

AI Model for Detecting Abnormal Behaviour

System Requirements Specification

GROUP 1-C


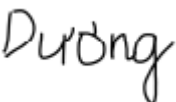



Name	Position	Email	Phone No
Tran Quoc Dung	Software Development	dungtqsw00420@fpt.edu.vn	0943910306
Pham Hoang Duong	Data Research	duongphsw00843@fpt.edu.vn	0902162467
Nguyen Thai Son	Data Research	sonntsw00509@fpt.edu.vn	0902950403
Duong Quoc Trung	AI Researcher	trungdqsw00902@fpt.edu.vn	0962579876
Do Tuan Dat	Software Development	datdtswh00592@fpt.edu.vn	0865411803

COS40005, Computing Technology Project A, May-2024, 11/06/2024

DOCUMENT CHANGE CONTROL

Version	Date	Authors	Summary of Changes
1.0	11/06	All	Create an initial draft document.
1.10	27/06	Dung, Duong, Son	Finalizing the Software Requirements Specification.

DOCUMENT SIGN OFF

Name	Position	Signature	Date
Tran Quoc Dung	Software Developer, Writer		21/06/2024
Pham Hoang Duong	Data Research, Software Dev		21/06/2024
Nguyen Thai Son	Data Researcher		21/06/2024
Duong Quoc Trung	AI Researcher		21/06/2024
Do Tuan Dat	Software Developer		21/06/2024

CLIENT SIGN OFF


Name	Position	Signature	Date
Le Van Khang	AI Engineer		28/06/2024
Organisation			
Quy Nhon AI Creative Alley			

Table Of Contents

1. Introduction.....	4
1.1. Purpose	4
1.2. Scope	4
1.3. Definitions, Acronyms and Abbreviations	4
2. Overall Description.....	5
2.1. Product Features	6
2.2. System Requirements	6
2.3. Acceptance Criteria	6
2.4. Documentation	6
3. Functional Requirements	7
4. Non-Functional (Quality) Requirements.....	8
5. Interface Requirements	9
5.1. System in Context.....	9
5.2. User Interfaces	9
5.3. Hardware Interfaces	10
5.4. Software Interfaces	10
5.5. Communication Interfaces	10
6. References (if any).....	11

1. Introduction

In this project, the Abnormal Behaviour Detection AI Model aims to investigate, analyse the video transferred from CCTV cameras with a view to identifying anomalous activities. By means of machine learning techniques, the system will enhance security and safety in various contexts, such as public spaces, workplaces, and residential areas.

1.1. Purpose

The SRS (System Requirements Specification) defines the requirements for developing the Abnormal Behaviour Detection AI Model. It illustrates the functionalities, constraints, along with the performance expectations of the system.

1.2. Scope

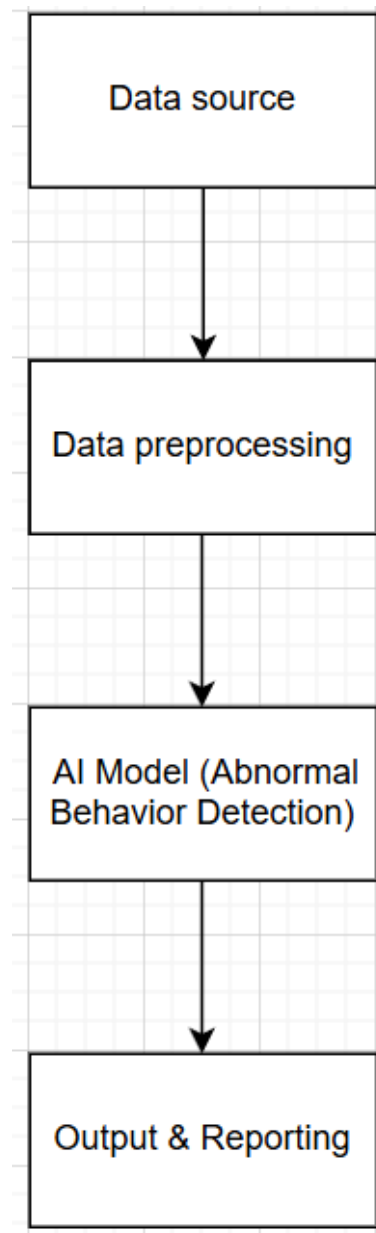
- Access and process the recorded videos from CCTV cameras
- Based on researched activities in survey, utilize the AI algorithms to analyse and detect the abnormal behaviours (e.g. fighting, fainted, etc.)
- Alerts, notify the users if any is detected
- Provide an intuitive & friendly UI for configuration and monitoring

1.3. Definitions, Acronyms and Abbreviations

- AI: Artificial Intelligence
- CCTV: Closed-Circuit Television
- SRS: System Requirements Specifications
- UI: User Interface
- ML: Machine Learning
- FPS: Frames Per Second
- API: Application Programming Interface
- IoT: Internet of Things

2. Overall Description

The project involves the development of a new AI model designed to detect abnormal behaviour in datasets. This system is being developed from scratch and is not an upgrade or replacement of an existing product. The AI model serves as a prototype with the potential for future integration into larger systems for behaviour analysis and anomaly detection.



2.1. Product Features

- Use machine learning algorithms to identify abnormal behavioural patterns in data sets
- Includes data cleaning and normalization procedures to prepare raw data for analysis
- Provides a user-friendly interface for viewing results
- Allows users to adjust model parameters to fine-tune detection accuracy
- Generate comprehensive reports of detected anomalies, including detailed analysis and visualization

2.2. System Requirements

The system requirements define the necessary conditions for deploying the AI model. These requirements ensure that the software functions correctly and efficiently.

- Software Requirements:
 - Operating System: Window 10 - 64 bits
 - Programming Language: Python 3.9.x
 - Algorithms: CNN, RNN, LSTM, Deep Learning, Computer Vision, Visual Studio Code
 - Frameworks: TensorFlow / PyTorch
 - Libraries: Keras, Numpy, Tensorflow, Object Detection
- Hardware Requirements:
 - Operating System: Window 10, 64 bits
 - RAM: 16GB
 - Processor: AMD Ryzen-5 2400G
 - Storage: Minimum of 50GB (project dataset might require 40GB of the storage)
- Resources: About more than 500 video clips (being analysed & detected)

2.3. Acceptance Criteria

- The AI model must achieve at least 80% detection accuracy based on a predefined test data set
- The model must satisfy the chosen metric
- The system must process and analyse data within an acceptable time frame (e.g. less than 2 minutes for a typical data set)
- The interface should be intuitive and easy to use, with clear instructions and a responsive design
- The system must operate stably, without errors or serious problems during standard operation

2.4. Documentation

- Detailed instructions for installing, configuring and using the software
- Step-by-step instructions for common tasks and features
- In-depth documentation of the system's architecture, components, and codebase
- Documentation of internal and external audits, including findings and corrective actions taken
- Reports from team members reflecting on work completed, challenges faced, and knowledge gained

3. Functional Requirements

Functional requirements with use case analysis for the system:

Use case 1: Input training data

Goal: The user inputs a set of videos to be used as training data for the model.

Actors: User, System

Pre-condition: The training video dataset is organized on the user's machine.

Post-condition: The system successfully retrieves the training video dataset.

Main scenario:

1. User interacts with the system's interface and requests the system to start the operation of retrieving the training video dataset.
2. The system prompts the user to provide the location to where the training video dataset is stored.
3. The system retrieves the training video dataset and informs the user of the status of the operation.

Use case 2: Data processing

Goal: Adjust training data so that the learning process would be more efficient and effective.

Actors: User, System

Pre-condition: The training video dataset is retrieved by the system.

Post-condition: Changes are made to the original training dataset on the system that supports the learning process.

Main scenario:

1. The system makes changes to the training dataset by applying different methods with the goal of supporting the learning process (example: frame extraction, normalization, feature extraction, ...).
2. These changes are then saved into the system for further learning.

Use case 3: Learning from data

Goal: Analyze data from the data processing operation and make predictions of anomalous events/objects.

Actors: User, System

Pre-condition: The data outputted from the data processing operation is retrieved by the system.

Post-condition: The system can predict whether an event/object is an anomaly or not.

Main scenario:

1. The system recruit models with algorithms to learn normal patterns (unsupervised) OR recruit algorithms to classify normal patterns and anomalies patterns (supervised) provided by data from the data processing operation.
2. During this learning process, any variables created by the models (example: weight, bias) are saved into the system for future refinements of the algorithms.

Use case 4: Model optimization

Goal: Continuously making changes to variables used within the algorithms of the model to find the best variables for anomaly detection.

Actors: User, System

Pre-condition: The variables generated by the algorithms from the learning process are retrieved by the system.

Post-condition: These variables are fine tuned to give the most accurate results for anomaly predictions.

Main scenario:

1. The system calculates how well the model is performing by comparing the results generated from the model with the desired results (for example using loss function).
2. The system makes changes to the variables of the models' algorithms so that the results generated from the model approaches the desired results.
3. The optimization process stops once the difference between the results generated from the model and the desired results falls below a certain threshold.

4. Non-Functional (Quality) Requirements

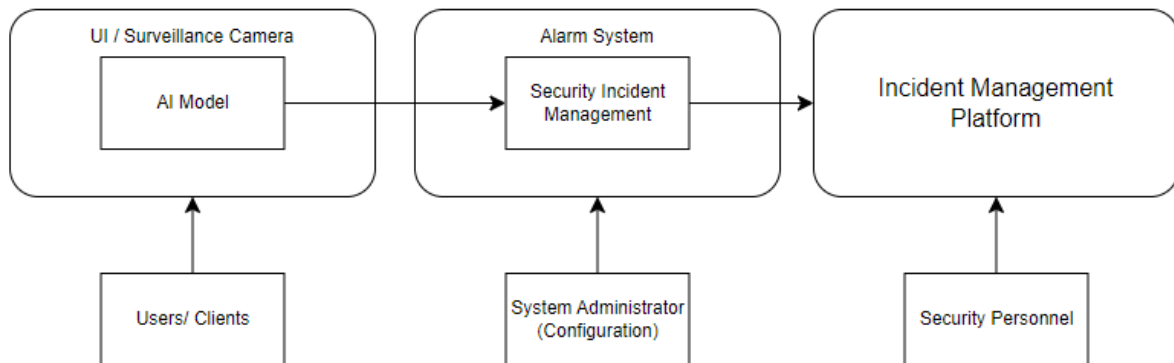
The non-functional requirements follow the ISO 9126 practice with the following criteria:

- **Functionality:**
 - Input videos for training model can't be any higher than 360p resolution so as not to consume too much performance, better for processing
 - Output data should be formatted in JSON format for ease of use commercially like in web applications
 - Variables generated from algorithms that are used for anomaly detections need to free in memory if needed (if creating new dynamic memory) in the code for better performance
 - Data saved by the system for future refinement of the model should not exceed 5GB, manual deletion or by algorithm should be considered
- **Reliability:**
 - Live footage streaming uses variable bitrate to lower processing power needs in high speed, chaotic environments with an average of 360p quality normally if no such chaotic
 - Backup network connection for main Wi-Fi network error like Bluetooth connection for IoT connections
- **Usability:**
 - Users can navigate and successfully finish a task within 20 seconds of operating
 - No more than 3 main colors being used to avoid cognitive overload
- **Efficiency:**
 - Video feed captured from security camera is processed within 3 seconds
 - Connection between software and IoT devices is with a delay around 5 seconds
- **Maintainability:**
 - Code will follow the team coding standards specified in the SQAP documents
 - Each iteration of the software will be separated by version with clear change logs
 - Weekly manual freeing of video storage, maximum 50 videos/week
- **Portability:**
 - The software can run in Windows 10/11 with no performance loss or software incompatibility

5. Interface Requirements

5.1. System in Context

- Users and stakeholders:
 - Security personnel: Monitoring the system, notice the abnormal behaviour alert
 - System Administrators: Maintaining, upgrading the AI Model for smooth experience
 - Users/ Clients: Interact with the system through the UI



- Descriptions:
 - The **UI/ Surveillance Camera** is accessed by the clients or users, capturing the video streams; The **AI Model** then processes the video frames, detecting abnormal behaviors. In case something is detected. If any is detected, alerts are sent to the **Alarm System**.
 - The **Alarm System** can be configured by the System Administrator, which receives alerts, trigger notifications; The **Security Incident Management Platform** logs and manages incidents reported by the system.
 - The **Incident Management Platform** is an Alarm Display UI, which provides real-time alerts to the security personnels.

5.2. User Interfaces

- Users log in with their credentials (username/ password).
- Display live video streams from surveillance cameras, overview of system status and recent alerts.
- Users can switch between different angles of live cameras
- Displaying bounding boxes around detected objects, then highlight them.
- Displays real-time alerts, timestamps, severity levels
- Can access history detections
- Securely log out from the system.

5.3. Hardware Interfaces

- Surveillance Cameras: Providing video feeds over Real-Time Streaming Protocol, must establish connections to camera IPs and specific ports
- Alarm Systems: Communicates with alarm systems via APIs or custom protocols; Define endpoints (URLs) for sending alerts
- Security Incident Management Platform: Uses RESTful APIs or database connection; Specify authentication tokens, endpoints and data formats for incident logging.
- User Interface: Web-based UI uses HTTPS for secure communication; Define routes for different UI components
- System Administrator UI: Uses HTTPS; Provide secure access to admin-specific routes
- Video Display Hardware: Typically use HDMI or DisplayPort; Ensure compatibility with common display standards.

5.4. Software Interfaces

- Database system can be used for storing links to videos that is in main administrator pc and also logs from the program using MySQL database bundled within Xampp for easier access with pre-installed web UI phpMyAdmin (8.0.30 Xampp from apachefriends.org)
Xampp also comes with pre-installed Apache HTTP as a web server for hosting website for the administrator access
- Transferring videos file or logs file from users computer to the main administrator computer for management can be done using WinSCP manually or using scripts for computers within a same local networks like small offices or home (6.3 from winscp.net)

5.5. Communication Interfaces

- Using HTTP that is prominent in RESTful API to use for IoT system involving the main software and physical devices like surveillance cameras, alarm system
- Real-time Streaming Protocol (RTSP) for live video capture from security camera to live video analysis
- HTTPS for secured access to administrator UI for advanced control settings like downloading videos or freeing video storage, logs, accounts management, etc

6. References (if any)

- Anomaly Detection in Surveillance Videos:
<https://ieeexplore.ieee.org/document/9001731>
- ISO/IEC 9126 (Software Product Quality):
https://en.wikipedia.org/wiki/ISO/IEC_9126
- Violence Detection using Computer Vision Approaches:
<https://ieeexplore.ieee.org/document/9817374>