Department of Computer Science
Technical University of Cluj-Napoca

Structure of computer systems

Laboratory activity 2018-2019

Project title: Webcam motion detection system

Name: Lukacsffy Levente
Group:30433

Contents

1 Introduction **3**

1.1 Definition 3

2 Introduction **4**

2.1 Objectice 4

2.2 Use cases 4

2.3 Chosen language 4

3 General Aspects **5**

4 Implementation **7**

4.1 Development environment 7

4.2 Setting up environment 8

4.3 Accessing hardware: OpenCV 9

4.4 Other steps 12

4.5 Winsound 14

4.6 Pygame 15

5 Code **16**

5.1 HTML 16

5.2 Python 16

6 Testing **19**

6.1 Examples 19

6.2 Refining the visuals 20

6.3 Error cases 21

7 Developement possibilities **22**

8 Conclusion **23**

9 Bibliography **24**

Chapter 1

Introduction

1.1 Definition

Motion detection is the process of detecting a change in the position of an object relative to its surroundings or a change in the surroundings relative to an object. This relative change can be found by the differentiation of the standard and the current image. Motion detection is widely used on multiple branches:

1. Alarms
2. Ticket gates
3. Lightning systems
4. Security
5. Automation

The project is about developing a system based on more subsystems capable of complex functions like detecting movements, detecting changes in environment, streaming live footage onto browser. The motion detection is the base step of creating complex home surveillance systems. This project can be used for further development as a fundamental function of a security system. The following chapters introduce the program creation, algorithms, basic principles and the complex process of motion detection. It is needless to mention the possible errors which may occur based on the environmental variables like light, colours and position of objects. These will be presented in later chapters.

Chapter 2

Introduction

2.1 Objectice

Create a system which has the following functionalities:

1. Visualization of the object via Internet

2. Alert in case of detected motion

2.2 Use cases

1. Starting the application

2. Visualising the observed object

3. Getting alerted when a movement occurs

2.3 Chosen language

Python is a general-purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. For the implementation I chose Python 3.7.4 as this language has a lot of benefits:

1. As it is a new language for me I preferred a language which can be easily learnt. Python does not have a specific syntax and also has a lot of libraries. This is the main idea of choosing Python. For this project a huge amount of libraries are used which help the programmer (for ex.: cv2, flask, winsound, datetime)

2. Python is also well known for it's usability in different domains so it can be considered as an important programming language.

3. [2]Python is a simple and easily readable language

Chapter 3

General Aspects

Steps of motion detection:

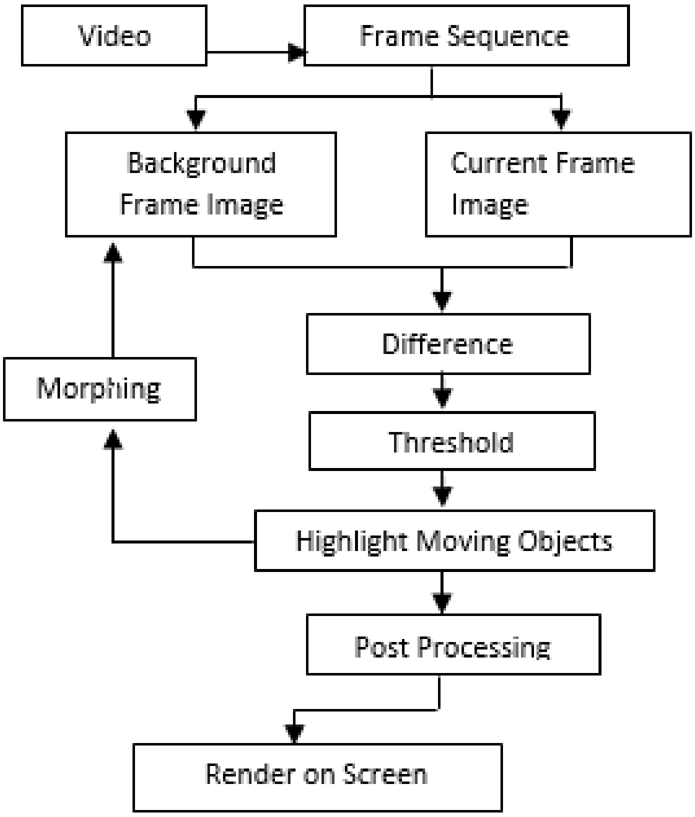


Figure 3.1: Steps

This flowchart identifies the basic algorithm of motion detection. These steps will be further discussed but first the algorithm must be understood. Motion detection is not based on a video. As we know videos are images (frames) played a speed (framerate) which seems to us as a video which is created by our mind. So the first step in the algorithm is to divide the footage into separate frames. Then all these frames are analysed. Algorithm steps of getting movement:

1. Creating a standard frame (first frame), set to none, which will be used for calculating the absolute difference. This frame is the first frame after the program is started. This frame is considered as it there was no movement. Any change in the following frames can be considered as movement.

2. . Modifying the current frame in the following manner
    (a) Making a greyscale copy of it for the checking the threshold
    (b) Making a blurred copy of the greyscale frame
    (c) If the standard frame isn't set a frame, then set it to the greyscale image
3. Calculate the absolute difference between the base frame and the greyscale image
4. Creating a threshold based on the light conditions of the room
5. Creating a dilate image based on the threshold which means that the difference will be expanded as white pixels on black background
6. If a white area is bigger the a limit then consider it as movement

Chapter 4

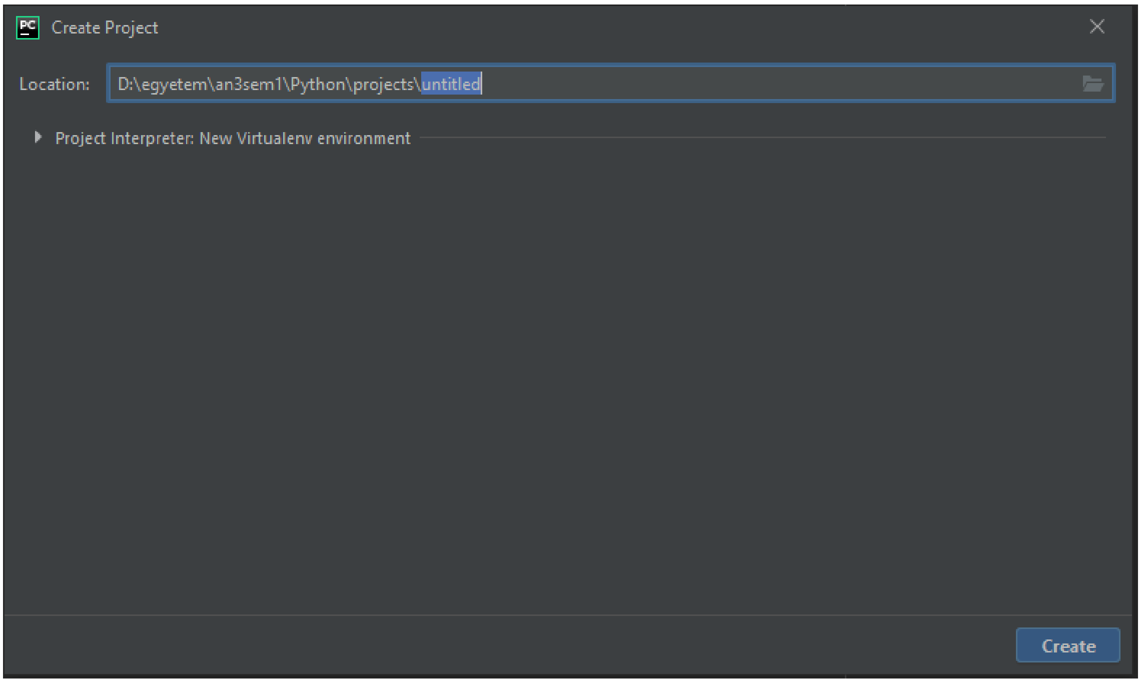
Implementation

In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages.

4.1 Development environment

For developing such a complex project an IDE (Integrated Development Environment) must be chosen in order to help us. The choice this time is Pycharm. Pycharm is a tool used for developing Python applications. This choice is easily confirmed by the fact that this IDE is free despite the fact that it offers a huge amount of advantages[5], like:

1. Intelligent Coding Assistance
2. Intelligent Code Editor
3. Smart Code Navigation
4. Fast and Safe Refactorings
5. Built-in Developer Tools
6. Debugging, Testing and Profiling
7. VCS, Deployment and Remote Development
8. Database tools
9. Web Development
10. JavaScript and HTML
11. Customizable UI
12. Plugins



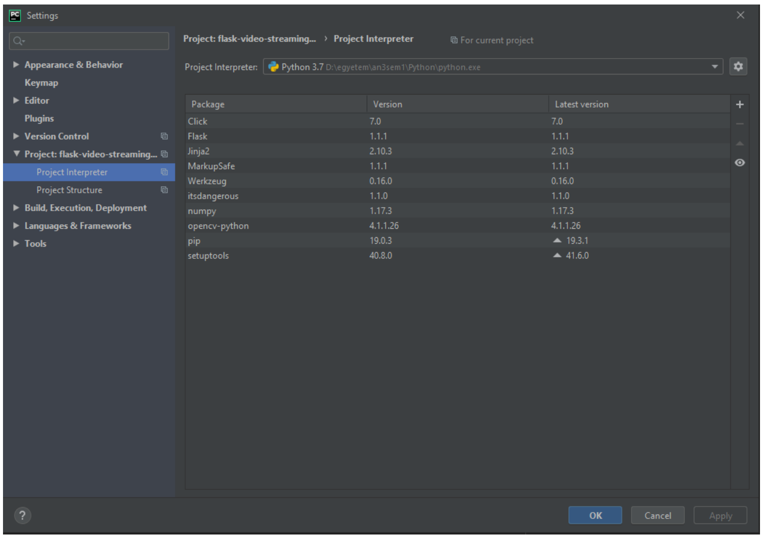
Location: D:\egyetem\an3sem1\Python\projects\untitled

Project Interpreter: New Virtualenv environment

Figure 4.1: Creating a new project

4.2 Setting up environment

After [1] Pycharm is installed a new project is created. [4]In this IDE the libraries must be installed in the Interpreter in order to used them. For example openCV can be installed in the following way: File - Settings - Project interpreter. In this window we click on the "+" marked



Project: flask-video-streaming... › Project Interpreter For current project

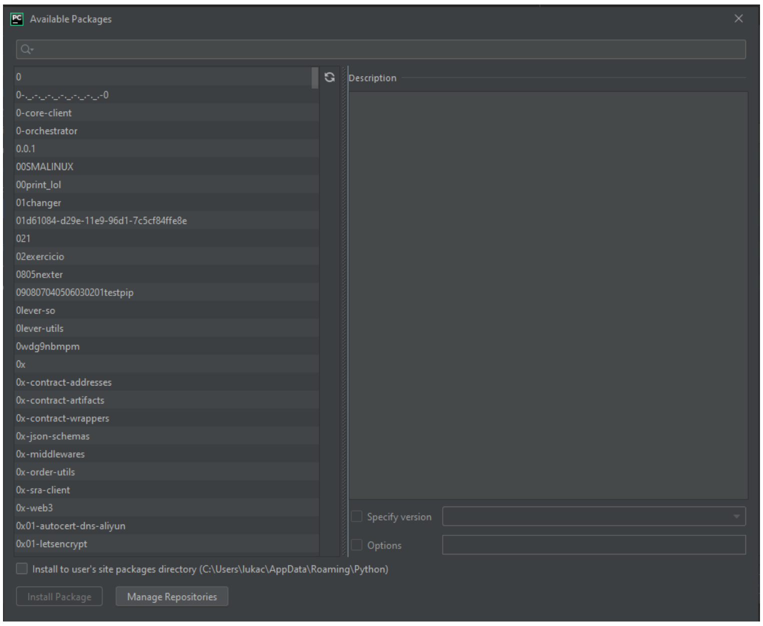
Project Interpreter: Python 3.7 D:\egyetem\an3sem1\Python\python.exe

Package	Version	Latest version
Click	7.0	7.0
Flask	1.1.1	1.1.1
Jinja2	2.10.3	2.10.3
MarkupSafe	1.1.1	1.1.1
Werkzeug	0.16.0	0.16.0
itsdangerous	1.1.0	1.1.0
numpy	1.17.3	1.17.3
opencv-python	4.1.1.26	4.1.1.26
pip	19.0.3	▲ 19.3.1
setuptools	40.8.0	▲ 41.6.0

Figure 4.2: Project interpreter

by the arrow.

The new window has a search bar where we can find all the libraries. Select a library then click the "Install Package" button. This will install the selected packages. If more packages are neded the actions must be repeated.



0
0-._.-._.-._.-._.-._.-._.-0
0-core-client
0-orchestrator
0.0.1
00SMALINUX
00print_lol
01changer
01d61084-d29e-11e9-96d1-7c5cf84ffe8e
021
02exercicio
0805nexter
090807040506030201testpip
0lever-so
0lever-utils
0wdg9nbmpm
0x
0x-contract-addresses
0x-contract-artifacts
0x-contract-wrappers
0x-json-schemas
0x-middlewares
0x-order-utils
0x-sra-client
0x-web3
0x01-autocert-dns-aliyun
0x01-letsencrypt

□ Install to user's site packages directory (C:\Users\lukac\AppData\Roaming\Python)

Figure 4.3: Steps

4.3 Accessing hardware: OpenCV

This Python library is the base of the implementation as it is created for solving computer vison problems. This library was first developed by Intel, mainly used for real time computer vision problems. CV2 library offers the following simple functions like: VideoCapture (access to hardware (camera)), cvtColor (used for changing the colour of the frames), imshow (returning the frames)... The first step of the development is to have access to the hardware, in our case the Webcam. This can be achieved by calling the "cv2.VideoCapture(0)" function. This function selects the default camera (value 0 stands for the default hardware as camera). After this a baseframe variable is created which then will stand for the standard frame(the one which will be used to check if a movement occured). Implementing the next steps of the algorithm:

1. The following steps must be done for multiple times with each frame so a while loop is used. This loop is used for analysing all the frames separately. Get the current frame and store it in a variable. With cv2 having the $cvtColor(frame, cv2.COLORRGB2GRAY)$ every frame is made greyscale and a greyscale copy of them is made. You may ask why are the frames greyscale? Well the answer is simple. It is less expensive (computationally) to operate on single-channel image than on 3-channel images like RGB.

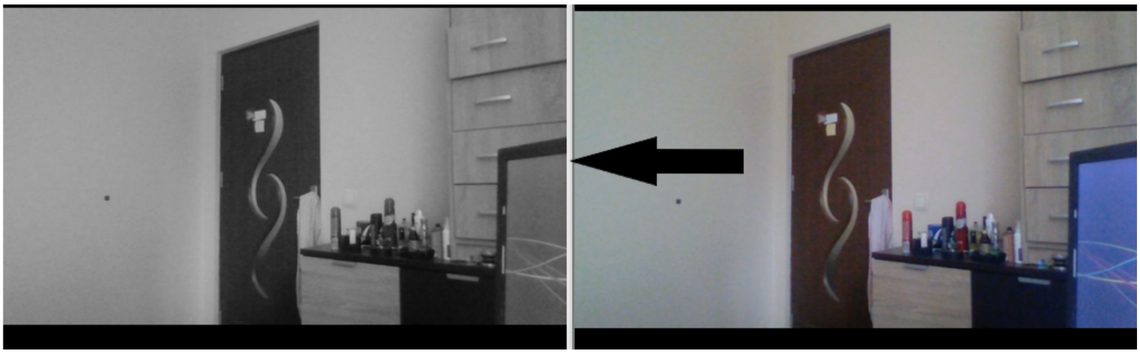


Figure 4.4: Grayscaling

2. The greyscale image is blurred by a cv2 function which is called GaussianBlur. This function uses a Kernel (A kernel is nothing more than a (square) array of pixels (a small image so to speak). Usually, the values in the kernel add up to one. This is to make sure no energy is added or removed from the image after the operation.) Each pixel is multiplied by this kernel. The kernel size must be chosen as a valid odd integer (in order to have a centre). And the standard deviation is set to 0. cv2.GaussianBlur(greyscalef, (15,15),0) The higher the number,the more pixels are washed together

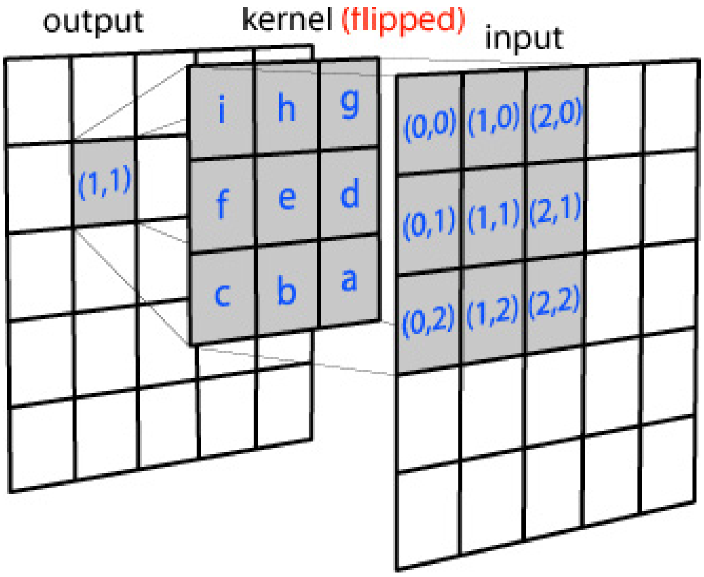


Figure 4.5: Kernel explanation

3. We check if the base frame is set or, none. If it is none, we set it to the greyscale image.

4. The next step is based on cv2 again. We calculate the[8] absolute difference between the current frame and the greyscale, base frame. This returns a "delta frame". cv2.absdiff(X,Y) subtracts each element in array Y from the corresponding element in array X and returns the absolute difference in the corresponding element of the output array Z. Example:

5. [3]The following step is based on calculating the threshold for the motion detection: The cv2.threshold() function is used: Simple Thresholding: Here, the matter is straight forward. For every pixel, the same threshold value is applied. If the pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value. The function cv.threshold is used to apply the thresholding. The first argument is the source image, which should be a grayscale image. The second argument is the threshold value which is used to classify the pixel values. The third argument is the maximum value which is assigned to pixel values exceeding the threshold.

6. The next step is to represent the effects of the [9]threshold which can be achieved by cv2.dilate. It is basically an image filtering algorithm which allows to expand the differences between the images. This is used in order to make small movements visible too. This example shows us that even small surfaces can be expanded. These expansions are done on a binary image (black background, white foreground). It also uses a kernel which is by default a 3*3 kernel. In order to make the expansion bigger the number of iterations of dilating can be changed. If iterations=2 it means that the expansion will be further expanded. The inverse operation is called Erosion. The function is called to the threshold created: cv2.dilate("threshold image", None, iterations =nr)

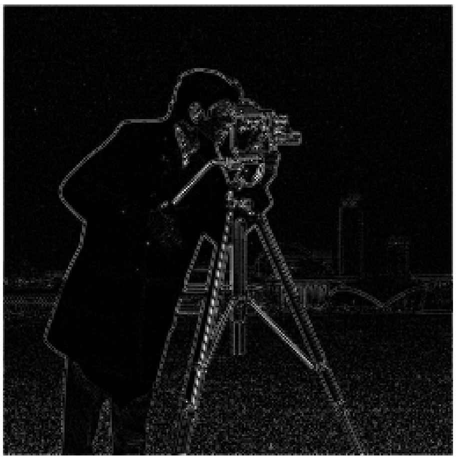


Figure 4.6: Absolut difference

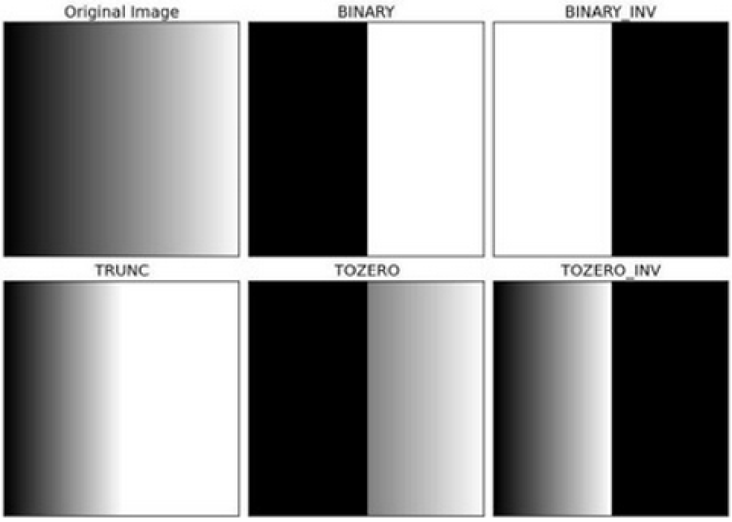


Figure 4.7: Treshold

7. [6]Next we must find the contours of the binary image. What are contours? Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same colour or intensity. The contours are a useful tool for shape analysis and object detection and recognition. The findContours function has 3 parameters: first one is source image, second is contour retrieval mode, third is contour approximation method. The following are the chosen ones in the implementation: $cv2.RETR_EXTERNAL$ and $cv2.CHAIN_APPROX_SIMPLE$. These allow the following functionality: save memory by removing all points which are redundant (compression). For example if we have a triangle then we don't need all the point from the triangle but only the 3 straight lines. This creates a huge memory save. This algoritm returns two values. For theese we only use the first value so simply we can set the second value to an unused variable.

8. In the next step we use the knowledge gathered and make the decision: All the contour areas are traversed and checked. If a contour area is greater than a specified threshold limit then a bounding rectangle is created on the image. This rectangle is drawn to the image.

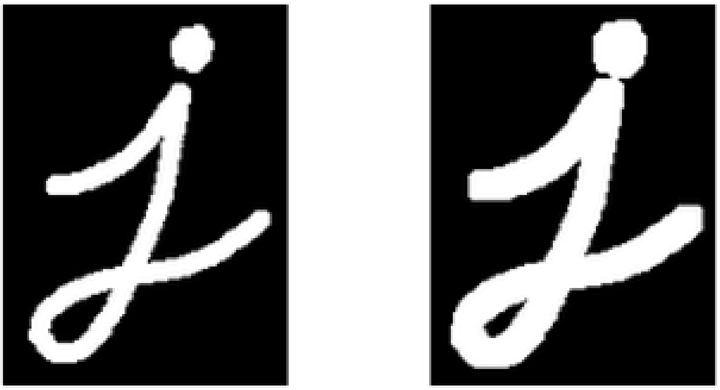


Figure 4.8: Expansion

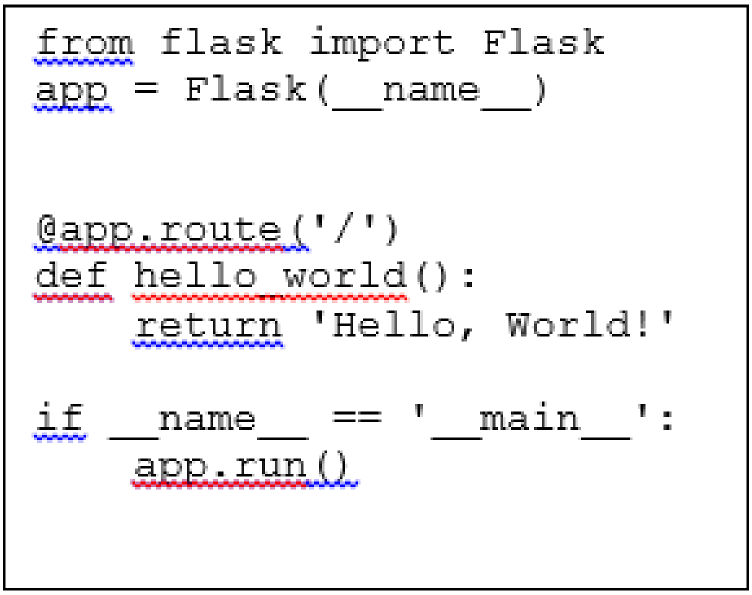
In this decision we can also add notifications like a beep noise. The easiest way of doing this is by selecting a frequency and a duration and call the winsound.Beep(frequency, duration) function. The usage of this function: winsound.Beep(frequency, duration): Beep the PC's speaker. The frequency parameter specifies frequency, in hertz, of the sound, and must be in the range 37 through 32,767. The duration parameter specifies the number of milliseconds the sound should last. If the system is not able to beep the speaker, Runtime Error is raised.

4.4 Other steps

Figure 4.9: Flask

Flask (source code) is a Python web framework built with a small core and easy-to-extend philosophy. Why is Flask a good web framework choice? Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running. Example of flask application:

The above code shows "Hello, World!" on localhost port 5000 in a web browser when run with the python app.py command and the Flask library installed. In our project Flask is used in order to create a temporary web server which is able to stream the live footage of the webcam to a browser. The aim of the Flask server is the implementation of the following: In order to make this possible first we need to make a basic html page. What is HTML? [12]Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. The latest version is HTML5. Basic templates can be found on the internet but for us a very simple one is enough. Our web application is made in a similar manner as the last Flask application presented. Only we need to add a few lines of



```
from flask import Flask
app = Flask(__name__)


@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run()
```

Figure 4.10: Flask example

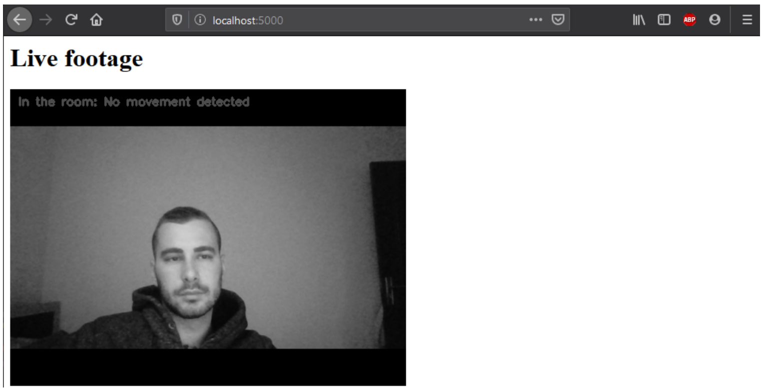
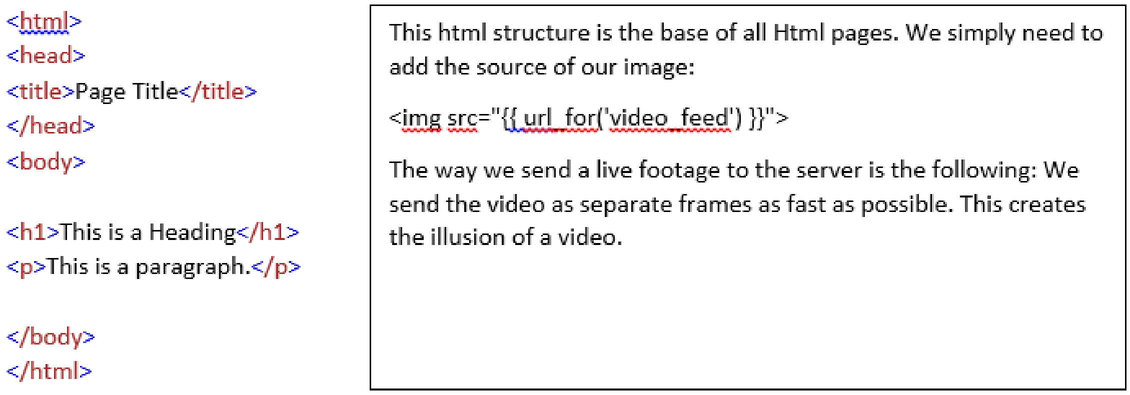


Figure 4.11: HTML aim

code in order to send the frames: The app must return the frames, This can be implemented in the following way: We extend the “@app.route('/')” with one more functionality. It will be capable of responding: return Response(gen(), mimetype='multipart/x-mixed-replace; boundary=frame'). Here the gen function has the motion detection but also has something in plus. It has a functionality which “yeilds” the frames to the image source. This function however does not accept simply pictures. So to say, we have to convert the images into bytes. This can be achieved in the following manner:We use cv2 build in imencode and tobytes functions. What does “imencode” do? As the name suggests it is used for image encoding. So as its name suggests it will simply encode the frames to jpg then this image can be further converted ino bytes by the tobytes function. We can further develop our application by showing the different “steps” of the algorithm. This can be easily done because all the variables used in the creation of the motion detection are available: the baseframe, the greyscale frame, the blurred frame, the dilated image and the delta frame. To show these frames in real time we can simply call a function in the main while loop. To show a frame on the desktop we use an OpenCV function called imshow. This function displays an image in the specified window.
Parameters:

1. winname – Name of the window.

2. image – Image to be shown.



```
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This html structure is the base of all Html pages. We simply need to add the source of our image:

`<img src="{{ url_for('video_feed') }}">`

The way we send a live footage to the server is the following: We send the video as separate frames as fast as possible. This creates the illusion of a video.

Figure 4.12: HTML template

Example: cv2.imshow('Example', image)

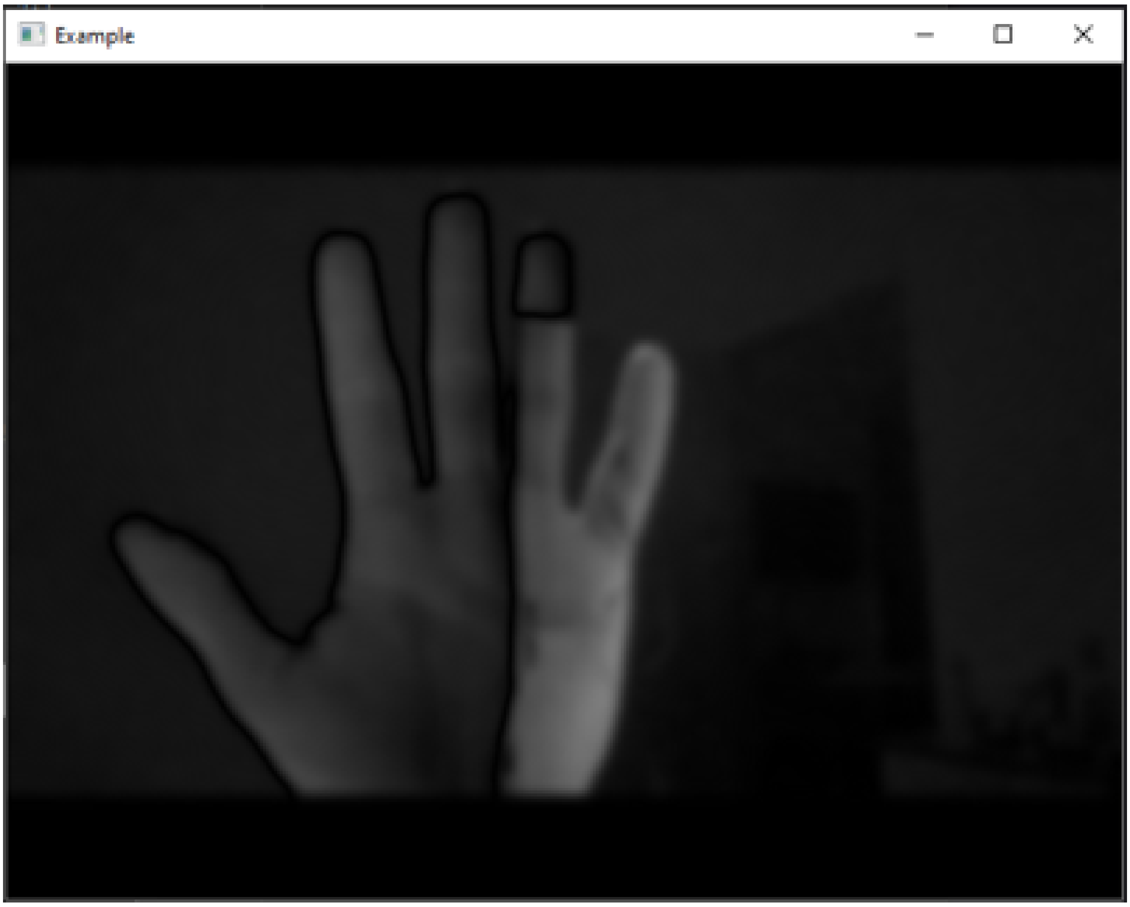


Figure 4.13: Imshow example

4.5 Winsound

The [11]winsound module provides access to the basic sound-playing machinery provided by Windows platforms. It includes functions and several constants. The function used for generationg sound is the "Beep" function. This function takes as arguments the frequency and the duration. The frequency parameter specifies frequency, in hertz, of the sound, and must be in the range 37 through 32,767. The duration parameter specifies the number of milliseconds the sound should last. If the system is not able to beep the speaker, RuntimeError is raised. For this function we also need to includ a library called winsound. However Winsound is not a good soultion as it is a "synchronous" function. So as the sound is played the video starts lagging and even if we stop the application the sounds continue to play on.

4.6 Pygame

We can use another library called pygame which has a "mixer". This plays the sounds asynchronously. This is the perfect solution for us. In order to create a beep effect we need to download a [18]wav file. To set up the mixer we must use:

```
pygame.mixer.init()
pygame.mixer.music.load(open('beep.wav'))
```

The file must be in the same folder as the python file. To play the sound we use the following function.

```
pygame.mixer.music.play()
```

Chapter 5

Code

5.1 HTML

```
<html>
  <head>
    <title>Live footage</title>
  </head>
  <body>
    <h1>Live footage</h1>
    <img src="{{ url_for('video_feed') }}">
  </body>
</html>
```

5.2 Python

```
#!/usr/bin/env python
from flask import Flask, render_template, Response

import cv2
import motdec
import imutils
import threading
import pygame
import time
frequency=2500
duration=5
pygame.mixer.init()
pygame.mixer.music.load(open('beep.wav'))
app = Flask(__name__)


@app.route('/')
def index():
    """Streaming page."""
    return render_template('index.html')

def gen():
    videocap = cv2.VideoCapture(0)
    basef = None
    hasrect=False
    beeped=0
    frame = videocap.read()[1]
    time.sleep(3)

    while True:
        frame = videocap.read()[1]
        text = ’No movement detected’
        beeped += 1
        greayscalef = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        gaussian = cv2.GaussianBlur(greayscalef, (21, 21), 0)

        if basef is None:
            basef = gaussian
        else:
            pass

        #frame = imutils.resize(frame, width=500)
        delta = cv2.absdiff(basef, gaussian)
        thresh = cv2.threshold(delta, 100, 255, cv2.THRESH_BINARY)[1]
        dilated = cv2.dilate(thresh, None, iterations=1)
        #  cnt = cv2.findContours
#the following line must continue the previous line
(dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[1]
        cnts, notused= cv2.findContours
#the following line must continue the previous line
(dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

        for i in cnts:
            beeped += 1
            if cv2.contourArea(i) > 400:
                (x, y, w, h) = cv2.boundingRect(i)
                if not hasrect:
                    cv2.rectangle
#the following line must continue the previous line
(frame, (x-50, y-50), (x + w-50, y + h-50), (255, 255, 255), 3)
                text = ’Movement detected’
                if beeped % 10 == 0:
                    pygame.mixer.music.play()
                hasrect=True
            else:
                pass
       # pygame.mixer.music.stop()
        font = cv2.FONT_ITALIC
        cv2.putText
#the following line must continue the previous line
(frame, ’In the room: %s’ % (text), (10, 20), cv2.FONT_ITALIC, 0.5, (0, 0, 255), 2)
        cv2.imshow(’Security Feed’, frame)
        cv2.imshow('Threshold(foreground mask)', dilated)
        cv2.imshow('Delta', delta)
        hasrect=False
        key = cv2.waitKey(1) & 0xFF
        if key == ord('q'):

            cv2.destroyAllWindows()
            break

        cv2.imwrite(filename='saved_img.jpg', img=frame)
        img_new = cv2.imread('saved_img.jpg', cv2.IMREAD_GRAYSCALE)
        yield (b'--frame\r\n'b'Content-Type: image/jpeg\r\n\r\n'
#the following line must continue the previous line
+ cv2.imencode('.jpg', img_new)[1].tobytes() + b'\r\n')
        #yield cv2.imencode('.jpg', img_new)[1].tobytes()

@app.route('/video_feed')
def video_feed():
    """Video streaming route. Put this in the src attribute of an img tag."""
    return Response(gen(), mimetype='multipart/x-mixed-replace; boundary=frame')


if __name__ == '__main__':
   # x = threading.Thread(target=thread_function)
    # x.start()
    app.run(host='0.0.0.0', debug=True, threaded=True)
```

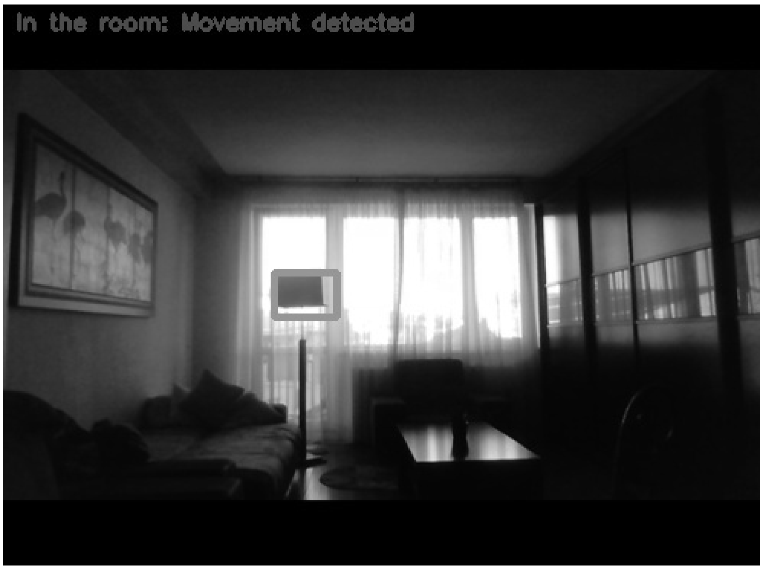
Chapter 6

Testing

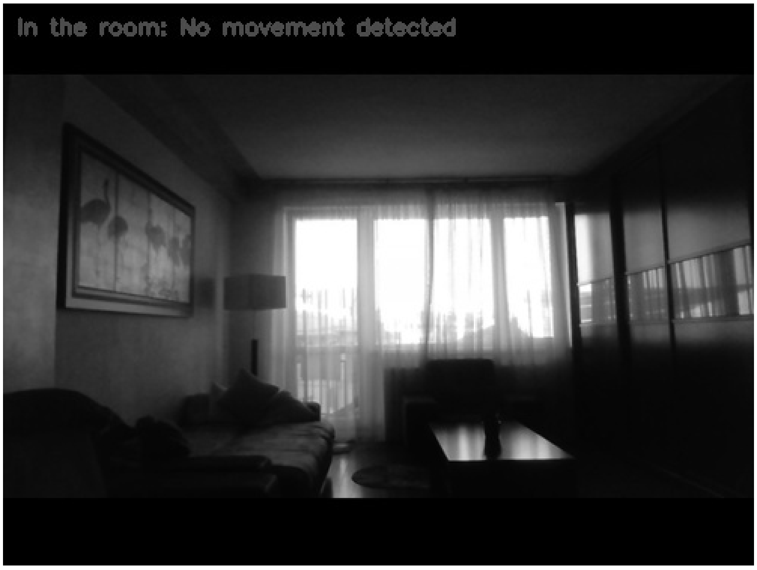
"Program testing can be used to show the presence of bugs, but never to show their absence!"
— Edsger W. Dijkstra
As this quote shows us that[15] testing can never prove the perfectness of a program but can help us get rid of the most bugs. Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.[1] Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use. In order to test my product I created some pecific inputs that could help me in further development and even to find bugs. In some steps of the development I noticed that light is a high "risk" factor in motion detection as it influences the results of the detection. Too much or insufficient light can affect the treshold. Multiple inputs were tested in order to find the correct treshold limits.

6.1 Examples

Figure 6.1 and 6.2 show the correct way the boundig rectangle is placed and also that the algorithm works correctly and is able to detect motion. After the first tests an error occured. When a movement appeared the rectangle which showed the movement had a positive offset from the movement. In order to face this problem this offset needed to be edited. This was done in the function which draws the rectangle. To make the function draw the rectangle correctly we added to the starting and ending positions -50 pixels. This made the function work correctly.



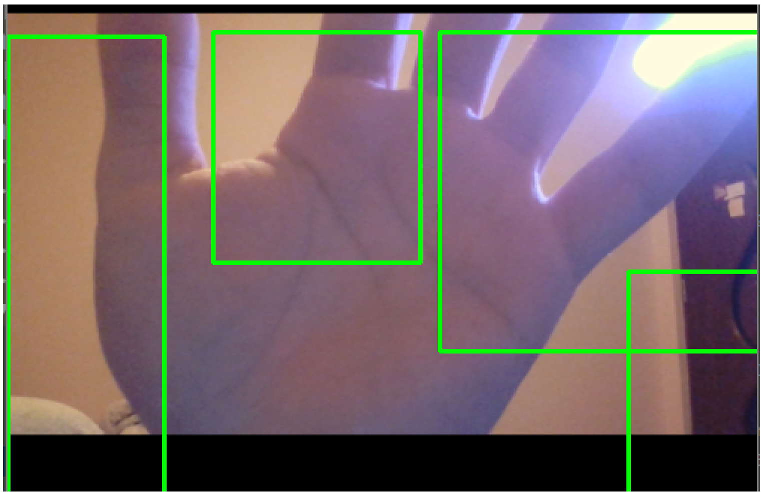
In the room: Movement detected



In the room: No movement detected

6.2 Refining the visuals

The aim of this project is to detect motion. However it doesn't matter how many objects are moving. This offers a possibility to develope the project. By adding a boolean variable we can detect whether a rectangle is added to the image or not. This way we can guarantee that only one rectangle will be added to the frame. This makes it look cleaner and also in some cases when the light conditions are not ideal for the detection and loads of rectangles appear on the frame are stopped. Only one rectangle will be visible.



6.3 Error cases

This motion detection system has 2 cases when the motion is not detected.

1. When the background and the moving object has similar color darkness (example: white and light yellow) the system fails to read the motion as there is not enough difference. However this can be changed by editiing the treshold limits. The treshold must be edited to match the environment.

2. An important factor in the efficiency of the system is the brightness of the environment. Too low or too bright environments cause errors in the detection as the system becomes uncapable of bringing results from the differentiation.

3. On the second start of the program, the system works perfectly, but on the first boot we can encounter errors. These errors are caused because of the webcam. It requires some time to ”heat up” before the first use. For this a trick is used. Before the main loop the ”sleep” function from the ”time” library is used in order to quarantee time for the webcam to heat up. It is the equivalent to the Bash shell’s sleep command. Before calling this function a frame is read from the webcam, to start it, then the sleep function allows it to heat up. This function has only one argumet: the number of seconds that the system will wait.

4. [16]Moving background error: background subtraction method for detecting moving foreground objects from a nonstationary background. While background subtraction has traditionally worked well for a stationary background, the same cannot be implied for a nonstationary viewing sensor. To a limited extent, motion compensation for the nonstationary background can be applied. However, in practice, it is difficult to realize the motion compensation to sufficient pixel accuracy, and the traditional background subtraction algorithm will fail for a moving scene. The problem is further complicated when the moving target to be detected/tracked is small, since the pixel error in motion that is compensating the background will subsume the small target.

Chapter 7

Developement possibilities

In order to further develop this project we have several possibilities:

1. Make the html page accept multiple images at the same time (like the threshold, or grayscale image).
2. Posibillity to see more steps of the algorithm (more steps show live).
3. Use of multiple sensors for motion detection like nightvision camera for the use of this system in any environment.
4. Changing the default camera to an exterior webcam.

Chapter 8

Conclusion

In conclusion, this reference explains the development of a “Webcam motion detection” system. Created in the Pyhton programming language using libraries like OpenCV, Flask, Imutils. The implementation is based on basic motion detection algorithms. This project may be used as fundamentals of a more complex system with higher accuracy and better performance. This project is great for learning new skills, learning the basics of image processing, the basics of Python. It also offers the possibility to learn about sockets and servers. Using the Flask library it prepares you for basic web development and the first steps of servers. It is highly recommended to read this paper understand the code and learn new skills!

Chapter 9

Bibliography

[1]https://www.python.org/downloads/
[2]https://www.geeksforgeeks.org/python-programming-examples/
[3]https://www.ijser.org/paper/Motion-detection-algorithm-based-on-Background-Subtraction.html/
[4]https://www.jetbrains.com/pycharm/
[5]https://www.jetbrains.com/pycharm/features/
[6]https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html
[7]https://computergraphics.stackexchange.com/questions/39/how-is-gaussian-blur-implemented
[8]https://www.mathworks.com/help/images/ref/imabsdiff.html
[9]https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html
[11]https://docs.python.org/2.6/library/winsound.html?highlight=sound
[12]https://www.w3schools.com/html/
[13]https://en.wikipedia.org/wiki/HTML
[14]https://docs.opencv.org/2.4/modules/highgui/doc/user_interface.html
[15]https://en.wikipedia.org/wiki/Software_testing
[16]https://docs.python.org/2/library/winsound.html
[17]https://link.springer.com/article/10.1007/s00138-002-0091-0
[18]https://www.soundjay.com/beep-sounds-1.html