AKADEMIA GÓRNICZO-HUTNICZA

Documentation for the project

Closed-Circuit TeleVision system

For the subject:

Design Laboratory

Elektronika i telekomunikacja 3 rok

Bartłomiej Kisielewski, Kacper Śliwa

Tuesday 8:00

Course leader Dr. Inż. Jacek Kołodziej

29.01.2025r.

The aim of the project is to design a simple CCTV system with motion, face detection.

Comprehensive Documentation of the CCTV System

Table of Contents

1. Introduction to the project

- **Project objectives**

The goal of this project is to develop a scalable CCTV system that provies:

  - Real-time video surveillance
  - Motion detection and automated recording
  - Local storage for captured footage

- Brief description of the System's funcionality

  The system uses two cameras conncted to a laptop for live monitoring and motion detection. When motion is detected, recording is triggered and saved locally. The user can view combined camera feeds and manage recordings through a basic GUI.

2. Project repository

- **Project structure**

  The repository is organized as follows:

  - **src/:** Contains the source code for the CCTV system, including system detection and video recording logic
  - **docs/:** Holds documentation files such as system design, user guides, and technical specifications
  - **README.md:** Provides an overview of the project, setup instructions, and usage guidelines
    https://github.com/Darned99/CCTV-System-Design-Lab-PY

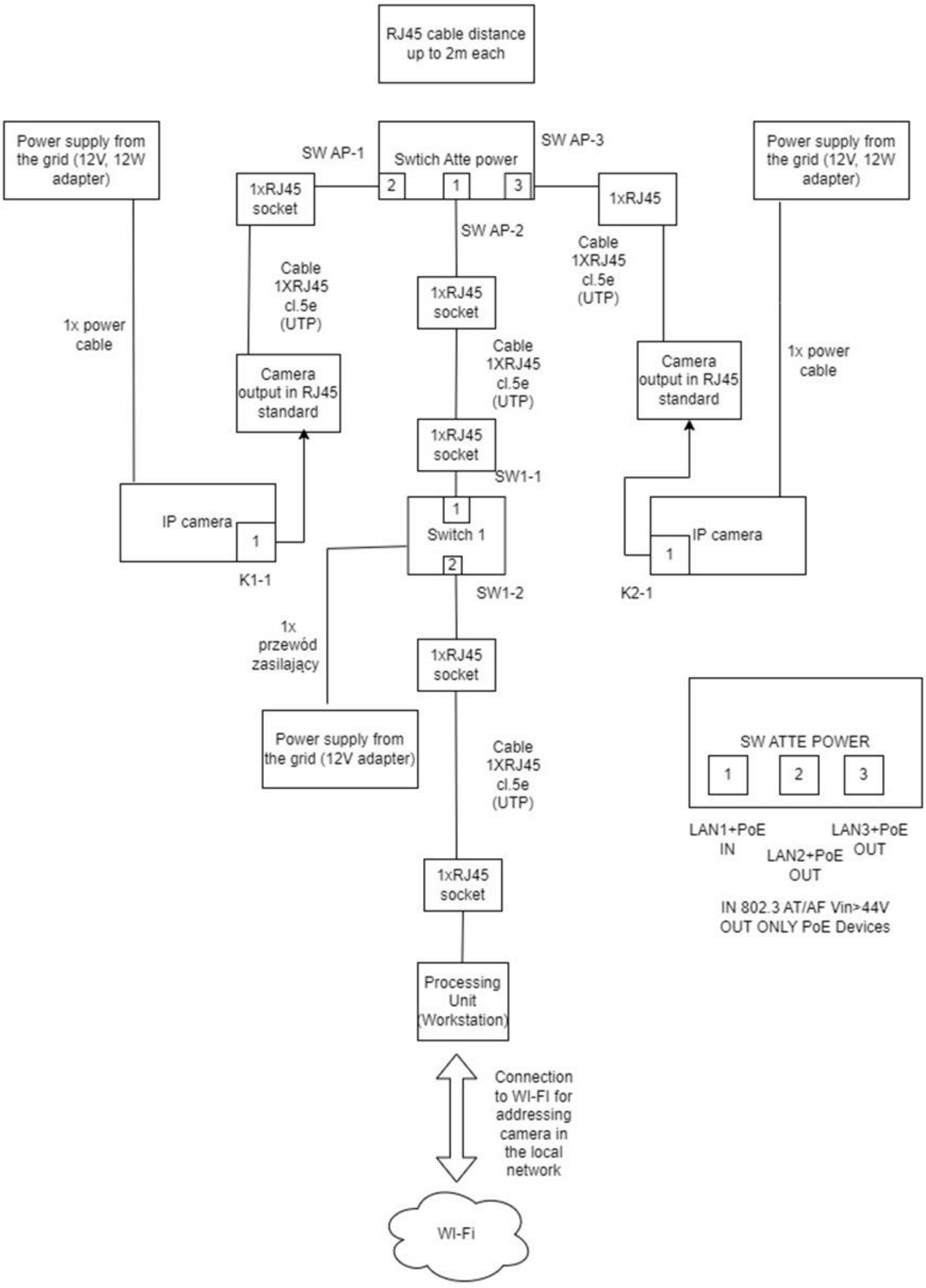
3. Project assumptions

- **Key system requirements**
  - **Cameras:** Two cameras with 720p resolution for live monitoring and recording.
  - **Processing Unit:** A laptop capable of real-time processing.
  - **Power Supply:** Reliable power source for uninterrupted operation
  - **Storage:** At lease 10 GB of available space for saving recorded videos in MP4 format
- **Development environment description**
  - **Programming language:** Python 3.8 or later
  - **IDE:** Visual Studio Code with Python extensions or PyCharm
  - **Platform:** Windows system for compatibility with OpenCV library.
- **Technologies and libraries used**
  - **OpenCV:** for video capture, motion detection and image processing
  - **Haar Cascade Classifiers:** For detecting faces and bodies in the video feed
  - **Datetime and OS Modules:** For timestamping recordings and managing storage
  - **VideoWriter:** For saving video recordings in MP4 format

4. System Block Diagram

System Components and Their Functions:

- **Camera Module (IP Camera):**
    - Captures real-time video footage.
    - Compatible with USB webcams or IP cameras using RTSP streams.
    - Each camera is initialized with an ID or RTSP URL.
- **Processing Unit (Workstation):**
    - Analyzes video frames from both cameras using pre-trained Haar cascades for face and body detection.
    - Operates in two modes:
        - Motion detection mode: triggers recording when motion is detected
        - Continuous recording mode: Records video continuously
    - Handles frame resizing and merging for GUI display
- **Storage Module:**
    - Saves all recordings in a local folder name Videos
    - File naming format: camera_id_date_hour.mp4
- **GUI Application (User Interface):**
    - Displays live video feeds from all connected cameras in 2x2 grid
    - Allows mode switching using keyboard inputs
        - 1 for motion detection mode
        - 2 for continuous recording mode
        - q exit the program


RJ45 cable distance up to 2m each
Power supply from the grid (12V, 12W adapter)
SW AP-1
Swtich Atte power
2
1
3
SW AP-3
Power supply from the grid (12V, 12W adapter)
1xRJ45 socket
1xRJ45
SW AP-2
Cable 1XRJ45 cl.5e (UTP)
Cable 1XRJ45 cl.5e (UTP)
1x power cable
1xRJ45 socket
Cable 1XRJ45 cl.5e (UTP)
1x power cable
Camera output in RJ45 standard
Camera output in RJ45 standard
1xRJ45 socket
SW1-1
IP camera
1
1
Switch 1
2
1
IP camera
K1-1
SW1-2
K2-1
1x przewód zasilający
1xRJ45 socket
Power supply from the grid (12V adapter)
Cable 1XRJ45 cl.5e (UTP)
SW ATTE POWER
1
2
3
LAN1+PoE IN
LAN2+PoE OUT
LAN3+PoE OUT
IN 802.3 AT/AF Vin>44V
OUT ONLY PoE Devices
1xRJ45 socket
Processing Unit (Workstation)
Connection to WI-FI for addressing camera in the local network
WI-Fi

5. Data flow

- Each camera captures video frames and sends them to the processing unit.
- Frames are converted to grayscale for motion analysis. Haar Cascade classifiers detect faces and bodies.
- If motion is detected in **Mode 1**, recording is triggered and continues for a predefined duration after motion stops.
- In **Mode 2**, the system records continuously without motion detection.
- All video streams are displayed in separate OpenCV windows for live monitoring.
- The program can be terminated manually by pressing the q key, which releases all resources.

6. Technical specification of components

Hardware

- **Cameras:**
  - USB webcams or IP cameras (RTSP supported).
  - Example RTSP URL: rtsp://username:password@192.168.1.67:554/Streaming/Channels/2.

Software

- **Motion Detection Algorithm:**
  - Haar Cascade Classifiers:
    - haarcascade_frontalface_default.xml (for face detection).
    - haarcascade_fullbody.xml (for body detection).
  - Resolution: Frames processed with at leat 480p resolution, 720p recommended
- **File Management:**
  - Videos saved in MP4 format using OpenCV's VideoWriter.
  -

7. System Implementation

Core Modules and Their Roles

- **CameraMonitor Class (cctv_project.py):**
  - Manages cameras, motion detection, and recording.
  - Supports two modes:
    - **Motion Detection Mode:** Detects faces and bodies using Haar Cascade and triggers recording.
    - **Continuous Recording Mode:** Records video streams without analysis.
  - Saves recordings in the Videos folder with timestamps.
- **Main Script (run_cameras.py):**
  - Initializes cameras (local or RTSP) and manages their operation.
  - Enables dynamic switching between modes via keyboard inputs (1, 2).
  - Displays live camera feeds in OpenCV windows.

8. User Guide

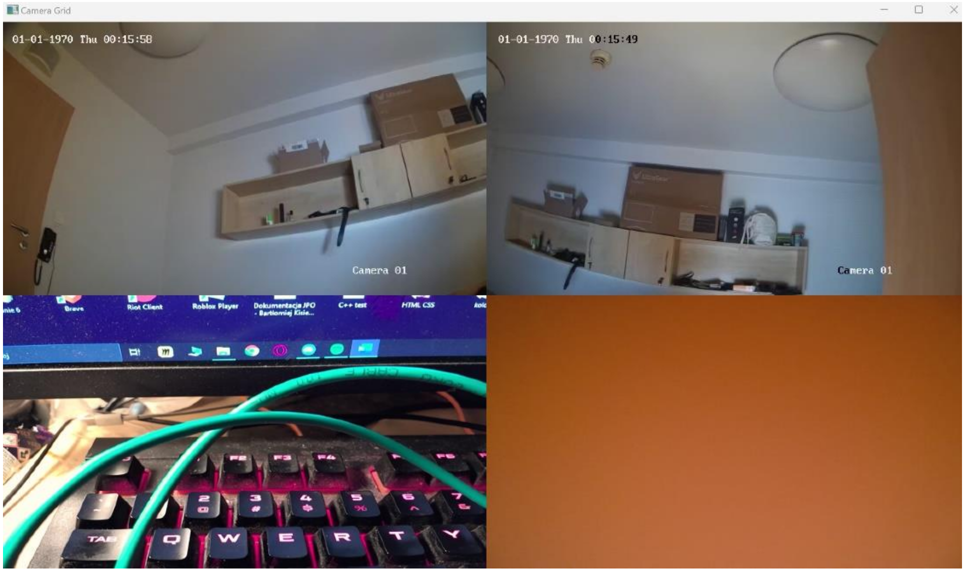
How to run the system

- **Installation requirements**
  - Ensure Python 3.8 or later is installed
  - Install required dependencies using:
    - **pip install opencv-python**
    - **pip install numpy**
  - Connect cameras (USB or IP) and ensure they are operationl
- **Running:**
  - Navigate to the project folder and execute:
    - **python src/run_cameras.py**
  - Enter the number of cameras and provide their indentifiers (0, 1 for USB or for example RSTP URL for IP cameras)
- **Instruction for Using GUI application:**
  - The system displays a grid view for all camera feeds.
  - Use the following keyboard commands:
    - 1 – enable motion detection mode, it is a basic mode
    - 2 – enable continuous recording mode
    - q – exit the application and release resources
  - Recorded videos are stored in the Videos/ directory, after 7 days they are automatically deleted.
- **Troubleshooting:**
  - If an IP camera does not load, test its stream
  - If recordings are not saved, check the Videos/ directory permissions
  - If recordings are saved, but they're damaged, try to change in the CameraMonitor class located in cctv_project width and height
  - Debugging logs are printed in the terminal for further issue diagnosis

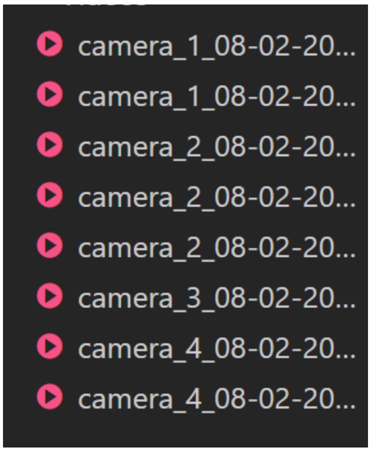
9. Brief presentation

The following are important elements of the system during commissioning such as:

- **GUI with the grid of 2x2**



- **Saved video files in the Videos directory**



camera_1_08-02-20...

camera_1_08-02-20...

camera_2_08-02-20...

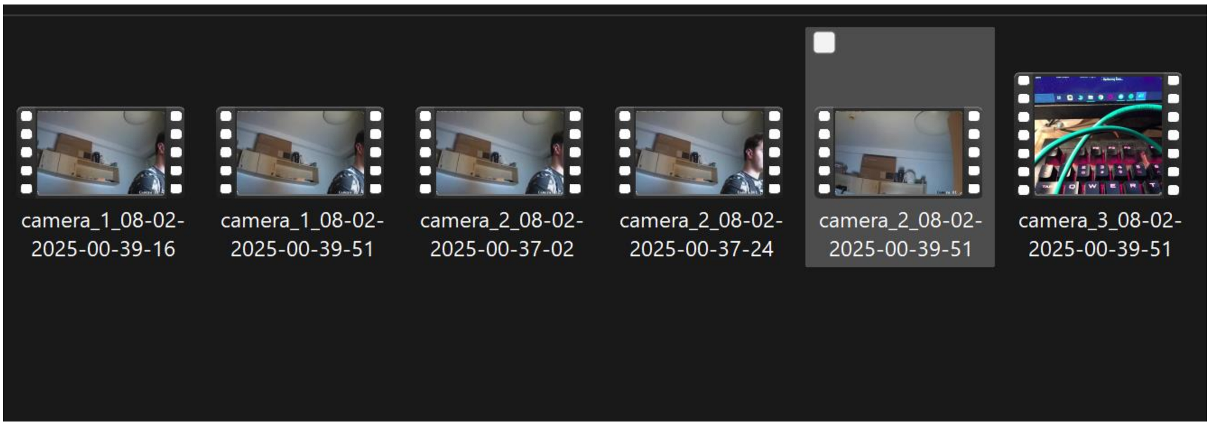
camera_2_08-02-20...

camera_2_08-02-20...

camera_3_08-02-20...

camera_4_08-02-20...

camera_4_08-02-20...



camera_1_08-02-2025-00-39-16

camera_1_08-02-2025-00-39-51

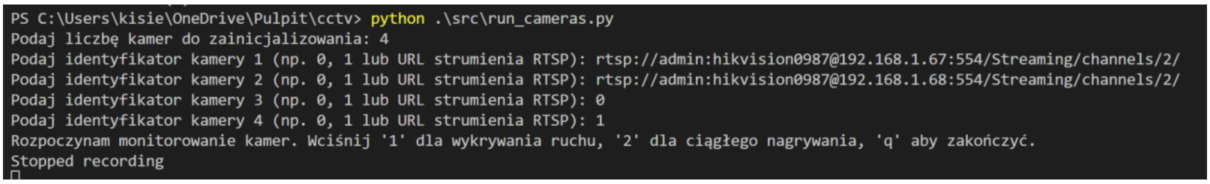
camera_2_08-02-2025-00-37-02

camera_2_08-02-2025-00-37-24

camera_2_08-02-2025-00-39-51

camera_3_08-02-2025-00-39-51

❖ Display of system logs



```
PS C:\Users\kisie\OneDrive\Pulpit\cctv> python .\src\run_cameras.py
Podaj liczbę kamer do zainicjalizowania: 4
Podaj identyfikator kamery 1 (np. 0, 1 lub URL strumienia RTSP): rtsp://admin:hikvision0987@192.168.1.67:554/Streaming/channels/2/
Podaj identyfikator kamery 2 (np. 0, 1 lub URL strumienia RTSP): rtsp://admin:hikvision0987@192.168.1.68:554/Streaming/channels/2/
Podaj identyfikator kamery 3 (np. 0, 1 lub URL strumienia RTSP): 0
Podaj identyfikator kamery 4 (np. 0, 1 lub URL strumienia RTSP): 1
Rozpoczynam monitorowanie kamer. Wciśnij '1' dla wykrywania ruchu, '2' dla ciągłego nagrywania, 'q' aby zakończyć.
Stopped recording
□
```

In addition, the appendix contains sample recordings presenting the operation