

Biztonsági kamera:

Security camera with motion detection and tracking.

Dependencies:

- Python 3.9.7
- OpenCV-Python
- Numpy
- Matplotlib.Pyplot
- Argparse
- os
- sys

Installing dependencies

Use `pip install -r requirements.txt`

Modules:

- `main.py`
 - The software can be started with the `main.py` file, that takes additional arguments.
- `bgsb.py`
 - This module uses background subtraction to show motion.
- `occupation.py`
 - This module helps with detecting motion.
- `track_motion.py`
 - Tracks motion, using CamShift algorithm.
- `track_motion2.py`
 - Tracks motion using contour detection.

User guide:

```
usage: main.py [-h] [--input INPUT] [--bsAlgo {KNN,MOG2}] [--sensAlgo
{camshift,contour}]
```

Security Camera with motion detection.(quit with 'q')

optional arguments:

```
-h, --help            show this help message and exit
--input INPUT          path to the video source (webcam is default).
--bsAlgo {KNN,MOG2}    Choose the background subtraction algorithm MOG2 or KNN
--sensAlgo {camshift,contour}
                        Choose what type of motion detection to use: camshift or
contour. Contour detection can track multiple objects
```

main.py

This is the user interface of the program. The user can specify the video source input, the algorithm of the background subtraction and the type of the motion detecting. And the library i have used for this, is `argparse`.

bgsub.py

This is the brain of the program. The `main.py` module calls `bgsub.py` to start the whole process.

`bgsub.py` takes 3 agruments `bgsub(vsrc, bsAlgo, sensAlgo)`. `vsrc` is the path to the video file or the number of the camera. `bsAlgo` is the backgroundsubtraction algorith, it can be MOG2 or KNN. `sensAlgo` is the type of motion detecting, Camshift can be used for detecing a predefined ROI (range of interest). This technique is not fully implemented yet. The other type of motion detecting is with finding the contours of objects moving on the video. This technique i find more effective for stationary security cameras.

Using the Backgroundsubtractor classes:

```
backSub = cv.createBackgroundSubtractorMOG2() # for MOG2

backSub = cv.createBackgroundSubtractorKNN() # for KNN

fgMask = backSub.apply(frame, learningRate=-1) # obtain forground mask of video
stream
```

These constructors can take various arguments for fine tune your background subtraction quality. But they do the job witouth any arguments aswell.

`backSub.apply(frame, learningRate=-1)`

frame: is a frame from the video learningRate: specify the learning rate of background model (0 - 1), -1 is for automatically chosen

learning rate

VideoCapture

```
capture = cv.VideoCapture(vsrc) # get stream from source
ret, frame = capture.read() # obtain frame from video stream
cv.imshow('VideoFrame', frame) # show the frame obtained from the videostream
```

occupation.py

The purpose of this module, to make sure if theres any major object moving in the frame, for example: animals, humans, bicycles, cars, etc...

This way we dont have to apply any contour detection algorithm, so we can save precious resources.

`is_occupied(frame, threshold)`

Frame: the input image, this should be a foreground mask

Threshold: with this argument, we can adjust the sensitivity of the detection in range 0 - 10000.

the smaller the number, the smaller motion can be detected.

```
hist = cv.calcHist([frame], [0], None, [2], [0,256]) # calculating the histogram
of the binary image

summa = sum(hist[:,0]) # sum of the pixel intensity

if (hist[1]/summa)*10000 > threshold:
    return True
return False
```

In this code example, we calculate the histogram of the foreground mask.

And if there is enough white pixels in the mask (this we can adjust with the threshold argument) the function returns True.