event of a security incident or system failure.
   These plans include procedures for quickly restoring system functionality and data integrity with minimal downtime.

The PyroGuard Fire Detection System's software architecture takes a streamlined approach to access control and security, focusing on protecting the system and its data from unauthorized access and security threats. This results in a safe and effective framework for handling fire detection.

3.6 Global software control:

Key Aspects and Functionality:

1. **Centralized Control System:**
   a. Functionality: Acts as the command center for the entire system, overseeing the operation of different software components, from the fire detection algorithms to user interface interactions.
   b. Implementation: Utilized through a main control software that monitors and manages the activities and statuses of all subsystems.
2. **Software Coordination:**
   a. Subsystem Interaction: Ensures smooth communication and data transfer between subsystems like the fire detection algorithm, alarm system, and notification system.
   b. Data Flow Management: Controls the flow of data across the system, ensuring timely processing and response, particularly crucial for real-time fire detection and alerts.

3. **Control Algorithms:**
   a. Decision-Making Logic: Implements sophisticated algorithms to make system-wide decisions, such as when to trigger alarms or escalate alerts to emergency services.

    b. Adaptive Responses: Capable of adapting responses based on the severity or type of fire detected, optimizing the system's reaction to different scenarios.

4. **User Access and Control:**
    a. Access Levels: Defines various access levels for different users – from system administrators to end-users, ensuring that each user has appropriate control capabilities and data access.
    b. Interface Control: Allows users to interact with the system, adjust settings, view camera feeds, and receive alerts, all coordinated through a central interface.

5. **System-Wide Updates and Maintenance:**
    a. Regular Updates: Centralized control for deploying system-wide software updates, ensuring all components are up-to-date and functioning optimally.
    b. Maintenance Scheduling: Coordinates maintenance activities across different subsystems, minimizing downtime and ensuring ongoing reliability.

6. **Emergency Protocols:**
    a. Fail-Safe Mechanisms: Inbuilt fail-safe and redundancy protocols to handle system failures, ensuring that fire detection and alerting capabilities are maintained even in adverse conditions.
    b. Manual Override Functions: Includes options for manual intervention in case of system malfunctions or false alarms, allowing human control over automated processes when needed.

7. **Performance Monitoring and Analytics:**
    a. System Health Checks: Continuous monitoring of system performance, detecting and addressing potential issues before they impact functionality.
    b. Analytics and Reporting: Generates performance reports and analytics, providing insights into system operations, incident responses, and areas for improvement.

In essence, global software control in the PyroGuard Fire Detection System is about ensuring cohesive operation across all software components, maintaining system integrity, and providing efficient and reliable fire detection and response capabilities. This control framework is pivotal to the system's overall effectiveness, ensuring that every aspect from detection to response works seamlessly under a unified software umbrella.

3.7 Boundary conditions:

The operational, environmental, and technical parameters that the software is intended to operate within to provide optimal performance for the PyroGuard Fire Detection System are specified in the boundary conditions of the proposed software architecture.

1. **Operational Boundaries:**
    a) System Capacity and Scalability: The architecture is designed to handle a specific range of data inputs, including the number of simultaneous video feeds from cameras. It should be scalable to accommodate future expansions or increased data load.
    b) Concurrent User Limit: The software is optimized to support a certain number of users accessing the system simultaneously without performance degradation.

2. **Environmental Boundaries:**
   a) Operational Environment: The program needs to work well in a variety of environments, including those with different lighting and temperature settings that are common in places where the cameras are placed.
   b) Adaptability to Camera Specifications: In order to ensure compatibility and efficient processing of video feeds, the software must be able to adapt to a variety of camera kinds and specifications.

3. **Technical Boundaries:**
   a) Network Limitations: The architecture accounts for potential network bandwidth limitations and latency issues, ensuring efficient data transmission from cameras to servers.
   b) Integration with Existing Systems: The software is designed to integrate with existing hardware and network infrastructure, considering compatibility and interoperability with various systems and protocols.

4. **Performance Boundaries:**
   a) Detection Accuracy: Setting benchmarks for the fire detection algorithm's accuracy, including false positive and false negative rates.
   b) Alarm Response Time: Defining the maximum time allowed from fire detection to alarm activation.

5. **Integration Boundaries:**
   a) Compatibility with Existing Systems: Detailing the system's compatibility with various camera models and types, as well as integration capabilities with existing alarm systems.
   b) Software Interoperability: Specifying the requirements for software integration, including operating systems, third-party software, and API availability for integration with other systems.

6. **Safety and Compliance Boundaries:**
   a) Regulatory Compliance: Ensuring the system meets all relevant fire safety and electronic communication regulations.

7. **Maintenance and Update Boundaries:**
   a) Update Frequency and Downtime: The architecture includes considerations for the frequency of software updates and maintenance, aiming to minimize system downtime.
   b) Backward Compatibility: Ensures that new updates are backward compatible with previous versions, avoiding disruptions in system functionality.

4. Subsystem services:

Fire Detection Subsystem
The fire detection subsystem is tasked with continuously receiving and analyzing live video footage, extracting 15-30 frames per second. The extracted frames are processed utilizing image processing techniques and a deep learning model to see if there are any emergency fire situations. To train this deep learning model, a diverse dataset is used so that the model is able to identify fire in various fire scenarios, and different stages of fire alongside several weather and lighting conditions. The model is trained using annotation with the use of classification to label the data in the previously mentioned dataset.The end product should have a working model that has an accuracy rate of approximately 80-90%, as well as a model that provides an early response as it is trained with data that is representative of different phases of fire situations.

Alarm Activation Subsystem
When the fire detection subsystem detects fire, it immediately triggers this subsystem, so that the fire can be dealt with as soon as possible. Visual and audible alarms are activated, and alarms will not stop unless a previously determined security pin is used to override the alarm, or it is manually deactivated through the system by an authorized user.

Location Identification Subsystem
The images that contain the detected fire are analyzed using image processing techniques such as normalizing and scaling in order to find the placement of the fire. After the placement of the fire is detected with respect to the camera locations that can be accessed through the database, the location of the fire is then compared with the locations of fire extinguishers to see which one is the closest using Euclidean distance.

Notification Subsystem
The contact information for the administrators and necessary personnel are kept in the database so that they can be reached out in the event of an emergency. As soon as the location of the fire is identified, a clearly written and easy-to-understand notification is sent, containing the information of where the fire is and where the nearest fire extinguishing station is. The notifications can be customized by administrative staff to meet their specific needs.

Data Processing and Management Subsystem
The visual data that is received from multiple cameras are processed and stored efficiently for system improvement. Contact information for the notification subsystem, camera and fire extinguishing locations for the location identification subsystem are also stored. Cloud-based storage is used so that the system is scalable and secure.

User Interface and Interaction Subsystem

The system has a dynamic and user-friendly user interface. The users may interact with the system through the login system. Once logged in, the users will be presented with a list of locations they have access to that they may choose from for a more detailed look. Once a location is selected, the user can access live camera feeds and fire detection information of that location. They can also view previous incidents through a log of fire detections that can be filtered and searched through.

Security and Access Control Subsystem

The user access to the system will be regulated through a secure authentication process, and the system will be carefully managed through a role-based access control mechanism to manage different levels of access. All data that goes through the system stores will be encrypted, and relevant safety and security measures will be met through the necessary protocols. Privacy requirements will be enforced for all live camera footage.

Maintenance and Update Subsystem

Compatibility with various cameras and different operating systems will be ensured, alongside the compatibility with specific versions of Python, Keras, TensorFlow, PyTorch, and tools like Roboflow. An up-to-date database will be maintained for contact information, camera locations, and fire extinguishing stations. The system will be scalable to incorporate additional cameras and fire extinguishing stations when necessary. Administrative users will be able to update the system software. Routine maintenance checks will be done to guarantee system reliability.

5. Glossary A:

- Euclidean Distance: A mathematical measure used to calculate the distance between two points in a space.
- Roboflow: Roboflow is a development platform that streamlines the process of building and deploying computer vision models. It provides tools for annotating images, managing datasets, and automating the creation of machine learning pipelines, enabling both novice and expert users to improve and apply their models efficiently.
- YOLO (You Only Look Once) Algorithm: The YOLO algorithm is a state-of-the-art, realtime object detection system that stands out for its speed and accuracy. It 'looks once' at the image by processing it in a single pass, predicting both bounding boxes and class probabilities directly from full images in one evaluation, which makes it exceptionally fast compared to other detection algorithms.
- Current System: Refers to any existing fire detection setup that PyroGuard is intended to replace or augment. This could include traditional smoke detectors, heat sensors, or manual surveillance systems that currently monitor for signs of fire.
- Proposed System: This term denotes the newly designed PyroGuard Fire Detection System, which utilizes advanced machine learning algorithms for the early detection of fire and smoke through camera feeds.
- Functional Requirements: Specific functionalities that the PyroGuard system is expected to perform, such as real-time fire detection, smoke identification, alert generation, and integration with existing camera systems.
- Nonfunctional Requirements: Requirements that define the system's quality attributes, including performance criteria, security standards, usability, reliability, and maintainability of the PyroGuard system.
- Pseudo Requirements: Hypothetical or proposed conditions that the system may encounter, used for planning and testing purposes to ensure the robustness of the PyroGuard system.
- System Models: Conceptual and technical diagrams and models that represent the structure, behavior, and interaction of the system's components within PyroGuard.
- Scenarios: Detailed narratives or use cases that describe specific instances of user interaction with the PyroGuard system, typically for the purpose of detecting fire or responding to alerts.
- Use Case Model: A diagrammatic representation of the system’s functionalities and the interactions between the user and the system, illustrating how the PyroGuard system fulfills its intended use.

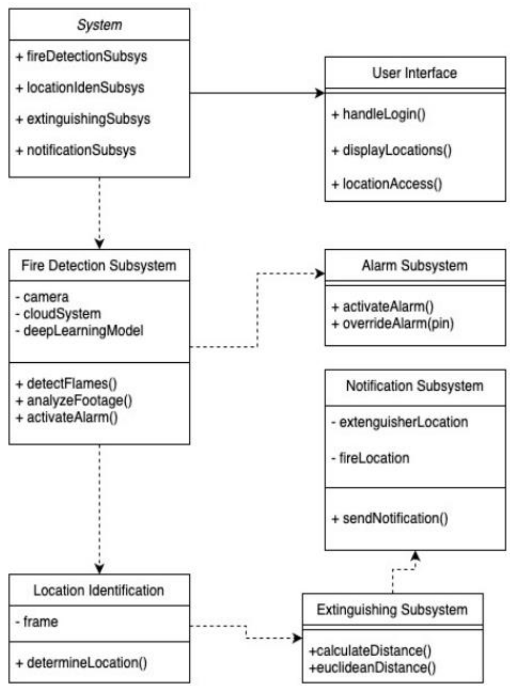
- Object and Class Model: Diagrams and descriptions that outline the objects, classes, and their relationships within the PyroGuard system, providing a blueprint of the system's architecture.

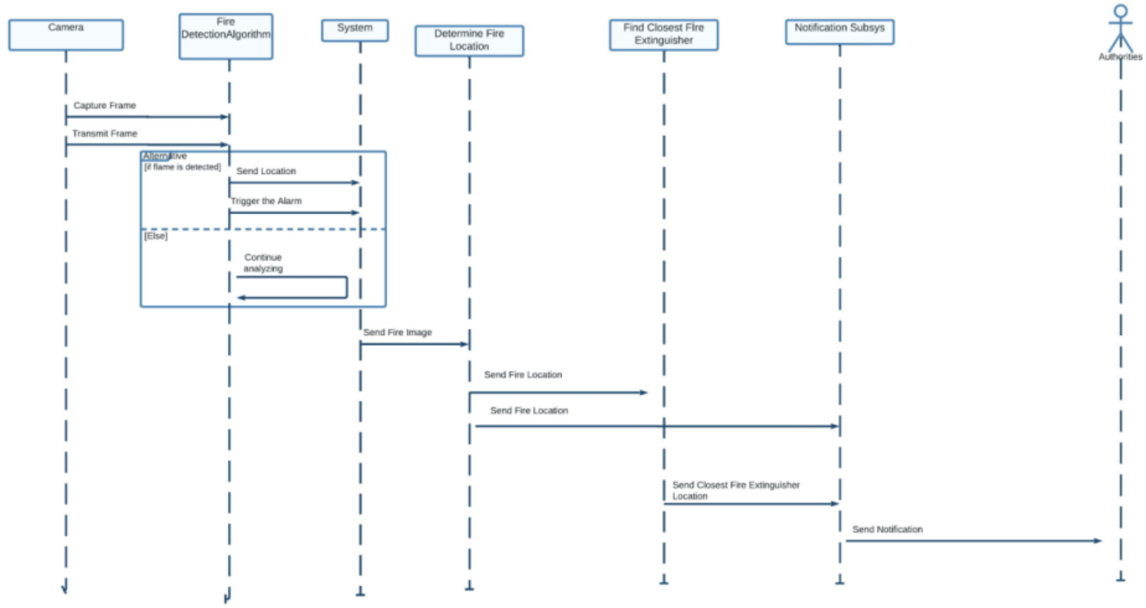
- Dynamic Models: Representations that detail the behavior of the PyroGuard system over time, including state diagrams that illustrate how the system responds to different events or conditions.

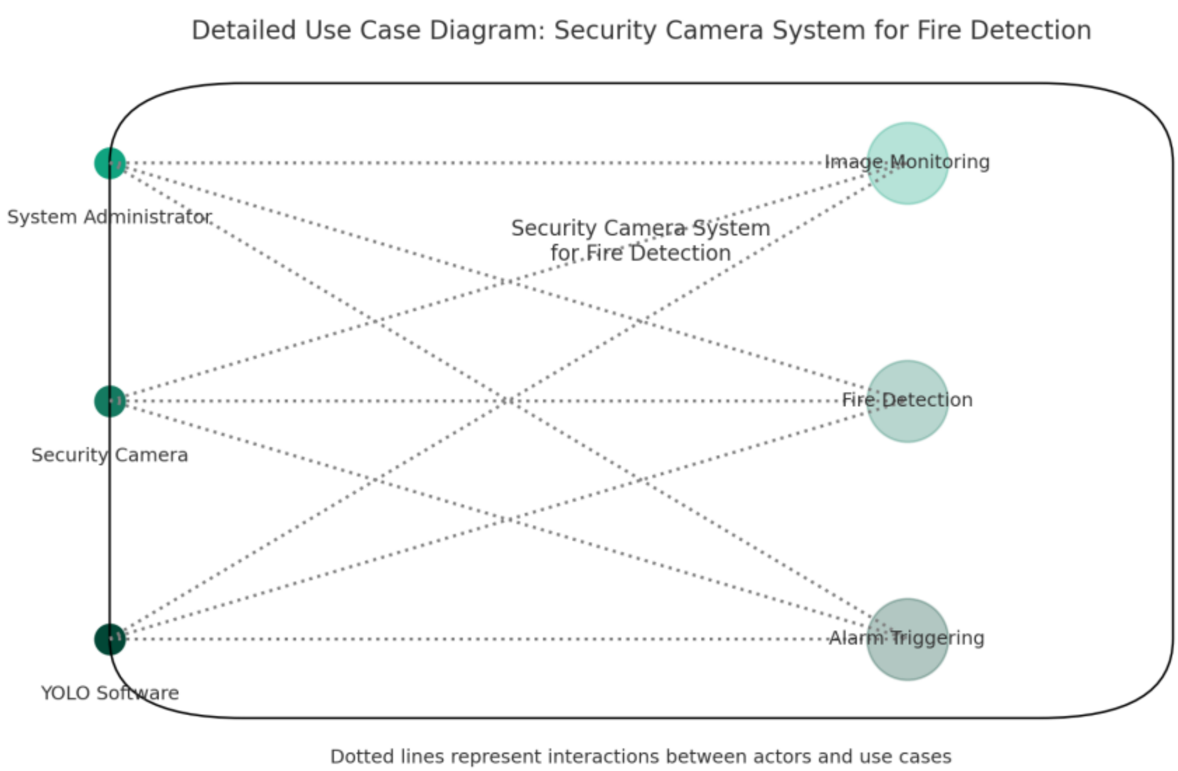
- User Interface: Navigational Paths and Screen Mock-ups: Visual designs and flowcharts that depict the user journey through the PyroGuard system's interface, illustrating the steps a user takes to complete tasks and the layout of the UI elements.

6. Glossary B:

Object and class model




Camera
Fire DetectionAlgorithm
System
Determine Fire Location
Find Closest Fire Extinguisher
Notification Subsys
Authorities
Capture Frame
Transmit Frame
Alternative
[if flame is detected]
Send Location
Trigger the Alarm
[Else]
Continue analyzing
Send Fire Image
Send Fire Location
Send Fire Location
Send Closest Fire Extinguisher Location
Send Notification


Detailed Use Case Diagram: Security Camera System for Fire Detection
System Administrator
Security Camera
YOLO Software
Security Camera System
for Fire Detection
Image Monitoring
Fire Detection
Alarm Triggering
Dotted lines represent interactions between actors and use cases

7. References:

[1] National Institute of Standards and Technology. (n.d.). Role-Based Access Control. Retrieved from https://csrc.nist.gov/glossary/term/role_based_access_control

[2] IBM. (n.d.). What is an API? Learn the Types of APIs and How They Work. Retrieved from https://www.ibm.com/topics/api

[3] TechTarget. (n.d.). Internet of Things (IoT). Retrieved from https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

[4] N. H. Zulkifli, M. F. A. Rasid, M. K. M. Salleh, & K. S. Sim. (2021). Dynamic Object Tracking on Autonomous UAV System for Surveillance Applications. ResearchGate. Retrieved from https://www.researchgate.net/publication/356672238_Dynamic_Object_Tracking_on_Autonomous_UAV_System_for_Surveillance_Applications