Technical University of Cluj-Napoca
Faculty of Automation and Computer Science

Design of a surveillance system using webcams

Diaconu Călin
Group: 30432

Table of Contents

Task Description

Design a surveillance system which uses webcams.

The system must allow the online visualisation of the video stream and it must alert the user by means of sound in case of an unexpected situation (e.g.: in a room that should be empty, a moving object can be detected).

Project Design

For motion detection, the system uses an algorithm based on background subtraction.

This kind of algorithm works according to the following steps:

1) Accumulating the weighted average of the previous N frames (in this case 32 being the default value)
2) Taking the current frame and subtracting it from the weighted average of frames
3) Thresholding the output of the subtraction to highlight the regions with substantial differences in pixel values ("white" for foreground and "black" for background)
4) Applying basic image processing techniques such as erosions and dilations to remove noise
5) Using contour detection to extract the regions containing motion

To perform background subtraction and motion detection, a class named "SingleMotionDetector" is implemented. The class is developed inside the "singlemotiondetector.py" file, found in the "motion_detection" submodule of "pyimagesearch".

The "webstreaming.py" file will use OpenCV to access the webcam, perform motion detection via "SingleMotionDetection" and serve the output frames to the web browser via the Flask web framework.

In order for the web browser to have something to display, the content of "index.html" needs to be populated with HTML used to serve the video feed. Flask will actually handle the sending of the video stream to the browser.

Tools and languages

HTML - as the application needs to run inside a web browser, the HTML language is used. HTML (Hypertext Markup Language) is the standard markup language for documents designed to be displayed in a web browser. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

Python - the main language, used in developing the motion detection algorithm and generating the web server through Flask. Python is an interpreted, high-level, general-purpose programming language. Python uses significant whitespace and has an object-oriented approach. It is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented and functional programming.

Flask - used to generate the web server used to access the generated html file by the browser. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation or any other components where pre-existing third-party libraries provide common functions. Flask supports extensions that can add application features as if they were implemented in Flask itself.

OpenCV - used for the motion detection and image processing part of the project. OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use. It supports deep learning frameworks and is written in C++.

PyAudio - used for generating the alarm sound when an intruder is detected. It is a cross-platform audio input/output stream library. It provides Python bindings for PortAudio, the cross-platform audio I/O library. It can be used to play and record audio on a variety of platforms.

How to use

Prerequisites: Python 3, Flask, Numpy, OpenCV, Imutils, PyAudio, a web browser.

Instructions:

1. Open a terminal inside the folder containing the project
2. Execute the command “python webstreaming.py --ip 0.0.0.0 --port 8000” (ip will be the address of the device, port is the ephemeral port number of the server between 1024 and 65535; optionally, the --f argument could be added, which will change the number of frames used to construct the background model, with the default value being 32)
3. Open a web browser and access the address http://127.0.0.1:8000/. Now you should be able to get a visual output from the camera with a red bounding box around the moving element and also hear an alarm sound if such an element is present.

Bibliography

- https://pyimagesearch.com/
- https://stackoverflow.com/
- https://wikipedia.org/
- https://pypi.org/
- https://people.csail.mit.edu/