TED UNIVERSITY

CMPE491/SENG491
Computer Engineering Senior Project
Software Engineering Senior Project

PyroGuard Fire Detection System

Project Analysis Report
26.11.2023

Team Members:

Ecem Sıla Gök (1118505616)

Ece Selin Adıgüzel (1988345282)

Zeynep Beyza Uçar (1004263330)

Doruk Aydoğan (1354737307)

Table of Contents

1. Introduction

The PyroGuard Fire Detection System represents a transformative approach in the realm of fire safety and emergency response. In the contemporary landscape, where industrial and commercial settings like harbors, docks, and factories are becoming increasingly complex, the urgency for a reliable, efficient, and quick fire detection mechanism is paramount. This report delves into a comprehensive analysis of the PyroGuard System, a solution designed to address these emergent needs effectively.

The inception of the PyroGuard project is rooted in the recognition of a critical gap in existing fire detection methodologies - the need for early detection and rapid response in environments where every second counts. This analysis report serves as a detailed examination and validation of the PyroGuard system. It operates under the paradigm of bridging the developer ("us") and the user/customer, ensuring that the system not only meets the specified requirements but also transcends them to offer a robust, reliable, and user-friendly fire detection solution. The report adheres to the principles of correctness, completeness, consistency, and verifiability in modeling the system.

The report details the system's functional and nonfunctional requirements, pseudo-requirements, and various system models including scenarios, use case models, object and class models, dynamic models, and user interface designs. These elements collectively offer a holistic view of the system’s capabilities, efficiency, and user interaction.

2. Current system (if any)

The current phase of the PyroGuard Fire Detection System project is a testament to the power of artificial intelligence and machine learning in improving public safety. Our team embarked on a mission to develop a highly advanced AI model that can accurately and quickly detect the presence of fire and smoke, distinguishing it from other similar visual patterns that can lead to false alarms.

To this end, we collected a robust dataset of 1000 images specifically designed for training a deep learning model. The first foray into the solution space was marked by the implementation of the YOLO (You Only Look Once) algorithm, specifically the yolov8s model, for its renowned speed and accuracy in object detection tasks. Our preliminary model was trained on a carefully annotated set of 100 images, a process facilitated by Roboflow, an annotation and dataset management tool.

Initial results were promising but there were too many false alarms. Thus it was clear that a larger dataset and more training iterations were needed to improve the model's performance. Hence, a second iteration of the model was trained on approximately 200 images, enhanced with data augmentation techniques such as rotation, shear, blur, and saturation to improve the model's robustness against various lighting conditions and angles. The model's classes are rigorously categorized into 'fire', 'smoke' and 'other'; the "other" played a critical role in reducing false positives caused by objects with a red tint, such as sunsets or car headlights. Thus we started the second training with 510 images with augmentation (rotation, shear, blur and saturation).

The development did not stop at model training; We designed a custom code base to use this model for real-time detection and tagging of fire in images and videos. We wrote code that takes each image, processes it, and detects and tags the fire. This led to the creation of an improved interface to test the effectiveness of the model on visual inputs.

YOLO's predicted image shows a collage of various scenes where the PyroGuard Fire Detection system is used to detect the presence of fire and smoke. Accordingly, our conclusions are:

- High Confidence in Fire Detection: Various images show a high confidence level for fire detection (e.g., 0.9 or 0.7), indicated by labels such as “fire-E3Md.” These images depict large, visible flames often associated with structural fires or wildfires. The model appears to perform well in scenarios where there are clear visual indicators of fire.

- Smoke Detection: There are cases labeled "Smoke 0.3" where the system has a lower confidence level in detecting smoke. This may be due to the dispersed nature of the smoke; This makes the flames more difficult to detect rather than their more obvious appearance.

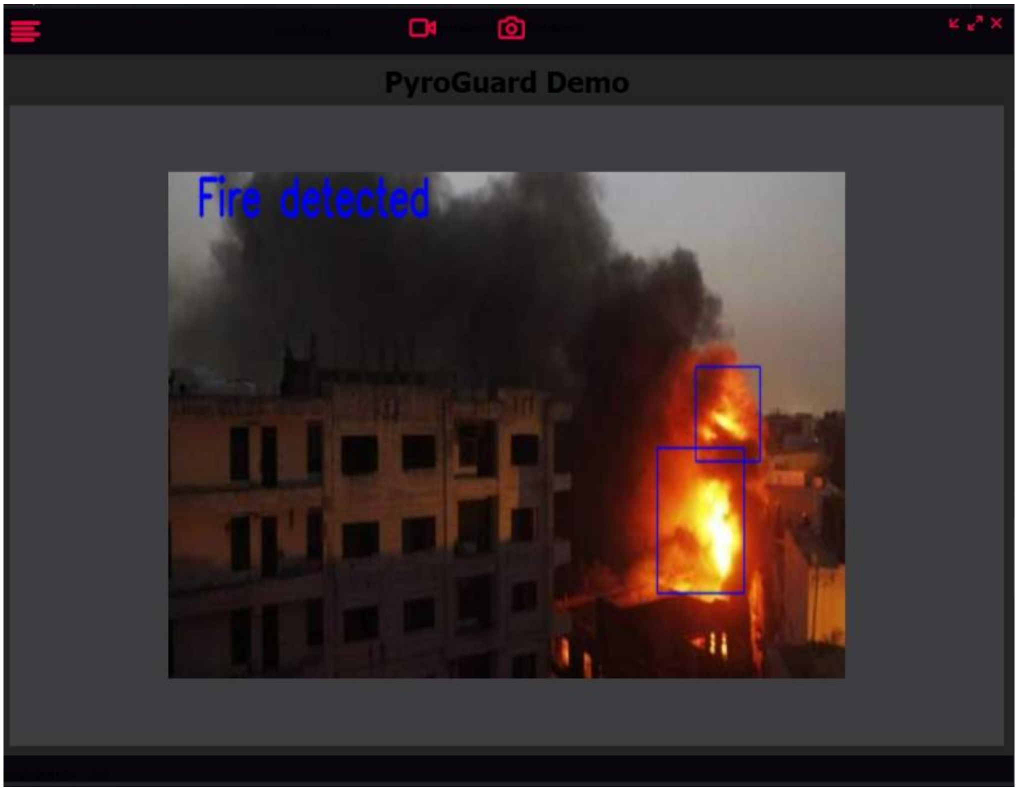
- Challenges with Ambiguous Objects: The “Other 0.5” label indicates that the system recognizes objects that could potentially be mistaken for fire due to their color, shape, or brightness. This highlights the importance of the 'other' class in the model, which helps distinguish between actual fire and objects that may lead to false positives.

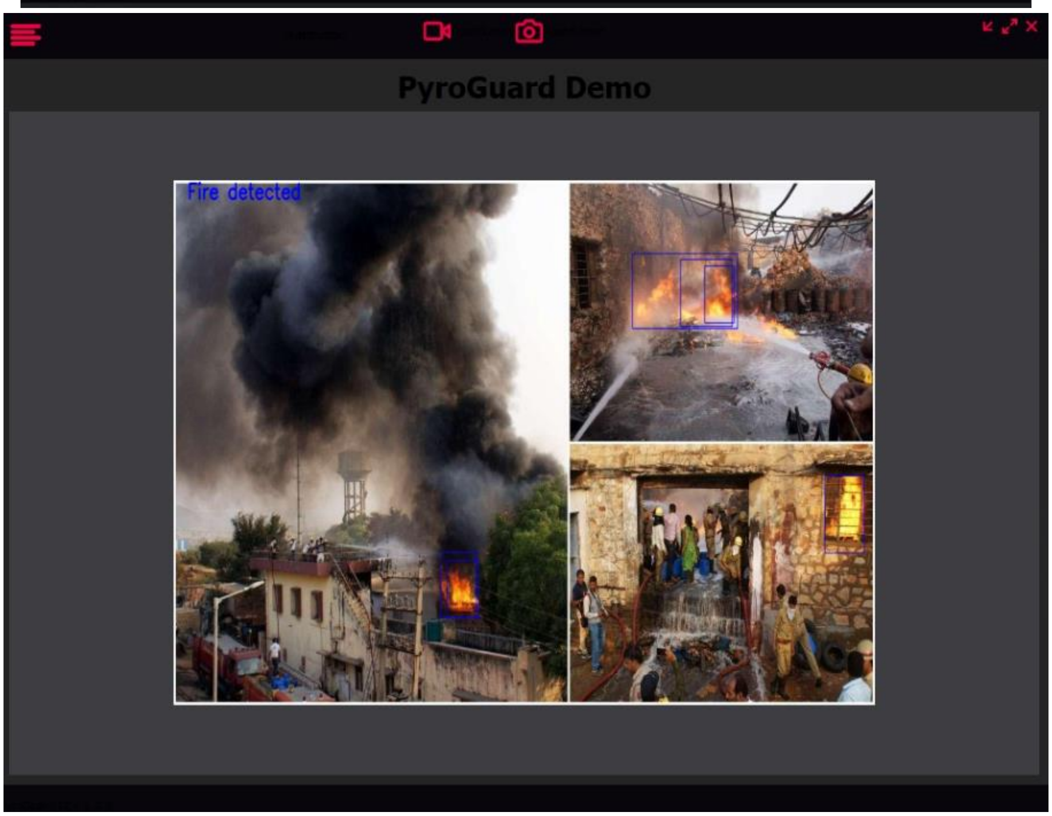
- Potential False Positives and Negatives: Presence of both high- and low-confidence detections in the same image (for example, "fire-E3Md 0.5" and "fire-E3Md 0.5" together, "fire-E3Md 0.3" and " fire-E3Md 0.4) "in other) may indicate potential false positives or negatives. It is important to investigate these further to better understand the model's decision-making process.

- Detection in Different Conditions: Collecting images also demonstrates the system's ability to detect fire and smoke under various conditions (day, night, indoor and outdoor). This highlights the model's versatility and its application in diverse environments.

- Need for More Training: The emergence of the 'other' class and lower levels of confidence in smoke detection suggests the need for further model training and refinement, particularly to improve smoke detection and reduce false positives.

The observations point to a promising start for our PyroGuard Fire Detection system, with the ability to detect fire with high confidence in many situations. However, it also highlights areas for improvement, such as enhancing smoke detection accuracy and reducing false positives, which can be addressed with more training, testing other AI algorithms, and a larger data set, as planned in the next phase of the project.





Our observations during testing revealed a major problem: the model's occasional inability to distinguish between fire and red car headlights. This insight spurred a hypothesis that by analyzing consecutive frames, the temporal consistency of objects like car headlights — as opposed to the dynamic, fluctuating nature of fire and smoke — could be used to differentiate between them.

The project is now preparing to enter a new phase where we will harness a larger dataset to retrain the YOLO model and improve its accuracy and reliability. This initiative is not just about technical excellence, but also about ensuring the safety and security of environments susceptible to fire-related disasters.

3. Proposed system

3.1 Overview

The PyroGuard Fire Detection System is designed to be an end-to-end system that will be able to detect flames as soon as possible, even just moments after the first instant of ignition. The system prioritizes being able to use already existing camera systems as a less expensive option to take an extra security measure and to increase affordability.

The system is intended to be deployed for especially critical environments such as harbors, docks, and factory settings, as they are mostly highly flammable environments and even a few seconds are extremely critical in how much damage is done in case of a fire. With this said, this system can be easily used by any individuals or entities that wish to take an extra security measure.

The system will have a user-friendly interface that will be used to ensure a secure login system which will regulate access. After a user is logged in, a list of the locations they have access to will be presented to them which will be particularly useful in cases where the user has access to multiple locations.

Apart from these, the system will consist of the following integrated subsystems:

Fire Detection Algorithm:

The camera footage will be streamed to a cloud-based system so that it can be continuously analyzed for the possible presence of flames. A deep learning model will be used to examine the frames extracted from the video footage. It works by detecting any flame-like objects and returns a percentage that shows how likely they are to be flames. If the probability returned by the algorithm is over a certain threshold, the alarm system will be activated.

Alarm Activation:

When the algorithm detects the presence of a fire, an installed alarm system will be triggered. The alarm system intends to alert any personnel or individuals that may be close to the fire site. Once activated, the alarm system may be overridden or manually turned off by entering a predetermined security pin to the alarm panel. Overriding the alarm can only be done in person and is not permitted to be accessed remotely for security purposes.

Location Identification:

Using camera perspectives that are able to detect flames, the approximate location and bounds of the fire will be detected.

Fire-Extinguishing Stations:

As the location of the fire is determined, the system will compare the determined location to compare with the location of predetermined fire extinguishing stations. This will be done by utilizing a simple calculation of Euclidean distance. Since this will not be a complicated computation process, it is expected that it will lead to an efficient integration.

Notification System:

The system will notify the administrative and security personnel upon encountering fire. The notification will include information of the location of the fire and the nearest fire extinguishers that may be used, in order to ensure a more swift and informed response.

3.2 Functional Requirements

User Interface:

- The system shall have a login system the users can access the system through.
- The user interface shall present the users a list of all the locations that user has access to.
- When the user selects a location from the offered list, the live feeds from cameras in that location can be seen alongside any previous or active fire situations.

Fire Detection Algorithm:

- The system shall be connected to multiple cameras that provide live footage.
- The live footage shall continually be analyzed to ensure that there isn't an emergency fire situation.
- 15-30 frames per second should be extracted from the live footage and analyzed.
- The system shall use image processing techniques and a deep learning model to detect the presence of flames from the live footage.
- The model used to detect the flames shall be trained with a diverse dataset that includes images from various scenarios and environments, at different stages of the fire.
- The model shall use annotation to label the data and shall be trained with the use of classification.
- The developed model should be able to detect flames right after it ignites, before turning into a full-blown fire.
- The results the algorithm produces should hold a minimum accuracy rate of approximately 80-90%.

Alarm Activation:

- The system shall trigger the alarm system immediately after a flame is detected.
- The alarm shall continue ringing until it is manually deactivated or overridden using a predetermined security pin.

Location Identification:

- The approximate location of the flames shall be detected from the images extracted.

Notification System:

- The system shall hold the contact information of the people that will be notified when fire is detected.
- The necessary people shall be notified once fire is detected, the notification shall include the information about the location of the fire and the nearest fire extinguisher's placement.
- The notifications should be sent within 15-20 seconds once the fire is detected.

Fire-Extinguishing Stations:

- The system shall have a database of predetermined fire extinguishing stations.
- The system shall compare the detected location with the nearest fire extinguishing stations and detect the closest one using Euclidean distance.

3.3 Nonfunctional Requirements

Performance:

- The system should be able to detect the presence of flames with a minimal delay after it is visible.
- The system should be able to handle up to 9 video feeds simultaneously.
- The system should be able to provide accurate results with different weather conditions.
- The system should be able to provide accurate results with little regard to lighting conditions.
- The notification system shall have a fast response time to ensure timely intervention.

Reliability:

- The algorithm shall work with a minimum number of false positives and false negatives, leading to an overall performance accuracy of approximately 80-90%.
- The alarm system shall be activated when fire is detected.

Scalability:

- The alarm system shall be scalable to incorporate more cameras and fire extinguishing stations.

Usability:

- The user interface shall be dynamic and user-friendly.
- The notifications shall be worded in an easy-to-understand and brief manner.
- The notification alerts should be customizable to meet customer needs.

Security and Safety:

- As data safety is a top concern, all data that is transmitted shall be encrypted using industry-standard protocols.
- Relevant preventative measures shall be taken to ensure cloud safety, in cases that the system utilizes cloud operations for the video footage.
- Relevant safety and regulatory standards shall be met by the system such as Civil Defense Law (No.7126), KVKK (No. 6698), TSE etc.
- Privacy requirements shall be followed in the usage of live camera feedback.
- The system should have clear and effective emergency protocols in place in case of a false alarm or system failure.

Compatibility:

- The system should be compatible with a variety of cameras.
- The system should be compatible with a variety of operating systems.

3.4 Pseudo requirements

- It may be suggested to install an additional thermal camera for extra accuracy where it is deemed necessary.
- It is expected that the model will be developed using Python, especially with the Keras and Tensorflow libraries and the PyTorch framework. It is important that the versions used are compatible, so that no issues occur later on.
- It is suggested that as long as the system utilizes a YOLO algorithm, Roboflow to be used for the datasets as it is an important tool for the diversity of data, allows more control over the used datasets, and can be easily used for data annotation.

3.5 System models

3.5.1 Scenarios

Image Monitoring Scenario

- Description: The security camera continuously captures real-time video footage and transmits it to the Deep Learning algorithm for analysis.
- Actors Involved: Security Camera, Deep Learning Algorithm.
- Process Flow:
- The security camera captures live video footage.
- The captured footage is transmitted to the Deep Learning algorithm.
- The Deep Learning algorithm performs analysis for fire detection.

Fire Detection Scenario

- Description: The Deep Learning algorithm processes the incoming video feed, employing its fire detection algorithm to identify potential fire hazards.
- Actors Involved: Deep Learning Algorithm.
- Process Flow:
- The Deep Learning algorithm processes the video stream.
- It applies its fire detection algorithm to identify potential fire hazards.
- If fire is detected, the algorithm triggers an alert.

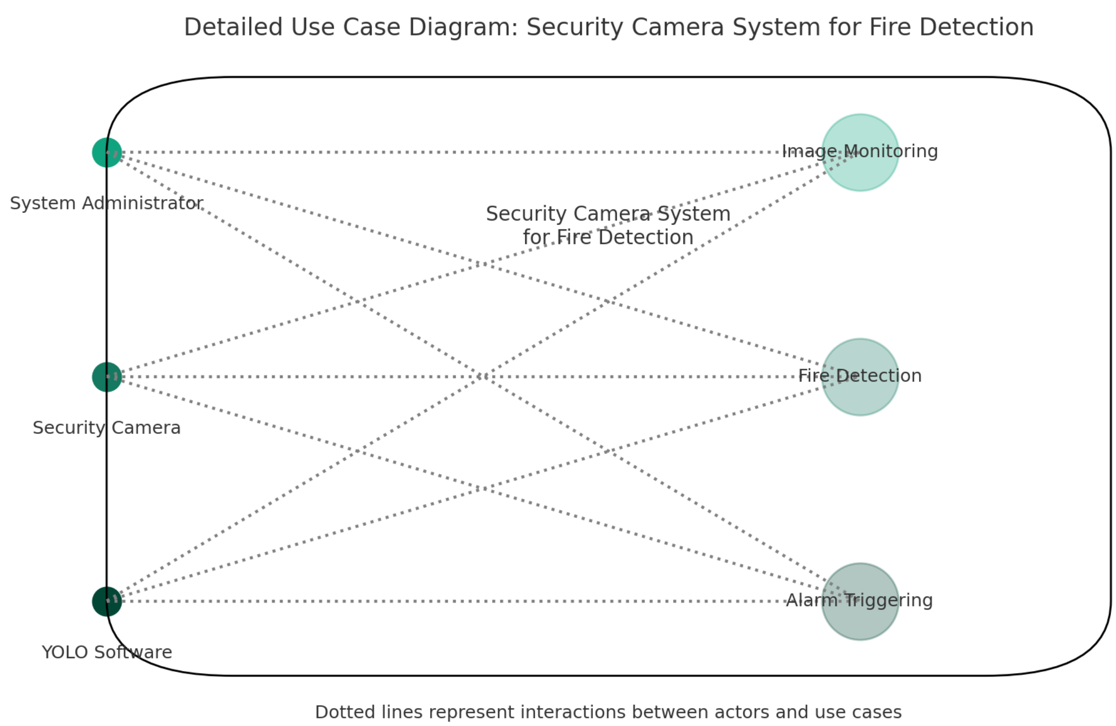
Alarm Triggering Scenario

- Description: Upon fire detection, the system activates an alarm and notifies the system administrator.
- Actors Involved: Deep Learning Algorithm, System Administrator.
- Process Flow:
- Upon detecting fire, the Deep Learning algorithm sends a signal to trigger the alarm.
- An alert notification is sent to the system administrator.
- The system administrator receives the alert and takes appropriate action.

System Configuration and Maintenance Scenario

- Description: The system administrator periodically configures and maintains the system for optimal performance.
- Actors Involved: System Administrator.
- Process Flow:
- The system administrator accesses the system settings.
- Configuration parameters, such as sensitivity and alert thresholds, are adjusted, and software updates are applied as needed.
- Regular maintenance checks are performed to ensure the reliability of the system.

3.5.2 Use case model



Use Case Name	Monitoring the Security Cameras
Description	The security cameras continuously monitors the environment.
Primary Actors	Security Camera
Secondary Actors	-
Priority	High
Preconditions	System Active
Main Flow	1. Camera captures an image. 2. Image is sent to the system to be analyzed.

Use Case Name	Detecting Active Fire
Description	Image processing techniques and a trained deep learning model uses the received security footage and attempts to detect the presence of fire.
Primary Actors	Trained Model
Secondary Actors	-
Priority	High
Preconditions	Receiving the security footage from the installed security cameras.
Main Flow	1. The deep learning model analyzes the image. 2. It returns probabilities for any fire-like objects it finds. 3. If the probability for any object is above a pre-determined threshold, the system consider it as fire detected. 4. The alarm system gets triggered. 5. The system starts the subsystem to determine the placement of the fire.

Use Case Name	Alarm Triggering
Description	When fire is detected, the system automatically triggers an alarm.
Primary Actors	Alarm system
Secondary Actors	-
Priority	High
Preconditions	Fire is detected.
Main Flow	1. Alarm is triggered by the fire detection subsystem. 2. An audible alarm is activated. 3. The alarm rings until it is manually turned off or overridden using a security pin.

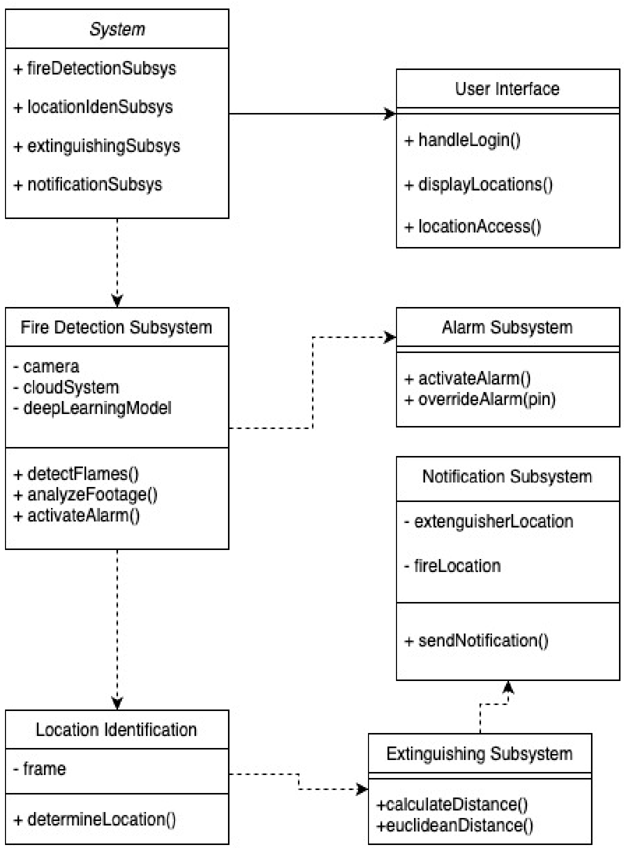
Use Case Name	Determining the Location of the Fire
Description	When fire is detected, the system uses the images that has flames in them to detect the placement of the fire.
Primary Actors	Location Detection Algorithm
Secondary Actors	-
Priority	Medium
Preconditions	Fire is detected.
Main Flow	1. Once fire is detected, the algorithm uses the images that have fire detected in them to analyze the location of the fire. 2. The system activates the subsystem to find the closest predetermined fire extinguishing station.

Use Case Name	Detecting the Closest Fire Extinguishing Station
Description	After the location of the fire is known, the nearest fire extinguishing station is calculated to decrease the time of intervention.
Primary Actors	Distance Comparison Algorithm
Secondary Actors	-
Priority	Medium
Preconditions	Fire is detected, the location of the fire is found.
Main Flow	1. The relative distance of each fire extinguisher is calculated to the active fire. 2. The one that is the closest is returned.

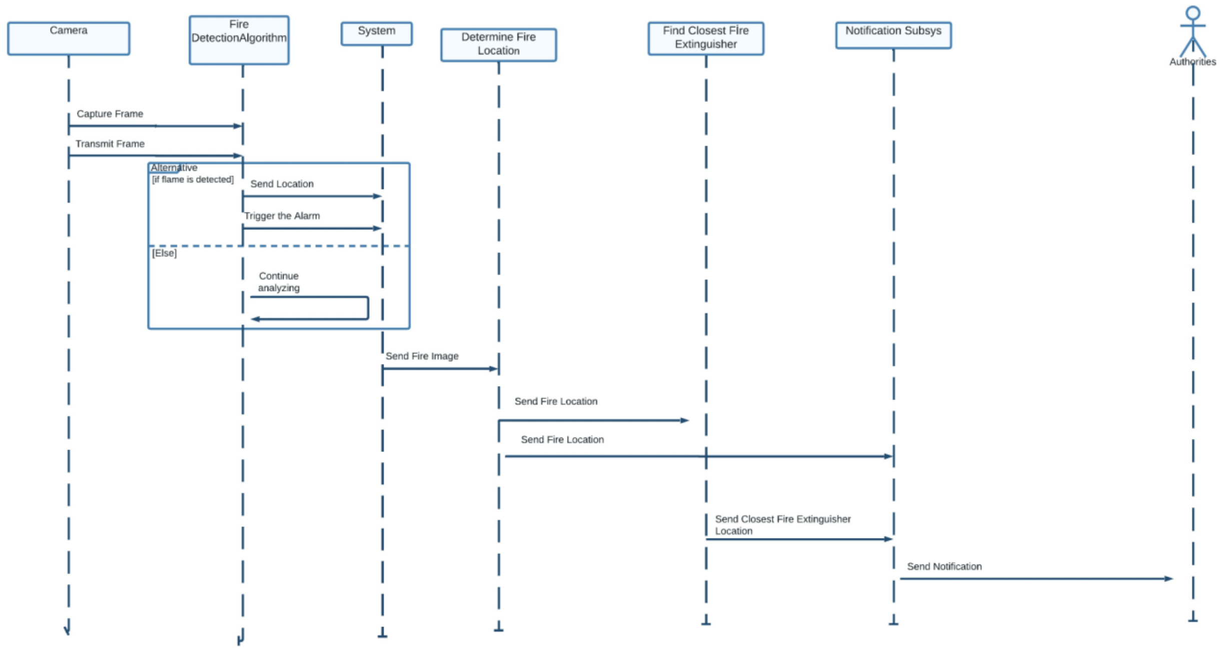
Use Case Name	Sending Notification to Administrative and Security Personnel
Description	The system notifies the administrator and the necessary security personnel to let them know there is an active fire, with the extended information of the placement of the fire and the closest fire extinguishing station.
Primary Actors	System Administrators, Security Personnel
Secondary Actors	Location Detection Algorithm, Distance Comparison Algorithm
Priority	High
Preconditions	Fire is detected, the location of the fire is found, the nearest fire extinguishing station is detected
Main Flow	1. The notification subsystem receives the information about the placement of the fire. 2. The notification subsystem receives the information about the fire extinguishing station that is closest to the fire. 3. The notification system sends a notification informing the necessary people that includes the abovementioned information.

Use Case Name	Configuration and Maintenance
Description	The system administrator configures the deep learning model and performs system maintenance.
Primary Actors	System Administrator
Secondary Actors	-
Priority	Low
Preconditions	-
Main Flow	1. Administrator makes configuration settings. 2. Performs necessary maintenance.

3.5.3 Object and class model



3.5.4 Dynamic models



3.5.5 User interface - navigational paths and screen mock-ups

This section would illustrate how users interact with the system, the layout of the UI elements, and the journey a user takes from one part of the application to another, often referred to as the user flow or navigational path.

User Interface

Design Philosophy: The UI of PyroGuard will be designed to be intuitive and user-friendly, allowing users from various technical backgrounds to operate the system efficiently. The design should incorporate a clean, straightforward layout that emphasizes quick access to the system's core features—monitoring and detection alerts.

Color Scheme and Icons: Given that the application deals with fire detection, the color scheme would likely include red and orange tones, symbolizing urgency and alertness, while ensuring that the interface does not cause visual strain during extended use. Icons and buttons will be designed to be self-explanatory with standard symbols for settings, alerts, and status indicators.

Navigational Paths

Login and Authentication: The first screen after application launch would be a secure login page to authenticate users. After successful login, users would be directed to the main dashboard.

Main Dashboard: The dashboard would provide a quick overview of the system's status, including the number of cameras active, current detections, and system health. Users could navigate to different areas of the application from here.

Location Access: A side panel or dropdown menu would list all the locations the user has access to. Selecting a location would bring up live feeds from cameras in that area and any active fire detections.

Alerts and History: Another section would show a log of all detections, categorized by date, time, and location. Users could filter results or search for specific incidents.

Settings: In the settings menu, users could configure system parameters, manage camera feeds, set up notification preferences, and adjust detection sensitivity or other AI model parameters.

Screen Mock-ups

Live Feed Screen: This would show live camera feeds with detection overlays. If the AI detects a fire, the corresponding section of the video feed would be highlighted with a bounding box, and an alert would be displayed.

Incident Review Screen: Here, users could review past incidents. Each incident would have a thumbnail image from the detection, timestamp, and options to view video footage or export the incident report.

Analytics Screen: A section for analytics would display graphical data on detection frequency, response times, and system accuracy, helping users to monitor and evaluate the system's performance.

The aim is to ensure that the user interface facilitates easy navigation, quick access to critical information, and a streamlined process for monitoring and responding to fire incidents.

4. Glossary

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, real-time 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.

5. References

https://sparxsystems.com/resources/tutorials/uml/dynamic-model.html
https://app.diagrams.net
https://lucid.app/documents#/documents?folder_id=recent
https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/amp/
https://app.roboflow.com/
https://docs.ultralytics.com/