

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Bakalářská práce**

**Domácí kamerový systém za použití Raspberry Pi**

**Ladislav HÁJEK**

**© 2019 ČZU v Praze**

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Ladislav Hájek

Informatika

Název práce

**Domácí kamerový systém za použití Raspberry Pi**

Název anglicky

**Household camera system using Raspberry Pi**

---

### Cíle práce

Cílem práce je vytvořit kamerový systém řízený počítačem Raspberry Pi, který by se mohl stát možnou alternativou pro monitorování domácnosti oproti běžným kamerovým systémům. Uživatel, který nakonfiguruje tento počítač spolu se všemi potřebnými komponenty, bude mít částečný přehled o dění ve své domácnosti, i když v ní nebude fyzicky přítomen.

### Metodika

Teoretická část bakalářské práce představuje počítač Raspberry Pi jako takový. Popisuje jednotlivý software a hardware, které jsou za účelem vytvoření kamerového systému zapotřebí. Metodika teoretické části bakalářské práce je z větší části založena na studiu a analýze jednotlivých informačních zdrojů.

Praktická část je věnována postupu celkového projektu. Nejdříve je popsána instalace a konfigurace jednotlivého software a hardware. Následně je kamerový systém podroben testování, které je zaměřeno na efektivitu objektové a pohybové detekce v jednotlivých modelových situacích, kvalitu pořizovaných snímků a složitost kamerového systému jako celku

Na základě syntézy poznatků z teoretické části práce a výsledků projektu z části praktické je formulován závěr bakalářské práce.

---

### **Doporučený rozsah**

#### **práce**

30-40stran

### **Klíčová slova**

Raspberry Pi, Camera Module V2, OpenCV, Python, Movement detection, Webcamera, Pushbullet, Face recognition, Body

---

### **Doporučené zdroje informací**

Monk, S. Raspberry Pi Cookbook. O'Reilly Media 2013 ISBN 978-1-449-36522-6

ROBINSON, A. -- COOK, M. *Raspberry Pi projects* .: Chichester, England: Wiley, 2014. ISBN 978-1-118-55543-9.

UPTON, E. -- GONER, J. -- HALFACREE, G. Raspberry Pi : uživatelská příručka (2.vydání). Brno: Computer ISBN 978-80-251-4819-8

---

### **Předběžný termín obhajoby**

2019/20 ZS – PEF (únor 2020)

### **Vedoucí práce**

Ing. Marek Pícka, Ph.D.

### **Garantující pracoviště**

Katedra informačního inženýrství

---

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

---

**Ing. Marín Pelikán, Ph.D.**

Děkan

V Praze dne 28. 11. 2019

## Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Domácí kamerový systém za použití Raspberry Pi" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.11.2019

---

## **Poděkování**

Rád bych touto cestou poděkoval Ing. Marku Píckovi za vedení bakalářské práce. Bez jeho věcných rad a připomínek by tato bakalářská práce nenabyla současné podoby.

# Domácí kamerový systém za použití Raspberry Pi

## Abstrakt

Tato bakalářská práce pojednává o domácím kamerovém systému, řízeném počítačem Raspberry Pi. V teoretické části je představen počítač Raspberry Pi spolu s jednotlivými komponenty, které jsou k sestrojení kamerového systému zapotřebí. Následně je popsán základní princip objektové detekce.

V praktické části je popsán proces zavádění a instalace kamerového systému spolu s testováním jeho efektivity. Raspberry Pi bude pomocí svého kamerového modulu či webkamery monitorovat daný objekt a při detekci obličeje narušitele je vyfotografován snímek, který je uložen jak do fyzické paměti počítače Raspberry Pi, tak odeslán na telefonní zařízení uživatele kamerového systému pomocí aplikace Pushbullet. Fotografie uživatele je pořízena i v případě, kdy je jeho pohyb zaznamenán externím senzorem pohybu. Ve chvíli, kdy je fotografie narušitele pořízena, zazní alarm. Po zaznění alarmu je do mobilního zařízení uživatele zasláno SMS upozornění, které jej informuje o narušiteli v domácnosti a odkáže jej do aplikace Pushbullet.

**Klíčová slova:** Raspberry Pi, Camera Module V2, OpenCV, Python, Movement detection, Webcamera, Pushbullet, Face recognition, Body recognition

# Household camera system using Raspberry Pi

## Abstract

This bachelor thesis deals with a household camera system controlled by a Raspberry Pi computer. In the theoretical part of the thesis is the Raspberry Pi computer along with components, which are required for the construction, introduced. Afterwards is the basic principle of object detection described.

In the practical part is the process of the camera system installation described along with testing its effectiveness. Raspberry Pi will monitor a given object using Camera Module or Webcam and when detecting intruder's face, the intruder will be photographed, and the photo stored in Raspberry Pi's physical memory and also sent to the camera system user's phone using mobile application Pushbullet. The intruder's photo is taken even when his movement is detected by the external motion sensor. The moment the intruder's photograph is shot, the alarm is sound. After the alarm is sound, the SMS notification is sent to the user's mobile device. It informs the user about household's intruder and refers to check the Pushbullet application.

**Keywords:** Raspberry Pi, Camera Module V2, OpenCV, Python, Movement detection, Webcam, Pushbullet, Face recognition, Body recognition, Piezo Buzzer

# Obsah

<b>1 Úvod</b>	<b>13</b>
<b>2 Cíl práce a metodika</b>	<b>14</b>
Cíl práce	14
Metodika	14
<b>3 Teoretická východiska</b>	<b>15</b>
3.1. Raspberry Pi	15
3.1.1. Raspberry Pi Model 3 B	16
3.1.1.1. Technická specifikace	16
3.2. Alternativy Raspberry Pi	16
3.2.1. Banana Pi	17
3.2.2. Orange Pi	17
3.3. Port GPIO	18
3.3.1. GPIO piny	18
3.3.2. VVC piny	18
3.3.3. Ground piny	18
3.3.4. I <sup>2</sup> C	18
3.3.5. UART	19
3.4. Všestranné možnosti použití počítače Raspberry Pi	19
3.5. Použití počítače Raspberry Pi jako kamerový systém	19
3.6. Použitý Software	20
3.6.1. OpenCV	20
3.6.2. NumPy	20
3.6.3. Imutils	20
3.6.4. Raspbian	20
3.6.5. Python	21
3.6.6. Pushbullet	22
3.6.7. VNC Viewer	22
3.7. Použitý hardware	23
3.7.1 Raspberry Pi Camera Module	23
3.7.1.1. Technická specifikace	23
3.7.2. Webkamera	24
3.7.2.1. Technická specifikace	24
3.7.3. Piezo Buzzer	24
3.7.4. PIR detektor pohybu	25
3.7.4.1. Princip detekce pohybu	25
3.8. Obličejová detekce	27



3.8.1.	Klasifikátor.....	27
3.8.2.	Boost.....	27
3.8.2.1.	Adaboost .....	28
3.8.3.	Detektor Viola Jones .....	28
3.8.3.1.	Kaskáda klasifikátorů.....	29
3.8.3.2.	HAAR příznaky .....	29
3.8.3.3.	Integrální obraz .....	31
3.8.4.	Shrnutí procesu obličejové detekce.....	32
3.8.5.	Intel klasifikátor .....	33
<b>4</b>	<b>Praktická část.....</b>	<b>34</b>
4.1.	Instalace Software.....	34
4.1.1.	Vytvoření virtuálního prostředí .....	34
4.1.2.	Instalace python balíčků .....	34
4.1.2.1.	Balíčky .....	35
4.1.3.	Instalace OpenCV .....	35
4.2.	Instalace Hardware .....	35
4.2.1.	Instalace kamerového modulu.....	35
4.2.1.1.	Vyfočení fotografie .....	36
4.2.1.2.	Nahrání videa .....	36
4.2.1.3.	Přehrání videa .....	36
4.2.2.	Zapojení Piezo Buzzer .....	36
4.2.3.	Zapojení PIR senzoru .....	37
4.3.	Programování.....	37
4.3.1.	IDE .....	37
4.3.1.1.	Naprogramování metody detekce pohybu .....	38
4.3.1.2.	Naprogramování zasílání SMS upozornění .....	38
4.3.2.	Pojmenování obrázků .....	38
4.3.3.	Výpis data do obrázku .....	39
4.3.4.	Pojmenování adresáře .....	39
4.3.5.	Uložení obrázků do souboru .....	40
4.3.6.	Poslání obrázku .....	40
4.3.7.	Poslání SMS upozornění .....	41
4.3.8.	VNC .....	42
4.4.	Testování .....	43
4.4.1.	Před testováním .....	43
4.4.1.1.	Situování Raspberry Pi .....	43
4.4.1.2.	Výběr klasifikátoru .....	43
4.4.2.	Modelová situace.....	43

4.4.2.1.	Snímání ve dne.....	44
4.4.2.2.	Snímání v noci .....	44
<b>5</b>	<b>Výsledky testování kamerového systému .....</b>	<b>46</b>
5.1.	Efektivnost klasifikátorů.....	46
5.2.	Kvalita pohybového senzoru .....	46
5.3.	Kvalita snímků.....	47
5.4.	Celková cena kamerového systému.....	47
<b>6</b>	<b>Diskuse.....</b>	<b>48</b>
6.1.	Návrhy k vylepšení.....	48
6.1.1.	Kamery.....	48
6.1.2.	Počet kamer.....	48
6.1.3.	MotionEyeOS .....	49
6.1.4.	Napájení .....	49
6.1.5.	Domácí mazlíčci .....	50
6.2.	Nabyté zkušenosti.....	51
<b>7</b>	<b>Závěr.....</b>	<b>52</b>
<b>8</b>	<b>Seznam zdrojů.....</b>	<b>53</b>
<b>9</b>	<b>Přílohy.....</b>	<b>58</b>
	Příloha A: Kód cam_system.py .....	58
	Příloha B: Výběr fotografií získaných při testování .....	62
	Příloha C: Ceník služeb Sistel International k 25.11.2019.....	64

## Seznam obrázků

Obrázek 1 Raspberry Pi Model 3 B .....	15
Obrázek 2 Banana Pi DDR3.....	17
Obrázek 3 Orange Pi One.....	17
Obrázek 4 GPIO piny Raspberry Pi 3 Model B .....	19
Obrázek 5 Raspbian s prostředím PIXEL .....	21
Obrázek 6 Pushbullet API.....	22
Obrázek 7 Camera Module V2.....	23
Obrázek 8 Webcamera .....	24
Obrázek 9 Piezo Buzzer .....	25
Obrázek 10 Detekce pohybu .....	26
Obrázek 11 PIR detektor pohybu HC-SR501 .....	26
Obrázek 12 Haarovy příznaky.....	30
Obrázek 13 Detekce obličeje Haarovými příznaky.....	30
Obrázek 14 Haarův příznak (vlevo) a hodnoty detekované v obraze (vpravo).....	31
Obrázek 15 Integrální obraz.....	32
Obrázek 16 Python balíčky .....	35
Obrázek 17 Zapojení Piezo Buzzer .....	37
Obrázek 18 Zapojení PIR senzoru .....	37
Obrázek 19 Kód metody pohybové detekce .....	38
Obrázek 20 Kód metody zasílání SMS upozornění .....	38
Obrázek 21 Názvy adresářů .....	40
Obrázek 22 Názvy snímků .....	40
Obrázek 23 Pushbullet chat.....	41
Obrázek 24 Testovací obrázek (USB kamera).....	44
Obrázek 25 Testovací obrázek (USB kamera).....	44
Obrázek 26 Testovací snímek (PiCamera).....	44
Obrázek 27 Testovací snímek (PiCamera).....	45
Obrázek 28 Testovací obrázek (PiCamera).....	45
Obrázek 29 Testovací snímek (PiCamera).....	45

## Seznam použitých tabulek

Tabulka 1 Technická specifikace Raspberry Pi Model 3 B .....	16
Tabulka 2 Technická specifikace Raspberry Pi Camera Module V2.....	23
Tabulka 3 Technická specifikace Webkamery.....	24
Tabulka 4 Technická specifikace PIR senzoru.....	26
Tabulka 5 Celková cena Raspberry Pi kamerového systému .....	47

## Seznam použitých zkratek

BYOD	Bring Your Own Device
DNS	Domain Name System
FM	Frequency Modulation
GPIO	General Purpose Input/Output
IDE	Integrated Development Environment
IoT	Internet of Things
IP Camera	Internet Protocol Camera
OpenCV	Open source Computer Vision
PIR	Passive Infrared Sensor
PIXEL	Pi Improved Xwindow Environment, Lightweight
RAM	Random Access Memory
SATA	Serial Advanced Technology Attachment
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SoC	System on Chip
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VPN	Virtual Private Network
XML	Extensible Markup Language

# 1 Úvod

V dnešní době přestávají být pojmy jako je IoT, VPN, DNS či BYOD velkou neznámou, nýbrž se tyto pojmy čím dál více stávají záležitostmi našeho každodenního života. Přítomnost nejrůznějších elektronických zařízení nás dennodenně provází takřka na každém kroku, jsou našimi pomocníky a společníky v soukromém i veřejném životě, mohou nám například: usnadňovat práci, využíváme je jako edukační či bezpečnostní prostředky.

Pojem krádež je však znám snad úplně každým. Krade se od nepaměti a lidé se odjakživa v obavě o svůj majetek snažili najít co nejúčinnější způsob, jak svůj majetek ochránit, aby předešli ztrátě.

Cílem této bakalářské práce je vytvoření kamerového systému za použití počítače Raspberry Pi, pro účely monitorování domácnosti s posouzením jeho účinnosti a konkurenceschopnosti oproti běžným kamerovým systémům.

K sepsání této práce mě vedla zvědavost po větším poznání počítače Raspberry Pi a myšlenka jeho využití k monitorovacím účelům, která by mohla oslovit takovou skupinu uživatelů, kteří se chystají zavést účinný kamerový systém s vynaložením co nejmenšího množství finančních prostředků.

## **2 Cíl práce a metodika**

### **Cíl práce**

Cílem práce je vytvoření kamerového systému řízeném počítačem Raspberry Pi, který by se mohl stát možnou alternativou pro monitorování domácnosti oproti běžným kamerovým systémům.

Uživatel Raspberry Pi, který nakonfiguruje tento počítač spolu se všemi potřebnými komponenty, pak bude mít možnost mít částečný přehled o dění ve své domácnosti, i když v ní nebude fyzicky přítomen.

Výsledkem práce je zhodnocení účinnosti vytvořeného kamerového systému spolu s návrhy možností jeho vylepšení.

### **Metodika**

V teoretické části je představen počítač Raspberry Pi jako takový. Dále je představen jednotlivý hardware a software, který je za účelem vytvoření kamerového systému zapotřebí.

Praktická část je věnována postupu celkového projektu. Nejdříve je popsána instalace a konfigurace jednotlivého hardware a software. Následně je kamerový systém podroben testování, které je zaměřeno na efektivitu objektové a pohybové detekce v jednotlivých modelových situacích, kvalitu pořizovaných snímků a složitost kamerového systému jako celku.

## 3 Teoretická východiska

### 3.1. Raspberry Pi

V roce 2006 se britský technický ředitel Eben Upton rozhodl aspirovat k vytvoření počítače velikosti platební karty, který by byl lehce dostupný, ovladatelný a přispíval ve výuce informatiky na britských školách.<sup>1</sup> Netrvalo dlouho a Eben Upton založil oficiální nadaci Raspberry Foundation, v roce 2012 vznikl první model počítače Raspberry Pi, který se vyvíjel na půdě univerzity v Cambridge.<sup>2</sup> V současnosti je na trhu k dostání nejnovější Raspberry Pi 4 Model B.<sup>3</sup>

Mezi hlavní změny, které v průběhu vývoje nastaly, můžeme zařadit: přidání ethernetového adaptéru RJ45, výkonnější procesor, vyšší operační paměť, větší počet USB portů či větší počet pinů na patici P1, které slouží k připojení periferních zařízení jako například snímačů pohybu, teplotních senzorů, senzorů zvuku apod.

Raspberry Pi je obecně vnímán jako počítač velikosti kreditní karty, který je na trhu k dostání přibližně za tisíc korun. K jeho užívání stačí vlastnit klávesnici, myš a monitor. Uprostřed všech základních desek se nachází polovodičový čip *Broadcom* (jedná se o tzv. System on a Chip), který vykonává jak obecné výpočty (CPU), tak vykreslování grafiky (GPU) či jiné funkce. Specifická je i paměť RAM, která je nestálá. Tato vlastnost se projevuje tím, že při přerušení napájení nebo vypnutí počítače Raspberry Pi se veškerý obsah paměti Raspberry Pi ztratí. V rámci projektu této bakalářské práce je použit Raspberry Pi Model 3 B.



Obrázek 1 Raspberry Pi Model 3 B

---

<sup>1</sup> HICKEY, Shane. This article is more than 5 years old The Raspberry Pi computer: how a bright British idea took flight. In: *Theguardian* [online]. 9.3.2014 [cit. 2019-11-05]. Dostupné z: <https://www.theguardian.com/technology/2014/mar/09/raspberry-pi-computer-eben-upton-cambridge>

<sup>2</sup> tamtéž

<sup>3</sup> Raspberry Pi 4 Model B - 4GB RAM. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/raspberry-pi-4b/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>

### 3.1.1. Raspberry Pi Model 3 B

#### 3.1.1.1. Technická specifikace

Tabulka 1 Technická specifikace Raspberry Pi Model 3 B

Parametr	Hodnota
Datum vydání	2016-02-01
Architektura	ARMv8-A (64/32-bit)
SoC	Broadcom BCM2837
Procesor (CPU)	1.2 GHz 64-bit quad-core ARM Cortex-A53
Video (GPU)	Broadcom VideoCore IV @ 250 MHz (BCM2837: 3D part of GPU @ 300 MHz, video part of GPU @ 400 MHz)
	OpenGL ES 2.0 (BCM2835, BCM2836: 24 GFLOPS / BCM2837: 28.8 GFLOPS)
	MPEG-2 and VC-1 (with license), 1080p30 H.264/MPEG-4 AVC high-profile decoder and encoder (BCM2837: 1080p60)
Paměť (SDRAM)	1 GB (sdílená s GPU)
USB 2.0 porty	4 (přes zabudovaný pětiportový USB hub; jeden USB port vnitřně propojen s ethernet portem)
Video vstup	15-pinový MIPI konektor kamerového rozhraní (CSI)
Video výstup	HDMI (rev 1.3 & 1.4), 14 HDMI rozlišení od 640×350 do 1920×1200 plus různé PAL a NTSC standardy, kompozitní video (PAL a NTSC) via 3,5 mm TRRS jack sdílený s výstupem zvuku, MIPI konektor rozhraní displeje (DSI)
Zvukový vstup	Přes I2C rozhraní
Zvukový výstup	Analogový (přes 3,5mm jack), digitální (přes HDMI), I2S
Interní paměť	MicroSDHC, USB Boot Mode
Integrovaná síť	10/100 Mbit/s Ethernet + WiFi 802.11n a Bluetooth 4.1
Nízkoúrovňové periferie	17× GPIO plus stejné funkce a HAT ID sběrnice
Jmenovitý výkon	1,34 A (6,7 W) při maximálním zatížení (připojený monitor, klávesnice, myš a WiFi)
	1,34 A (6,7 W) při maximálním zatížení (připojený monitor, klávesnice, myš a WiFi)
Rozměry	85,60 mm × 56,5 mm × 17
Hmotnost	45g

Zdroj: *RPishop.cz*, 2019

### 3.2. Alternativy Raspberry Pi

Raspberry Pi Ebena Uptona byl první jednodeskový mikropočítač, který byl díky své relativně nízké ceně pro širokou veřejnost finančně dostupný. Po vstupu Raspberry Pi na trh začaly vznikat další jednodeskové mikropočítače, kterými jsou například Orange Pi a Banana



Pi, jež v minulosti i současnosti dokážou i přes svou často vyšší cenu konkurovat Raspberry Pi svými komponenty a výkonem.

### 3.2.1. Banana Pi

Banana Pi se od Raspberry Pi na první pohled příliš neliší. To je dáno tím, že design Banana Pi byl počítačem Raspberry Pi silně ovlivněn. Dokonce jsou Banana Pi s Raspberry Pi navzájem kompatibilní, což se projevuje možností instalace Raspbianu, Debianu či Androidu na Banana Pi.

Navzdory silné inspiraci, dokázali výrobci vybavit počítače Banana Pi novinkami, jako je implementace rozhraní SATA pro případné napojení pevného disku, či integrovaný mikrofon, který se stal součástí základové desky počítače.<sup>4</sup>



*Obrázek 2 Banana Pi DDR3*

### 3.2.2. Orange Pi

Orange Pi se podobně jako Banana Pi od Raspberry Pi liší hned v několika aspektech, a to především ve výchozím prostředí operačního systému\* Armbian, který je založený na OS Debianu a OS Ubuntu. Instalace jiných OS jako Ubuntu, Android, Raspbian a Fedora je taktéž možná.<sup>5</sup>



*Obrázek 3 Orange Pi One*

---

\*dále operační systém taktéž OS

<sup>4</sup> Banana Pi Series Comparison. In: *Wiki.banana-pi* [online]. 8.7.2019 [cit. 2019-11-05]. Dostupné z: [http://wiki.banana-pi.org/Banana\\_Pi\\_Series\\_Comparison](http://wiki.banana-pi.org/Banana_Pi_Series_Comparison)

<sup>5</sup> Armbian. In: *Techrepublic* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.armbian.com/>

### 3.3. Port GPIO

Jednodeskový počítač Raspberry Pi disponuje patičí P1, na které se nachází čtyřicet programovatelných pinů. Na tyto piny lze pomocí speciálních propojovacích kabelů napojit externí zařízení jako například senzory pohybu, plynový senzor, laserový modul nebo radiový přijímač.<sup>6</sup>

Piny jsou ve dvou řadách číslovány střídavě, kde v dolní řadě jsou piny označovány lichými čísly a v horní řadě čísla sudými.<sup>7</sup> GPIO piny se dělí do různých typů v závislosti na jejich specifických vlastnostech (viz podkapitoly 3.3.1. – 3.3.5.).

#### 3.3.1. GPIO piny

Těchto pinů nalezneme na patiči P1 počítače Raspberry Pi Model 3 B největší množství. „Tyto piny lze přepínat mezi dvěma stavy, stav vysoký (kdy poskytují kladné napětí 3,3 V) a nízký (kdy odpovídají zemnění neboli napětí 0 V)“<sup>8</sup>. Tyto stavy odpovídají hodnotám 1 a 0 binární logiky. Díky jejich pomoci můžeme vypínat a zapínat jiné komponenty.<sup>9</sup>

#### 3.3.2. VVC piny

Dále se na patiči nachází piny, které připojeným zařízením poskytují konstantní napětí 3,3 V, nebo 5 V.<sup>10</sup>

#### 3.3.3. Ground piny

Ground piny se používají k uzemnění zařízení, externím zařízením tedy nepřinášejí žádné napětí.<sup>11</sup>

#### 3.3.4. I<sup>2</sup>C

I<sup>2</sup>C je sběrnice, která zprostředkovává komunikaci mezi více integrovanými obvody jako například mezi sběrnicí a Pi procesorem BCM2835, který je jádrem systému.<sup>12</sup>

---

<sup>6</sup> HEMMINGS, Megan. What is a Jumper Wire? In: *Sparkfuneducation* [online]. 30.1.2018 [cit. 2019-11-05]. Dostupné z: <http://blog.sparkfuneducation.com/what-is-jumper-wire>

<sup>7</sup> UPTON, E. -- GONER, J. -- HALFACREE, G. Raspberry Pi : uživatelská příručka (2.vydání). Brno: Computer Press, 2016. ISBN 978-80-251-4819-8, s. 218

<sup>8</sup> cit. UPTON, E. -- GONER, J. -- HALFACREE, G. Raspberry Pi : uživatelská příručka (2.vydání). Brno: Computer Press, 2016. ISBN 978-80-251-4819-8, s. 219

<sup>9</sup> tamtéž

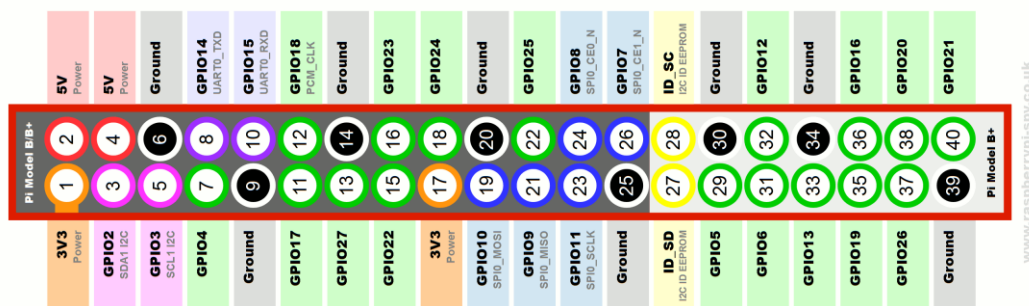
<sup>10</sup> Raspberry Pi GPIO. In: *Rpi.science.uoit* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpi.science.uoit.ca/lab/gpio/>

<sup>11</sup> tamtéž

<sup>12</sup> UPTON, E. -- GONER, J. -- HALFACREE, G; s. 219-220

### 3.3.5. UART

„Sériová sběrnice UART poskytuje jednoduché sériové rozhraní mezi dvěma vodiči. Pokud je v `cmdline.txt` nastaven sériový port, používá se jako port pro zprávy právě tato sériová sběrnice.“<sup>13</sup>



Obrázek 4 GPIO piny Raspberry Pi 3 Model B

## 3.4. Všestranné možnosti použití počítače Raspberry Pi

Počítač Raspberry Pi od svého základu vyniká tím, že může být používán jako klasický desktopový počítač. Ačkoliv nedosahuje tak vysokého výpočetního výkonu, je pro běžnou práci na internetu, poslech hudby či pro základní programování naprosto dostačující.

Použití Raspberry Pi je vskutku všestranné, jelikož na jeho V/V piny lze připojit širokou škálu externích zařízení, jako například senzory pohybu, plynové senzory, laserové moduly a radiové přijímače. Lze tak například provozovat tzv. chytrou domácnost (regulace teploty, automatické zalévání květin, automatické krmení domácích mazlíčků či chytrou televizi).<sup>14</sup>

Samo o sobě může počítač Raspberry Pi pracovat jako bezdrátový tiskový server, radiová FM stanice, Webserver nebo herní server.<sup>15</sup>

## 3.5. Použití počítače Raspberry Pi jako kamerový systém

Za účelem monitorování objektů lze k počítači Raspberry Pi připojit velké množství USB a IP kamer. Obvykle se nestává, že by tyto kamery nebyly s počítačem Raspberry Pi vzájemně kompatibilní. Jako alternativa k těmto kamerám se nabízí Raspberry Pi Camera Module, oficiální kamerový modul vytvořený pro Raspberry Pi přímo na míru.

<sup>13</sup> cit. UPTON, E. -- GONER, J. -- HALFACREE, G; s. 219

<sup>14</sup> FROMAGET, Patrick. 25 awesome Raspberry Pi project ideas at home. In: *Raspberrytips* [online]. [cit. 2019-11-05]. Dostupné z: <https://raspberrytips.com/raspberry-pi-projects-for-home/>

<sup>15</sup> Top Uses of Raspberry Pi. In: *Educba* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.educba.com/uses-of-raspberry-pi/>

Na piny počítače Raspberry Pi je pak možné připojit širokou škálu externích zařízení jako například pohybový sensor či spínač osvětlení, které potencionálně mohou efektivitu kamerového systému výrazně navýšit.

### 3.6. Použitý Software

V této kapitole je uveden nejpodstatnější software, který byl v rámci projektu použit.

#### 3.6.1. OpenCV

OpenCV je otevřená multiplatformní knihovna pro manipulaci s obrazem, která je zaměřena především na počítačové vidění a zpracování obrazu v reálném čase. Tuto knihovnu je možné využít z prostředí programovacích jazyků C či C++, a s generátorem rozhraní SWIG také programovací jazyk Python, který je v rámci projektu primárně využíván. V rámci projektu je používána OpenCV verze 4.12.<sup>16, 17</sup>

#### 3.6.2. NumPy

NumPy je knihovna, která slouží k práci s poli, při niž používá složité matematické postupy. Tyto postupy jsou v rámci bakalářské práce použity, a to především při zpracovávání objektové detekce. V projektu je využívána NumPy verze 1.16.5.<sup>18</sup>

#### 3.6.3. Imutils

Imutils je balíček funkcí, které usnadňují knihovnám jako OpenCV proces zpracování obrazu, především co se týče otázek změn velikosti obrazu, třízení obrysů, detekování hran či vizualizace dat. V rámci projektu je využívána Imutils verze 0.5.3.<sup>19</sup>

#### 3.6.4. Raspbian

Raspbian je OS optimalizovaný pro jednodeskový počítač Raspberry Pi vytvořený v roce 2012 společností Raspbery Foundation.<sup>20</sup> Jedná se o OS odvozený z populární linuxové distribuce Debian. V tomto projektu je využívána verze Raspbianu, která nese název *Raspbian Stretch*.

---

<sup>16</sup> OpenCV. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 13.4.2019 [cit. 2019-11-05]. Dostupné z: <https://cs.wikipedia.org/wiki/OpenCV>

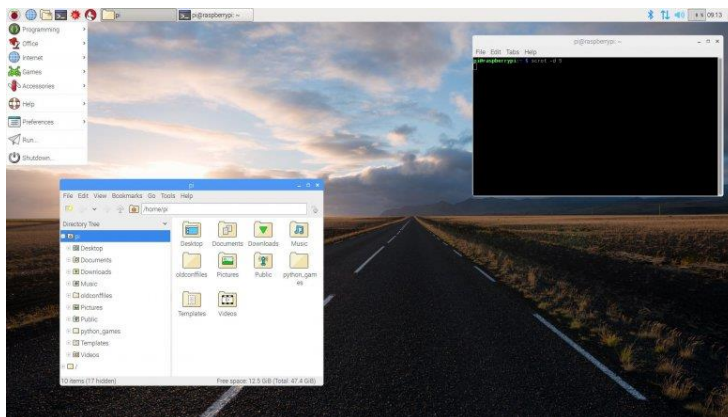
<sup>17</sup> Welcome to SWIG. In: *Swig* [online]. [cit. 2019-11-05]. Dostupné z: <http://www.swig.org/>

<sup>18</sup> NumPy Tutorial. In: *www.tutorialspoint.com* [online]. [cit. 2019-11-07]. Dostupné z: <https://www.tutorialspoint.com/numpy/index.htm>

<sup>19</sup> *Imutils 0.5.3* [online]. In: . [cit. 2019-11-07]. Dostupné z: <https://pypi.org/project/imutils/>

<sup>20</sup> Welcome to Raspbian. In: *Raspbian* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.raspbian.org/>

OS Raspbian od roku 2016 používá PIXEL jako desktopové prostředí. Pro uživatele, kteří s užíváním Linuxu a jeho distribucí nemají zkušenosti, je PIXEL mnohem přívětivější (na rozdíl od dříve čistě scrollovacího textu), jelikož přináší prostředí podobné populárnímu OS Windows.<sup>21</sup>



Obrázek 5 Raspbian s prostředím PIXEL

### 3.6.5. Python

Programovací jazyk Python je řazen mezi skriptovací jazyky. Navržen byl v roce 1991 Nizozemcem Guido von Rossumem. V rámci této bakalářské práce je používána Python verze 3.7.<sup>22</sup>

Jednou z hlavních předností programovacího jazyka Python je zejména jeho interpretační povaha, přehlednost a jednoduchý syntax. Dále disponuje širokou škálou rozšiřujících balíčků a knihoven jako například již zmíněný Imutils a NumPy. Jedná se o open-source, takže jej lze pořídit bezplatně. Může být rozšiřován dalšími programovacími jazyky jako například C nebo C++.<sup>23</sup>

Za slabou stránku Pythonu lze považovat jeho pomalejší rychlost, vyšší spotřebu paměti či nevhodnost jeho používání při vývoji mobilních aplikací.<sup>24</sup>

<sup>21</sup> BARNES, Russell. PIXEL: the brand new desktop for the Raspberry Pi. In: *Magpi.raspberrypi* [online]. 5.11.2016 [cit. 2019-11-05]. Dostupné z: <https://magpi.raspberrypi.org/articles/introducing-pixel>

<sup>22</sup> shivamsaraswat. Python Language Introduction. In: *Geeksforgereeks* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.geeksforgereeks.org/python-language-introduction/>

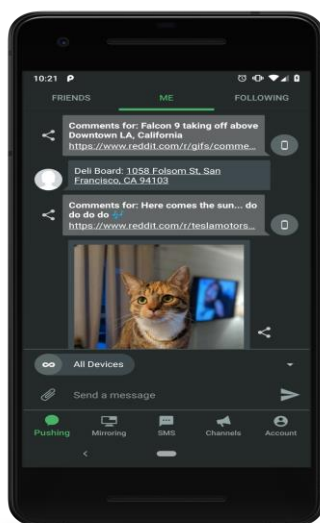
<sup>23</sup> Python Programming Language. In: *Educba* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.educba.com/python-programming-beginners-tutorial/>

<sup>24</sup> vartika02. Disadvantages of Python. In: *Geeksforgereek* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.geeksforgereeks.org/disadvantages-of-python/>

### 3.6.6. Pushbullet

Pushbullet je aplikace vytvořená společností Pushbullet, sídlící v San Franciscu. Tato aplikace je dostupná na OS Windows, Android a iOS. Podporován je webovými prohlížeči Safari, Opera, Mozilla Firefox a Google Chrome. Umožňuje sdílení datových souborů mezi mobilními zařízeními, osobními počítači a mobilním zařízením s osobním počítačem.<sup>25</sup>

V rámci projektu této bakalářské práce slouží Pushbullet primárně k zaslání kamerou vyfotografovaného snímku z počítače Raspberry Pi do uživatelského mobilního zařízení.



Obrázek 6 Pushbullet API

### 3.6.7. VNC Viewer

VNC Viewer je open-source program, jenž umožňuje vzdálené připojení jednoho počítače ke grafickému uživatelskému rozhraní druhého počítače prostřednictvím počítačové sítě. Vytvořen byl v roce 2002 na půdě cambridgeské univerzity.<sup>26</sup> VNC funguje v režimu klient-server, kdy server vytváří grafickou plochu a prostřednictvím sítě komunikuje s klientem, který si plochu zobrazuje. Pro komunikaci se používá protokol RFB, jehož cílem je minimalizovat objem přenášených dat mezi klientem a serverem, což vyniká zejména při pomalejších mobilních připojení.<sup>27</sup> VNC Viewer je dostupný na OS Windows, macOS, Linux, Raspbian, iOS, Android atd. V rámci projektu této bakalářské práce bude VNC Viewer sloužit

<sup>25</sup> About Us. In: *Pushbullet* [online]. [cit. 2019-11-30]. Dostupné z: <https://www.pushbullet.com/about>

<sup>26</sup> A Cambridge success story. In: *Realvnc* [online]. [cit. 2019-11-25]. Dostupné z: <https://www.realvnc.com/en/company-profile/>

<sup>27</sup> DVOŘÁK, Jiří. Co je to VNC a jak se s ním připojit k virtuálnímu serveru. In: *Coolhousing* [online]. 6.7.2017 [cit. 2019-11-25]. Dostupné z: <https://www.coolhousing.net/cz/co-je-to-vnc-a-jak-se-s-nim-pripojit-k-virtualnimu-serveru>

k připojení mobilního zařízení ke vzdálenému grafickému uživatelskému rozhraní počítače Raspberry Pi.

### 3.7. Použitý hardware

V této kapitole je představen jednotlivý hardware, který byl v rámci projektu použit.

#### 3.7.1 Raspberry Pi Camera Module

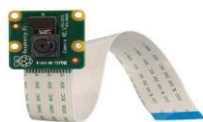
Hlavní komponent, bez kterého by nemohl projekt fungovat, je kamera. Pro účely projektu je použit Raspberry Pi Camera Module V2. Od běžných kamer se na první pohled liší svým širokým žilovým kabelem, jehož koncovka se zapojuje do speciálního CSI konektoru, který se nachází přímo na desce počítače Raspberry Pi. Cena této kamery se na trhu v současné době pohybuje okolo 800 Kč.<sup>28</sup>

##### 3.7.1.1. Technická specifikace

Tabulka 2 Technická specifikace Raspberry Pi Camera Module V2

Parametr	Hodnota
VideoFPS	1080p30, 720p/60
Fotky	3280x2464 px
Čip	8Mpx Sony IMX219 CCD
Konektor	CSI
Rozměry	25 × 20 × 9 mm
Noční vidění	NE
Verze	2
IR filtr	ANO
Montážních otvory	4
Zpětně kompatibilita s verzí 1	ANO

Zdroj: RPIshop.cz, 2019



Obrázek 7 Camera Module V2

<sup>28</sup> Raspberry Pi kamera V2. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/kamery/329-raspberry-pi-kamera-modul-v2.html>

### 3.7.2. Webkamera

Alternativou k Raspberry Pi kamerovému modulu je zvolena webová kamera Creative Live! Cam Sync HD. K počítači Raspberry Pi je zapojena do jednoho ze čtyř USB 2.0 portů Raspberry Pi Model 3 B.<sup>29</sup>

#### 3.7.2.1. Technická specifikace

Tabulka 3 Technická specifikace Webkamery

Parametr	Hodnota
Snímač	obrazový snímač s rozlišením HD 720p (1280 x 720)
Rozlišení videa	HD 720p (1280 x 720 pixelu)
Rozlišení obrazu	3,7 megapixelu**
Rychlost snímání	až 30 snímku/s při kvalitě HD 720p
Integrovaný mikrofon s funkcí potlačení hluku	
Funkce Plug & Chat v systému Windows, Mac OS X 10.5 a Linux 2.6	
Pevná ohnisková vzdálenost	
Víceúčelový podstavec	
Délka kabelu	1,5 m
USB 2.0 Hi-Speed	

Zdroj: *cs.creative.com*, 2019



Obrázek 8 Webcamera

### 3.7.3. Piezo Buzzer

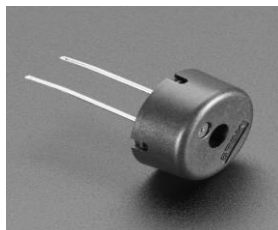
Piezo Buzzer je externí zvukové zařízení, které slouží pro vyluzování pískavých jednoduchých vysokých tonů. V rámci projektu kamerového systému slouží jako alarm. Dokáže vyvinout zvuk o síle mezi 2 kHz a 10kHz. Frekvenci zvuku je možné modulovat dle libosti ve zmíněném intervalu. Pro nejvýraznější tony je výrobcem doporučeno nastavit frekvenci na hladinu okolo 4 kHz.<sup>30</sup>

<sup>29</sup> Creative Live! Cam Sync HD. In: *Creative* [online]. [cit. 2019-11-05]. Dostupné z: <https://cs.creative.com/p/web-cameras/live-cam-sync-hd>

<sup>30</sup> Piezo Buzzer - PS1240. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/bzucaky/813-piezo-buzzer-ps1240.html>



Jedná se o buzzer pasivní. To znamená, že neprodukuje zvuk automaticky sám, ale k jeho vytvoření vyžaduje vstupní kmitavý signál.<sup>31</sup>



Obrázek 9 Piezo Buzzer

#### 3.7.4. PIR detektor pohybu

PIR (Passive infrared sensor) detektor pohybu HC-SR501 je externí zařízení. Disponuje třípinovým konektorem (Ground, VCC, GPIO pin), který se pomocí propojovacích kabelů napojuje na totožné piny počítače Raspberry Pi.

##### 3.7.4.1. Princip detekce pohybu

Detekce probíhá na bázi emitování tepelné energie ve formě infračerveného záření, kterým vyzařují objekty, jejichž teplota je vyšší než hodnota absolutní nuly (-273,15 °C). Čím teplejší objekt je, tím větším množstvím infračerveného záření vyzařuje.<sup>32</sup>

PIR detektor pohybu byl vytvořen za účelem detekování infračerveného záření pomocí infračerveného senzoru a Frenselovy čočky, která zaostřuje infračervené signály na infračervený senzor.<sup>33</sup>

Infračervený senzor disponuje dvěma pravoúhlými drážkami, vyrobených z materiálu, který umožňuje infračervenému záření pronikat skrz. Za těmito drážkami se nachází dvě oddělené infračervené senzorové elektrody. Jedna z těchto elektrod slouží k produkci pozitivního výstupu, druhá k produkci výstupu negativního. Důvodem produkce těchto výstupů je hledání změny v hodnotách úrovní infračervené intenzity, nikoliv infračervené intenzity úrovní okolí.<sup>34</sup>

---

<sup>31</sup> Passive Buzzer. In: *Addicore* [online]. [cit. 2019-11-07]. Dostupné z: <https://www.addicore.com/Passive-Buzzer-p/ad319.htm>

<sup>32</sup> How HC-SR501 PIR Sensor Works: Interface It With Arduino. In: *Lastminuteengineers* [online]. [cit. 2019-11-05]. Dostupné z: <https://lastminuteengineers.com/pir-sensor-arduino-tutorial/>

<sup>33</sup> tamtéž

<sup>34</sup> tamtéž

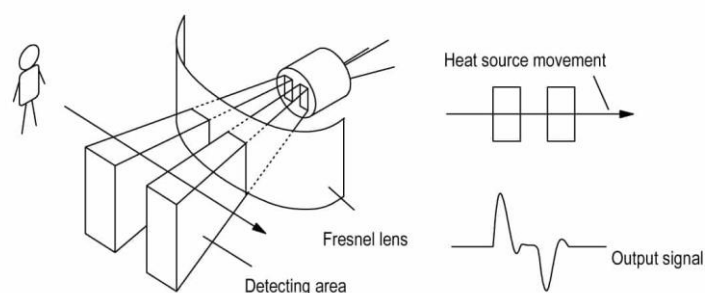
Obě elektrody jsou zapojeny tak, že se navzájem ruší. Pokud jedna polovina uvidí více, nebo méně infračerveného záření než polovina druhá, bude výstup kolísat a nabývat hodnot High, nebo Low.<sup>35</sup>

### 3.7.4.2. Technická specifikace PIR senzoru

Tabulka 4 Technická specifikace PIR senzoru

Parametr	Hodnota
Napájecí napětí	5-20V
Doba sepnutí	5s - 200s
Dosah detekce	cca 7m
Úhel detekce	menší než 120°
Klidový proud	méně než 50uA
Pracovní teplota	-15°C - +70°C
Rozměr	32x24x24mm

Zdoj: *laskarduino.cz*, 2019



Obrázek 10 Detekce pohybu



Obrázek 11 PIR detektor pohybu HC-SR501

<sup>35</sup> How HC-SR501 PIR Sensor Works: Interface It With Arduino. In: *Lastminuteengineers* [online]. [cit. 2019-11-05]. Dostupné z: <https://lastminuteengineers.com/pir-sensor-arduino-tutorial/>

### 3.8. Obličejová detekce

Obličejová detekce je problematika spadající do oblasti počítačového vidění. Za desetiletí své existence prošla rozsáhlým vývojem od počátečních manuálních metod klasifikace obličejů, až po dnešní automatické metody. V současné době se s obličejovou detekcí, potažmo obličejovou rekognicí, setkáváme například v rámci biometrických aplikací (autentizace uživatele při odemykání chytrých telefonů) či bezpečnostních systémech (bezpečnostní kontroly na letištích).<sup>36</sup>

#### 3.8.1. Klasifikátor

V rámci obličejové detekce jsou používány takzvané klasifikátory. Úkolem klasifikátorů je rozdělování vstupních dat s hodnotami atributů (rysů) do výstupních tříd. Vstupními daty (souborem trénovaných dat) je obsáhlý soubor obrázků, ve kterých se lidský obličej nachází (pozitivní vzorek), nebo nenachází (negativní vzorek). Klasifikátor disponující těmito daty podstupuje procesem *trénování*, při kterém je na základě vyhledávání rysů lidského obličeje v obraze rozhodnuto, zdali je obraz souboru trénovaných dat pozitivní, nebo negativní. Požadovaná výstupní třída je v tomto případě lidský obličej.<sup>37</sup>

Při detekci lidských obličejů jsou jednotlivé kamerou zaznamenané snímky analyzovány. Na jejich základě je hledána potencionální shoda rysů pořízených snímků s rysy snímků pozitivních vzorků souboru trénovaných dat. Jedna z aplikací knihovny OpenCV (*opencv\_visualisation*) je schopna zviditelnit, které rysy byly v jednotlivých snímcích nalezeny.<sup>38</sup>

#### 3.8.2. Boost

Metoda Boost slouží k sekvenčnímu vylepšování (trénování) přesnosti méně účinných klasifikátorů, které jsou označovány jako *weak learners*. Za weak learner (slabý žák) je považován takový klasifikátor, jenž klasifikuje data modelu s přesností menší než 50 % a který je vytvářen výběrem vzorků z trénovací množiny.<sup>39</sup>

---

<sup>36</sup> DAVIS WEST, Jess. A BRIEF HISTORY OF FACE RECOGNITION. In: *Facefirst* [online]. 1.8.2017 [cit. 2019-11-09]. Dostupné z: <https://www.facefirst.com/blog/brief-history-of-face-recognition-software/>

<sup>37</sup> *Cascade Classifier Training* [online]. In: Docs.opencv 21.11.2019 [cit. 2019-11-21]. Dostupné z: [https://docs.opencv.org/master/dc/d88/tutorial\\_traincascade.html](https://docs.opencv.org/master/dc/d88/tutorial_traincascade.html)

<sup>38</sup> tamtéž

<sup>39</sup> FABIEN, Maël. Boosting and AdaBoost clearly explained. In: *Towardsdatascience* [online]. 14.2.2019 [cit. 2019-11-05]. Dostupné z: <https://towardsdatascience.com/boosting-and-adaboost-clearly-explained-856e21152d3e>

Úspěšně vytrénovaní weak learners mohou být uspořádáni do jednoho koncového *strong learner*. Za strong learner (silný žák) je považován takový klasifikátor, jehož přesnost klasifikace se již může pohybovat nad mezí 50 %.<sup>40</sup>

### 3.8.2.1. Adaboost

Algoritmus Adaboost (adaptivní boosting) je řazen mezi jednu z nejznámějších variant metody Boost. K vytvoření Adaboost nejvíce přispěli Yoav Freund a Robert E. Schapire<sup>41</sup>, kteří za svou AdaBoost práci získali v roce 2003 prestižní Gödelovu cenu, která se od roku 1993 každoročně uděluje autorům prací v oblasti teoretické informatiky.<sup>42</sup>

Algoritmus metody boost probíhá tak, že base weak learner (*prvotní slabý žák*) nejdříve přečte vstupní data modelu a každému vzorku přiřadí rovnou váhu. Následně jsou data analyzována a dle výsledku analýzy je vytvořen jednoúrovňový (*optimální*) rozhodovací strom, podle nějž algoritmus posoudí, zdali nastala chybová predikce. Pokud k chybové predikci došlo, pak je právě tato chybová predikce sekvenčně přidělena nově přidanému slabému žáku, který se zaměřuje na nápravu chyby předchozí predikce, jenž nabývá vyšší váhy a je na ni dán větší důraz. Pokud chybová predikce nastala, pak je další predikci přidána váha nižší a je na ni dán menší důraz. Iterační přidávání slabých žáků trvá do té doby, než je chybovost predikce klasifikátoru snížena natolik, že jej můžeme považovat za silný klasifikátor.<sup>43</sup>

### 3.8.3. Detektor Viola Jones

Algoritmy detekce obličejů, které jsou založené na rysech, jsou rychlé, přesné a používají se již mnoho let. Mezi nejznámější techniky bezpochyby patří technika kaskádových klasifikátorů (viz kapitola 3.8.3.1), která je podrobně popsána v publikaci „*Rychlá detekce objektů užitím boostované kaskády funkcí*“ Paulem Violou a Michaelem Jonesem v roce 2001.<sup>44</sup>

---

<sup>40</sup> Saucecat. Boosting algorithm: AdaBoost. In: *Towardsdatascience* [online]. 29.4.2017 [cit. 2019-11-30]. Dostupné z: <https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c>

<sup>41</sup> GROVER, Puneet. *Clearing air around “Boosting”* [online]. In: *Towardsdatascience* . 18.3.2019 [cit. 2019-11-29]. Dostupné z: <https://towardsdatascience.com/clearing-air-around-boosting-28452bb63f9e#935b>

<sup>42</sup> Gödel Prize. In: *Sigact* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.sigact.org/prizes/g%C3%B6del.html>

<sup>43</sup> S. BRID, Rajesh. Introduction to Boosting. In: *Medium* [online]. 1.11.2018 [cit. 2019-11-09]. Dostupné z: <https://medium.com/greyatom/boosting-ce84639a805d>

<sup>44</sup> BROWNLEE, Jason. *How to Perform Face Detection with Deep Learning* [online]. In: . 3.6.2019 [cit. 2019-11-09]. Dostupné z: <https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/>

### 3.8.3.1. Kaskáda klasifikátorů

Algoritmus AdaBoost vytváří kaskádu klasifikátorů tak, že sekvenčním propojováním vícero slabých klasifikátorů vytváří silné klasifikátory, které jsou hierarchicky seřazovány do úrovní dle jejich komplexnosti. Každou jednotlivou úroveň kaskády lze tedy považovat za jeden ucelený silný klasifikátor. Počáteční úroveň kaskády disponuje malým množstvím slabých klasifikátorů s tím, že další úroveň kaskády dědí všechny slabé klasifikátory úrovně předešlé a navíc nabývá dalšího slabého, nebo většího množství slabých klasifikátorů. Čím je úroveň kaskády vyšší, tím je silný klasifikátor daného stupně přesnější, jelikož se za chodu učí eliminovat ty oblasti, na kterých se obličej nenachází a více se soustředit na oblasti, na kterých by se lidský obličej nacházet mohl.<sup>45</sup>

V praxi jsou jednotlivé kamerové snímky prohledávány pomocí suboken. Úrovně kaskády na základě polohy subokna posuzují, zdali se v sobokně lidský obličej nachází, či nikoliv. Pokud úroveň kaskády usoudí, že se subokno ve snímku nachází v oblasti s obličejem, pak je detekce provedena. Pokud úroveň kaskády usoudí, že se subokno v oblasti s obličejem nenachází, pak je subokno posunuto do jiné oblasti snímku a další pokus o detekci je řízen další úrovní kaskády.<sup>46</sup>

Implementace kaskádových klasifikátorů je používána v rámci knihovny OpenCV, která umožňuje výběr předškoleného modelu detekce obličeje a jeho možného trénování dle vlastního datového souboru.<sup>47</sup>

### 3.8.3.2. HAAR příznaky

Publikace: „*Rychlá detekce objektů užitím boostované kaskády funkcí*“<sup>48</sup> Paula Violy a Michaela Jonese představila množinu rysů nazývaných jako *Haarovy příznaky*, jež jsou neměnnou součástí Viola Jones algoritmu. Rozlišovány jsou tři základní typy: „edge features“ (pro detekci rohů), „line features“ (pro detekci čar) a „four rectangle features“ (pro detekci šikmých čar).<sup>49</sup>

---

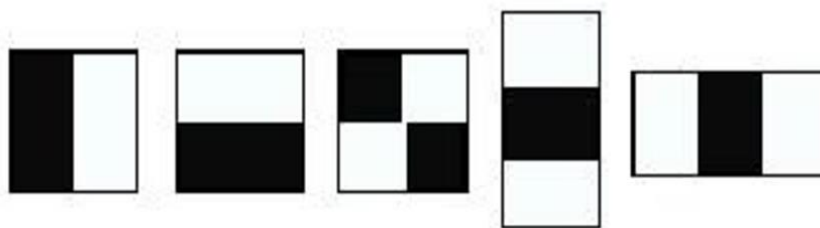
<sup>45</sup> BERGER, Will. Deep Learning Haar Cascade Explained. In: *Willberger* [online]. [cit. 2019-11-28]. Dostupné z: <http://www.willberger.org/cascade-haar-explained/>

<sup>46</sup> tamtéž

<sup>47</sup> BROWNLEE, Jason. *How to Perform Face Detection with Deep Learning* [online]. In: *Machinelearningmastery*. 3.6.2019 [cit. 2019-11-09]. Dostupné z: <https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/>

<sup>48</sup> tamtéž

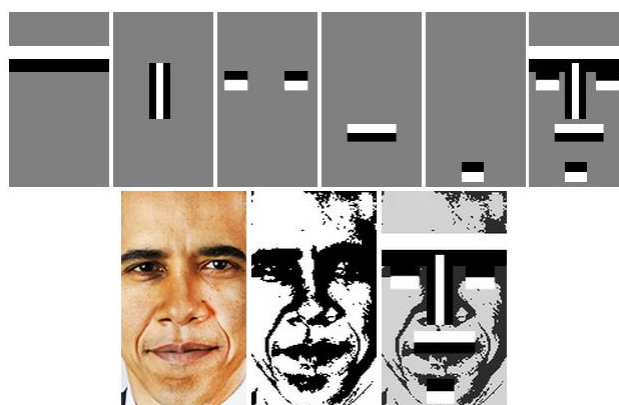
<sup>49</sup> Global Software Support. Face Detection with OpenCV – Computer Vision. In: *Globalsoftwaresupport* [online]. 24.2.2018 [cit. 2019-11-10]. Dostupné z: <https://www.globalsoftwaresupport.com/face-detection-opencv-computer-vision/>



Obrázek 12 Haarovy příznaky

Všechny lidské obličej se vyznačují podobnými rysy. Například oblast očí je obecně tmavší nežli horní líce a nosní můstek světlejší než oblast očí. Umístění a velikost očí, nosu a úst či vzdálenost mezi nimi je též specifická.<sup>50</sup>

Haarovy příznaky primárně slouží k detekci čar a hran obrazu, u kterého chceme určit, zdali se jedná o lidský obličej, či nikoliv. Prokládáním do greyscale konvertovaného obrazu jednotlivými haarovými příznaky můžeme jednotlivé specifické rysy lidského obličej odhalit, pakliže je těchto odhalení dostatečně velké množství, tak můžeme obrázek klasifikovat jako obrázek, na němž se lidský obličej nachází.<sup>51</sup>



Obrázek 13 Detekce obličej Haarovými příznaky

Proces prokládání obrazu haarovými příznaky a odhalování rysů lidského obličej probíhá za pomoci porovnávání hodnot intenzity jednotlivých pixelů příznaku s hodnotami intenzity pixelů greyscale obrazu. Tento proces provádí Viola Jones algoritmus.<sup>52</sup>

<sup>50</sup> Global Software Support. Face Detection with OpenCV – Computer Vision. In: *Globalsoftwaresupport* [online]. 24.2.2018 [cit. 2019-11-10]. Dostupné z: <https://www.globalsoftwaresupport.com/face-detection-opencv-computer-vision/>

<sup>51</sup> tamtéž

<sup>52</sup> tamtéž

1	1	0	0	0,7	0,6	0,4	0,2
1	1	0	0	0,6	0,7	0,2	0,1
1	1	0	0	0,8	0,7	0,4	0,1
1	1	0	0	0,7	0,8	0,2	0,2

Obrázek 14 Haarův příznak (vlevo) a hodnoty detekované v obraze (vpravo)

Černé pixely haarova příznaku nabývají hodnoty 1, bílé pixely hodnoty 0. Jednotlivé odstíny šedi nabývají hodnot z intervalu  $\langle 0,01; 0,99 \rangle$ .<sup>53</sup>

Haarův příznak je nalezen ve chvíli, kdy je výsledná hodnota Viola Jones algoritmem provedeného porovnání rovna 1. Tento případ však nemůže reálně plně nastat, jelikož fotografie, ať už v greyscale, či nikoliv, nenabývají tak ostře černých a bílých barev, přesto se může výsledná hodnota k hodnotě 1 přibližovat. Čím blíže k hodnotě 1 výsledná hodnota je, tím je větší šance, že byl v obrázku haarův příznak nalezen.<sup>54</sup>

Výsledná hodnota se počítá rozdílem průměru sumy hodnot intenzity tmavých oblastí a průměru sumy hodnot intenzity světlých oblastí. Pro názorný příklad lze zvolit hodnoty pravé tabulky obr. 14, jejíž výpočet vypadá následovně:  $0,7 - 0,225 = 0,475$ . Podobnost části obrázku a haarova příznaku tedy činí 47,5 %.<sup>55</sup>

### 3.8.3.3. Integrální obraz

Integrální obraz navazuje na problematiku hledání haarových příznaků v greyscale obrazu. Přichází totiž s řešením, které proces vyhledávání jednotlivých příznaků výrazně usnadňuje.<sup>56</sup>

Celkový proces je výpočetně zatěžován tím, že při prokládání greyscale obrazu haarovými rysy všech tří typů, různých velikostí a různých umístění v obrazu, je zapotřebí propočítávat obrovské množství numerických hodnot intenzity pixelů, aby bylo možné určit,

<sup>53</sup> Global Software Support. Face Detection with OpenCV – Computer Vision. In: *Globalsupport* [online]. 24.2.2018 [cit. 2019-11-10]. Dostupné z: <https://www.globalsupport.com/face-detection-opencv-computer-vision/>

<sup>54</sup> tamtéž

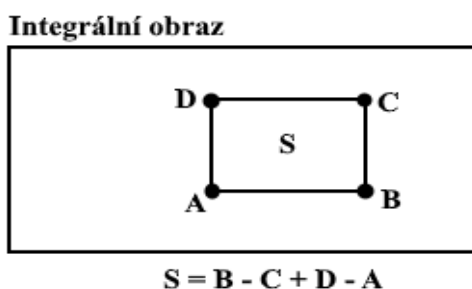
<sup>55</sup> tamtéž

<sup>56</sup> tamtéž

zdali se v obrazu haarův příznak nachází, či nikoliv. Počet propočítaných hodnot v rámci jednoho obrazu může nabývat řádu desetitisíců až statisíců.<sup>57</sup>

Integrální obraz je takový výsledek transformace hodnot pixelů výchozího obrazu, že každý jednotlivý pixel uvnitř integrálního obrazu představuje součet hodnot pixelů výchozího obrazu doleva a nahoru. Pravý dolní pixel obrazu tedy obsahuje součet všech pixelů obrazu.<sup>58</sup>

K výpočtu obdélníkových oblastí výchozího obrazu pomocí obrazu integrálního není nutné počítat se všemi hodnotami jeho pixelů, ale pouze se čtyřmi. Výsledná plocha oblasti integrálního obrazu se rovná sumě všech jednotlivých pixelů oblasti obrazu výchozího.<sup>59</sup>



*Obrázek 15 Integrální obraz*

### 3.8.4. Shrnutí procesu obličejové detekce

Detekce obličejů probíhá na základě analýzy obrazu pomocí klasifikátorů, které v obrazu hledají specifické rysy lidského obličejů pomocí haarových příznaků. Specifické rysy jsou hledány v souboru trénovaných dat, což jsou příklady snímků, na kterých se lidský obličej nachází (pozitivní snímky), a na kterých se lidský obličej nenachází (negativní snímky). Vytrénované klasifikátory pak na základě obrazové analýzy poznají, zdali se na fotografii lidský obličej nachází, či nikoliv. Tyto klasifikátory jsou trénovány algoritmem Adaboost<sup>60</sup> a dle jejich síly jsou hierarchicky shlukovány do Haarovy kaskády klasifikátorů. Nejsilnější klasifikátory takové kaskády dokážou přítomnost lidského obličejů rychle a spolehlivě rozpoznat. Hledání haarových příznaků v obraze se zvyšující se úrovní kaskády klasifikátorů stává čím dál více komplexní, proto je využíváno práce s integrálními obrazy, které svým jednoduchým výpočtem plochy v obraze proces hledání příznaků v každém jednotlivém snímku videa výrazně ulehčí.

---

<sup>57</sup> Global Software Support. Face Detection with OpenCV – Computer Vision. In: *Globalsoftwaresupport* [online]. 24.2.2018 [cit. 2019-11-10]. Dostupné z: <https://www.globalsoftwaresupport.com/face-detection-opencv-computer-vision/>

<sup>58</sup> tamtéž

<sup>59</sup> tamtéž

<sup>60</sup> BROWNLEE, Jason. *How to Perform Face Detection with Deep Learning* [online]. In: . 3.6.2019 [cit. 2019-11-09]. Dostupné z: <https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/>



### 3.8.5. Intel klasifikátor

V rámci projektu této bakalářské práce je během chodu kamerového systému využíván jeden ze tří klasifikátorů společnosti Intel, které jsou licencované (*For Open Source Computer Vision Library*<sup>61</sup>) a vytvořené v letech 2000 a 2004. Konkrétně se jedná o klasifikátor pro detekci obličeje, detekci horní části těla a detekci celého těla.

Jedná se o rozsáhlé XML soubory, které obsahují kaskádu klasifikátorů, disponující tisíci obdélníkovými rysy. Během snímání kamerového systému je přes kamerou zaznamenané snímky spouštěna kaskáda klasifikátorů, která prozkoumává jednotlivé části obrazů ve více měřítcích, aby mohla ve snímku rychle a efektivně detekovat tváře a těla nejrůznějších tvarů a velikostí.

---

<sup>61</sup> Smart-Security-Camera. *GitHub* [online]. 13.9.2017 [cit. 2019-11-29]. Dostupné z: <https://github.com/HackerShackOfficial/Smart-Security-Camera/tree/master/models>

## 4 Praktická část

Jednotlivé software a hardware komponenty projektu již byly představeny v teoretické části. Praktická část je zaměřena na konstrukci kamerového systému spolu s testováním jeho efektivity. V kapitole 4.1 je popsána instalace jednotlivého software, který je pro účely fungování kamerového systému zapotřebí. Následující kapitoly jsou zaměřeny na správné zakomponování hardware a naprogramování klíčových funkcí, kterými se kamerový systém vyznačuje. V závěru je kamerový systém podroben testování v jednotlivých modelových situacích. Výsledky testování jsou následně shrnuty a vyhodnoceny spolu s návrhy vylepšení projektu.

### 4.1. Instalace Software

Kapitola software pojednává o návodu k dosažení instalace programového vybavení projektu. Jedná se především o vytvoření virtuálního prostředí, do kterého budou jednotlivé knihovny a balíčky, potřebné ke správnému fungování projektu, instalovány.

#### 4.1.1. Vytvoření virtuálního prostředí

Ze všeho nejdříve je dobré vytvořit takzvané virtuální prostředí. Virtuální prostředí je takové prostředí, do kterého se budou instalovat knihovny, které budou aktivní pouze v rámci tohoto prostředí. Využití virtuálního prostředí přináší výhodu ve formě možnosti nainstalování knihoven tak, aby tyto knihovny nezasahovaly do systémového nastavení, nebo do ostatních virtuálních prostředí. Tím je možné například oddělit jednotlivé projekty, jelikož v každém prostředí mohou být nainstalované odlišné verze knihoven, které mohou, nebo nemohou být v rámci těchto projektů navzájem kompatibilní.

Virtuální prostředí vytvoříme sekvencí terminálových příkazů:

```
sudo apt-get install virtualenv (instalace virtuálního prostředí)  
cd Projekt/ (vstoupení do adresáře projektu)  
virtualenv env --python /usr/bin/python3 (vytvoření virtuálního prostředí)  
source env/bin/activate (aktivace virtuálního prostředí)  
deactivate (deaktivace virtuálního prostředí)
```

#### 4.1.2. Instalace python balíčků

V rámci projektu jsou používány Python balíčky jako například Imutils a NumPy. Pro instalaci potřebných Python balíčků stačí napsat následující sekvenci příkazů:

```
cd Projekt/ (vstoupení do adresáře projektu)  
source env/bin/activate (aktivace virtuálního prostředí)  
pip install numpy atd. (instalace jednotlivých balíčků)
```

### 4.1.2.1. Balíčky

Celkový seznam balíčků adresáře Projekt je vyobrazen zde:

```
(env) pi@raspberrypi:~/Projekt/env/lib/python3.7 $ cd site-packages
(env) pi@raspberrypi:~/Projekt/env/lib/python3.7/site-packages $ ls
certifi                               pushbullet
certifi-2019.9.11.dist-info           pushbullet.py-0.11.0.dist-info
cv2.so                                 __pycache__
easy_install.py                       python_magic-0.4.15.dist-info
chardet                                requests
chardet-3.0.4.dist-info               requests-2.22.0.dist-info
idna                                   RPi
idna-2.8.dist-info                    RPi.GPIO-0.7.0.dist-info
imutils                                setuptools
imutils-0.5.3.dist-info               setuptools-42.0.0.dist-info
magic.py                               six.py
numpy                                  six-1.13.0.dist-info
numpy-1.16.5.dist-info                 urllib3
picamera                               urllib3-1.25.7.dist-info
picamera-1.13.dist-info                websocket
pip                                    websocket_client-0.56.0.dist-info
pip-19.3.1.dist-info                  wheel
pkg_resources                         wheel-0.33.6.dist-info
pkg_resources-0.0.0.dist-info
(env) pi@raspberrypi:~/Projekt/env/lib/python3.7/site-packages $
```

Obrázek 16 Python balíčky

### 4.1.3. Instalace OpenCV

Instalace knihovny OpenCV je v rámci celého projektu tou časově nejnáročnější instalací. Nejdříve je zapotřebí z github stáhnout .zip soubor obsahující OpenCV, který po stažení zkopírujeme do výchozího adresáře /home/pi/. Následně nastává fáze kompilování OpenCV z programovacího jazyka C do Pythonu pomocí CMake (software pro překlad programů v různých OS). Tento proces překladu může trvat i několik hodin. Po úspěšné kompilaci může proběhnout konečná instalace OpenCV. V poslední řadě je třeba mezi sebou pomocí symbolického linku propojit nainstalované OpenCV a virtuální prostředí. V kódu projektu je knihovna OpenCV importována příkazem: `import cv2`.<sup>62</sup>

## 4.2. Instalace Hardware

V této kapitole je popsán postup instalace a konfigurace hlavních hardwarových komponentů.

### 4.2.1. Instalace kamerového modulu

Za účelem instalace kamerového modulu je zapotřebí zajistit oprávnění k používání kamery v konfiguračním nastavení Raspberry Pi. Před tím, než je počítač spuštěn je nutné zapojit kamerový modul do CSI konektoru. Po spuštění počítače stačí do terminálu zadat příkaz `sudo raspi-config`, pomocí kterého se zobrazí konfigurační nástroj. V rámci nástroje

<sup>62</sup> ROSEBROCK, Adrian. *Install OpenCV 4 on your Raspberry Pi* [online]. In: . 26.9.2018 [cit. 2019-11-22]. Dostupné z: <https://www.pyimagesearch.com/2018/09/26/install-opencv-4-on-your-raspberry-pi/>

zvolíme *Interfacing options*, následně *Camera*, a nakonec zvolíme možnost *Enable*. Poté je potřeba počítač rebootovat.

#### **4.2.1.1. Vyfocení fotografie**

Následně je doporučeno vyzkoušet funkčnost zapojené kamery zadáním jednoduchého příkazu do terminálu: *raspistill -o obrazek.jpg*. Tento příkaz nařizuje kameře vyfotografovat snímek s názvem *obrazek.jpg*. Výsledný snímek je uložen do výchozího adresáře.

#### **4.2.1.2. Nahrání videa**

Alternativní možností vyzkoušení funkčnosti zapojené kamery je zadání terminálového příkazu: *raspivid -o video.h264*. Tento příkaz nařizuje kameře nahrávání krátkého videa, které bude stejně tak jako při získávání fotografie uloženo do výchozího adresáře.

#### **4.2.1.3. Přehráání videa**

Pro přehráání nahraného videa je zapotřebné nainstalovat videopřehrávač. Jedním z takových videopřehrávačů je *Omxplayer*. K instalaci tohoto přehrávače je zapotřebí zadání příkazu: *sudo apt-get install omxplayer*. Po úspěšné instalaci stačí vstoupit do výchozího adresáře pomocí *cd* a námi pojmenované video spustit příkazem: *omxplayer video.h264*.

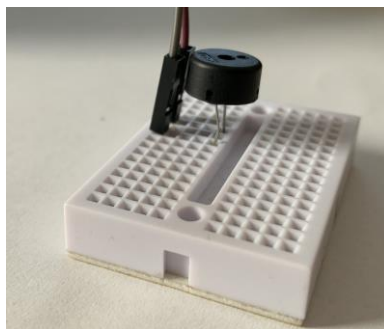
### **4.2.2. Zapojení Piezo Buzzer**

Pro správné zapojení Piezo Buzzer jsou použity dva propojovací kabely s male-female koncovkami a nepájivého pole.<sup>63</sup>Nejdříve jsou oba vývody Piezo opatrně vsunuty do dvou libovolných bodů nepájivého pole. Následně jsou propojovací kabely napojeny na specifické piny počítače Raspberry Pi.

Jeden z propojovacích kabelů je svou female koncovkou napojen na libovolný ground pin, druhý na libovolný GPIO pin počítače Raspberry Pi. Poté jsou male koncovky propojovacích kabelů vsunuty do bodů nepájivého pole tak, aby jejich vzájemná poloha byla rovnoběžná. (viz obr. 11)

---

<sup>63</sup> Barevné mini nepájivé pole. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/nepajiva-pole/73-427-barevne-mini-nepajive-pole.html#/63-barva-bila>



Obrázek 17 Zapojení Piezo Buzzer

### 4.2.3. Zapojení PIR senzoru

Pro správně zapojení PIR senzoru jsou použity tři propojovací kabely s female-female koncovkami. Těmito kabely jsou propojeny piny počítače Raspberry Pi s piny třípinového konektoru PIR senzoru. Typy jednotlivých pinů, umístěných na PIR senzoru jsou pevně dané a je nutné je propojit s totožnými pinovými typy počítače Raspberry Pi. (viz obr. 18)



Obrázek 18 Zapojení PIR senzoru

## 4.3. Programování

Po přípravě jednotlivého software a hardware je zapotřebí jednotlivé softwarové funkce a hardwarové komponenty naprogramovat. Kód projektu je k nalezení v příloze A.

### 4.3.1. IDE

Jako vývojové prostředí pro projekt bakalářské práce je zvolen *Thonny*. *Thonny* se pro mě s Pythonem začínající potřeby ukázal jako velice dobrá volba. Je jednoduchý, přehledný, disponuje jednoduchým debuggerem a poukazuje na chyby v syntaxi.<sup>64</sup>

---

<sup>64</sup> BARNES, Russel. Thonny on a Raspberry Pi: using the new Python IDE in Raspbian. In: *Dronebotworkshop* [online]. 5.11.2017 [cit. 2019-11-05]. Dostupné z: <https://magpi.raspberrypi.org/articles/thonny>

### 4.3.1.1. Naprogramování metody detekce pohybu

```
import RPi.GPIO as IO # Import GPIO modulu
class VideoCamera(object): # Třída pro v projektu používanou kameru
# Získávání obrázků
    def get_frame(self):
        frame = self.flip_if_needed(self.vs.read())
        return frame
# Horizontální převrácení kamery
video_camera = VideoCamera(flip=False) # U PiCamery: True, u Webkamery: False
IO.setwarnings(False) # Zablokovat varování
IO.setmode(IO.BCM) # Typ číslování pinů na patičce dle Broadcom SOC channel
motion_detector_pin = 4 # Číslo použitého GPIO pinu
IO.setup(motion_detector_pin, IO.IN) # Nastavit pin detektoru na vstup
SMS_UPDATE_INTERVAL = 600 # Interval posílání SMS upozornění je 600 vteřin. Po první detekci je SMS upozornění ihned posláno, další proběhne až po detekci,
# která bude časově vzdálena 600 vteřin od detekce puvodní
def check_for_movement(): # Snímání pohybu PIR senzorem
    lastSmsTime = 0 # Poslední čas posledního posílání SMS
    while True: # Nekonečný cyklus
        try: # Pokud kód proběhne správně = try, pokud nastane chyba = exception (Try-Catch)
            motion_detected = IO.input(motion_detector_pin) # Čti status pinu PIR senzoru a přidej mu proměnou "motion detected"
            if motion_detected: # Pokud je pohyb detekován
                print("Motion detected!") # Vypis upozornění
                frame = video_camera.get_frame() # Kamera poradí snímek
                object_processing(frame) # Pojmenování snímku, vložení data porizení snímku přímo do snímku, uložení snímku, posílání snímku Pushbullet
                buzz() # Zazní buzcauk
                print("Alarm!") # Vypis alarmu
                if time.time() - lastSmsTime > SMS_UPDATE_INTERVAL: # Pokud rozdíl aktuálního a posledního času odeslání SMS je vyšší než min. interval SMS
                    sendSMS() # Posli SMS upozornění
                    lastSmsTime = time.time() # Prepřepnutí času posledního posílání SMS
                else: # Jinak nečekej nic
                    pass
            else: # Pokud není zaznamenán pohyb, každou vteřinu vypisuj "No motion detected"
                print("No motion detected.")
                time.sleep(1)
        except: # Pokud nastane chyba, vypis chybovou zprávu
            logger.error("Error detecting motion: ", sys.exc_info()[0])
            raise
```

Obrázek 19 Kód metody pohybové detekce

### 4.3.1.2. Naprogramování zaslání SMS upozornění

```
# Import smtplib pro funkci zaslání zpravy
import smtplib
# Emailové balíčky, které budeme potřebovat
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Zaslání SMS upozornění z Gmail na mobilní zařízení
def sendSMS():
    sender = '?????????????????.com' # Gmail adresa
    sender_password = '?????????????' # Heslo emailové adresy odesílatele
    send_to_mobile = '+420yyyyxxxzzz@sms.cz.o2.com' # Telefonní číslo adresáta
    subject = 'Alarm!' # Podmet zpravy
    message = 'Zkontroluj Pushbullet' # Hlavní obsah zpravy

    msg = MIMEMultipart() # Container pro vnější zprávu emailu
    msg['From'] = sender # Od odesílatele
    msg['Subject'] = subject # Pro adresáta

    msg.attach(MIMEText(message, 'plain')) # Priložení zpravy
    server = smtplib.SMTP('smtp.gmail.com', 587) # Odeslání SMS pomocí (SMTP) Serveru (vyžaduje TLS, nebo SSL); 587 = port pro TLS/STARTTLS
    server.starttls() # TLS = Transport Layer Security (kryptografický protokol) pro bezpečnější posílání dat
    server.login(sender, sender_password) # Přihlášení se do Gmail
    text = msg.as_string() # Vložení textu zpravy jako textového řetězce
    server.sendmail(sender, send_to_mobile, text) # Zaslání mailu adresátovi
    server.quit() # Ukončení SMTP relace a ukončení připojení
```

Obrázek 20 Kód metody zaslání SMS upozornění

## 4.3.2. Pojmenování obrázků

Aby byl mezi kamerou vyfotografovanými snímky přehled, je zapotřebí, aby byly jednotlivé snímky navzájem odlišitelné. To je docíleno tím, že každý z obrázků obsahuje jedinečný název.

Tento jedinečný název je dán datem pořízení obrázku. Pro dostatečnou přehlednost je formát tohoto názvu stanoven na `%d%m%Y %H%M%S`, tedy na aktuální den, měsíc, rok, hodnu, minutu a sekundu. Toho je docíleno pomocí metody `strftime()`, která je pomocí objektů `date` nebo `datetime` schopná vracet datum jako textový řetězec tzv. *string*.

### 4.3.3. Výpis data do obrázku

Za účelem o něco větší přehlednosti bude kamerou vyfocený obrázek dále rozšířen o přidání data do obrázku samotného. To však není docíleno pomocí metody `strftime()`, nýbrž pomocí knihovny OpenCV a jejích funkcí. Konkrétně se jedná o funkci `putText()`.

Tato funkce obsahuje následující argumenty:

- Obrázek, do kterého bude text zapsán
- Konkrétní text, který bude vypsán
- Souřadnice umístění textu v obrázku
- Font písma
- Velikost písma
- Barva písma
- Tučnost písma
- „Type of line used“

### 4.3.4. Pojmenování adresáře

Vyfotografované snímky jsou ukládány do fyzické paměti paměťové karty microSD počítače Raspberry Pi, konkrétně jsou umístěny do podadresáře adresáře „*Projekt*“, nesoucí název „*photos*“. V podadresáři *photos* pak vznikají další podadresáře, které jsou pojmenovány podle data započatí snímání kamerového systému. Formát tohoto pojmenování je stanoven na `%Y%m%d_%H`, tedy na aktuální rok, měsíc, den a hodinu. Stejně tak jako tomu bylo při pojmenování obrázků, bylo tohoto formátu docíleno pomocí metody `strftime()`.

Všechny snímky pořízené v rámci jedné hodiny snímajícího kamerového systému jsou ukládány do jednoho uceleného adresáře. Pokud doba snímání kamerového systému přesáhne jednu hodnu, bude vytvořen adresář druhý, do které se budou ukládat další snímky.

```

pi@raspberrypi: ~/Projekt/photos
Soubor Upravit Karty Nápověda
pi@raspberrypi:~ $ cd Projekt
pi@raspberrypi:~/Projekt $ source env/bin/activate
(env) pi@raspberrypi:~/Projekt $ ls
buzzer_test.py count.py lul.jpg photos README.md
camera.py env main.py project.log requirements.txt
cam_system.py install_opencv.sh models __pycache__ sms.py
(env) pi@raspberrypi:~/Projekt $ cd photos
(env) pi@raspberrypi:~/Projekt/photos $ ls
20191124_23 20191125_16 20191125_20 20191128_20 20191129_12 20191130_11
20191125_00 20191125_18 20191125_21 20191128_23 20191129_15
20191125_09 20191125_19 20191126_20 20191129_11 20191129_17
(env) pi@raspberrypi:~/Projekt/photos $

```

Obrázek 21 Názvy adresářů

#### 4.3.5. Uložení obrázků do souboru

Pro uložení vyfoceného obrázku do souboru je, podobně jako v případě výpisu data do obrázku, použita funkce knihovny OpenCV. V tomto případě se jedná o metodu *imwrite()*. Tato funkce defaultně obsahuje následující argumenty:

- Název souboru (obsahuje formát obrázku, například *.jpg*)
- Obrázek, který bude uložen (např. *image*, *frame*)

```

pi@raspberrypi: ~/Projekt/photos/20191125_21
or Upravit Karty Nápověda
pberrypi:~ $ cd Projekt/
pberrypi:~/Projekt $ source env/bin/activate
pi@raspberrypi:~/Projekt $ cd photos
pi@raspberrypi:~/Projekt/photos $ ls
24_23 20191125_09 20191125_18 20191125_20
25_00 20191125_16 20191125_19 20191125_21
pi@raspberrypi:~/Projekt/photos $ cd 20191125_21
pi@raspberrypi:~/Projekt/photos/20191125_21 $ ls
25_210019.jpg 20191125_210742.jpg 20191125_212635.jpg
25_210036.jpg 20191125_210759.jpg 20191125_212653.jpg
25_210049.jpg 20191125_210812.jpg 20191125_212717.jpg
25_210103.jpg 20191125_211947.jpg 20191125_212718.jpg
25_210117.jpg 20191125_212014.jpg 20191125_212731.jpg
25_210131.jpg 20191125_212029.jpg 20191125_212818.jpg
25_210144.jpg 20191125_212248.jpg 20191125_212819.jpg
25_210158.jpg 20191125_212422.jpg 20191125_212835.jpg
25_210247.jpg 20191125_212430.jpg 20191125_212850.jpg
25_210347.jpg 20191125_212436.jpg 20191125_212944.jpg
25_210428.jpg 20191125_212529.jpg 20191125_213020.jpg
25_210447.jpg 20191125_212530.jpg 20191125_213022.jpg
25_210540.jpg 20191125_212603.jpg 20191125_214457.jpg
25_210553.jpg 20191125_212612.jpg
pi@raspberrypi:~/Projekt/photos/20191125_21 $

```

Obrázek 22 Názvy snímků

#### 4.3.6. Poslání obrázku

Proces zasílání snímků pomocí aplikace Pushbullet probíhá tak, že je vyfotografovaný snímek nejdříve odeslán aplikací Pushbullet na zpracování. Aplikace Pushbullet snímek zpracuje a odešle mobilnímu zařízení. Mobilní zařízení snímek přijme a dotazujícími příkazy jako *@snap* a *@check* se dotazuje na další potenciální příchozí snímky. Všechny requests (požadavky) a responses (odpovědi, odezvy) Pushbullet API jsou zapsány pomocí JSON.



Autentizace k uživatelskému účtu probíhá pomocí takzvaného *Token klíče*, který je pro každý Pushbullet účet jedinečný. Dále se Pushbullet vyznačuje svou *End-to-end encryption*, tedy že pouze uživatelé, kteří mezi sebou prostřednictvím Pushbullet komunikují, mohou číst své zprávy, nikoliv někdo neoprávněný. Zprávy jsou šifrovány odesílatelem, třetí strana (Pushbullet) nemá prostředky k jejich dešifrování a ukládá je šifrované. Příjemce (adresát zprávy) zašifrovaná data načte a sám je dešifruje.<sup>65</sup>

Na obrázku č. 21 je zobrazeno uživatelské prostředí aplikace Pushbullet. Konkrétně se jedná o sekci zasílání zpráv a fotografií v rámci jednoho Pushbullet účtu, ke kterému je přistupováno ze dvou zařízení – počítače Raspberry Pi a mobilního zařízení.

V rámci projektu je používána základní verze Pushbullet. Kapacita zasílaných souborů pro základní verzi je omezena na 25 MB a úložné místo na 2 GB. S verzí Pro, jejíž pořízení vyjde uživatele na 39,99 \$ ročně, či 4,99 \$ měsíčně, jsou omezení zmíněných kapacit výrazně navýšeny.<sup>66</sup>



Obrázek 23 Pushbullet chat

#### 4.3.7. Poslání SMS upozornění

Poté, co je kamerový snímek pořízen a odeslán na Pushbullet, je uživatel informován prostřednictvím SMS upozornění. Proces zasílání SMS upozornění je následující: po získaném snímku je z uživatelem zadané Gmail schránky odesláno SMS upozornění na

<sup>65</sup> Pushbullet API: End-to-End Encryption. In: *Pushbullet* [online]. [cit. 2019-11-25]. Dostupné z: <https://docs.pushbullet.com/>

<sup>66</sup> Pushbullet Pro. In: *Pushbullet* [online]. [cit. 2019-11-29]. Dostupné z: <https://www.pushbullet.com/pro>

telefonní číslo uživatele. Formát SMS upozornění je dán telefonním číslem a mobilním operátorem uživatele.<sup>67</sup>

Upozornění je uživateli poprvé zasláno po první pohybové, nebo obličejové/tělové detekci. Následně je zadán časový interval, po který nebude uživateli upozornění zasíláno. Jakmile interval skončí a objekt je opět detekován, je uživateli zasláno upozornění další. Obsah upozornění má za úkol uživatele informovat o detekovaném objektu a nasměrovat jej do aplikace Pushbullet, kde má možnost si snímek prohlédnout.

Nesporná výhodou tohoto způsobu informování je ta, že uživatel nemusí být svým mobilním zařízením připojen k internetu, aby se dozvěděl, že se v jeho domácnosti někdo nachází.

#### 4.3.8. VNC

Hlavní podmínkou pro úspěšné zajištění přístupu mobilního zařízení ke vzdálené ploše počítače Raspberry Pi je společná lokální síť. K tomu stačí znát IP adresu Raspberry Pi, povolit VNC v konfiguračním nastavení Raspberry Pi a nainstalovat VNC klient. Po spuštění VNC je nutné zadat adresu VNC serveru počítače Raspberry Pi.

Následně je uživatel vyzván, aby zadal přihlašovací údaje počítače Raspberry Pi. Po úspěšném zadání přihlašovacích údajů je uživateli mobilního zařízení schváleno oprávnění ke vstupu ke vzdálenému grafickému rozhraní počítače Raspberry Pi.<sup>68</sup>

Tohoto přístupu bude využíváno k příjemnějšímu ovládání kamerového systému. Uživatel kamerového systému bude mít možnost jednoduchého zapínání, či vypínání systému prostřednictvím svého mobilního telefonu.

Není tak nutná potřeba kamerový systém složitě spouštět skrze připojování, spouštění a odpojování Raspberry Pi od monitoru desktopového počítače, s jeho případným následným přemístěním na místo, odkud bude počítač monitorovat daný objekt.

Díky VNC stačí samotný počítač Raspberry Pi zapojit ke zdroji napájení a kamerový systém spustit zadáním terminálových příkazů prostřednictvím mobilního zařízení. Stejně tak lze za pomoci VNC činnost kamerového systému pohodlně ukončit.

---

<sup>67</sup> Formát adresy SMS upozornění u O2. In: *O2* [online]. [cit. 2019-11-27]. Dostupné z: <https://sms-o2.cz/sms-upozorneni-na-novy-mail-jak-si-ho-nastavit/>

<sup>68</sup> DVOŘÁK, Jiří. Co je to VNC a jak se s ním připojit k virtuálnímu serveru. In: *Coolhousing* [online]. 6.7.2017 [cit. 2019-11-25]. Dostupné z: <https://www.coolhousing.net/cz/co-je-to-vnc-a-jak-se-s-nim-pripojit-k-virtualnimu-serveru>

## **4.4. Testování**

Testování kamerového systému je zaměřeno na efektivitu objektové a pohybové detekce v jednotlivých modelových situacích, kvalitu pořizovaných snímků a složitost kamerového systému jako celku.

Před testováním byla stanovena hypotéza, kterou se pokusíme potvrdit, popřípadě vyvrátit. Vytyčená hypotéza zní následovně: *Raspberry Pi kamerový systém je konkurenceschopný oproti běžným kamerovým systémům.*

### **4.4.1. Před testováním**

Před začátkem testování kamerového systému je zapotřebí učinit několik klíčových rozhodnutí, která mají na průběh a testování projektu zásadní dopad. Jedná se především o otázky umístění Raspberry v prostoru objektu a výběr klasifikátoru.

#### **4.4.1.1. Situování Raspberry Pi**

K tomu, aby mohl být kamerový systém opravdu účinný je potřeba dbát na správné umístění počítače Raspberry Pi a všech jeho potřebných komponentů. Je vhodné jej v hlídaném objektu umístit tak, aby kamerový modul zabíral co největší úhel a sledoval klíčové body, jako vstupní dveře či majetek, jehož ztrátě chce uživatel kamerového systému předejít.

#### **4.4.1.2. Výběr klasifikátoru**

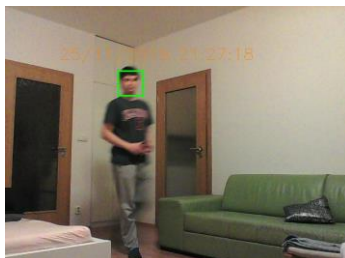
Správný výběr klasifikátoru je pro úspěšné detekování narušitele naprosto klíčový. Je nezbytné ho uzpůsobit několika faktorům, jako je například výšková poloha kamery v místnosti. V rámci projektu máme možnost použít jeden ze tří klasifikátorů (klasifikátor na rozpoznání obličeje, horní poloviny těla a celého těla). V procesu testování byl vždy zvolen právě jeden z těchto klasifikátorů.

### **4.4.2. Modelová situace**

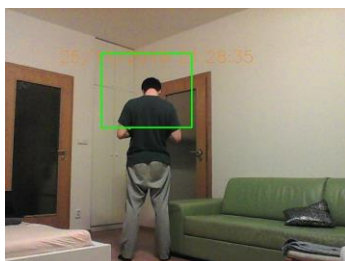
Modelové situace vychází z reálného prostředí mé domácnosti a popisuje takové situace, za kterých bude Raspberry Pi kamerový systém operovat, a které by mohly pravděpodobně nastat. Tyto situace nejsou striktně určeny pro domácnosti, lze je aplikovat i pro objekty menších skladů, či obchodů.

#### 4.4.2.1. Snímání ve dne

Jako první modelovou situaci jsem zvolil snímání kamerového systému za bílého dne. Díky tomu, že Raspberry Pi Camera Module V2 disponuje IR filtrem, mohou během dne vyfotografované snímky dosahovat vysoké kvality. Tato vysoká kvalita umožňuje účinnou detekci obličeje a horní části těla, jelikož je cíl snadno odlišitelný od svého okolí.



Obrázek 24 Testovací obrázek (USB kamera)



Obrázek 25 Testovací obrázek (USB kamera)

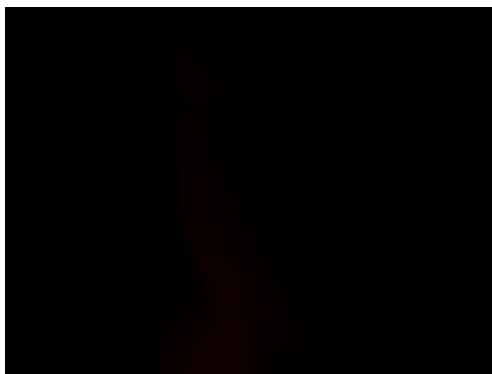


Obrázek 26 Testovací snímek (PiCamera)

#### 4.4.2.2. Snímání v noci

Snímání ve večerních hodinách se ukázalo být problematictější nežli snímání ve dne, jelikož za absence jakéhokoliv osvětlení zaznamenává kamerový modul nebo webkamera pouze tmou. To má za následek nemožnost detekovat obličej narušitele, jelikož nemá možnost jej odlišit od okolního prostředí. Pro úspěšnou detekci za tmy je zapotřebí přítomnost, alespoň malého množství přísunu světla.

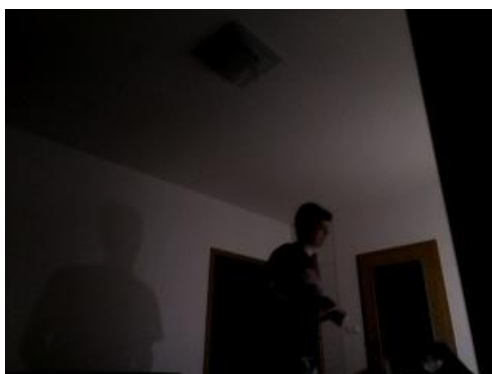
Díky senzoru pohybu je kamerový systém schopen pohyb narušitele zachytit a poskytnout fotografii. Tato fotografie však pravděpodobně k celkové identifikaci narušitele nebude kvůli temnému prostředí dostačující.



*Obrázek 27 Testovací snímek (PiCamera)*



*Obrázek 28 Testovací obrázek (PiCamera)*



*Obrázek 29 Testovací snímek (PiCamera)*

## **5 Výsledky testování kamerového systému**

### **5.1. Efektivnost klasifikátorů**

Raspberry Pi kamerový systém byl v rámci modelových situací podroben testování, během kterého bylo pořízeno několik fotografií narušitele (ukázkou podobných snímků naleznete v příloze B). Při pořizování těchto fotografií byla zjištěna některá omezení použitých klasifikátorů.

Jedno z omezení použitého klasifikátoru pro detekci obličeje tkví v jeho nemožnosti detekovat narušitele, který je ke kameře natočen z profilu, jelikož Viola Jones algoritmus pomocí haarových rysů vyhledává právě dvě oči, jedna ústa a jeden nos. Velmi často dochází k situacím, kdy mají narušitelé obličej zahalený v kukle či masce. Ani v tomto případě nebude algoritmus Viola Jones při detekci lidského obličeje úspěšný, jelikož ústa a nos bývají kuklou skrytá.

Použitý klasifikátor pro detekci horní poloviny lidského těla je v tomto ohledu účinnější. Narušitel je tímto klasifikátorem detekován i v případě, kdy mu obličej skrývá kukla. Dále může být narušitel detekován v situaci, kdy je ke kameře otočen zády. V případě, kdy je narušitel ke kameře natočen z profilu, tak k detekci nedojde. Jak klasifikátor pro detekci horní poloviny lidského těla, tak klasifikátor pro detekci celého lidského těla, se příliš nezaměřují na vyhledávání konkrétních vnitřních rysů lidského obličeje, nýbrž na tvaru siluety lidského těla.

Použit klasifikátor pro detekci celého těla se mi během testování nepodařilo, jelikož je zapotřebí, aby kamera operovala v takové výšce, ve které by celou postavu narušitele zachytila. Stejně jako je tomu u klasifikátoru pro detekci horní poloviny těla, je i u klasifikátoru pro detekci celého lidského těla nemožné detekovat narušitele, který je kameře natočen z profilu.

### **5.2. Kvalita pohybového senzoru**

Pohybový senzor částečně redukuje nedostatky použitých klasifikátorů. S jeho pomocí lze například detekovat pohyblivý cíl, který je ke kameře natočen zády, nebo z profilu. Během testování bylo zjištěno, že PIR senzor dokáže pohyblivý cíl spolehlivě detekovat na vzdálenost 5 metrů.

### 5.3. Kvalita snímků

Kvalitu pořizovaných snímků ovlivňují kamery použité v projektu. Jedná se o USB kameru CREATIVE a Raspberry Pi Camera Module V2. Snímky pořízené USB kamerou CREATIVE s 3,7 Mpx byly během testování pořízeny v rozlišení (640x480px), jejich velikost se pohybuje okolo 90 kB. Raspberry Pi kamerový modul V2 s 8 Mpx<sup>69</sup> během testování pořídil snímky s rozlišením až 320x240px, jejíž velikost se pohybuje okolo 20 kB.

Snímky těchto kamer jsou pro účely projektu kamerového systému dostačující, jelikož při denním snímání vyobrazují zachyceného narušitele v přijatelné kvalitě, při které jej lze dostatečně popsat. Disponováním IR filtru jsou však obě tyto kamery k identifikaci narušitele při nočním monitorování nedostatečné (viz obr. 25; 26).

### 5.4. Celková cena kamerového systému

Celková cena je získána sečtením cen všech jednotlivých hardwarových komponentů kamerového systému. Veškeré softwarové komponenty, výjimkou může být případná plná verze aplikace Pushbullet Pro (49,99\$/ročně, 4,99\$/měsíčně), byly získány bezplatně. Komponenty jako propojovací kabely či HDMI kabely jsem disponoval již před konstrukcí kamerového systému.

Tabulka 5 Celková cena Raspberry Pi kamerového systému

Produkt	Cena
Raspberry Pi Model 3 B	888 Kč
Raspberry Pi Camera Module V2	858 Kč
Piezo Buzzer	59 Kč
Nepájivé pole	69 Kč
Heatsink (passivní chladič)	99 Kč
PIR sensor	59 Kč
microUSB adaptér	264 Kč
microSD karta	247 Kč
<b>Celkem</b>	<b>2 543 Kč</b>

<sup>69</sup> RASPBERRY Pi Camera Module V2. In: *Alza* [online]. [cit. 2019-11-27]. Dostupné z: [https://www.alza.cz/raspberry-pi-camera-module-v2-d4256972.htm?kampan=adw1\\_prislusenstvi-pro-it-tv\\_pla\\_all\\_prislusenstvi-css\\_rozsirujici-karty\\_c\\_9062906\\_1o4\\_RK100c3&gclid=Cj0KCQiAt\\_PuBRDcARIsAMNIbDrYTH0-vH-2GIXbHf2QwLmiYbzNFGabJTtuaP7UA\\_nXUyAGO6qeBYQaAsAjEALw\\_wcB](https://www.alza.cz/raspberry-pi-camera-module-v2-d4256972.htm?kampan=adw1_prislusenstvi-pro-it-tv_pla_all_prislusenstvi-css_rozsirujici-karty_c_9062906_1o4_RK100c3&gclid=Cj0KCQiAt_PuBRDcARIsAMNIbDrYTH0-vH-2GIXbHf2QwLmiYbzNFGabJTtuaP7UA_nXUyAGO6qeBYQaAsAjEALw_wcB)

## 6 Diskuse

Vytvořený kamerový systém byl podroben testování jeho efektivnosti. Byla zjištěna omezení použitých klasifikátorů, detektoru pohybu, kamer a finančního zatížení projektu.

Nejkritičtější omezení tkví v nízké účinnosti jednotlivých klasifikátorů, které nejsou určeny pro detekci člověka, postaveného ke kameře z profilu. Kritické omezení lze spatřit při nočním působení kamerového systému, jelikož použité kamery disponují IR filtrem. Snímky poskytnuté těmito kamerami pak nemohou o narušiteli poskytnout dostatečné množství informací.

Za výhodu kamerového systému lze shledat jeho nízkou pořizovací cenu. Tato nízká cena však vyvstává na úkor použitelnosti celkového systému. Jednotlivé společnosti (viz příloha C), které zavádí kamerové systémy do domácností, či velkých podniků, jsou profesionální ve svém oboru. Jejich kvalitní kamery a software, navzdory násobně vyšší pořizovací ceně, Raspberry Pi kamerový systém výrazně překonává. Na základě výsledků z testování Raspberry Pi kamerového systému jsem však přesvědčen, že pro monitorování domácnosti je tento systém dostačující.

### 6.1. Návrhy k vylepšení

V rámci bakalářské práce byl vytvořen kamerový systém, který byl podroben testování jeho efektivnosti. Ačkoliv se kamerový systém pro domácí použití ukázal jako dostatečný, lze v kamerovém systému najít několik zásadních nedostatků, na které se zaměřují následující kapitoly (6.1.1. – 6.1.6.).

#### 6.1.1. Kamery

Jelikož se v průběhu testování projektu používané kamery ukázaly být během nočního monitorování nepříliš účinné. Nabízí se řešení ve formě Raspberry Pi kamerového modulu, který nedisponuje IR filtrem. Absencí IR filtru je umožněno mít lepší přehled o dění v neosvětlené místnosti, což je u běžného kamerového modulu jen stěží možné.<sup>70</sup>

#### 6.1.2. Počet kamer

V rámci projektu je používána právě jedna kamera. Ať už se jedná o Camera Module V2 nebo USB webkameru, je tato kamera schopna zaměřit pouze jeden určitý úhel hlídaného

---

<sup>70</sup> Raspberry Pi NoIR kamera V2. In: *Rpishop* [online]. [cit. 2019-11-07]. Dostupné z: <https://rpishop.cz/kamery/331-raspberry-pi-noir-kamera-modul-v2.html>



objektu. Přidáním většího množství dobře umístěných kamer bychom dosáhli většího pokrytí prostoru, a tím i potenciálního navýšení šance případného zachycení narušitele.

### 6.1.3. MotionEyeOS

Za účelem přehledu nad výstupem více kamer je možné využít web-based interface, který nese název MotionEyeOS. MotionEyeOS umožňuje podporu pro širokou škálu USB kamer, Raspberry Pi kamerových modulů či IP kamer.<sup>71</sup>

MotionEyeOS je software podporovaný napříč mikro počítači jako Banana Pi, Orange Pi, Raspberry Pi či Odroid. Vyznačuje se zejména svým web-based a mobile friendly interface, který je pro uživatele velice přívětivý, jelikož je práce s ním velice intuitivní.<sup>72</sup>

Dále MotionEyeOS umožňuje automatické ukládání pořízených fotografií na pevná uložení jako SD kartu a USB drive, případně na cloudová uložení jako například Dropbox či Google Drive. To vše lze jednoduše nastavit v rámci ukládání souborů uvnitř nastavení MotionEyeOS.<sup>73</sup>

Další podstatnou předností MotionEyeOS je jeho možnost detekovat pohyb bez potřeby užívání externího pohybového senzoru.<sup>74</sup>

V poslední řadě je MotionEyeOS schopno nejen nahrávat kamerou zachycené video, nýbrž i toto nahrané video zpětně přehrávat, což může analýzu videa výrazně ulehčit a přispět k identifikaci narušitele.<sup>75</sup>

### 6.1.4. Napájení

V rámci projektu jsem jako zdroj napájení počítače Raspberry Pi používal adaptér, který disponoval microUSB konektorem, poskytující výstupní proud 2,5A a výstupní napětí 5,1V. Celková délka jeho kabelu činí pouhých 150 cm, což se během testování projektu ukázalo jako velice omezující, jelikož jsem počítači nebyl schopen zajistit přístup k napájení na vyvýšených místech monitorovaného objektu.<sup>76</sup> V rámci svých možností jsem byl schopen rozsah napájení prodloužit pomocí prodlužovacího kabelu.

---

<sup>71</sup> DroneBot Workshop. Build a remote Surveillance Camera using motionEyeOS & Raspberry Pi. In: *Dronebotworkshop* [online]. 7.12.2018 [cit. 2019-11-05]. Dostupné z: [https://dronebotworkshop.com/motioneyeos-raspberry-pi/#Motion\\_Detection%20motioneye](https://dronebotworkshop.com/motioneyeos-raspberry-pi/#Motion_Detection%20motioneye)

<sup>72</sup> tamtéž

<sup>73</sup> tamtéž

<sup>74</sup> tamtéž

<sup>75</sup> tamtéž

<sup>76</sup> Raspberry Pi microUSB 5,1V=2,5A zdroj, EU/UK, černá. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/zdroje/192-25a-oficialni-microusb-napajeci-zdroj-cerny.html>

Jako alternativní metoda se nabízí použití Powerbank jakožto zdroj napájení. Díky své krátké kabelové délce je možné napájet počítač z krátké vzdálenosti, a tím mu dovoluje operovat na místech, kam microUSB adaptér nedosáhne. Tato výhoda však vyvstává na úkor omezené doby napájení, která závisí především na kapacitě powerbanky a spotřebě energie Raspberry Pi při procesorem prováděných výpočtech.

### 6.1.5. Domácí mazlíčci

V mnoha domácnostech chovají majitelé domácí mazlíčky, např. psa, kočku. V případě, kdy se zvířata mohou volně pohybovat v monitorované místnosti, může velice snadno dojít k situaci, kdy bude jejich pohyb PIR senzorem detekován. Mohlo by tak snadno dojít k přeplnění kapacity microSD karty velkým množstvím redundantních snímků.

### 6.1.6. Paměť

V rámci projektu byla použita SD karta o kapacitě 16 GB. Předinstalovaný OS Raspbian Stretch spolu s veškerým základním programovým příslušenstvím však zaujímá téměř 6 GB, čili základní množství volného místa činí přibližně 10 GB. Po instalaci pro projekt potřebných python balíčků a OpenCV bylo množství volného místa sníženo na 4,9 GB. Množství volného místa SD karty lze jednoduše zjistit terminálovým příkazem: `df -Bm`.

Velikost vyfotografovaných snímků, které jsou uloženy v adresáři, činí přibližně 20 kB (PiCamera) a 90 kB (USB kamera). Při dlouhodobém používání kamerového systému a zanedbávání vymazávání těchto snímků může být kapacita microSD karty naplněna.

Pokud bychom vlastnili SD kartu s větší kapacitou, bylo by možné kamerový systém používat delší dobu bez toho, aby byla potřeba snímky průběžně odstraňovat. Existuje například objemnější microSD karta o kapacitě 32 GB, jejíž užití by bylo praktičtější.<sup>77</sup>

---

<sup>77</sup> NOOBS & Raspbian + 32GB SanDisk EDGE microSDHC UHS-I U1 A1 + SD adaptér. In: *Rpishop* [online]. [cit. 2019-11-25]. Dostupné z: <https://rpishop.cz/sd-karty/1451-sandisk-edge-32gb-microsdhc-uhs-i-u1-a1-noobs-raspbian.html>

## **6.2. Nabyté zkušenosti**

Během vypracování této bakalářské práce jsem měl možnost více se seznámit s počítačem Raspberry Pi, poznat jeho softwarové i hardwarové možnosti a omezení. Naučil jsem se užívat OS Raspbian. Zdokonalil jsem své znalosti a zkušenosti s programovacím jazykem Python a prohloubil své znalosti v oblasti objektové detekce.

## 7 Závěr

Cílem projektu bakalářské práce bylo vytvoření kamerového systému řízeném počítačem Raspberry Pi, primárně určený pro domácí použití.

V teoretické části byl představen počítač Raspberry Pi, jednotlivé hardwarové a softwarové komponenty projektu a popis základního principu obličejové detekce.

V praktické části byl jednotlivý hardware a software naprogramován a podroben testování.

Testovací hypotéza zněla: *Raspberry Pi kamerový systém je konkurenceschopný oproti běžným kamerovým systémům.* Tato hypotéza je vyvrácena, jelikož použitý hardware (USB kamera a Raspberry Pi Camera Module V2) a software (klasifikátor) nedokáží profesionálním komplexním kamerovým systémům, např. uzavřeného televizního okruhu, plně konkurovat. Počítač Raspberry Pi není pro monitorovací účely primárně určený, přesto jej lze v objektech malého rozsahu (např. domácností, garáží, malých skladů apod.) efektivně použít.

## 8 Seznam zdrojů

- A Cambridge success story. In: *Realvnc* [online]. [cit. 2019-11-25]. Dostupné z: <https://www.realvnc.com/en/company-profile/>
- About Us. In: *Pushbullet* [online]. [cit. 2019-11-30]. Dostupné z: <https://www.pushbullet.com/about>
- Armbian. In: *Techrepublic* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.armbian.com/>
- Banana Pi Series Comparison. In: *Wiki.banana-pi* [online]. 8.7.2019 [cit. 2019-11-05]. Dostupné z: [http://wiki.banana-pi.org/Banana\\_Pi\\_Series\\_Comparison](http://wiki.banana-pi.org/Banana_Pi_Series_Comparison)
- Barevné mini nepájivé pole. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/nepajiva-pole/73-427-barevne-mini-nepajive-pole.html#/63-barva-bila>
- BARNES, Russel. Thonny on a Raspberry Pi: using the new Python IDE in Raspbian. In: *Dronebotworkshop* [online]. 5.11.2017 [cit. 2019-11-05]. Dostupné z: <https://magpi.raspberrypi.org/articles/thonny>
- BARNES, Russell. PIXEL: the brand new desktop for the Raspberry Pi. In: *Magpi.raspberrypi* [online]. 5.11.2016 [cit. 2019-11-05]. Dostupné z: <https://magpi.raspberrypi.org/articles/introducing-pixel>
- BERGER, Will. Deep Learning Haar Cascade Explained. In: *Willberger* [online]. [cit. 2019-11-28]. Dostupné z: <http://www.willberger.org/cascade-haar-explained/>
- BROWNLEE, Jason. *How to Perform Face Detection with Deep Learning* [online]. In: . 3.6.2019 [cit. 2019-11-09]. Dostupné z: [https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/Cascade\\_Classifier\\_Training](https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/Cascade_Classifier_Training) [online]. In: Docs.opencv 21.11.2019 [cit. 2019-11-21]. Dostupné z: [https://docs.opencv.org/master/dc/d88/tutorial\\_traincascade.html](https://docs.opencv.org/master/dc/d88/tutorial_traincascade.html)
- Creative Live! Cam Sync HD. In: *Creative* [online]. [cit. 2019-11-05]. Dostupné z: <https://cs.creative.com/p/web-cameras/live-cam-sync-hd>
- DAVIS WEST, Jess. A BRIEF HISTORY OF FACE RECOGNITION. In: *Facefirst* [online]. 1.8.2017 [cit. 2019-11-09]. Dostupné z: <https://www.facefirst.com/blog/brief-history-of-face-recognition-software/>
- DroneBot Workshop. Build a remote Surveillance Camera using motionEyeOS & Raspberry Pi. In: *Dronebotworkshop* [online]. 7.12.2018 [cit. 2019-11-05]. Dostupné z: [https://dronebotworkshop.com/motioneyeos-raspberry-pi/#Motion\\_Detection%20motioneye](https://dronebotworkshop.com/motioneyeos-raspberry-pi/#Motion_Detection%20motioneye)
- DVOŘÁK, Jiří. Co je to VNC a jak se s ním připojit k virtuálnímu serveru. In: *Coolhousing* [online]. 6.7.2017 [cit. 2019-11-25]. Dostupné z: <https://www.coolhousing.net/cz/co-je-to-vnc-a-jak-se-s-nim-pripojit-k-virtualnimu-serveru>
- FABIEN, Maël. Boosting and AdaBoost clearly explained. In: *Towardsdatascience* [online]. 14.2.2019 [cit. 2019-11-05]. Dostupné z: <https://towardsdatascience.com/boosting-and-adaboost-clearly-explained-856e21152d3e>
- Formát adresy SMS upozornění u O2. In: *O2* [online]. [cit. 2019-11-27]. Dostupné z: <https://sms-o2.cz/sms-upozorneni-na-novy-mail-jak-si-ho-nastavit/>
- FROMAGET, Patrick. 25 awesome Raspberry Pi project ideas at home. In: *Raspberrytips* [online]. [cit. 2019-11-05]. Dostupné z: <https://raspberrytips.com/raspberry-pi-projects-for-home/>
- Gödel Prize. In: *Sigact* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.sigact.org/prizes/g%C3%B6del.html>
- GROVER, Puneet. *Clearing air around "Boosting"* [online]. In: *Towardsdatascience* . 18.3.2019 [cit. 2019-11-29]. Dostupné z: <https://towardsdatascience.com/clearing-air-around-boosting-28452bb63f9e#935b>

HEMMINGS, Megan. What is a Jumper Wire? In: *Sparkfuneducation* [online]. 30.1.2018 [cit. 2019-11-05]. Dostupné z: <http://blog.sparkfuneducation.com/what-is-jumper-wire>

HICKEY, Shane. This article is more than 5 years old The Raspberry Pi computer: how a bright British idea took flight. In: *Theguardian* [online]. 9.3.2014 [cit. 2019-11-05]. Dostupné z: <https://www.theguardian.com/technology/2014/mar/09/raspberry-pi-computer-eben-upton-cambridge>

How HC-SR501 PIR Sensor Works: Interface It With Arduino. In: *Lastminuteengineers* [online]. [cit. 2019-11-05]. Dostupné z: <https://lastminuteengineers.com/pir-sensor-arduino-tutorial/>

*Imutils 0.5.3* [online]. In: . [cit. 2019-11-07]. Dostupné z: <https://pypi.org/project/imutils/>

Monk, S. Raspberry Pi Cookbook. O'Reilly Media 2013 ISBN 978-1-449-36522-6 NOOBS & Raspbian + 32GB SanDisk EDGE microSDHC UHS-I U1 A1 + SD adaptér. In: *Rpishop* [online]. [cit. 2019-11-25]. Dostupné z: <https://rpishop.cz/sd-karty/1451-sandisk-edge-32gb-microsdhc-uhs-i-u1-a1-noobs-raspbian.html>

NumPy Tutorial. In: *www.tutorialspoint.com* [online]. [cit. 2019-11-07]. Dostupné z: <https://www.tutorialspoint.com/numpy/index.htm>

OpenCV. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 13.4.2019 [cit. 2019-11-05]. Dostupné z: <https://cs.wikipedia.org/wiki/OpenCV>

Passive Buzzer. In: *Addicore* [online]. [cit. 2019-11-07]. Dostupné z: <https://www.addicore.com/Passive-Buzzer-p/ad319.htm>

Piezo Buzzer - PS1240. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/bzucaky/813-piezo-buzzer-ps1240.html>

Pushbullet API: End-to-End Encryption. In: *Pushbullet* [online]. [cit. 2019-11-25]. Dostupné z: <https://docs.pushbullet.com/>

Pushbullet Pro. In: *Pushbullet* [online]. [cit. 2019-11-29]. Dostupné z: <https://www.pushbullet.com/pro>

Python Programming Language. In: *Educba* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.educba.com/python-programming-beginners-tutorial/>

Python Programming Language. In: *Educba* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.educba.com/python-programming-beginners-tutorial/>

Raspberry Pi 4 Model B - 4GB RAM. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/raspberry-pi-4b/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>

RASPBERRY Pi Camera Module V2. In: *Alza* [online]. [cit. 2019-11-27]. Dostupné z: [https://www.alza.cz/raspberry-pi-camera-module-v2-d4256972.htm?kampan=adw1\\_prislusenstvi-pro-it-tv\\_pla\\_all\\_prislusenstvi-css\\_rozsirujici-karty\\_c\\_9062906\\_1o4\\_RK100c3&gclid=Cj0KCQiAt\\_PuBRDcARIsAMNIBdrYTH0-vH-2GIXbHf2QwLmiYbzNFGabJTtuaP7UA\\_nXUyAGO6qeBYQaAsAjEALw\\_wcB](https://www.alza.cz/raspberry-pi-camera-module-v2-d4256972.htm?kampan=adw1_prislusenstvi-pro-it-tv_pla_all_prislusenstvi-css_rozsirujici-karty_c_9062906_1o4_RK100c3&gclid=Cj0KCQiAt_PuBRDcARIsAMNIBdrYTH0-vH-2GIXbHf2QwLmiYbzNFGabJTtuaP7UA_nXUyAGO6qeBYQaAsAjEALw_wcB)

Raspberry Pi GPIO. In: *Rpi.science.uoit* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpi.science.uoit.ca/lab/gpio/>

Raspberry Pi kamera V2. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/kamery/329-raspberry-pi-kamera-modul-v2.html>

Raspberry Pi microUSB 5,1V=2,5A zdroj, EU/UK, černá. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/zdroje/192-25a-oficialni-microusb-napajeci-zdroj-cerny.html>

Raspberry Pi NoIR kamera V2. In: *Rpishop* [online]. [cit. 2019-11-07]. Dostupné z: <https://rpishop.cz/kamery/331-raspberry-pi-noir-kamera-modul-v2.html>

ROBINSON, A. -- COOK, M. *Raspberry Pi projects* . Chichester, England: Wiley, 2014. ISBN 978-1-118-55543-9.

ROSEBROCK, Adrian. *Install OpenCV 4 on your Raspberry Pi* [online]. In: . 26.9.2018 [cit. 2019-11-22]. Dostupné z: <https://www.pyimagesearch.com/2018/09/26/install-opencv-4-on-your-raspberry-pi/>

S. BRID, Rajesh. Introduction to Boosting. In: *Medium* [online]. 1.11.2018 [cit. 2019-11-09]. Dostupné z: <https://medium.com/greyatom/boosting-ce84639a805d>

Saucecat. Boosting algorithm: AdaBoost. In: *Towardsdatascience* [online]. 29.4.2017 [cit. 2019-11-30]. Dostupné z: <https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c>

shivamsaraswat. Python Language Introduction. In: *Geeksforgeeks* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.geeksforgeeks.org/python-language-introduction/>

Top Uses of Raspberry Pi. In: *Educba* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.educba.com/uses-of-raspberry-pi/>

UPTON, E. -- GONER, J. -- HALFACREE, G. *Raspberry Pi : uživatelská příručka* (2.vydání). Brno: Computer Press, 2016. ISBN 978-80-251-4819-8, s. 219

UPTON, E. -- GONER, J. -- HALFACREE, G. *Raspberry Pi : uživatelská příručka* (2.vydání). Brno: Computer Press, 2016. ISBN 978-80-251-4819-8, s. 218

UPTON, E. -- GONER, J. -- HALFACREE, G; s. 219

UPTON, E. -- GONER, J. -- HALFACREE, G; s. 219-220

varatika02. Disadvantages of Python. In: *Geeksforgeek* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.geeksforgeeks.org/disadvantages-of-python/>

Welcome to Raspbian. In: *Raspbian* [online]. [cit. 2019-11-05]. Dostupné z: <https://www.raspbian.org/>

Welcome to SWIG. In: *Swig* [online]. [cit. 2019-11-05]. Dostupné z: <http://www.swig.org/>

## Zdroje obrázků

- 1 Raspberry Pi 3 Model B 64-bit 1GB RAM. In: *Rpishop.cz* [online]. [cit. 2019-10-09]. Dostupné z: <https://rpishop.cz/raspberry-pi-3b/283-raspberry-pi-3-model-b-64-bit.html>
- 2 Banana Pi, DDR3 1 GB,, SinoVoip. In: *Distrelec.cz* [online]. [cit. 2019-10-09]. Dostupné z: <https://www.distrelec.cz/cs/banana-pi-ddr3-gb-sinovoip-banana-pi/p/30011509>
- 3 Orange Pi One H3 Quad-core 512MB RAM. In: *Rasel.cz* [online]. [cit. 2019-10-09]. Dostupné z: <https://www.rasel.cz/orange-pi-one-h3-quad-core-512mb-ram-p19333/>
- 4 Mat. Raspberry PI 2, 3, Zero & ZeroW GPIO explained. In: *Notenoughtech.com* [online]. 4.4.2016 [cit. 2019-11-05]. Dostupné z: <https://notenoughtech.com/raspberry-pi/rpi-gpio/>
- 5 JEŽEK, David. Raspberry Pi má nové desktopové prostředí Pixel. In: *Diit.cz* [online]. 4.10.2016 [cit. 2019-11-05]. Dostupné z: <https://diit.cz/clanek/raspberry-pi-ma-nove-desktopove-prostredi-pixel>
- 6 *Updated Material Design, Adaptive Icon, And Dark Mode Out Now* [online]. In: . 11.2.2019 [cit. 2019-11-13]. Dostupné z: <https://blog.pushbullet.com/>
- 7 Raspberry Pi kamera V2. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/kamery/329-raspberry-pi-kamera-modul-v2.html>
- 8 Creative Live! Cam Sync HD. In: *Creative* [online]. [cit. 2019-11-05]. Dostupné z: <https://cs.creative.com/p/web-cameras/live-cam-sync-hd>
- 9 Piezo Buzzer - PS1240. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z: <https://rpishop.cz/bzucaky/813-piezo-buzzer-ps1240.html>
- 10 DE BAKKER, Benne. How to use HC-SR501 PIR Motion Sensor with Arduino. In: *Makerguides* [online]. [cit. 2019-11-30]. Dostupné z: <https://www.makerguides.com/hc-sr501-arduino-tutorial/Arduino-PIR-detektor-pohybu-HC-SR501>
- 11 tutorial/Arduino PIR detektor pohybu HC-SR501. In: *Laskarduino* [online]. [cit. 2019-11-05]. Dostupné z: [https://www.laskarduino.cz/arduino-pir-detektor-pohybu-hc-sr501/?gclid=Cj0KCQjw0brtBRDOARIsANMDykbzApOJagatMLyX\\_sZFeLk7oVWZwN6jGDXtE4\\_wUoKA13EJ\\_xXoW4aAtReEALw\\_wcB](https://www.laskarduino.cz/arduino-pir-detektor-pohybu-hc-sr501/?gclid=Cj0KCQjw0brtBRDOARIsANMDykbzApOJagatMLyX_sZFeLk7oVWZwN6jGDXtE4_wUoKA13EJ_xXoW4aAtReEALw_wcB)
- 12 MARTÍNEZ ZARZUELA, Mario. Five types of rectangular Haar wavelet-like features. In: *Researchgate* [online]. [cit. 2019-11-10]. Dostupné z: [https://www.researchgate.net/figure/Five-types-of-rectangular-Haar-wavelet-like-features-A-feature-is-a-scalar-calculated-by\\_fig1\\_225367816](https://www.researchgate.net/figure/Five-types-of-rectangular-Haar-wavelet-like-features-A-feature-is-a-scalar-calculated-by_fig1_225367816)
- 13 JALAL, Mona. How can I understand Haar-like feature for face detection?: 21.5.2017. In: *Quora* [online]. [cit. 2019-11-10]. Dostupné z: <https://www.quora.com/How-can-I-understand-Haar-like-feature-for-face-detection>



## Zdroje tabulek

- 1 Raspberry Pi 3 Model B 64-bit 1GB RAM. In: *Rpishop.cz* [online]. [cit. 2019-10-09]. Dostupné z:  
<https://rpishop.cz/raspberry-pi-3b/283-raspberry-pi-3-model-b-64-bit.html>
- 2 Raspberry Pi kamera V2. In: *Rpishop* [online]. [cit. 2019-11-05]. Dostupné z:  
<https://rpishop.cz/kamery/329-raspberry-pi-kamera-modul-v2.html>
- 3 Creative Live! Cam Sync HD. In: *Creative* [online]. [cit. 2019-11-05]. Dostupné z:  
<https://cs.creative.com/p/web-cameras/live-cam-sync-hd>
- 4 Arduino PIR detektor pohybu HC-SR501. In: *Laskarduino* [online]. [cit. 2019-11-28]. Dostupné z:  
<https://www.laskarduino.cz/arduino-pir-detektor-pohybu-hc-sr501/>
- 5 Vlastní tabulka

## 9 Přílohy

### Příloha A: Kód cam\_system.py

```
import sys #System Specific Parameters and Functions - zajisti pristup ke specialnim
promennym
import cv2 # Importovani knihovny OpenCV
import numpy as np # Importovani balicku NumPy
from imutils.video.webcamvideostream import WebcamVideoStream # from
imutils.video.pivideoostream import PiVideoStream #from imutils.video.webcamvideostream
import WebcamVideoStream
# from picamera import PiCamera #Pouzit tuto tridu, pokud je pouzivan Raspberry Pi Camera
Module V2

# Logging (zaznamnik) pro lepsi kontrolu nad rizenim vyvoje projektu
import logging
# Vraci instanci RotatingFileHandler tridy. The specified file is opened and used as the
stream for logging.
from logging.handlers import RotatingFileHandler
import os
from os.path import join, dirname, abspath, exists

# Modul time poskytuje funkce spojene s casem
import time
from datetime import datetime

# Umoznuje spravovat vstupne/vystupni operace souvisejici se soubory
import io

import RPi.GPIO as IO #Import GPIO modulu
from pushbullet import PushBullet

# Import smtplib pro funkci zasilani zprav
import smtplib
# Eailove balicky, ktere budeme potrebovat
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Umoznuje beh vetsiho mnozstvi casi kodu najednou
from threading import Thread

# Trida pro v projektu pouzivanou kameru
class VideoCamera(object):
    def __init__(self, flip=False):
        self.vs = WebcamVideoStream().start() # PiVideoStream #WebcamVideoStream u
kameroveho modulu
        self.flip = flip
        time.sleep(300) # System zacne snimat po 300 sekundach

    def __del__(self):
        self.vs.stop()

    def flip_if_needed(self, frame):
        if self.flip:
            return np.flip(frame, 0)
        return frame

    # Ziskanvani obrazku
    def get_frame(self):
        frame = self.flip_if_needed(self.vs.read())
        return frame

    # Hledani obliceje, nebo ruznych objektu ve snimanem obrazu.
    def get_object(self, classifier):
        found_objects = False
        frame = self.flip_if_needed(self.vs.read()).copy()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Prechod do greyscale - Viola Jones

        objects = classifier.detectMultiScale( #Kaskáda klasifikátorů prozkoumává
jednotlivé části snímků ve více měřících, aby rychle detekovala obličeje nejruznejsich
velikosti
```

```

        gray,
        scaleFactor=1.3, # Označuje, jak moc se velikost snímku sníží v každém merítku
obrazu
        minNeighbors=2, # Muže zachytit až 3 lidské obličeje v jednom snímku
        minSize=(30, 30), # Minimalní velikost detekovaného objektu (30, 30) je
defaultní velikost. Menší objekty jsou v obrazu ignorovány.
        flags=cv2.CASCADE_SCALE_IMAGE
    )
    # Pokud len vrátil hodnotu "objects" větší než 0, pak je objekt považován za
nalezený
    if len(objects) > 0:
        found_objects = True

    # Vykreslí obdélník do snímku okolo nalezeného objektu
    for (x, y, w, h) in objects:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2) # souřadnice levého
horního a pravého spodního rohu detekovaného objektu v obrazu, barva rámečků, šířka čar rámečků
    return frame, found_objects

# Logging (zaznamník) pro lepší kontrolu nad řízením vývoje projektu
# Pokud například uživatel zavolá, aby nahlásil nějakou chybu, mohou být "logs" pro kontext
toto chyby snadno vyhledány, log zprávy mohou být vypisovány do vedlejšího souboru
logger = logging.getLogger() # Vytvoření loggeru objektu
# destinace log zprávy nastavena na standard output (stream do terminálu Raspberry Pi)
stream_handler = logging.StreamHandler(sys.stdout)
# Destinace log zprávy nastavena do souboru project.log
file_handler = RotatingFileHandler("project.log", maxBytes=10 * 1024 * 1024,
backupCount=10)
# Min úroveň zpracováváných log zpráv je stanovena na úroveň DEBUG, což je hodnota 10 -
(NOTSET = 0, DEBUG = 10, INFO = 20, WARNING = 30, ERROR = 40, CRITICAL = 50)
logger.setLevel(logging.DEBUG)
# Min úroveň zpracováváných log zpráv je stanovena na úroveň DEBUG, což je hodnota 10
stream_handler.setLevel(logging.DEBUG)
# Min úroveň zpracováváných log zpráv je stanovena na úroveň DEBUG, což je hodnota 10
file_handler.setLevel(logging.DEBUG)
# Přidání stream handleru
logger.addHandler(stream_handler)
# Přidání file handleru
logger.addHandler(file_handler)

# Horizontální převrácení kamery
video_camera = VideoCamera(flip=False) # U PiCamery: True, u Webkamery: False

# GPIO configuration of Piezo Buzzer & PIR sensor
IO.setwarnings(False) # Zablokovat varování
IO.setmode(IO.BCM) # Typ číslování pinů na patičce dle Broadcom SOC channel
motion_detector_pin = 4 # Číslo použitého GPIO pinu
buzzer_pin = 19 # Číslo použitého GPIO pinu
IO.setup(buzzer_pin, IO.OUT) # Nastavit pin detektoru na výstup
IO.setup(motion_detector_pin, IO.IN) # nastavit pin detektoru na vstup

# Buzzer frequency configuration
p = IO.PWM(buzzer_pin, 4000) # Ovládání hlasitosti

SMS_UPDATE_INTERVAL = 600 # Interval posílání SMS upozornění je 600 vteřin. Po první
detekci je SMS upozornění ihned posláno, další proběhne až po detekci,
# která bude časově vzdálena 600 vteřin od detekce původní

def check_for_movement(): # Snímání pohybu PIR senzorem
    lastSmsTime = 0 # Poslední čas posledního poslání SMS
    while True: # Nekonečný cyklus
        try: # Pokud kód proběhne správně = try, pokud nastane chyba = exception (Try-
Catch)
            motion_detected = IO.input(motion_detector_pin) # Čti status pinu PIR senzoru
a přidej mu proměnou "motion detected"
            if motion_detected: # Pokud je pohyb detekován
                print("Motion detected!") # Vypis upozornění
                frame = video_camera.get_frame() # Kamera porídí snímek
                object_processing(frame) # Pojmenování snímku, vložení data porovnání snímku
primo do snímku, uložení snímku, poslání snímku Pushbullet
                buzz() # Zazní bzucák
                print("Alarm!") # Vypis alarmu
                if time.time() - lastSmsTime > SMS_UPDATE_INTERVAL: # Pokud rozdíl
aktuálního a posledního času odeslání SMS je vyšší než min. interval SMS
                    sendSMS() # Posli SMS upozornění

```

```

        lastSmsTime = time.time() # Prepsani casu posledniho poslani SMS
    else: # Jinak nedelej nic
        pass
    else: # Pokud neni zaznamenan pohyb, kazdou vterinu vypisuj "No motion
detected"
        print("No motion detected.")
        time.sleep(1)
    except: # Pokud nastane chyba, vypis chybovou zpravu
        logger.error("Error detecting motion: ", sys.exc_info()[0])
        raise

FRAME_UPDATE_INTERVAL = 50 # Snimek porizen pomoci klasifikatoroveho urceni ma interval 50
sekund
BASE_PATH = dirname(abspath(__file__)) # Je absolutni cesta k adresari
object_classifier = cv2.CascadeClassifier(join(BASE_PATH, "model",
"facial_recognition_model.xml")) # Absolutni cesta k vyberu klasifikatoru

# Snimani a hledani lidskeho obliceje/horni poloviny lidskeho tela, cele lidske telo
def check_for_objects():
    """
    Loop - Find objects in front of the camera
    :return: None
    """
    actualTime = 0 # Aktualni cas
    lastSmsTime = 0 # Posledni cas posledniho poslani SMS
    lastObjFOund = 0 # Casova promena posledni detekce objektu v obrazu
    while True:
        time.sleep(0.5) # Aby vlakno neustale nezabiralo procesorovy cas, je treba jej
mirne pozastavit - prodleva mezi zpracovavanim obrazu
        try:
            frame, found_obj = video_camera.get_object(object_classifier) # Klasifikator,
ktery prohledava snimky kamery za ucelem nalezeni lidskeho obliceje
            if found_obj and (time.time() - lastObjFOund) > FRAME_UPDATE_INTERVAL: # Pokud
rozdil aktualniho casu a posledniho casu nalezeni objektu je vyssi nez interval
                logger.info("Face found") # Vypis o nalezeni obliceje
                object_processing(frame) #Pojmenovani snimku, vlozeni data porizeni snimku
primo do snimku, ulozeni snimku, poslani snimku Pushbullet
                print("Alarm!") # Vypis alarmu
                buzz() # Zazni bzucak
                lastObjFOund = time.time() # Cas od posledniho nalezeni objektu
                actualTime = time.time() # Aktualni cas

                if time.time() - lastSmsTime > SMS_UPDATE_INTERVAL: # Pokud rozdil
aktualniho a posledniho casu odeslani SMS je vyssi nez min. interval SMS
                    sendSMS() # Je zaslano SMS upozorneni
                    lastSmsTime = time.time() # Prepsani casu posledniho poslani SMS
                else: # Jinak nedelej nic
                    pass

                logger.info("Processing finished.")
        except:
            logger.error("Error getting frame: ", sys.exc_info()[0])
            raise

# Vlozeni popisku do obrazku
def insert_date(frame, date_text):
    """
    :param frame:
    :return:
    """
    cv2.putText(
        img=frame, #Obrazek, do ktereho ma byt text vypsan
        text=date_text, # Text, který bude vypsan
        org=(100, 100), # Souradnice textu v obrazku: Webcamera = 100, 100 Picamera =
24,30
        fontFace=cv2.FONT_HERSHEY_SIMPLEX, #font pisma
        fontScale=1, # Webcamera = 1 PiCamera = 0.7 velikost pisma
        color=(0, 128, 255) # Barva pisma
    )
    return frame

# Ulozeni obrazku do pameti Raspberry Pi
def save_frame(frame, full_path_filename):
    """

```

```

:param frame:
:param filename:
:return:
"""
cv2.imwrite(full_path_filename, frame)

# Pushbullet API KEY
API_KEY = "o.LhkmMdKZC4mUk3JHXpd9rb5qtDySHT1F"

# Odesilani snimku do aplikace Pushbullet
def mobile_push(frame, filename):
    pb = PushBullet(API_KEY) # Prihlaseni do Pushbullet
    now = datetime.now() #Aktualni datum
    date_text = now.strftime("%d/%m/%Y %H:%M:%S") # Datum porizeni snimku ve formatu -
den, mesic, rok hodina, minuta, sekunda
    pb.push_note("Alarm", "Narusitel detekovan dne : " + date_text) # Vypis do zpravy
    _, jpeg = cv2.imencode('.jpg', frame) # Zakoduje format obrazku za ucelem komprese dat
obrazku pro jejich rychlesji prenos a lepsi zpracovani

    # Zasobnik pro odesilnane snimky
    with io.BytesIO() as buf:
        buf.write(jpeg.tobytes()) # Vrat neupravena data snimku z interniho uloziste
        buf.seek(0) #Metoda seek () nastavuje aktualni pozici souboru ve file stream.
        file_data = pb.upload_file(buf, filename) #Uploaduje snimek
        pb.push_file(**file_data) # Zaslani snimku do Pushbullet

# Piezo Buzzer - Alarm
def buzz():
    p.start(1)
    p.ChangeDutyCycle(85)
    time.sleep(5) # Delka alarmu je 5 vterin
    p.stop()

# Zpracovani snimku, jejich pojmenovani podle casu porizeni a jejich ukladani do
specifickych adresaru
def object_processing(frame):
    """
    Base object processing (when an object is captured in front of the camera)
    :param frame: captured frame
    :return: None
    """
    now = datetime.now() # Aktualni cas
    filename = now.strftime("%Y%m%d_%H%M%S.jpg") # Format nazvu obrazku - aktualni rok,
mesic, den _ hodina, minuta a vterina
    date_text = now.strftime("%d/%m/%Y %H:%M:%S") # Format vkladaneho textu do obrazku -
aktualni rok, mesic, den _ hodina, minuta a vterina
    basedir = join(BASE_PATH, "photos", now.strftime("%Y%m%d_%H")) # Format vytvarenych
adresaru - aktualni rok, mesic, den a hodina, kterou kamerovy system bezi

    if not exists(basedir): # Pokud cilovy adresar neexistuje, je vytvoren adresar novy
        os.makedirs(basedir)

    frame = insert_date(frame, date_text) # Vlozeni data do obrazku
    save_frame(frame, join(basedir, filename)) # Ulozeni obrazku do pameti
    mobile_push(frame, filename) # Odeslani do Pushbullet

# Zasilani SMS upozorneni z Gmail na mobilni zarizeni
def sendSMS():
    sender = '?????????????????????????@gmail.com' # Gmail adresa
    sender_password = '?????????????????????' # Heslo emailove adresy odesilatele
    send_to_mobile = '+420xxxxxyyzzz@sms.cz.o2.com' # Telefonni cislo adresata
    subject = 'Alarm!' # Podmet zpravy
    message = 'Zkontroluj Pushbullet' # Hlavni obsah zpravy

    msg = MIMEMultipart() # Container pro vnejsi zpravu emailu
    msg['From'] = sender # Od odesilatele
    msg['Subject'] = subject # Pro adresata

    msg.attach(MIMEText(message, 'plain')) # Prilozeni zpravy
    server = smtplib.SMTP('smtp.gmail.com', 587) # Odeslani SMS pomoci (SMTP) Serveru
(vyzaduje TLS, nebo SSL); 587 = port pro TLS/STARTTLS

```

```

server.starttls() # TLS = Transport Layer Security (kryptograficky protokol) pro
bezpecnejsi posilani dat
server.login(sender, sender_password) # Prihlaseni se do Gmail
text = msg.as_string() # Vlozeni textu zpravy jakozto textoveho retezce
server.sendmail(sender, send_to_mobile, text) # Zaslani mailu adresatovi
server.quit() # Ukonceni SMTP relace a ukonceni pripojeni

# Spusteni systemu - dve vlakna, která budou "bezet" najednou
if __name__ == '__main__':
    logger.info("Start checking...") # Vypis o startu kameroveho systemu

    thread_classifier = Thread(target=check_for_objects) # Vlakno pro detekci podle
klasifikatoru
    thread_movement = Thread(target=check_for_movement) # Vlakno pro detekci podle detekce
pohybu

    try:
        for thread in (thread_classifier, thread_movement):
            thread.start()

        for thread in (thread_classifier, thread_movement):
            thread.join()

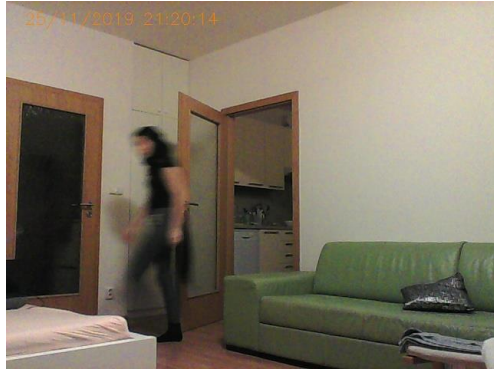
    except KeyboardInterrupt:
        print('Keyboard interrupt, stopping processing...') # vypis o konci kameroveho
systemu

IO.cleanup() # resetuje vsechny v projektu pouzite porty do input modu
logger.info("System finished.") # Kamerovy system byl ukoncen.

```

## Příloha B: Výběr fotografií získaných při testování





## Příloha C: Ceník služeb Sistel International k 25.11.2019

### Orientační ceník našich služeb

Jsme tradiční společnost působící na trhu **více než 25 let**. Přesto nezapomínáme na to nejdůležitější, zakládáme si na osobním přístupu ke každému našemu klientovi. Na základě našich zkušeností jsme schopni klientům poskytnout **profesionální služby** v oblasti zabezpečovacích systémů a služeb.

Druh služby	Popis služby	Cena
Ohlídka objektu	Fyzická ohlídka našim technikem pro navrzení nejefektivnějšího systému.	Zdarma
Návrh a rozpočet systému	Vyhotovení rozpočtu a návrhu, následná prezentace klientovi.	Zdarma
PZTS (EZS)	Popis služby	Cena
Montáž poplachového zabezpečovacího a tísňového systému	Odborná montáž, instalace a programování. Zaučení obsluhy.	Od 10.900 Kč
Střežení přes PCO	Měsíční paušální sazba za nepřetržitý monitoring objektu (cena v závislosti na velikosti objektu, lokalitě apod.).	Od 390 Kč
Připojení CCTV na video-pult	Měsíční paušál doplňkové služby ke střežení přes PCO - v případě poplachu operátor pomocí kamer vyhodnotí příčinu poplachu.	Od 190 Kč
Výjezd ZS	Výjezd zásahové skupiny v případě poplachu.	Od 290 Kč
Test funkčnosti	Testování spojení a aktivity PZTS vzdáleně přes PCO v pravidelných intervalech.	Zdarma
CCTV - kamerové systémy	Popis služby	Cena
Montáž CCTV	Odborná montáž, instalace a programování. Zaučení obsluhy.	Od 12.900 Kč
Návrh a rozpočet systému	Měsíční paušál za nepřetržitý monitoring kamer našimi operátory.	Od 1.900 Kč
Ostatní služby	Popis služby	Cena
Pronájem PZTS	Pronájem zabezpečovacího systému - Stavba, sklady apod. včetně připojení na PCO.	Od 150 Kč / den
Pronájem kamerového systému	Pronájem kamerového systému včetně příslušenství a připojení na Videopult. Vhodné pro stavby, sklady apod.	Od 250 Kč / den
<b>BEZPEČNÁ DOVOLENÁ</b>	Zapůjčení setu PZTS a připojení na PCO	100 Kč / den
<b>BEZPEČNÁ DOVOLENÁ PLUS</b>	Zapůjčení setu PZTS a připojení na PCO, služba fyzické kontroly objektu a vybírání poštovní schránky	150 Kč / den
Převoz cenin	Převoz finančních hotovostí, uměleckých sbírek apod. neoznačenými vozy a s ozbrojenou posádkou	Od 1190 Kč
Fyzická ostraha	Strážní, recepční služby. Ostraha malých i velkých objektů, jednorázové eventy.	DOHODOU

**Detailní rozpočet pro Vaše potřeby Vám rádi poskytneme na základě fyzické obhlídky objektu. Nenašli jste službu, kterou potřebujete? Nevadí. Ozvěte se nám a rádi s Vámi probereme případné možnosti.**

Sistel International s.r.o.  
 Podlipného 997/8, 180 00 Praha 8  
 E-mail: info@sistel.cz, web: www.sistel.cz  
 Tel.: 731 171 329  
 IČ:15886832. DIČ: CZ15886832

www.sistel.cz