Embedded Systems - Raspberry Pi Surveillance system

Richard Chomjak

May 2018

1 Project requirements

- System should behaves like surveillance system.
- System send an email with taken photography when motion was detected.
- System should responses on the following parameters
  - Email options SMTP server settings (username, password, address), client settings (from address, to address), timeout between emails
  - Camera options resolutions, frame rate, brightness, contrast
  - Computer visions options Gaussian Blur mask size
  - Miscellaneous - size of frame buffer
- User could defines the parameters e.g. frame buffer size, so, system changes internal queue size and calculates average frame.

2 Hardware

As platform, system uses Raspberry Pi model 3B.

2.1 Sensors

Pi camera

3 Software

Platform runs on Raspbian (GNU/Linux Debian based distribution) and OpenCV 3.3.0 (python)

3.1 Implementation

System is divided into three file **main.py**, **Dataloader.py** and **Mail.py**. In main is class **Surveillance**, which is the aggregate class.

3.1.1 Parse input arguments

System has two parameters -*h* help and –*config* path to the config file. Library **argparse** is used for parsing input parameters.

3.1.2 Input parameters

Configuration file is in JSON format, you can see an example in **config.json**, in repository. The content of file is parsed to Python's named tuples.

3.1.3 Reading frame from camera

Class **DataLoader** (file: *Dataloader.py*) is wrapper upper on device type OpenCV video and and PiCam. Choose of device type is based on parameter (*(*dev_type)) in **__init__** function. Parameter could has values **DEVICE_CAM** for OpenCV video or **DEVICE_CAM_RPI** for PiCamera.
Surveillance class initialize **DataLoader** and calls method **get_data_frame**

3.1.4 Manipulation with frame

For manipulating frame *OpenCV* library is used. Firstly, frame is converted to the gray scale by method **cvtColor**, then Gaussian Blur is applied on the frame by method **GaussianBlur** with mask parameter from config. file. For subtraction between frames, frame should be thresholded into only black and white colour spectrum, system calls function **threshold** for obtaining black and white colour frame. Optional stage is queue initialization for calculating an average frame, the frames are appended to the queue until queue is full. Length of queue depends on config. parameter. Next, average frame is calculated, thresholded frame is taken and subtracted from average frame to obtain frame without background. Then, method **erode** is called to clean noise followed by method **dilate** to use convolution to magnify differences at the frame. Method **findContours** is used for finding contours at the frame, when contour area is large enough, contour is drown to the original frame, next, frame is encoded to *PNG* format and appended to the image buffer. Finally, thresholded frame is appended to the queue and average frame is calculated.

3.1.5 Sending Email

Email class responds for sending email, method **send_email** initialize SMTP server and creates message to be sent. Email is sent in case, when time difference between sent emails is satisfied. Parameters for email are in config. file.