

# Detekce a klasifikace objektů v obraze z kamery

Bc. Aleš Vašíčka

---

Diplomová práce  
2018

 Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2017/2018

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Aleš Vašička**  
Osobní číslo: **A14495**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Automatické řízení a informatika**  
Forma studia: **kombinovaná**

Téma práce: **Detekce a klasifikace objektů v obrazu z kamery**  
Téma anglicky: **Object Detection and Classification from Video Camera Signal**

Zásady pro vypracování:

1. Seznamte se s úlohou detekce osob a zpracujte rešerši metod pro detekci pohybujícího se objektu ve videu z kamery umístěné nad prostorem.
2. Zpracujte rešerši metod pro klasifikaci detekovaných objektů a metod zpracování obrazu využitelných pro tuto úlohu.
3. Vytvořte software pro detekci a klasifikaci pohybujících se objektů ve videosignálu v prostředí MATLAB.
4. Implementaci metody ověřte na reálných datech a vyhodnoťte dosaženou přesnost.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. GONZALES, R.C., Woods, R.E., Eddins, S.L. Digital Image Processing Using MATLAB, Gatesmark Publishing, 2009. 609s. ISBN: 978-0982085400
2. RUSS, J.C. Image Processing Handbook, 6th Edition. CRC Press, ISBN 978-1439840450
3. HORÁK, Karel, KALOVÁ, Ilona, PETYOVSÝ, Petr , RICHTER, Miloslav. Počítačové vidění. [online]. Brno VUT v Brně. 2008. 132s. Dostupné z: [http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove\\_videni\\_S.pdf](http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove_videni_S.pdf)
4. DAVIES, E. R. Computer and machine vision: theory, algorithms, practicalities. 4th ed. Boston: Elsevier, 2012. ISBN 9780123869081
5. HLAVÁČ, Václav, SEDLÁČEK, Miloš, Zpracování signálu a obrazu. [online]. ČVUT v Praze. 1999.116s. Dostupné z: [http://www.ugn.cas.cz/kolcun/G/azro/txt/cvut\\_Hlavac\\_Sedlacek.pdf](http://www.ugn.cas.cz/kolcun/G/azro/txt/cvut_Hlavac_Sedlacek.pdf)
6. HÁJOVSKÝ, R., PUSTKOVÁ R., KUTÁLEK F. ZPRACOVÁNÍ OBRAZU V MĚŘICÍ A ŘÍDICÍ TECHNICE: učební text [online]. Vysoká škola báňská Technická univerzita Ostrava, 2012 [cit. 2017- 03-29]. Dostupné z: [http://www.person.vsb.cz/archivcd/FEI/ZOMRT/Zpracovani\\_obrazu\\_v\\_merici\\_a\\_ridici\\_technice.pdf](http://www.person.vsb.cz/archivcd/FEI/ZOMRT/Zpracovani_obrazu_v_merici_a_ridici_technice.pdf)

Vedoucí diplomové práce: **Ing. Jakub Novák, Ph.D.**

Ústav řízení procesů

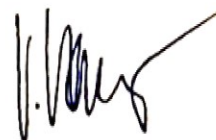
Datum zadání diplomové práce: **15. prosince 2017**

Termín odevzdání diplomové práce: **25. května 2018**

Ve Zlíně dne 15. prosince 2017



doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

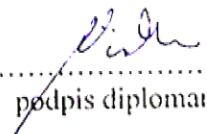
### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 24. 5. 2018

  
.....  
podpis diplomanta



## **ABSTRAKT**

Cílem diplomové práce je navrhnout a vytvořit program, který dokáže detekovat a rozpoznávat objekty v obraze přenášeném ze staticky umístěné kamery. Práce je rozdělena do dvou částí - teoretické a praktické. Teoretická část se věnuje analýze obrazu, pohybu a metodám rozpoznávání objektů v obraze. Praktická část se věnuje návrhu a realizaci programu v MATLABu, testování a vyhodnocení výsledků.

Klíčová slova: Počítačové vidění, detekce, segmentace, rozpoznání, objekt, klasifikace, MATLAB, SVM

## **ABSTRACT**

The aim of this master's thesis is to design and create a program which is able to detect and recognize objects in image transferred from a statically placed camera. The thesis is divided into two parts – theoretical and practical. The theoretical part deals with analysis of image, motion and methods of object recognition in the image. The practical part deals with design and implementation of MATLAB program, testing and evaluation of results.

Keywords: Computer vision, detection, segmentation, recognition, object, classification, MATLAB, SVM

V první řadě bych chtěl poděkovat Ing. Jakubu Novákovi, Ph.D. za rady, čas, odborné vedení a trpělivost, kterou mi věnoval při vypracovávání diplomové práce. Nakonec bych chtěl poděkovat mé přítelkyni a rodině, za jejich nekonečnou podporu a trpělivost.

*„Co slyším, to zapomenu. Co vidím, si pamatuji. Co si vyzkouším, tomu rozumím.“*

Konfucius

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD .....</b>	<b>9</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 ÚVOD DO PROBLEMATIKY.....</b>	<b>11</b>
<b>2 PRÁCE S OBRAZEM .....</b>	<b>12</b>
2.1 DIGITALIZACE.....	12
2.1.1 Interval vzorkování.....	12
2.1.2 Volba vzorkovací mřížky.....	13
2.1.3 Kvantování hodnot obrazové funkce.....	13
2.2 REPREZENTACE OBRAZOVÝCH DAT .....	14
2.2.1 RGB .....	15
2.2.2 HSV .....	16
2.2.3 Šedo tónový obraz .....	17
2.2.4 Binární obraz .....	17
<b>3 ANALÝZA OBRAZU A POHYBU .....</b>	<b>18</b>
3.1 ODSTRANĚNÍ ŠUMU .....	18
3.1.1 Konvoluční filtrace .....	18
3.2 DETEKCE OBJEKTŮ (SEGMENTACE OBRAZU).....	19
3.2.1 Prahování .....	19
3.2.2 Odečítání pozadí.....	21
3.2.3 Detekce hran.....	23
3.2.4 3D hloubková mapa.....	24
3.3 ODSTRANĚNÍ STÍNŮ .....	25
3.4 KLASIFIKACE OBJEKTU .....	26
3.4.1 Histograms of Oriented Gradients (HOG).....	26
3.4.2 Support vector machines (SVM).....	29
3.4.3 Surf (detekce významných bodů).....	32
3.4.4 Neuronové sítě.....	34
3.5 PROGRAM MATLAB .....	35
3.5.1 Popis prostředí MATLABu.....	36
3.5.2 Image Processing Toolbox.....	37
3.5.3 Image Acquisition Toolbox .....	37
3.5.4 Statistics and Machine Learning Toolbox .....	37
3.5.5 MATLAB Support Package for USB Webcams.....	38
<b>II PRAKTICKÁ ČÁST .....</b>	<b>39</b>
<b>4 POPIS WEBOVÉ KAMERY A SNÍMANÉHO PROSTŘEDÍ .....</b>	<b>41</b>
4.1 WEBOVÁ KAMERA - MICROSOFT LIFECAM HD-3000 .....	41
4.2 MODEL SNÍMANÉHO PROSTŘEDÍ.....	41
4.3 NASTAVENÍ KAMERY V PROGRAMU MATLAB.....	42
4.3.1 Program pro Záznam videa .....	43
<b>5 DETEKCE A KLASIFIKACE OSOB.....</b>	<b>45</b>

5.1	TRÉNOVÁNÍ OBJEKTŮ .....	47
5.2	AKTUALIZACE POZADÍ.....	50
5.3	ODEČÍTÁNÍ POZADÍ (DETEKCE POPŘEDÍ) .....	50
5.4	ODSTRANĚNÍ STÍNŮ .....	51
5.5	MORFOLOGICKÉ OPERACE .....	53
5.6	NALEZENÍ OBJEKTU .....	56
5.7	KLASIFIKACE OBJEKTU .....	57
<b>6</b>	<b>TESTOVÁNÍ A VYHODNOCENÍ VÝSLEDKŮ .....</b>	<b>62</b>
6.1	KLASIFIKACE OSOBY .....	62
6.2	KLASIFIKACE OSOBY S KOLEM.....	63
6.3	KLASIFIKACE OSOBY S KUFREM .....	65
6.4	DETEKCE NEZNÁMÉHO OBJEKTU .....	66
6.5	KLASIFIKACE ZA ZHORŠENÝCH PODMÍNEK .....	67
6.6	VYHODNOCENÍ VÝSLEDKŮ.....	69
6.6.1	Testovaný objekt 1.....	69
6.6.2	Testovaný objekt 2.....	70
6.6.3	Testovaný objekt 3.....	72
6.6.4	Testovaný objekt 4.....	73
6.7	MOŽNOSTI ROZŠÍŘENÍ.....	76
	<b>ZÁVĚR .....</b>	<b>77</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>78</b>
	<b>SEZNAM OBRÁZKŮ.....</b>	<b>82</b>
	<b>SEZNAM TABULEK .....</b>	<b>85</b>

## ÚVOD

Počítačové vidění a oblast digitálního zpracování obrazu je progresivně rozvíjející se odvětví výpočetní techniky, které je zaměřeno na získávání informací z videa nebo z obrazu snímaného kamerou. V současné době je trendem pořizovat si monitorovací a bezpečnostní kamery, například do městských veřejných prostor, ale i pro hlídání soukromých objektů. Záznamy z takových kamer se dají využít v oblasti počítačového vidění pro sledování a rozpoznávání objektů. Za použití počítačového vidění je v průmyslovém odvětví řízeno stále více strojů a velkých automatizovaných výrobních center.

Diplomová práce je zaměřena na segmentaci a rozpoznávání objektů ze statické kamery, která má sloužit jako doplněk vstupních turniketů, za účelem klasifikace vstoupivších osob. Dalšími využitími je mnoho, např.: hlídání objektů, obchodů, detekce a klasifikace podezřelých osob ve veřejném prostranství nebo detekce jiných objektů. Využití se odvíjí od toho, na jaké objekty či předměty je natrénován SVM klasifikátor.

Cílem této práce je navrhnout vhodnou programovou metodiku pro automatickou detekci a klasifikaci objektů. Nejprve bude potřeba nastudovat problematiku rozpoznávání obrazu, zpracovat literární rešerši a v neposlední řadě si objasnit základní postupy zpracování obrazových dat. V praktické části bude následovat výběr patřičných metod, realizace vlastního programu v MATLABu a vyhodnocení dosažených výsledků.

## I. TEORETICKÁ ČÁST



## 1 ÚVOD DO PROBLEMATIKY

Tématu detekce a klasifikace osob v obraze z kamery je v poslední době věnována spousta pozornosti. S tím souvisí i skutečnost, že obdobným tématům se v rámci zadání diplomové práce již věnovalo několik studentů. Jako příklad uvedu dvě diplomové práce. V první z nich s názvem „Detekce mobilního robota zpracováním obrazu“ se student zaměřil na lokalizaci robota pomocí dvou různých kamer, kde jako metodu segmentace zvolil adaptivní prahování, více viz [1]. Druhá diplomová práce zabývající se podobným tématem nese název „Rozpoznávání objektů s využitím programu Matlab“. V této práci se student věnoval klasifikaci jednoduchých objektů pomocí neuronových sítí, více viz [2].

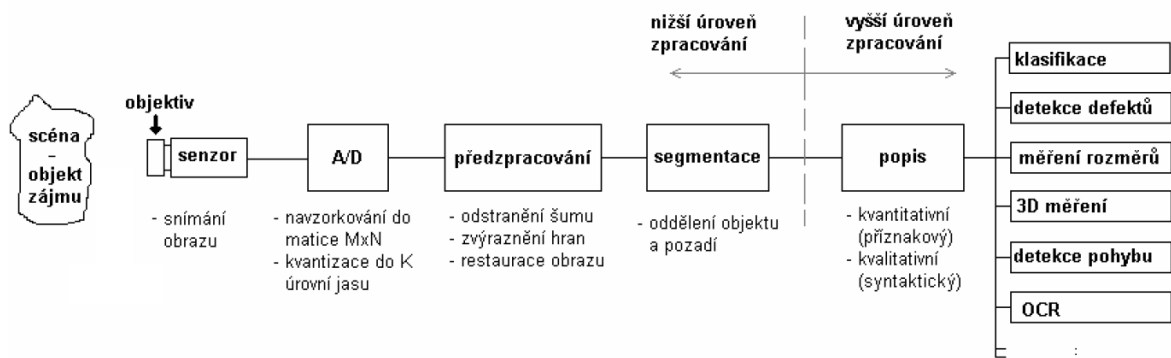
Cílem této diplomové práce je navržení metody pro vylepšení turniketů se světelnými bránami (IR čidly) pomocí kamerového systému.

Turnikety pracují na principu přerušování paprsků dopadajícího světla na senzor. Těchto senzorů je několik desítek a při průchodu objektu jsou paprsky postupně přerušovány. Data jsou průběžně zpracovávána softwarem, který vyhodnocuje vzniklé odezvy. Nicméně IR čidla nejsou zcela dostatečná, proto bylo nutné jejich doplnění o schopnost klasifikace objektů. Pro realizaci se nabízelo řešení prostřednictvím levných webových kamer až po velmi drahé 3D kamery za cenu desítek tisíc korun. Z důvodu co největší úspornosti v praxi byla pro navržení vhodné metody vybrána levnější varianta, tedy webová kamera. Umístění webové kamery bylo zvoleno přímo nad turnikety z důvodu nemožnosti zabránění detekci a následné klasifikaci objektů. Takto umístěná kamera má přehled nad celou scénou a nedá se nijak obejít.

## 2 PRÁCE S OBRAZEM

Pro možnost zpracování obrazu na počítači je nejprve nutné převést obraz do digitální podoby. Tím je myšleno převedení okolního (analogového) reálného světla, které je okem viditelné, do digitální podoby. Jedná se o konverzi optické veličiny na elektrickou. Oblast digitálního zpracování obrazu je dnes již velmi rozvinutou disciplínou, uplatňuje se hlavně v široké škále průmyslových využití, kde je velké množství automatizačních procesů založeno na metodách rozpoznání obrazu. [8]

Řetězec posloupností zpracování obrazové informace lze vyjádřit následujícím obrázkem (viz Obr. 1).



Obr. 1. Řetězec snímání a zpracování obrazu [5]

### 2.1 Digitalizace

Čidla a senzory pro vstup obrazové funkce jsou většinou zdrojem spojitého signálu. Pokud chceme obrazovou funkci zpracovávat na počítači, musíme ji digitalizovat. Digitalizace spočívá ve vzorkování obrazu v matici o rozměrech  $M * N$  bodů a v kvantování spojitě jasové funkce do  $K$  intervalů. Kvantováním začne nabývat jasová funkce v digitálních obrazech celočíselných hodnot. Čím jemnější je vzorkování a kvantování (čím větší je rozlišení matice  $M * N$ ), tím lépe je aproximován původní spojitý obrazový signál. [7][5]

V souvislosti se vzorkováním spojitě obrazové funkce je třeba vyřešit další otázky. Prvně je nutné určit interval vzorkování, následně vybrat vzorkovací mřížku. [5][4]

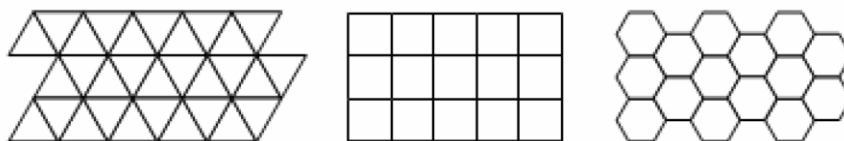
#### 2.1.1 Interval vzorkování

Intervalem vzorkování, jinak řečeno rozlišením obrazu, se rozumí vzdálenost mezi nejbližšími vzorkovacími body v obraze. Interval vzorkování je definován Shannonovým teoré-

mem zobecněným pro vícerozměrné signály. Interval vzorkování je nutné volit tak, aby byl menší nebo roven polovině rozměru nejmenších detailů v obraze. [7][5]

### 2.1.2 Volba vzorkovací mřížky

Volba vzorkovací mřížky je chápána jako výběr plošného uspořádání bodů při vzorkování. Nejčastěji je používána pravidelná čtvercová mřížka, kromě té však existují další tři pravidelné mnohoúhelníky, jejichž síť úplně pokrývá rovinu. Typy vzorkovacích mřížek jsou zobrazeny na následujícím obrázku (Obr. 2). [5]



Obr. 2. Typy vzorkovacích mřížek [5]

Jednomu vzorkovacímu bodu odpovídá v digitalizovaném obraze obrazový element, tj. pixel. Po uspořádání do vzorkovací mřížky pokrývají pixely celý digitalizovaný obraz. [7]

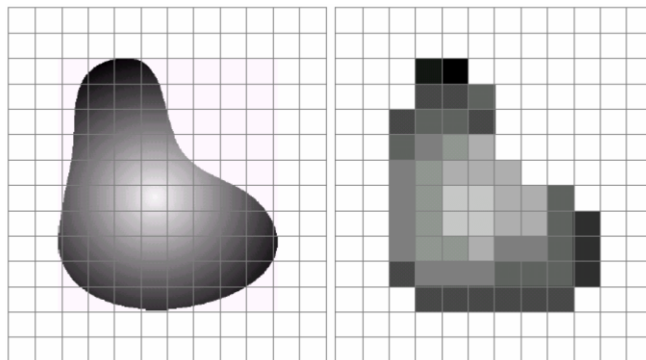
### 2.1.3 Kvantování hodnot obrazové funkce

Kvantování představuje diskretizaci jasových úrovní, tedy rozdělení spojité jasové úrovně všech obrázků do  $K$  úrovní. Jedná se o jasové rozlišení pixelů (viz Obr. 3). [8]

Pro počítačové zpracování obrazu je dán počet intervalů druhou mocninou.

$$K = 2^b \quad (1)$$

Počet kvantovacích úrovní udává, jak přesně budou vyjádřeny jemné detaily obrazu. Obraz musí být proto dostatečně velký, aby nevznikaly nepravdivé obrysy a citlivost zařízení se blížila citlivosti lidského oka.



Obr. 3. Spojitý obraz ve vzorkovací mřížce (vlevo) a výsledný obraz po vzorkování a kvantování hodnot (vpravo) [8]

Hodnota kvantování u černobílého obrazu může nabývat pouze hodnot 0 a 1 ( $K = 2^1$ ), v případě RGB obrazu může nabývat hodnot od 0 až 256 ( $K = 2^8$ ).

## 2.2 Reprezentace obrazových dat

Obraz (obrazová data) lze reprezentovat jako spojité rozložení jasu (intenzity a optické hustoty) v rovině. Rozložení hodnot jasu je definováno obrazovou funkcí:

$$f(x, y) \quad \text{pro statický obraz} \quad (2)$$

$$f(x, y, t) \quad \text{pro dynamický obraz (proměnný v čase } t) \quad (3)$$

Funkce  $f$  je reálná spojitá funkce s reálnými argumenty ( $x, y$  definují bod v rovině a  $t$  představuje hodnotu v čase).

Pro další zpracování je obraz prakticky vždy diskretizován v čase a lze si jej představit jako nespojitou sekvenci obrazových funkcí  $f(x, y)$ .

Definiční obor  $D$  hodnot obrazové funkce je určen následovně:

$$x \in (x_{min}, x_{max}), y \in (y_{min}, y_{max}) \quad (4)$$

Obor hodnot  $H$  obrazové funkce vyjádříme jako:

$$0 \leq f(x, y) \leq f_{max} \quad (5)$$

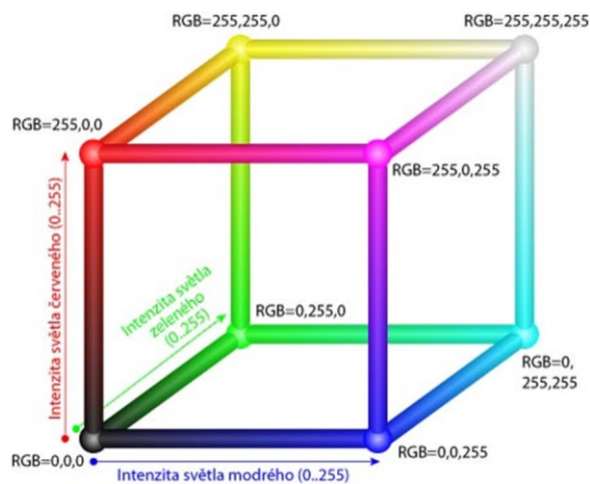
Takto definovaná obrazová funkce je schopná reprezentovat pouze šedo tónový (monochromatický) obraz (obraz s proměnnou intenzitou jedné barvy). Barevný obraz lze definovat jako soubor několika obrazových funkcí:

$$f_R(x, y); f_G(x, y); f_B(x, y) \quad (6)$$

Zde jednotlivé funkce reprezentují intenzitu jasu pro danou barevnou složku ve zvoleném barevném modelu (RGB, HSV, atd.). [5]

### 2.2.1 RGB

RGB je aditivní (sčítací) model, kde barvu vidíme jako kombinaci tří složek Red (červené), Green (zelené) a Blue (modré). Tento model pracuje na principu přidávání barev do černé. U RGB modelu nejde o fyzické míchání barev, ale o míchání tří světelných záření. Při sloučení těchto tří barev (možná intenzita 0 až 255) dojde k sečtení, po kterém vznikne výsledná barva. Při plné intenzitě všech tří barev vzniká bílá barva (255;255;255), při nulové hodnotě vzniká černá (0;0;0). Výše uvedené je zobrazeno na následujícím obrázku modelu krychle RGB (viz Obr. 4).[4][14]



Obr. 4. RGB model krychle [13]

Na obrázku níže (Obr. 5) je snímek zachycený kamerou, který je zobrazený v RGB modelu.



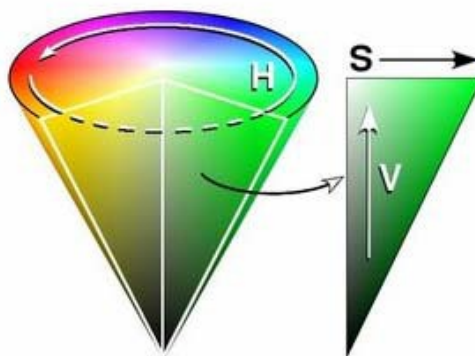
Obr. 5. Obrázek zobrazený v RGB modelu

### 2.2.2 HSV

Model HSV vyplňuje pravidelný šestiboký jehlan s vrcholem obráceným dolů. Ve vrcholu je umístěna barva s minimálním jasem, tedy černá barva. Tento model se nejvíce přibližuje fyzickému míchání barev, kdy se nové odstíny barev vytvářejí přidáním bílé a černé do jiných barev, za účelem vytvoření jejich nádechu, viz Obr. 6.[14]

Model HSV skládá barvu ze tří složek:

- Hue (barevný tón) – určuje převládající spektrální barvu. Udává se jako poloha na barevném kole ( $0^\circ$  až  $360^\circ$ ).
- Saturation (sytnost barvy) – určuje příměs jiné barvy, představuje množství šedi v poměru k odstínu v procentech (0% šedá až 100% plně sytá barva).
- Value (hodnota jasu) – určuje množství bílého světla. Vyjadřuje, kolik světla barva odráží. [13]



Obr. 6. HSV Barevný model [13]

Následující obrázky představují (Obr. 7) oddělené a zobrazené jednotlivé složky HSV modelu.



Obr. 7. Obrázky se zobrazenými jednotlivými složkami HSV (zleva: Hue, Saturation, Value)



### 2.2.3 Šedo tónový obraz

Šedo tónové obrazy, někdy označovány jako monochromatické, jsou většinou reprezentovány 8 bity na pixel. Mohou tedy nabývat hodnot 0 až 256, kde 0 představuje černou barvu, hodnota 256 představuje bílou barvu a rozmezí mezi těmito hodnotami jsou odstíny šedi, viz následující obrázek (Obr. 8). Šedo tónové obrazy jsou využívány v technice prahování (threshold) za účelem analýzy histogramu v digitalizovaném obrazu. [4][6]



Obr. 8. Šedo tónový obraz

### 2.2.4 Binární obraz

Binární obraz je z hlediska barevné hloubky vždy reprezentován maticí s hodnotami 0 a 1. Bitová hloubka u binárních obrazů je vždy rovna jedné. Ačkoliv binární obrazy obsahují mnohem méně informací než šedo tónové obrazy, obsahují informace o tvaru a velikosti, které jsou důležité pro lokalizaci objektu. Binární obraz se nejčastěji používá pro definici obrazové masky. Násobením prvků obrazu a masky se stejnou velikostí lze jednoduše odfiltrovat odpovídající části obrazu, viz následující obrázek s binární mapou objektu (viz Obr. 9). [3][6]



Obr. 9. Binární maska

### 3 ANALÝZA OBRAZU A POHYBU

V následujících kapitolách jsou popsány metody pro odstranění šumu, detekci pohybu a rozpoznání objektů. Jedná se o literární rešerši konkrétních metod, které by s ohledem na zadání bylo možné využít pro tuto diplomovou práci.

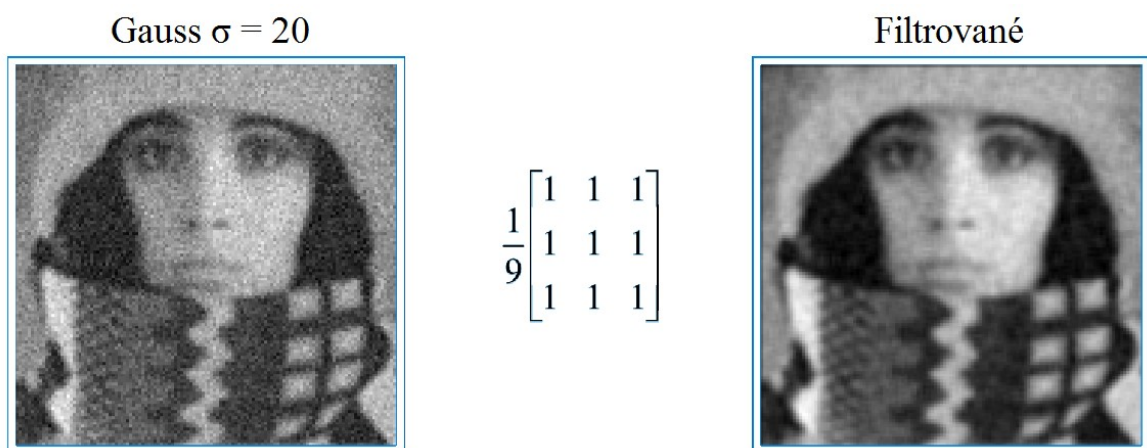
#### 3.1 Odstranění šumu

Odstranění šumu je hlavním úkolem předzpracování obrazu. Cílem odstranění šumu je zredukovat různé obrazové poruchy a jiné rušivé elementy. Při odstranění šumu se zlepšuje kvalita obrazu, která je podstatná pro následující zpracování obrazu. [19]

##### 3.1.1 Konvoluční filtrace

Konvoluční filtrace patří k nejjednodušším metodám pro potlačení šumu. Jedná se o obyčejné průměrování okolních bodů (pixelů). Tato metoda je založena na odfiltrování kmitočtových složek mimo pásmo užitečného signálu. Největší nevýhodou konvoluční filtrace je zhoršení prostorové rozlišovací schopnosti. Výhodou je jednoduché konvoluční jádro provádějící průměrování. [15][16]

Průměrování pomocí konvoluční masky, kde nová hodnota každého pixelu je určena jako průměr z hodnot pixelů v okolí masky (např. 3x3), je zobrazeno na Obr. 10. [5]



Obr. 10. Filtrace obrazu maskou 3x3 [16]

Pro průměrování je možné využít několik dalších masek, u kterých je dokonce možné upravovat váhu okolních bodů, viz matice  $h_2, h_3$ .

$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; h_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; h_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (7)$$

U masky  $h_3$  je použit separabilní filtr, který je možné rozložit na dva 1D filtry.

$$h_{31} = [1 \quad 2 \quad 1]; h_{32} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}; g = \frac{1}{16} [h_{31} \times (h_{32} \times f)] \quad (8)$$

## 3.2 Detekce objektů (segmentace obrazu)

Segmentace je důležitý proces, který slouží k rozpoznávání a oddělování objektů od pozadí a detekci celých oblastí, kterými jsou objekty reprezentovány v obraze. Segmentace se snaží obraz rozdělit do částí, které úzce souvisí s předměty či oblastmi reálného světa. Hlavním cílem je oddělit objekt od pozadí a zredukovat zpracovaná data. Jakmile je obraz segmentován, výsledné jednotlivé objekty (nebo oblasti) mohou být popsány, analyzovány a klasifikovány. [3][8]

Segmentaci obrazu lze získat různými metodami. Každá z metod je pro své vlastnosti aplikovatelná na různé situace. V rámci této kapitoly se zaměřím pouze na metody, které by bylo možné aplikovat na zadání práce, tj. prahování, odečítání pozadí, detekce hran a 3D hloubková mapa.

### 3.2.1 Prahování

Prahování je metodou nejjednoduššího možného rozpoznání obrazu, využívaného především pro předměty na bílém pozadí. Důležitou roli u této metody hrají histogramy. Po převedení obrázků do stupňů šedi vznikne histogram intenzity. [12]

Histogram udává grafickou informaci o rozložení jednotlivých jasových složek v obraze. Histogram digitálního obrazu s hodnotami jasu  $\langle 0, L - 1 \rangle$  je diskrétní funkcí,  $L$  udává počet úrovní šedi. [10]

$$h(r_k) = n_k \quad (9)$$

$r_k$  je  $k$ -tá úroveň šedi,  $n_k$  je celkový počet pixelů v obraze s určitou úrovní šedi  $r_k$

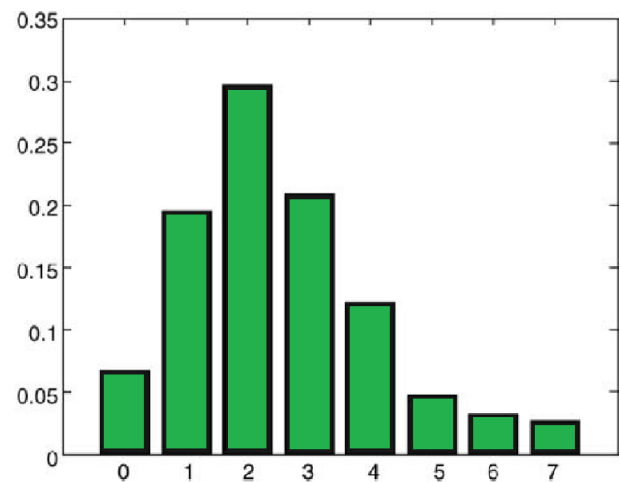
U normalizovaného tvaru jsou jednotlivé složky histogramu poděleny celkovým počtem pixelů  $n$  v obraze, viz následující rovnice: [8]

$$p(r_k) = \frac{n_k}{n} \quad (10)$$

$k = 0, 1, 2, \dots, L - 1$  a  $p(r_k)$  udává pravděpodobnost výskytu úrovně šedi  $r_k$

Histogramy jsou obvykle zobrazeny pomocí sloupcového grafu, kde každý sloupec představuje jednu hodnotu úrovně šedi. Výška sloupce je úměrná počtu pixelů, které odpovídají konkrétní úrovni šedi, viz Obr. 11. [10]

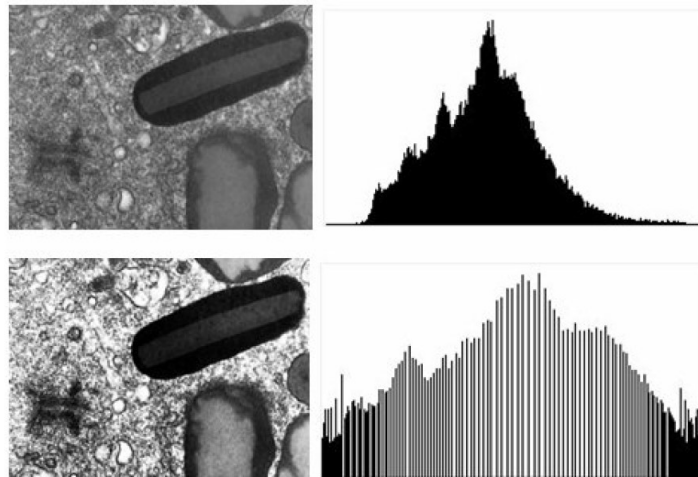
Gray Level ( $r_k$ )	$n_k$	$p(r_k)$
0	1120	0.068
1	3214	0.196
2	4850	0.296
3	3425	0.209
4	1995	0.122
5	784	0.048
6	541	0.033
7	455	0.028
Total	16,384	1.000



Obr. 11. Příklad histogramu s osmi úrovní šedi [10]

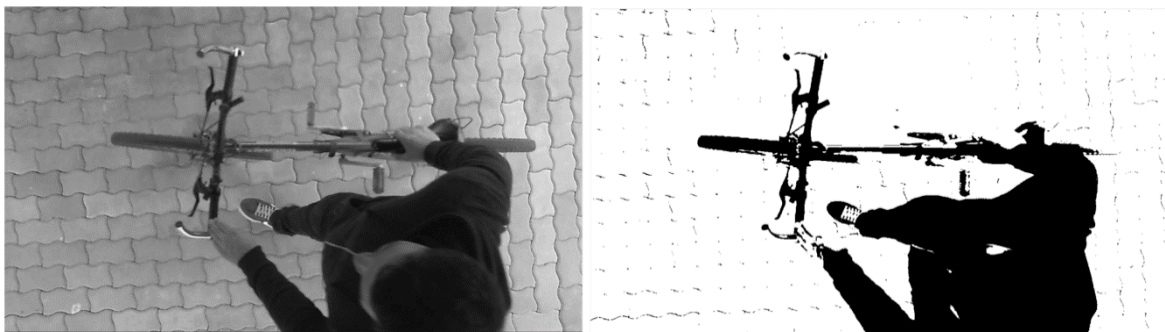
Ekvalizace histogramu (vyrovnání histogramu) je algoritmus, který změní rozložení intenzit v obraze tak, aby se v něm vyskytovaly intenzity přibližně se stejnou četností. Jeho účelem je tedy zvýšení kontrastu. [5]

Ekvalizace zvyšuje kontrast pro úrovně jasu blízko maxim histogramu a snižuje kontrast blízko minim histogramu. Histogram před a po ekvalizaci je uveden na Obr. 12.[8]



Obr. 12. Původní obraz s histogramem (nahore)  
a ekvalizovaný obrázek s histogramem (dole) [5]

Následující obrázek (Obr. 13) vykresluje situaci před a po prahování.



Obr. 13. Šedo tónový obraz před prahováním (vlevo) a po prahování (vpravo)

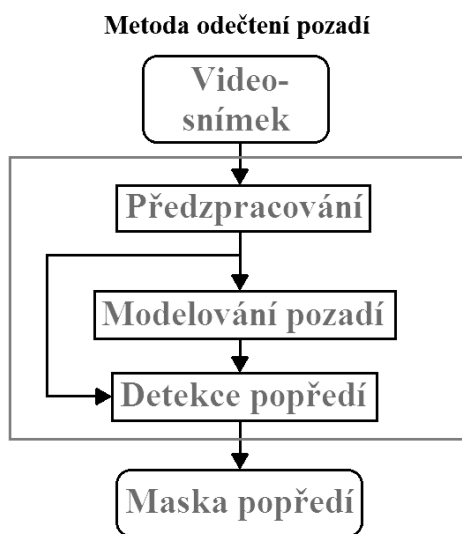
Tato metoda nebyla vybrána pro realizaci z důvodu nemožnosti zcela odfiltrout pozadí, v případě nastavení většího prahu by se odfiltrovala i část postavy.

### 3.2.2 Odečítání pozadí

Odečítání pozadí, známé také jako detekce popředí, je metodou v oblasti zpracování obrazu a počítačového vidění, kde se popředí obrazu extrahuje pro další zpracování obrazu.

Odečítání pozadí je široce využívaná technika pro vytváření masky v popředí (konkrétně binární obraz obsahující pixely patřící k pohybujícím se objektům ve scéně) pomocí statické kamery. Metoda odečítání pozadí odečítá mezi aktuálním snímkem a modelem pozadí obsahujícím statickou část scény nebo obecně vše, co lze považovat za pozadí, a to vzhledem k charakteristice pozorované scény. Následně vypočítá masku v popředí.

Níže na obrázku (Obr. 14) je znázorněno základní schéma algoritmu se třemi hlavními kroky metody odečítání pozadí.

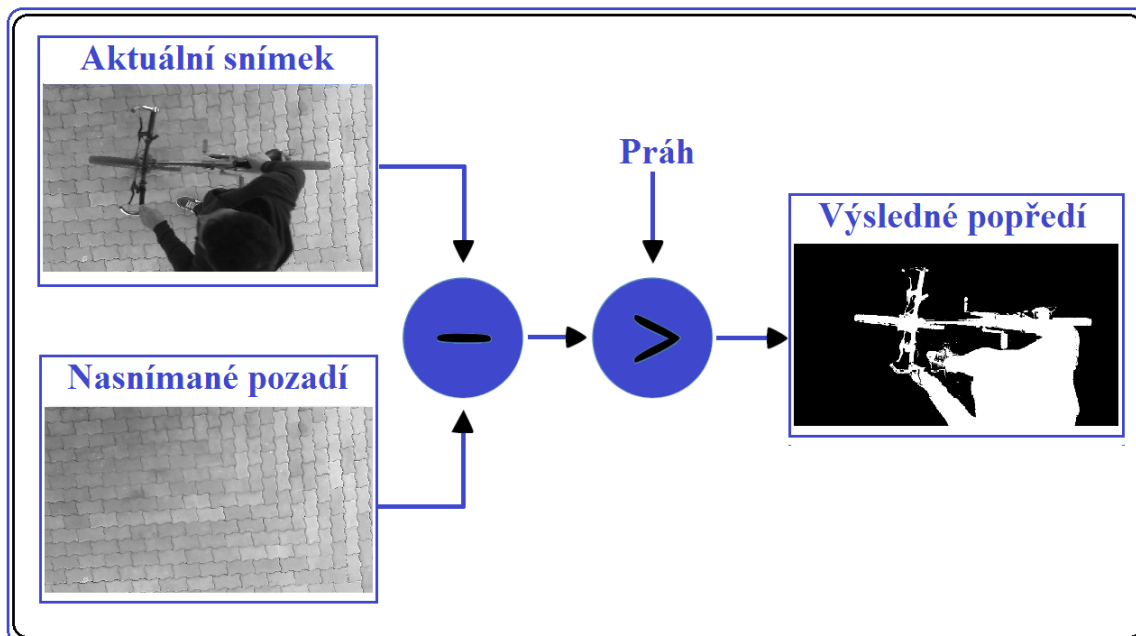


Obr. 14. Schéma algoritmu odečítání pozadí

Aplikací metody odečítání pozadí je získána binární maska, která vyjadřuje dělení bodů aktuálního snímku na body popředí a body pozadí. Černé pozadí zastupuje hodnota 0, bílé popředí zastupuje hodnota 255.

Pro detekci popředí je nutné správně nastavit prahovou hodnotu. Prahová hodnota určuje, zda ještě patří pixely k pozadí, či nikoliv. Odečítání pozadí lze provádět i se snímky v RGB, ale bylo by nutné odečítat každý kanál zvlášť. Z toho důvodu se používá odečítání pozadí s šedo tónovými snímky, kde je pouze jeden kanál. Na obrázku (Obr. 15) je zobrazeno ideálně odečtené výsledné popředí.





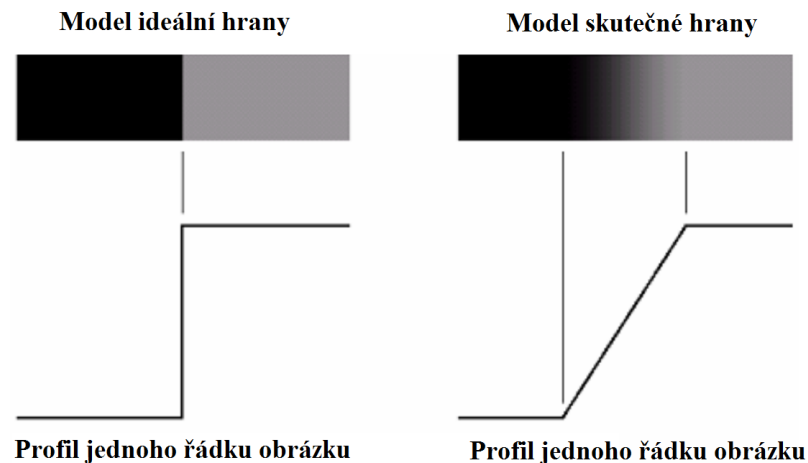
Obr. 15. Algoritmus pro získání masky popředí

Metoda byla zvolena pro realizaci v programu MATLAB z důvodu jednoduchosti a efektivity.

### 3.2.3 Detekce hran

Detekce hran je postupem v digitálním zpracování obrazu, který slouží k nalezení oblastí pixelů, u kterých se podstatně mění jas. Hrana může být definována jako hranice mezi dvěma obrazovými oblastmi, které mají odlišné charakteristiky podle některých vlastností (např. úroveň šedé, barvy nebo textury). Hrana v objektu se nemusí krýt s hranicí mezi objekty ve scéně. Hrany mohou vznikat a zanikat v závislosti na pohybu osob nebo věcí. [3][8]

Níže na obrázku (Obr. 16) je uveden model ideální hrany a model skutečné hrany, která je vlivem vzorkování a jiných vlivů rozmazána. Tloušťka hrany je určena délkou skoku. [8]



Obr. 16. Model ideální a skutečné hrany [8]

Na následujícím obrázku (Obr. 17) je vyobrazena aplikace detektoru hran na šedo tónový obraz a na binární mapu po odečtení pozadí.



Obr. 17. Detekce hran aplikovaná na šedo tónový obraz (vlevo) a na binární mapu (vpravo) Metodu by bylo možné použít stejně jako odečítání pozadí, pouze s tím rozdílem, že by se k učení klasifikátoru použily místo binárních map obrysy objektů v podobě hran. Detekce hran je ale metoda časově náročnější a na výpočty by bylo zapotřebí lepšího hardwaru, proto nebyla zvolena pro realizaci.

### 3.2.4 3D hloubková mapa

Nejjednodušší způsob získání 3D hloubkové mapy je pomocí 3D kamery. Pasivní 3D kamery využívají triangulace a aktivní 3D kamery promítají do scény matici IR světla a následně vyhodnocují, jak se toto světlo odráží od objektů.

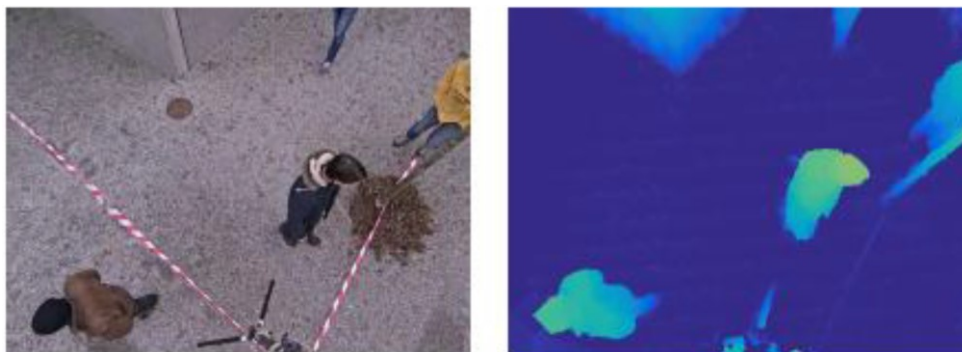
U pasivních 3D kamer je snímání založeno na stereo snímání, obraz je získáván ze dvou oddělených kamer. Obě kamery musí být prvně zkalibrovány, aby bylo následně možné využít triangulace pro určení přesných souřadnic ve 3D scéně. Zjištěním přesných souřadnic v jednotlivých snímcích je získána informace o vzdálenosti pixelu od senzoru.

Čím větší je vzdálenost objektu od kamer, tím je v hloubkové mapě tmavší pixel (viz Obr. 18). [23]



Obr. 18. Obraz z kamery a hloubková mapa [23]

Relativně novým zařízením jsou speciální aktivní 3D kamery s označením Time-of-Flight, které využívají detekce zpoždění mezi vyslaným modulovaným světlem a odraženým signálem. Přesnost zařízení je omezena výkonem vestavěného IR světla. Kamery jsou buď barevné (RGB), nebo černobílé. Níže na obrázku (Obr. 19) je výstup z barevné RGB kamery s odpovídající hloubkovou mapou. [23]



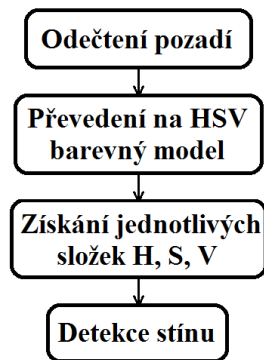
Obr. 19. Obraz barevné RGB aktivní kamery a odpovídající hloubková mapa [23]

Jedná se o metodu, u které se nemusí získávat maska popředí. Výstupem z kamery je hloubková mapa, která by se dala ihned využít k detekci a klasifikaci. Tato metoda nebyla vybrána z důvodů vysokých cen 3D kamer, když cena vhodných kamer pro tuto úlohu se pohybuje v řádech desítek tisíc korun.

### 3.3 Odstranění stínů

Stíny v obraze zabraňují správnému vyhodnocování objektů, proto je nutné je z video snímku odfiltrovat. Pro odstranění stínů se nejčastěji využívá převodu na takový barevný

model, ve kterém je možné jednotlivé složky oddělit, jako je tomu např. u HSV modelu (viz kapitola 2.2.2). Tato metoda pracuje na principu toho, že stín dopadající na nějakou plochu nezmění barvu, ale mění jen sytost a jas. Algoritmus detekce stínu je patrný na Obr. 20. [10]



Obr. 20. Algoritmus detekce stínu

Kamera sejme obraz v barevném modelu RGB. Pro odstranění stínů je nutné obraz převést do modelu HSV, u kterého dochází k filtraci jednotlivých složek. Následně je obraz převeden na šedo tónový snímek, který je odečten od pozadí, jehož výstupem je binární mapa.

### 3.4 Klasifikace objektu

Posledním krokem při zpracování obrazu je klasifikace již segmentovaných objektů.

Úkolem klasifikace je zařazení všech objektů nalezených v obraze do skupiny předem známých tříd. Klasifikace probíhá na základě předem naučeného klasifikátoru s trénovací množinou.

V následujících kapitolách bylo vybráno několik nejznámějších metod pro klasifikaci objektu. Vybrány byly metody HOG, SVM, SURF a Neuronové sítě.

Pro tuto úlohu byla zvolena SVM z důvodu rychlosti a možného budoucího využití pro klasifikaci v reálném čase.

#### 3.4.1 Histograms of Oriented Gradients (HOG)

Metoda HOG je založena na histogramech orientovaných gradientů, byla vyvinuta za účelem rozpoznávání lidských postav v obraze. Vychází z myšlenky popisu objektů pomocí gradientů, kde jednotlivé gradienty jsou reprezentovány svou velikostí a směrem. Obraz se rozdělí na malé oblasti (buňky). Pro každou buňku se vypočítá 1D histogram, který je vy-

počítán ze všech oblastí buňky. Výsledkem metody HOG mohou být jak barevné, tak šedo tónové obrazy. [17]

Celý obraz je rozdělen do bloků o stejné velikosti, kde každý z těchto bloků je reprezentován pomocí několika buněk viz Obr. 22. Každá buňka zachycuje určitou oblast, např. 8 x 8 pixelů, a je hlavním nositelem informace. V každé buňce se určí orientace gradientu, výsledné gradienty všech buněk by měly vymezovat hledaný objekt.

### Výpočet gradientu

Výsledný deskriptor je vysoce ovlivněn procesem výpočtu gradientu. Vstupní obraz se v prvním kroku filtruje Gaussovým jádrem o různém parametru  $\sigma$ . Následně se aplikují masky pro výpočet diskrétního gradientu. [17]

1-D masky:

$$[-1, 1], [-1, 0, 1], [1, -8, 0, 8, -1] \quad (2)$$

2-D masky:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3)$$

Z vyhodnocených výsledků testů se volí nejlepší kombinace parametru  $\sigma$  a filtrační masky pro stanovení gradientu metodou HOG, která získá gradient pomocí konvoluce Gaussovsky filtrovaného obrazu  $I$ ,  $\sigma = 0$ , maska  $[-1, 0, 1]$ , resp.  $[-1, 0, 1]^T$ :

$$I_x = I * [-1, 0, 1] \quad (1)$$

$$I_y = I * [-1, 0, 1]^T \quad (2)$$

Kde  $*$  znamená konvoluce. [17]

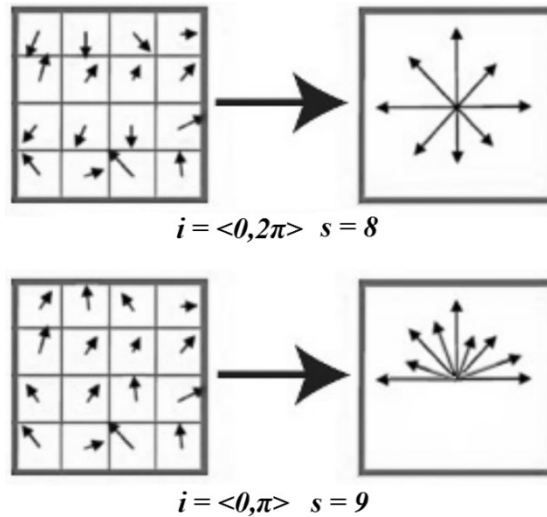
### Histogram orientací gradientů

V dalším kroku po získání obrazů  $I_x$  a  $I_y$  bude pro každý pixel vypočtena velikost gradientu  $m(x, y)$  a směr gradientu  $\theta(x, y)$ :

$$m(x, y) = \sqrt{I_x^2 + I_y^2} \quad (3)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{I_y}{I_x} \right) \quad (4)$$

Výše je uveden výsledný histogram orientací, kde počet binů je určen intervalem  $i$  a počtem sektorů  $s$ . Interval  $i$  udává rozsah, pro který se zjišťuje směr gradientu  $\theta(x, y)$ . Počet sektorů  $s$ , definuje, kolik je stejně velkých výsečí, na něž je interval  $i$  rozdělen viz Obr. 21.

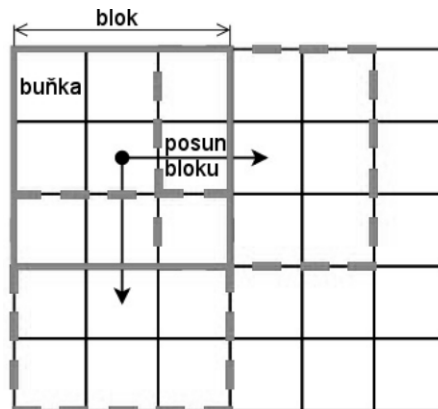


Obr. 21. Histogram orientací gradientů pro různé hodnoty intervalu  $i$  a počtu binů  $s$  [17]

### Konstrukce deskriptoru

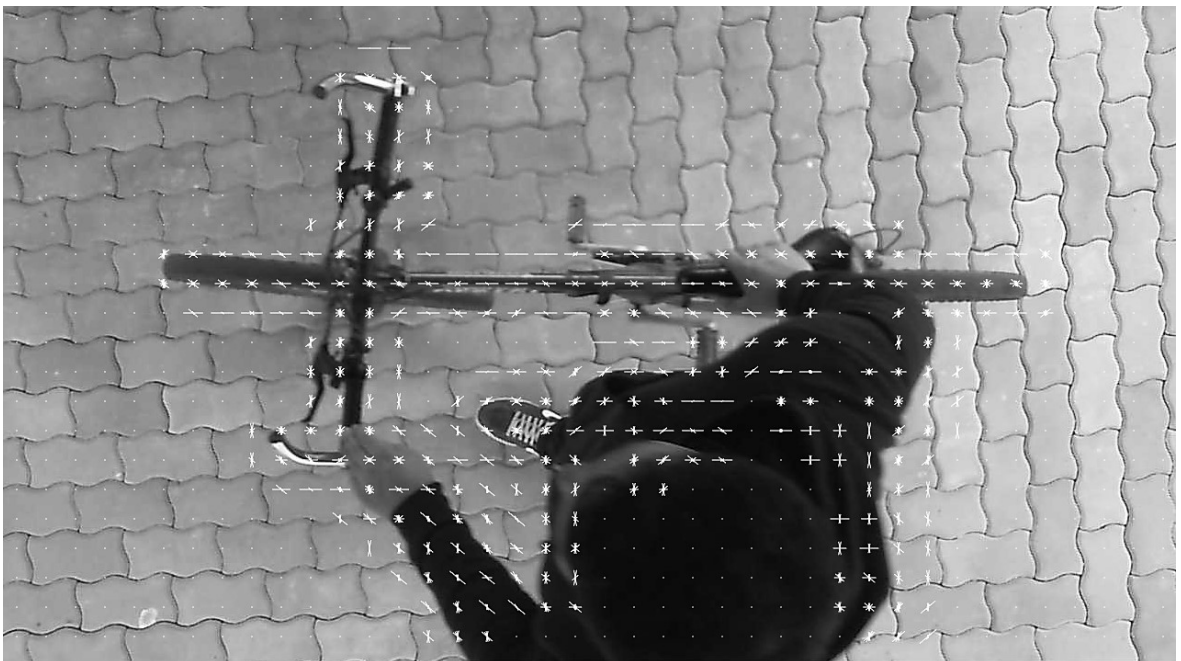
Rovnoměrným rozdělením vstupního obrazu do čtvercových bloků o hraně  $k$  pixelů je získán deskriptor HOG. Vzniklé bloky jsou dále rozčleněny na buňky o hraně  $l$  pixelů ( $l < k$ ), pro které jsou vypočteny normované histogramy orientací gradientů. Následně jsou výsledné histogramy buněk zprůměrovány v jeden, který tvoří popis bloku HOG. V dalším kroku se blok posune o zvolený počet pixelů. Konečný HOG deskriptor je zřetěžením všech popisů jednotlivých bloků uvnitř obrazu, viz Obr. 22. [17]





Obr. 22. Princip konstrukce HOG deskriptoru [17]

Na Obr. 23 je vyobrazena vizualizace metody HOG vytvořená v programu MATLAB.



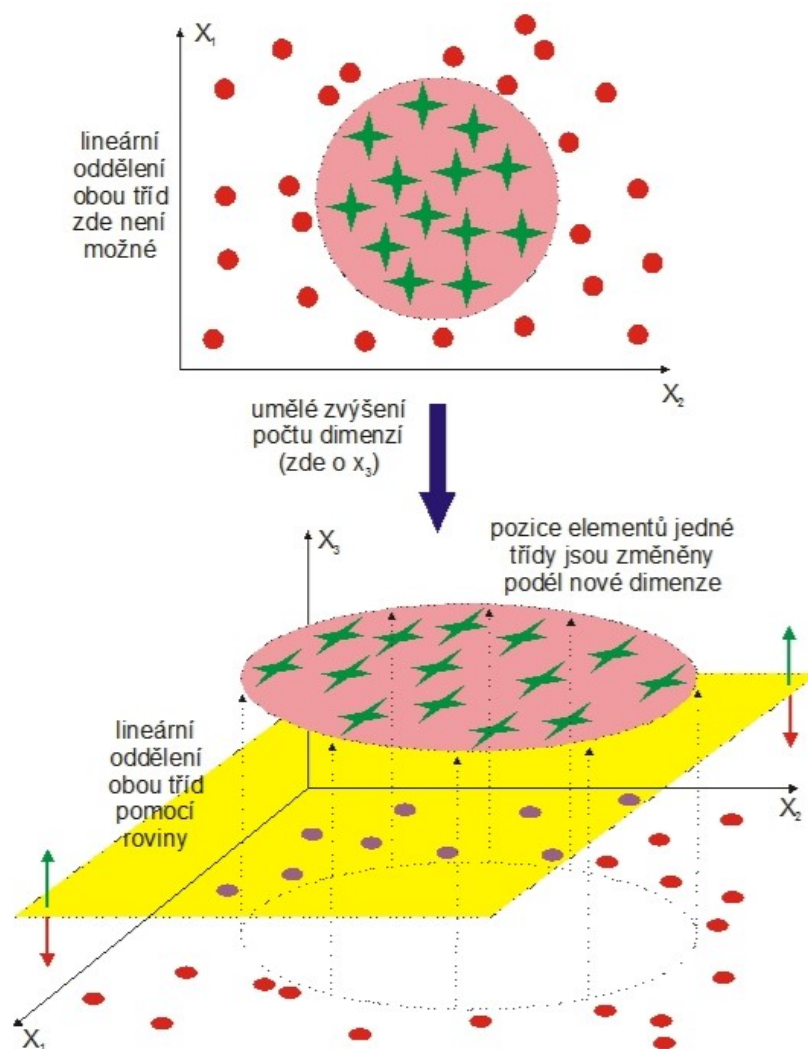
Obr. 23. Vizualizace metody HOG při pohledu shora

### 3.4.2 Suport vector machines (SVM)

Suport vector machines (metoda podpůrných vektorů) patří k novým metodám strojového učení, které spadají do skupiny tzv. jádrových algoritmů. Úkolem této metody je využití efektivních algoritmů k nalezení lineární hranice a zároveň schopnost reprezentovat vysoce složité nelineární funkce. Základním principem metody je převod daného původního

vstupního prostoru do jiného, vícedimensionálního, kde se již dají od sebe dělit třídy lineárně. [18]

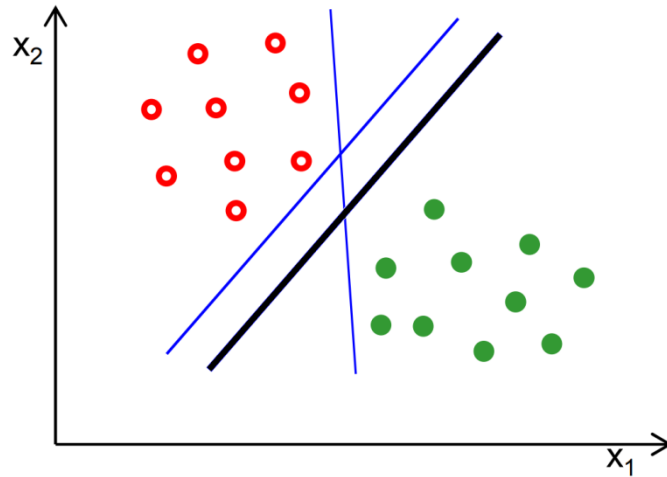
Princip metody je vyobrazen na následujícím obrázku (viz Obr. 24). V původním dvourozměrném prostoru se nacházejí dvě třídy, které jsou odděleny nelineárně kružnicí. Přidáním nové dimenze vznikne možnost přidat prvkům třídy uvnitř kružnice další souřadnici, která je posune nahoru, nebo dole podél osy  $x_3$ . Proto je možné pro oddělení obou tříd použít rovinu rovnoběžnou s rovinou danou osami  $x_1$  a  $x_2$ . Lineární hranici je nutné umístit tak, aby byla co nejefektivnější z hlediska kategorizace budoucích dat, které nebylo možné použít při tréninku. [18]



Obr. 24. Princip vzniku možnosti lineárního oddělení dvou tříd s nelineárními hranicemi pomocí přidané dimenze. [18]

V případě, že jsou klasifikační data lineárně separabilní, existuje nekonečně mnoho hranic, jimiž je možno rozdělit prostor tak, aby byly objekty rozděleny přímkou. Přímka udává

rozdělení objektů z jedné a druhé třídy. Nejlepším a nejrobustnějším řešením pro klasifikaci nových objektů, které nejsou součástí trénovací množiny, je hranice (přímka) značena silnou černou čarou (viz Obr. 25 a Obr. 26). [20]



Obr. 25. Ukázka dělení hranic trénovací množiny. [20]

Obecná definice rovnice:

$$h(x) = w^T x_k + w_0 \quad (5)$$

Orientace dělicí přímky je značena vektorem  $w$ , poloha vektoru je značena  $w_0$ . Klasifikace objektu  $x$  do jedné ze tříd  $\omega_1$ , nebo  $\omega_2$  bude dána tím, zda je rovnice (1.17) menší nebo rovna nule.

$$w^T x_k + w_0 > 0 \quad (6)$$

Objekt bude zařazen do třídy  $\omega_1$ .

$$w^T x_k + w_0 < 0 \quad (7)$$

Objekt bude zařazen do třídy  $\omega_2$ .

Vzdálenost bodu  $x$  od hranice je dána:

$$d = \frac{|w^T x + w_0|}{\|w\|} \quad (8)$$

Kde  $\|w\|$  je velikost vektoru  $w$ .

Změna velikosti vektoru  $\|w\|$  neovlivňuje výslednou klasifikaci, je možné jej volit libovolně. Příklad pro toleranční pásmo o šířce  $\frac{2}{\|w\|}$ , ve které se nesmí nacházet žádný bod.

$$w^T x_k + w_0 \geq 2 \quad (9)$$

$$w^T x_k + w_0 \leq 2 \quad (10)$$

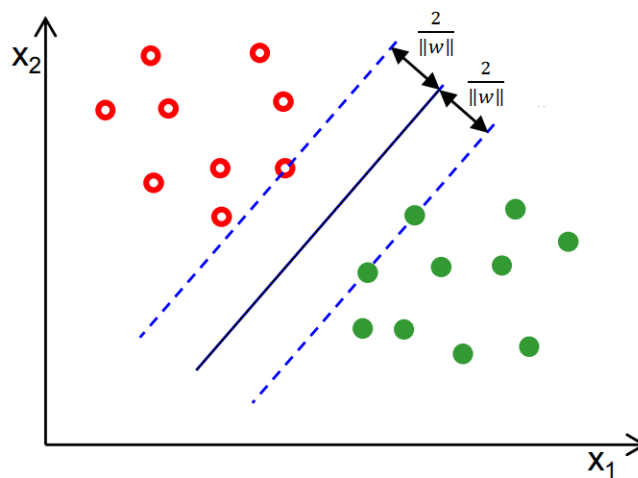
Lze zapsat zkráceně jako:

$$\delta_{Xk}(w^T x_k + w_0) \geq 2 \quad (11)$$

Pro  $k = 1, \dots, N$ , kde  $\delta_{Xk} = 2$  pro  $x_k$  ze třídy  $\omega_1$  a  $\delta_{Xk} = -2$  pro  $x_k$  ze třídy  $\omega_2$ .

Z důvodu nalezení co nejrobustnější klasifikace objektů je nutné najít takové hodnoty

$w$  a  $w_0$ , aby byla vzniklá celková tolerantní šířka  $\frac{2}{\|w\|} + \frac{2}{\|w\|} = \frac{4}{\|w\|}$  co největší.



Obr. 26. Hranice tolerantního pásma, hranice (černá přímka), tolerance (modrá přerušovaná přímka). [20]

### 3.4.3 Surf (detekce významných bodů)

SURF (Speeded-Up Robust Features) je metoda, která umí detekovat významné body nezávisle na změnách měřítka obrazů. Hlavním záměrem této metody byla konstrukce výpočetně rychlého a stabilního deskriptoru, který by mohl pracovat v reálném čase. Detekce významných bodů probíhá přímo pomocí determinantu Hessianovy matice. Výpočet determinantu je umožněn aplikací integrálního obrazu. Integrální obraz je struktura vybudovaná nad vstupním obrazem. Tato struktura umožňuje rychlé vypočtení součtu hodnot uvnitř libovolné oblasti obrazu, která je dána následující rovnicí. [20]

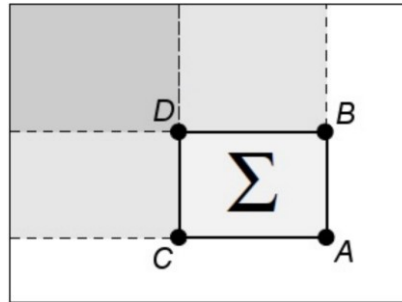
$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (12)$$

Vstupní obraz  $I(i, j)$ , obraz integrální  $I_{\Sigma}(x, y)$

Součty hodnot obdélníkových bloků uvnitř obrazu lze vyjádřit dosazením do vztahu:

$$\Sigma = A - B - C + D \tag{13}$$

Hledaný součet značí  $\Sigma$  a hodnoty integrálního obrazu v daných souřadnicích jsou A, B, C, D (viz následující obrázek Obr. 27).



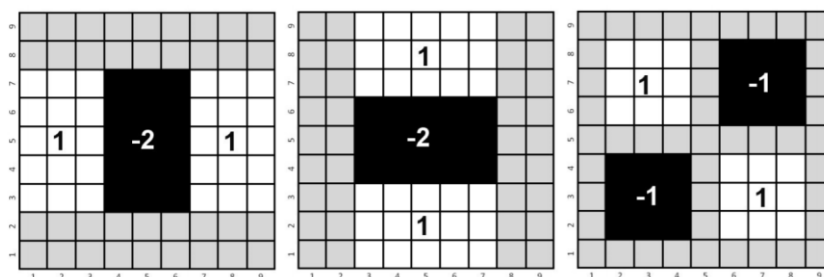
Obr. 27. Součet hodno v bloku  $\Sigma$  [22]

**Hessianova matice** v bodě  $(x, y)$  a v daném měřítku  $\sigma$  je definována:

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{yx}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \tag{14}$$

$L_{xx}$  je hodnota druhé parciální derivace podle  $x$ , konvoluce vstupního obrazu  $I(x, y)$  s Gaussovou funkcí  $G(x, y, \sigma)$  s měřítkem  $\sigma$  v bodě  $(x, y)$ .

Výpočet prvků Hessianovy matice využívá obdélníkových funkcí. Tyto funkce představují aproximaci druhé derivace v diskrétní formě, jde tedy o konvoluci obrazu s filtry (viz Obr. 28).



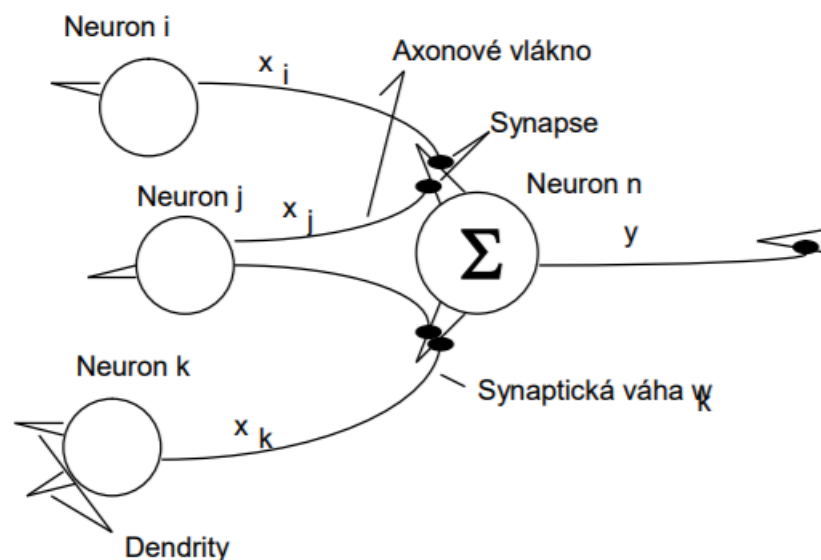
Obr. 28. Aproximace podle druhé derivace Gaussovy funkce podle  $x$ ,  $y$  a dle  $xy$  [22]

Metoda SURF se používá především pro klasifikaci obrazů.

### 3.4.4 Neuronové sítě

Vývoj konceptu umělé neuronové sítě užitečné pro rozpoznávání obrazu začal v 50. letech a pokračoval až do 60. let 20. století. Bledsoe a Browning v r. 1959 vyvinuli typ "n-tuple" klasifikátoru, který zahrnoval bitové nahrávání a vyhledávání v binárních datech. Největší vliv na vývoj neuronové sítě měl Rosenblattův tzv. perceptron (r. 1958, 1962, viz Obr. 29). [6]

Neuronové sítě jsou založeny na napodobování chování nervových soustav živých organismů, nepostupují podle předem naprogramovaného algoritmu. Během procesu učení si vytváří vlastní algoritmus, který se v čase vyvíjí, mění se stav neuronů a adaptují se váhy k řešení požadované úlohy. [3]



Obr. 29. Schématický model neuronu [24]

#### Model neuronové sítě – Perceptron

Perceptron je definován jako vážený součet všech vstupujících signálů. Pokud dojde k překonání prahové hodnoty  $\vartheta$ , neuron se excituje na hodnotu 1. V případě, že nedojde k překonání prahové hodnoty, neuron je inhibován, hodnota se sníží na 0. [24]

Za perceptron je považovaná neuronová síť s  $n$  vstupy ( $x_1, x_2, \dots, x_n$ ) a jedním pracovním neuronem spojeným s ostatními vstupy. Pro každé spojení je přiřazena váhová hodnota ( $w_1, w_2, \dots, w_n$ ). Matematický postup lze vyjádřit následující rovnicí pomocí funkce sig-num:

$$y = \text{Sgn} \left( \sum_{i=1}^n w_i x_i - \vartheta \right) \quad (15)$$

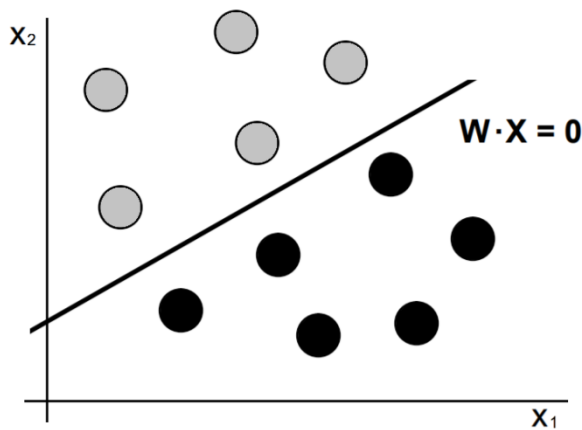
Po zavedení stálého neuronu s hodnotou excitace  $x_0 = 1$  a vazbou k neuronu  $w_0 = -\vartheta$  lze předchozí zápis zjednodušit:

$$y = \text{Sgn} \left( \sum_{i=0}^n w_i x_i \right) \quad (16)$$

Položením hodnoty v závorce rovno nule získáme rovnici nad-roviny:

$$\sum_{i=0}^n w_i x_i = \bar{W} \cdot \bar{X}, \quad \bar{W} \cdot \bar{X} = 0 \quad (17)$$

Vzniklá rovina rozděluje vstupní prostor na dva prostory (viz Obr. 30).



Obr. 30. Dvou rozměrný případ rozpoznávání pomocí perceptronu.

Schopnost neuronové sítě správně přiřazovat do tříd a rozpoznávat spočívá ve správném nastavení vah. Perceptron je proto nutné stále adaptovat na základě trénovací množiny. [24]

### 3.5 Program MATLAB

Pro realizaci vlastního programu bylo vybráno prostředí MATLAB, z důvodu dostupnosti na školních počítačích a podpoře toolboxů využitelných pro tuto úlohu.

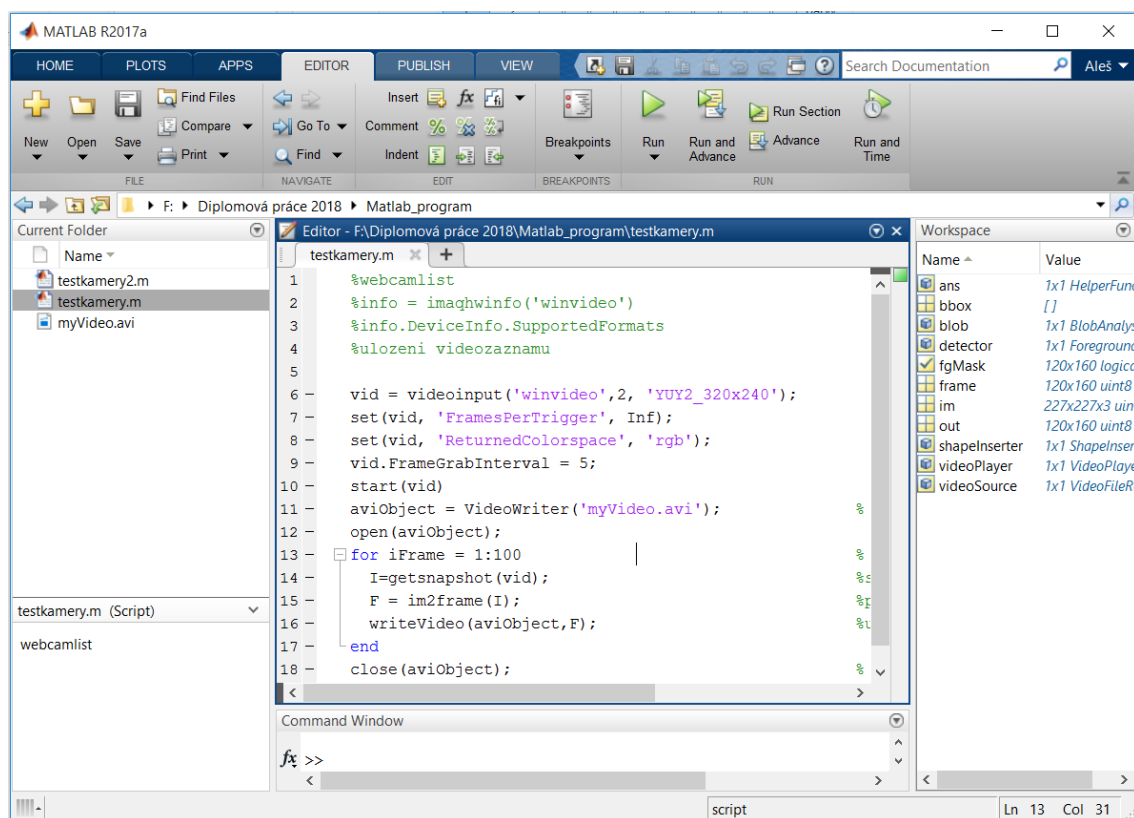
MATLAB (MATrix LABoratory) je interaktivní programové prostředí a skriptovací programovací jazyk, který je vhodný pro vědecké a inženýrské výpočty, modelování, simulaci, vývoj algoritmů a vizualizaci dat.

Výhodou MATLABu jsou tzv. toolboxy (rozšiřující knihovny funkcí) se soubory M-funkcí zaměřených na specifické účely (např. zpracování obrazu, signálů, statistiku apod.) [21]

### 3.5.1 Popis prostředí MATLABu

Základním prvkem grafického rozhraní MATLABu je příkazové okno (Command Window), dále okno pro správu aktuální složky (Current folder), pracovní prostor (Workspace) a okno s historií příkazů (Command history). Grafické rozhraní je zobrazeno na následujícím obrázku (viz Obr. 31).

- Command window – příkazový řádek, který slouží ke spuštění skriptů, funkcí, jednotlivých příkazů a zobrazování výstupů
- Current folder – zobrazuje informace o souborech v aktuální složce, kde lze měnit obsah adresáře a spouštět jednotlivé funkce
- Workspace – zobrazuje informace o definovaných proměnných, proměnná lze rozkliknout a zobrazit její obsah (typ, rozměr, atd.)
- Command history – je tvořen seznamem všech dříve použitých funkcí, v seznamu je možno hledat, případně znovu nějakou položku spustit. [21]



Obr. 31. Prostředí MATLAB R2017a



## Použité toolboxy v MATLAB

MATLAB je tvořen jádrem obsahujícím základní funkce a prostředky. Pro jednotlivé oblasti použití je možné tento základ rozšířit o problémově (na dané oblasti) orientované funkce soustředěné do toolboxu. Toolbox lze připodobnit ke knihovně, ve které se řeší specializované problémy dané oblasti. [9]

### 3.5.2 Image Processing Toolbox

Image Processing Toolbox poskytuje obsáhlý soubor algoritmů, funkcí a grafických uživatelských rozhraní pro zpracování, analýzu a vizualizaci obrazových dat a vývoj algoritmů pro práci s obrazem. Běžné úlohy zahrnují analýzu snímků, segmentaci a zdokonalení obrazu, převody mezi barevnými prostory, redukci šumu, geometrické transformace nebo registraci snímků. Mnoho funkcí Image Processing Toolboxu podporuje paralelní zpracování. [10]

### 3.5.3 Image Acquisition Toolbox

Image Acquisition Toolbox obsahuje nástroje pro snímání obrazových dat a videa ze standardních průmyslových zařízení. Slouží ke snímání obrazových dat a videí ze snímacích zařízení přímo do MATLABu a Simulinku. Hardware je možno detekovat automaticky, lze také konfigurovat jeho nastavení, zobrazit náhled a uložit obraz nebo video do paměti či přímo na disk. Z pokročilejších úloh umožňuje Image Acquisition Toolbox uživatelsky triggerovat snímání ve smyčce, snímat data na pozadí nebo synchronizovat vzorkování mezi několika multimodálními zařízeními. [11]

### 3.5.4 Statistics and Machine Learning Toolbox

Nástroj Statistics and Machine Learning Toolbox poskytuje funkce a aplikace pro popis, analýzu a modelování dat. Může se využít pro popisné statistiky a grafy, pro analýzu průzkumných dat, rozdělení pravděpodobností do dat a provádění testů hypotéz. Regresní a klasifikační algoritmy umožňují vyhodnotit výsledky z dat a sestavit prediktivní modely.

V analýze dat nástroj Statistics and Machine Learning nabízí výběr funkcí, postupnou regresi, analýzu hlavních komponent (PCA), regularizaci a další metody redukce dimenzí, které umožňují identifikovat proměnné nebo funkce, které ovlivňují model.

Panel nástrojů poskytuje supervizované a neověřené algoritmy pro strojové učení, včetně podpůrných vektorových strojů (SVM), posunutých a zavalených rozhodovacích

stromů, k-nearest neighbor, k-means, k-medoids, hierarchického shluku a Gaussových modelů. Statistické metody a algoritmy strojového učení mohou být použity pro výpočty datových sad, které jsou příliš velké pro uložení do paměti.[10]

### 3.5.5 MATLAB Support Package for USB Webcams

Nástroj MATLAB Support Package for USB Webcams se zaměřuje na podporu videokamer a webkamer připojených pomocí USB nebo integrovaných kamer v noteboocích. Po nainstalování balíčku MATLAB Support Package dokáže program kamery detekovat, rozpoznávat a rozlišovat. Seznam připojených kamer se zobrazí zadáním příkazu `webcamlist`. Poté je možné zaznamenávat obraz ve smyčce nebo zachycovat jednotlivé obrázky. U uložených obrázků je možné měnit jejich parametry a vlastnosti. [11]

## **II. PRAKTICKÁ ČÁST**

V následující kapitole popíši implementaci vlastního programu pro detekci a rozpoznávání grafických objektů. Pro realizaci a programování vlastní aplikace byl určen program MATLAB, který je předmětem zadání diplomové práce.

Program v MATLABu bude rozdělen na tři části, první část se bude zabývat pořízením video záznamu z webové kamery, druhá část bude využívat metodu strojového učení SVM k natrénování klasifikátoru a poslední část se bude věnovat detekci a klasifikaci objektů.

Videa budou zaznamenány v HD rozlišení se snímkovou frekvencí 20 snímků za vteřinu. Pro každou třídu bylo vytvořeno 30 binárních obrázků, ze kterých byl klasifikátor natrénován. Pro trénování budou vytvořeny binární obrázky 30 snímků pro každou kategorii. O segmentaci obrazu se bude starat metoda odečítání pozadí. Výstupem odečtení pozadí bude binární maska, která se bude dále upravovat pomocí morfologických operací. Upravená maska pozadí bude vstupem pro klasifikátor metody strojového učení SVM. Klasifikace nalezeného objektu bude probíhat pouze v určeném místě, aby se zabránilo špatné klasifikaci, pokud by objekt nebyl celý v záběru kamery. Program bude dále schopen klasifikované objekty zařadit do tabulky příchozích, respektive odchozích, osob a dále rozdělit dle tříd.

V poslední části bude provedeno vyhodnocení výsledků na celkem čtyřech různých objektech (osobách). Pro vyhodnocení budou použity i situace, pro které nebyl klasifikátor natrénovaný. V samotném závěru práce bude tabulka s vyhodnocením použité metody pro klasifikaci objektu.

## 4 POPIS WEBOVÉ KAMERY A SNÍMANÉHO PROSTŘEDÍ

Pro diplomovou práci byla vybrána webová kamera Microsoft LifeCam HD-3000 zobrazena na obrázku níže (Obr. 32). Jedná se o běžnou webovou HD kameru s níže popsanými vlastnostmi.

### 4.1 Webová kamera - Microsoft LifeCam HD-3000

Vlastnosti kamery:

- Maximální rozlišení videa 720p (1366 × 768 pixelů)
- Snímkovací frekvence až 30 snímků za vteřinu
- Poměr stran pořizovaného videa 16:9
- Technologie TrueColor pro jasný a výrazný obraz za všech světelných podmínek
- Automatické nastavení jasu
- Univerzální stavitelný a tvarovatelný stojánek
- Rozhraní připojení USB 2.0
- Rozměry kamery 109×44,5mm

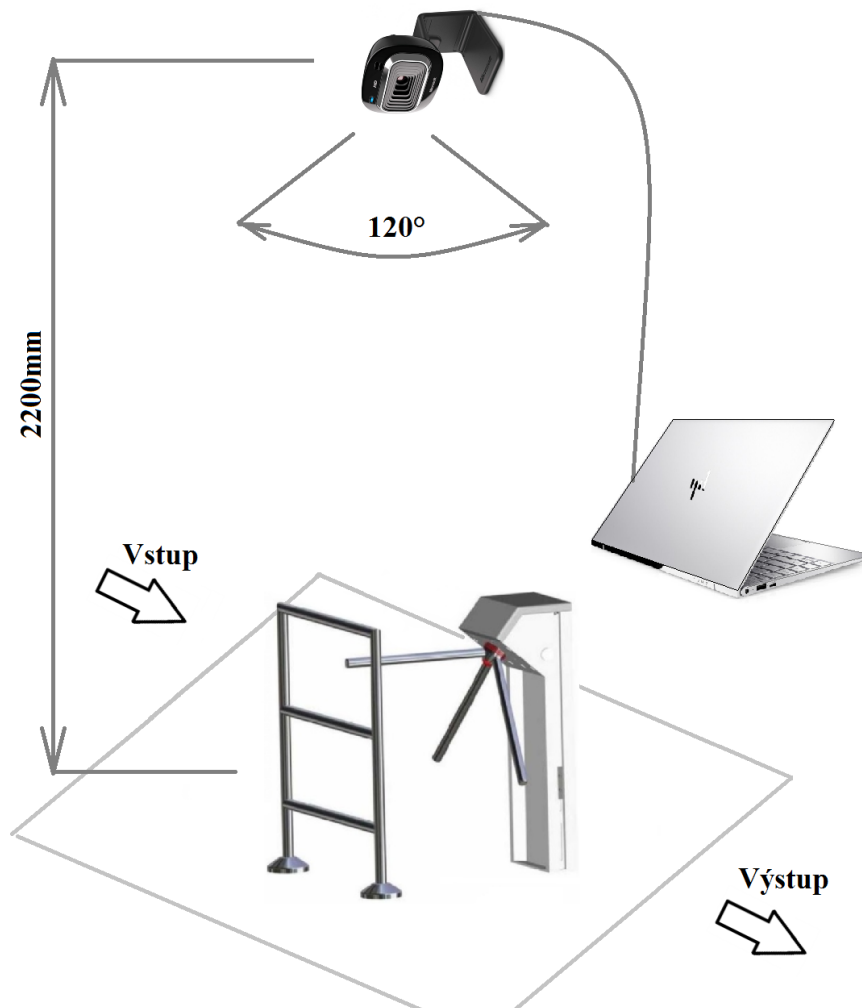


Obr. 32. Webová kamera Microsoft LifeCam HD-3000

### 4.2 Model snímaného prostředí

Na jednoduchém modelu níže (Obr. 33) je vyobrazena scéna, kde kamera zabírá průchod mezi vstupními turnikety. Kamera je připojena k počítači, na kterém je spuštěna aplikace pro záznam videa.

Kamera je otočena tak, aby zabírala co nejvíce ze vstupního prostoru před turnikety vzhledem k formátu 16:9.



Obr. 33. Model prostředí snímaného webovou kamerou

### 4.3 Nastavení kamery v programu MATLAB

Správná funkce a chod webové kamery v programu MATLAB je zajištěna instalací balíčku pro podporu webových kamer připojených pomocí USB, viz kapitola 3.5.5.

Pro natáčení videa v programu MATLAB je nutné provést následující kroky:

- nastavit rozlišení pořizovaného videa,
- přepnout expozici do manuálního režimu (Exposure Mode - manual),
- nastavit pevnou hodnotu expozice,
- přepnout vyvážení bílé taktéž do manuálního režimu (White Balance Mode - manual),

- nastavit pevnou hodnotu vyvážení bílé,
- nastavit kontrast.

Expozice udává množství světla dopadajícího na CMOS sensor kamery a je nutné jej nastavovat manuálně. Automatická expozice totiž mění světelné podmínky ihned při průchodu pod kamerou a vzhledem k použité metodě odečítání pozadí, která je náchylná právě na velkou změnu světelných podmínek, je tato vlastnost nežádoucí.

Všechny vlastnosti kamery, které je možné upravovat v programu, jsou zobrazeny na obrázku, viz Obr. 34.

```
Name: 'Microsoft® LifeCam HD-3000'  
Resolution: '640x480'  
AvailableResolutions: {1x12 cell}  
Zoom: 0  
WhiteBalanceMode: 'auto'  
Tilt: 0  
Exposure: -6  
Sharpness: 25  
Brightness: 133  
Saturation: 83  
BacklightCompensation: 1  
WhiteBalance: 4500  
Contrast: 5  
ExposureMode: 'auto'  
Pan: 0
```

Obr. 34. Vlastnosti kamery zobrazené pomocí programu MATLAB

#### 4.3.1 Program pro Záznam videa

Nejprve bylo nutné nastavit parametry a hodnoty nahrávání, jak bylo zmíněno v kapitole: 4.3.

V našem případě jsou zvoleny hodnoty v závislosti na světelných podmínkách v natáčeném prostředí. Jas byl nastaven na hodnotu 150, vyvážení bílé bylo nastaveno na hodnotu 3000 a pro pevné nastavení expozice byla zvolena hodnota -8.

```
src = getselectedsource(video_Obj);  
src.Brightness = 150;  
src.WhiteBalanceMode = 'manual';  
src.WhiteBalance = 3000;  
src.ExposureMode = 'manual';
```

```
src.Exposure = -8;
```

Z důvodu náročnosti výpočetních metod pro získání masky popředí, morfologické operace a následné identifikace objektu, byla zvolena snímkovací frekvence 20 snímků za vteřinu.

Výstupem programu je video soubor s příponou .avi. Název zaznamenaného videa je vždy aktuální čas vytvoření souboru, při opětovném spuštění tak nedochází k přepisu souboru a je možné data zálohovat.

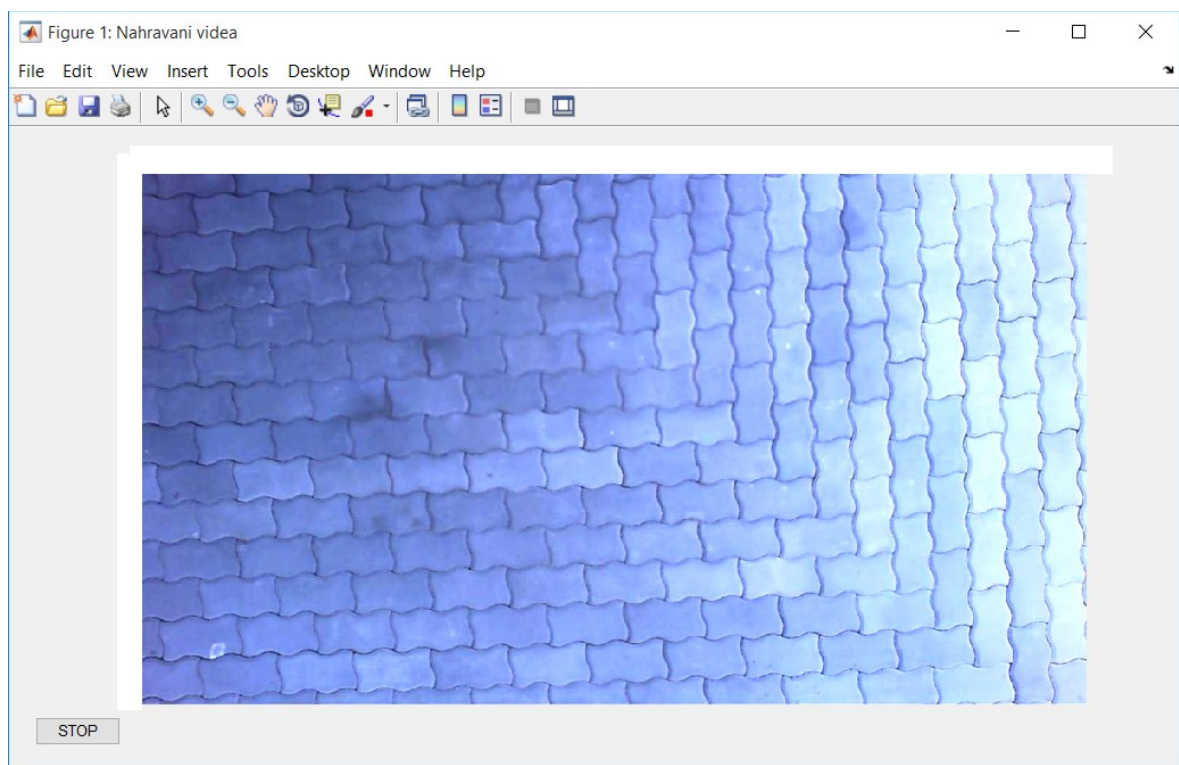
Přesný formát zápisu:

*zaznam\_video\_hh\_MM\_dd\_mm\_yy.avi* (aktuální hodina, minuta, den, měsíc, rok).

Následující obrázek (Obr. 35) zobrazuje jednoduché grafické rozhraní, ve kterém se nachází náhled aktuálně zaznamenávaného snímku a tlačítko stop.

Zapnutím programu se spustí nahrávání videa. Pro zastavení programu lze využít dva způsoby:

- zavření náhledu (zavření okna),
- stisknutí tlačítka stop.



Obr. 35. Program pro záznam videa – náhled

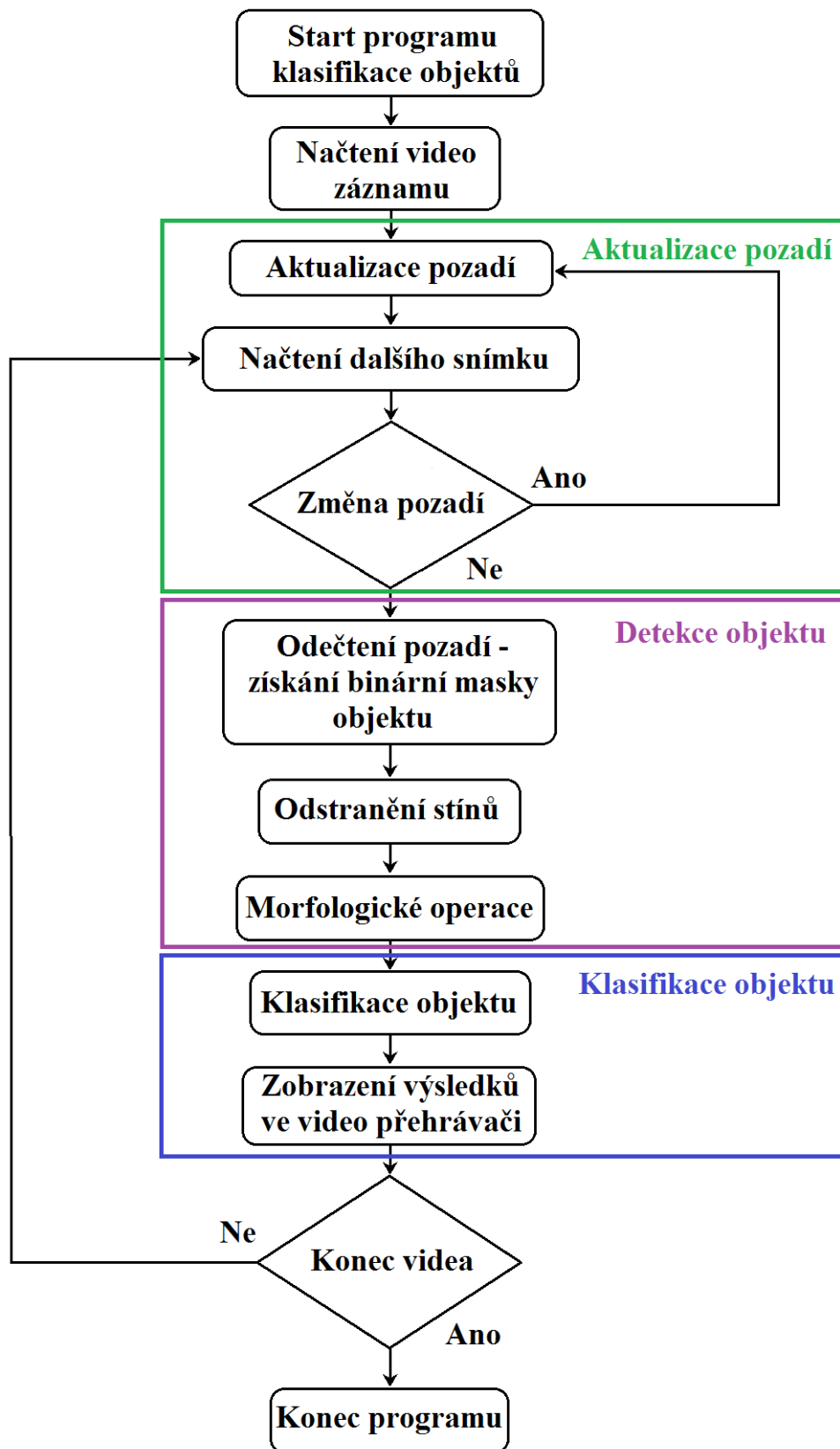
Ihned po ukončení programu je video soubor uložen do předem nastaveného adresáře.



## 5 DETEKCE A KLASIFIKACE OSOB

V první řadě bylo nutné si představit a popsat blokové schéma programu pro detekci a klasifikaci osob.

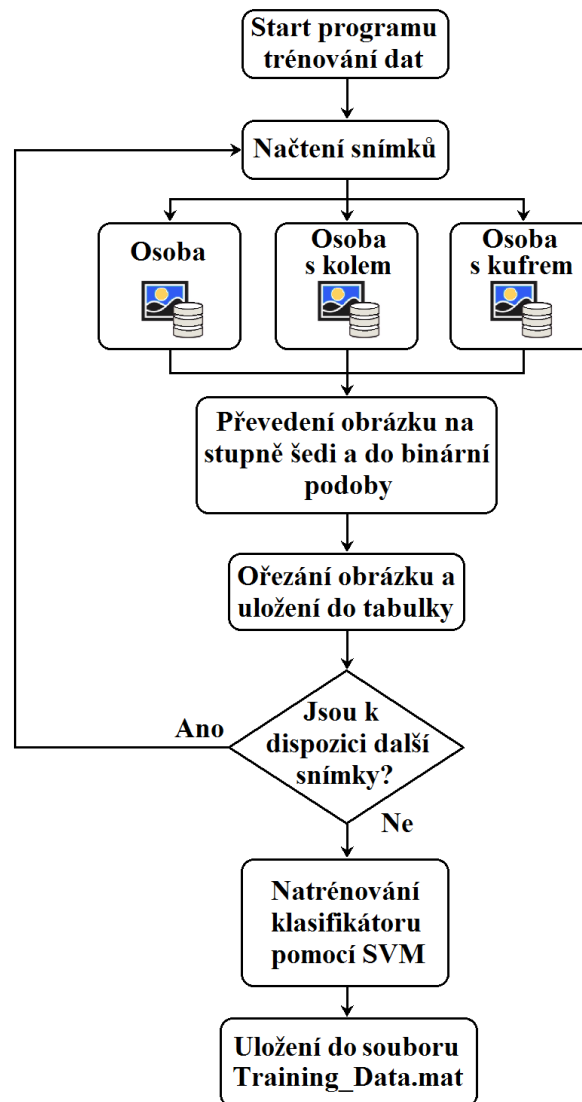
Popis blokového schématu programu klasifikace osob je vyjádřen na obrázku níže (viz Obr. 36). Po zapnutí programu a následného vybrání video záznamu se spustí přehrávání videa. První snímek se ukládá jako pozadí, které je dále vyhodnocováno s aktuálními snímky, v případě změny bude spuštěna aktualizace pozadí. Pokud není aktualizace pozadí vyžadována, přistoupí se k odečítání pozadí, kterým se nalezne binární maska objektu. V následujícím kroku je přistoupeno k odstranění stínů a aplikaci morfologických operací. Takto upravená binární maska přistupuje ke klasifikaci objektu. V posledním kroku jsou klasifikované objekty zobrazeny ve video přehrávači. Pokud není zjištěn konec videa, program pokračuje načtením dalšího snímku, v opačném případě program skončí.



Obr. 36. Blokové schéma programu pro klasifikaci objektu

## 5.1 Trénování objektů

Pro trénování objektů je vybrána metoda SVM (metoda podpůrných vektorů), která je uvedena v literární rešerši viz kapitola 3.4.2. Tato metoda byla vybrána z důvodu trénování binárních (černobílých) obrázků. Odečtením pozadí vznikne binární mapa, ve které vypadají všechny obrysy osob, kol, kufrů atd. se shora podobně. Je možné tedy rozpoznávat i osoby, které nebyly předmětem trénovací množiny binárních obrázků.



Obr. 37. Blokové schéma trénování dat a učení objektů

Před-trénování objektu je nastaveno pro tři kategorie:

1. Osoba
2. Osoba s kolem
3. Osoba s kufrem

Pro učení SVM bylo využito třicet obrázků z každé kategorie. Pro představu byly vybrány tři obrázky z každé kategorie, sloužící pro učení SVM.

### **Osoba**

Na obrázcích je vidět obrys postavy (objektu), který je v různém stádiu pohybu. Podobných obrázků, ze kterých je učen klasifikátor SVM, je celkem třicet (viz Obr. 38, Obr. 39, Obr. 40).



Obr. 38. Binární mapa s obrysem postavy, určena k trénování SVM.

### **Osoba s kolem**

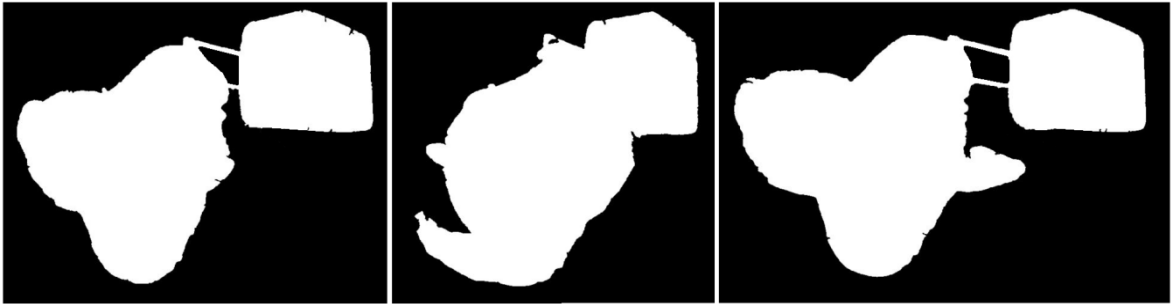
Totéž je i na snímcích dalšího objektu, kde je zachycen obrys postavy s kolem. Pro učení klasifikátoru SVM bylo rovněž použito třicet snímků.



Obr. 39. Binární mapa s obrysem postavy a kolem, určena k trénování SVM.

### **Osoba s kufrem**

Jako poslední je osoba s kufrem. Na obrázcích je zachycen ostrý obrys postavy v různých stádiích pohybu. Pro natrénování SVM bylo použito třicet snímků.



Obr. 40. Binární mapa s obrysem postavy a kufrem, určena k trénování SVM.

Trénovací program SVM klasifikátoru tvoří 3 smyčky *for* pro každý z identifikovaných objektů, které musí splňovat podmínku  $i=1:X$ , kde  $X$  je počet obrázků. V dalším kroku je nastavena podmínka *if*, tak, aby se načetly všechny obrázky bez rozdílu na číslování souboru. Jelikož je černobílý vstupní obrázek v modelu RGB, je nutné ho nejprve převést na stupně šedi a následně do binární podoby. V posledním kroku stačí už jen obrázek ořezat a uložit do tabulky, viz následující kód:

```
for i=1:30
    if i<10
        im=(imread(strcat...
            ('TrainingData/Osoba_kufr/Os_k_0', num2str(i), '.jpg')));
    else
        im=(imread(strcat...
            ('TrainingData/Osoba_kufr/Os_k_', ...
            num2str(i), '.jpg')));
    end
    %preved do cernobile a nasledne do binarni podoby
    im=imbinarize(rgb2gray(im));
    %orez
    souc_radky=find(sum(im,2));
    soucet_sloupce=find(sum(im,1));
    im_orezany=im(souc_radky(1):souc_radky(end), ...
        soucet_sloupce(1):soucet_sloupce(end), :);

    %preved na 100x100
    im_orezany = imresize(im_orezany, [100 100]);
    %uloz jako 1 radek
```

```
im_orezany = reshape(im_orezany', 1, 100*100);
%pridej do tabulky
Table=[Table;double(im_orezany)];
Y=[Y;1];
```

end

Pro naučení SVM klasifikátoru se využívá funkce MATLABu *fitcsvm*, která je určena pro obecnou klasifikaci. Vstupními daty pro klasifikátor je řádek (vektor), který obsahuje uloženy binární obrázky po řádcích. Následně jsou uloženy do tabulky - *Table* a kategorie *Y* (1 - osoba, 2 - osoba s kolem, 3 - osoba s kufrem).

```
SVMStruct = fitcsvm(Table,Y);
```

Naučený SVM klasifikátor je nutné uložit do souboru, aby se mohl využít pro další klasifikaci.

```
save('Training_Data.mat', 'SVMStruct');
```

## 5.2 Aktualizace pozadí

Pro aktualizaci pozadí se využívá příkazu „all“, který hledá v celé binární mapě všechny nenulové hodnoty matice. Následně se pozadí aktualizuje a to tak, že se nechá 70% starého pozadí a přičte se 30% nového pozadí. Poté se aktualizuje čas pro změnu pozadí.

```
odchylka=all((gray_Background(:)-gray_videoFrame(:))<25);
if odchylka == 1
    gray_Background = 0.7*gray_Background + 0.3*gray_videoFrame;
    time_Background = datestr(now, 'HH:MM:SS');
```

end

## 5.3 Odečítání pozadí (detekce popředí)

Pro detekci pozadí byla vybrána metoda odečítání pozadí, z důvodu použití statické kamery. Metoda pracuje na principu odečítání aktuálního snímku od modelu pozadí při nastaveném prahu. Rozdíl, který vznikne, je možné považovat za popředí.

V prvním kroku si vezmeme první snímek a uložíme ho jako pozadí a následně ho převedeme na stupně šedi.

```
background = step(videoReader);  
gray_Background = rgb2gray(background);
```

V následujícím kroku provedeme totéž pro snímek, který se bude od pozadí odečítat.

```
videoFrame = step(videoReader);  
gray_videoFrame = rgb2gray(videoFrame);
```

Nyní použijeme příkaz `imabsdiff`, který nám tyto snímky odečte v absolutní hodnotě s nastaveným prahem na hodnotu 50.

```
odectene_pozadi=imabsdiff(gray_Background,gray_videoFrame)>50;
```

Výsledkem je odečtené pozadí, tedy binární maska, na kterou se následně aplikují morfologické operace, nebo se ještě odstraní stíny.

## 5.4 Odstranění stínů

Stíny ve video snímku znamenají zhoršení rozlišovacích schopností klasifikátoru a následného rozpoznání objektu. Je tedy nutné je alespoň částečně eliminovat. V programu byla zvolena metoda odstranění stínů pomocí převedení barev z RGB modelu na HSV model. Po převedení na HSV model, je možné zjistit jednotlivé hodnoty pixelů. Vše je založeno na tom, že vzniklý stín nemění barvu pozadí, ale mění se pouze sytost a hodnota jasu. Zjištěním těchto hodnot dopadajícího stínu na podloží je možné stín částečně, až úplně eliminovat.

### Postup částečné eliminace stínu

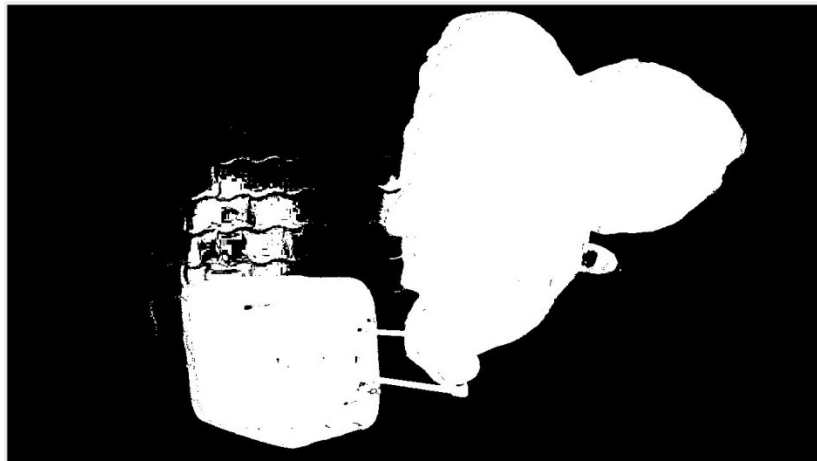
Nejprve provedeme zobrazení binárního snímku po odečtení pozadí. Metoda odečtení pozadí nedokáže rozlišit mezi objektem a stínem, proto je stín odečten od pozadí stejně jako objekt. Této situaci se dá předejít převedením aktuálního snímku na již zmíněný HSV model a v místech, kde se nachází stín, zjistíme hodnoty pro sytost barvy (Saturation) a pro hodnoty jasu (Value).

V našem případě byly hodnoty stínu pro sytost barvy  $0,43$  po  $0,50$  a hodnoty pro jas v rozmezí od  $0,50$  po  $0,61$ . Tímto postupem jsme detekovali stín a následně částečně eliminovali, viz následující zdrojový kód programu.

```
hsv_videoFrame = rgb2hsv(videoFrame);  
hsv_background = rgb2hsv(background);
```

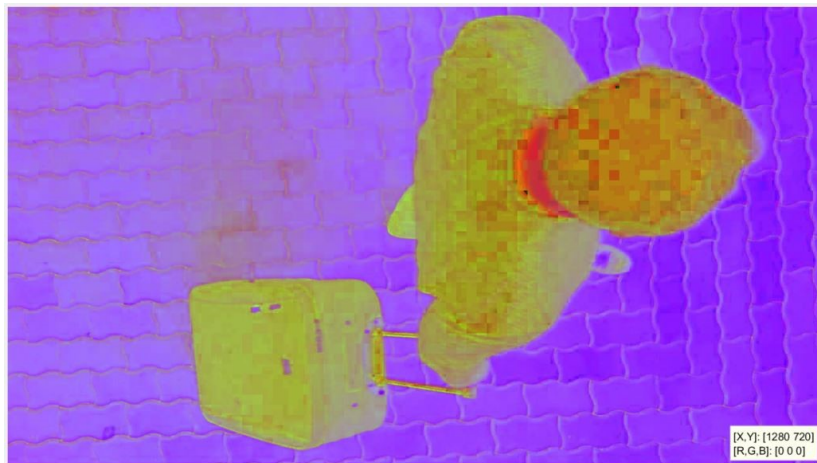
```
Hue = hsv_videoFrame(:,:,1);  
Sat = hsv_videoFrame(:,:,2);  
Val = hsv_videoFrame(:,:,3);  
detekovany_Stin = Sat > 0.43 & Sat < 0.5 & Val > 0.61 & Val <  
0.81;  
odecet_Stinu = odectene_pozadi & ~detekovany_Stin;
```

Níže je zobrazena binární mapa vzniklá odečtením snímku pozadí od snímku popředí pro nastavený práh. Výsledkem je vykreslený objekt se stínem.



Obr. 41. Binární mapa po odečtení pozadí.

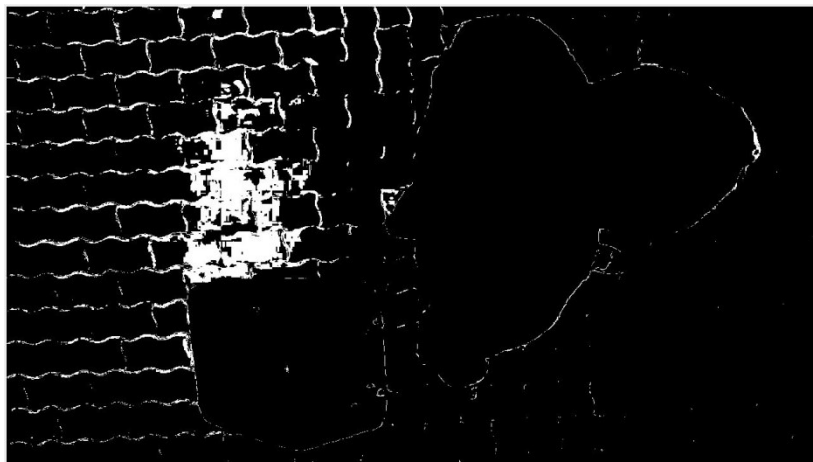
Následující snímek (Obr. 42) byl převeden na HSV model ještě před odečtem pozadí, ze kterého bylo možné zjistit odpovídající hodnoty sytosti a jasu pro stín.



Obr. 42. Snímek převedený z RGB na HSV model a zobrazený v RGB.

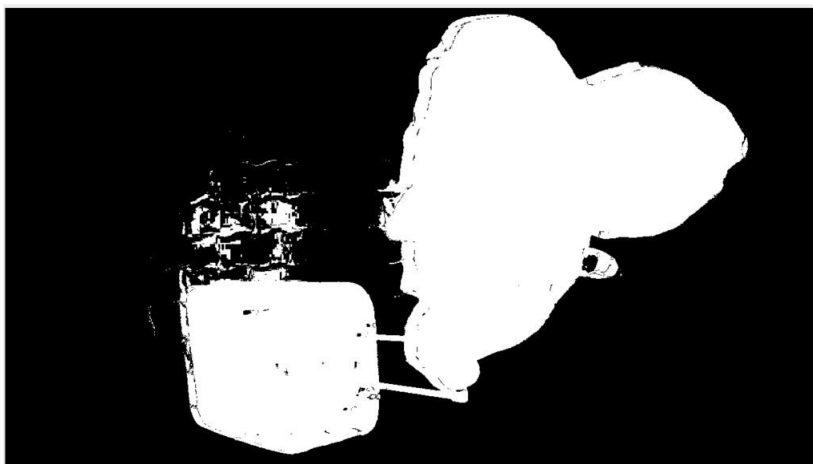


Na následujícím obrázku (Obr. 43) je detekovaný stín v nastavených mezích pro hodnoty sytosti.



Obr. 43. Nalezený stín pro zadané hodnoty sytosti a jasu v programu.

U následujícího obrázku (Obr. 44) je stín částečně eliminován, tak aby bylo možné ho ještě lépe odstranit morfologickými operacemi. Stín není odstraněn zcela z toho důvodu, že pokud by nějaká barva objektu měla stejné hodnoty pro sytost barvy a stejnou hodnotu jasu, byla by odstraněna stejně jako stín. Žádoucí je tedy udržovat co nejmenší meze pro hodnoty sytosti a jasu při odstranění stínu.

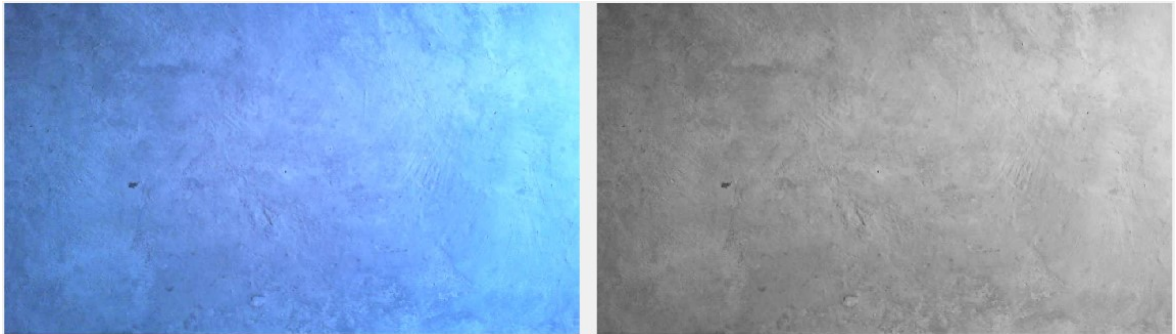


Obr. 44. Odečtený stín od binární mapy.

## 5.5 Morfologické operace

Morfologické operace se aplikují na binární obrázky ještě před detekcí a klasifikací osob, jedná se tedy o předzpracování obrazu. Výstupem morfologických metod jsou obrázky s dokonalými obrysy objektů, které usnadňují klasifikaci. V této kapitole si popíšeme jednotlivé filtry a funkce, kterými bylo dosaženo finálního obrázku.

Model pozadí v RGB a obrázek převedený do stupňů šedi (viz Obr. 45).



Obr. 45. Model pozadí v RGB (vlevo) a ve stupních šedi (vpravo).

Na obrázku (Obr. 46) níže je vyobrazen aktuální snímek v RGB a následně ve stupních šedi.



Obr. 46. Model pozadí s objektem, obrázek v RGB (vlevo) a ve stupních šedi (vpravo).

K detekci popředí (masky) objektu se využívají obrázky ve stupních šedi, které jsou od sebe následně odečteny příkazem *imabsdiff*.

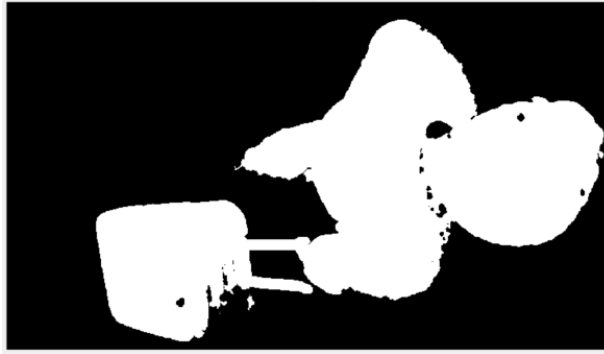
Příkaz *Imabsdiff* je totožný s rovnicí:

$$Maska = Abs(x - y) > threshold \quad (18)$$

,kde  $x$  značí pozadí,  $y$  je aktuální snímek a  $threshold$  je práh.

```
finalFrame =  
imabsdiff(thresholdBackground, thresholdCurrentFrame) > 55;
```

Následující obrázky (viz Obr. 47, Obr. 48, Obr. 49, Obr. 50) představují posloupnost použití morfologických operací.



Obr. 47. Binární maska vzniklá odečtem obrázku z pozadí a aktuálního snímku.

Mediánový filtr je použit k odstranění nežádoucích bodových rušení. V MATLABu je filtr dostupný pod označením *medfilt2*.

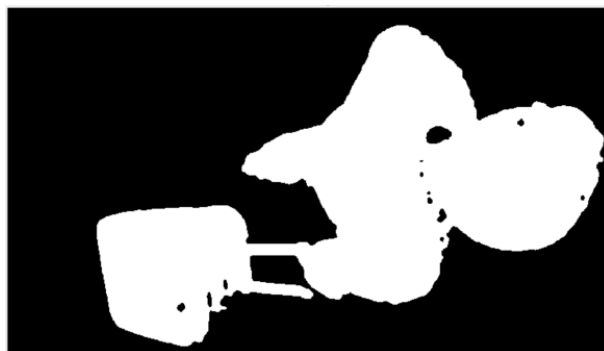
```
finalFrame = medfilt2(finalFrame,[3 3]);
```



Obr. 48. Upravená binární maska pomocí mediánového filtru.

Vznikající malé objekty po odečtení snímku je nutné odfiltrovat, proto jsou v dalším kroku odstraněny objekty menší než 1000px.

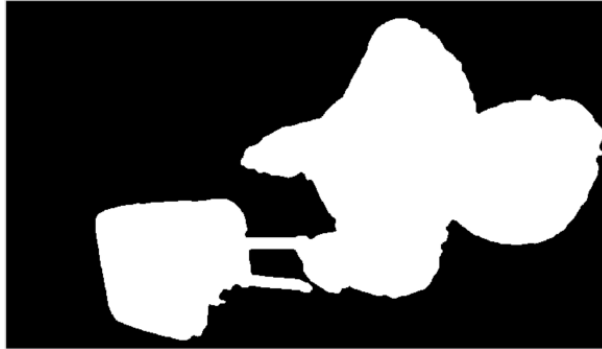
```
finalFrame = bwareaopen(finalFrame, 1000);
```



Obr. 49. Binární maska s odstraněnými objekty menšími než 1000px.

Na závěr jsou odstraněny díry z binárního obrázku, které by byly nežádoucí pro strojové učení SVM a klasifikaci.

```
finalFrame = imfill(finalFrame, 'holes');
```



Obr. 50. Finální snímek po odfiltrování děr z obrázků.

## 5.6 Nalezení objektu

Pro nalezení objektu byla zvolena vestavěná funkce MATLABu konkrétně blob analysis, která je založená na detekci oblastí v obraze, které mají odlišné okolí. V našem případě je okolí zcela odlišené od objektu vzhledem k tomu, že okolí má černou barvu a objekt barvu bílou. Tímto způsobem je objekt pouze detekován. Pro detekci je nastavena minimální a maximální plocha, kterou musí objekty splňovat, aby byly brány v potaz.

```
blobAnalysis = vision.BlobAnalysis...  
    ('BoundingBoxOutputPort', true, 'AreaOutputPort',...  
    true, 'CentroidOutputPort', true, ...  
    'MinimumBlobArea', 100000, 'MaximumBlobArea', 400000);
```

Hledání ohraničujícího rámečku (bbox - bounding box), u kterého se krokuje blobAnalysis a binární obrázek s odečteným pozadím.

```
[area,centroid,bbox] = step(blobAnalysis, finalFrame);
```

Zobrazení rámečků kolem detekovaných osob:

```
resultFrame = insertShape(videoFrame, 'Rectangle', bbox,  
    'Color',...  
    'green', 'Opacity', 0.1, 'LineWidth', 5);
```

Zjištění počtu objektů je velmi snadné, stačí zjistit velikost vektoru bbox (bounding box) které splňují všechna požadovaná kritéria:

```
pocetObjektu = size(bbox, 1);
```

## 5.7 Klasifikace objektu

V případě zjištění objektu je možné přistupovat k dalšímu kroku - ke klasifikaci. V samotném úvodu zdrojového kódu je provedeno načtení předtrénovaného klasifikátoru SVM, který je uložen v souboru Training\_Data.mat. Následně je možné volat proměnnou SVMStruct.

```
load('Training_Data.mat');
```

Klasifikace se spouští pouze po překročení jedné ze svislých čar (viz Obr. 51), tím je ošetřena klasifikace, aby byl celý objekt v záběru kamery. Levá svislá čára slouží pro klasifikaci příchozích osob. Podmínka je formulovaná tak, že se kontroluje středová souřadnice binární mapy - x a hledá se taková hodnota, kdy předchozí snímek měl x-sovou souřadnici menší než 600 a následující snímek má x-sovou souřadnici větší než 600.

$$\text{Předchozí snímek, } x < 600 \ \& \ \text{Aktuální snímek, } x > 600 \quad (19)$$

Pro klasifikaci odchozích osob slouží druhá svislá čára (viz Obr. 51). Podmínka pro klasifikaci je podobná jako u předchozího případu s tím rozdílem, že aktuální snímek musí mít hodnotu souřadnice x menší než u předchozího snímku.

$$\text{Předchozí snímek, } x > 600 \ \& \ \text{Aktuální snímek, } x < 600 \quad (20)$$

Pokud byl nalezen nějaký objekt a prošel zleva nebo zprava přes svislé čáry, spustí se klasifikace. V opačném případě se v okně video přehrávače zobrazí hlášení: „nebyl nalezen žádný objekt“. Pokud se aplikace spustí, aktuální snímek se nejprve ořeže a upraví, stejně jako tomu bylo pro obrázky v sekci trénování objektů (viz kapitola 5.1).

```
klasifikace = double(finalFrame);
souc_radky = find(sum(klasifikace,2));
soucet_sloupce = find(sum(klasifikace,1));
orezVideoFrame = klasifikace(souc_radky(1):souc_radky(end), ...
soucet_sloupce(1):soucet_sloupce(end), :);


```

```
orezVideoFrame = reshape(orezVideoFrame',1, 100*100);
```

Proměnná SVMStruct obsahuje řádkový vektor pro každou z tříd, ve kterém jsou zapsány binární obrázky. Ve stejném formátu jsou i snímky z videa a pomocí funkce predict je vy-  
pisováno skóre.

```
score = predict(SVMStruct, double(orezVideoFrame));
```

Klasifikace pracuje na principu uvedeném v následující tabulce, kde jsou uvedeny hodnoty skóre pro určité objekty.

*Tab. 1. Hranice klasifikovatelných ka-  
tegorií.*

	Skóre	
	Od	do
<b>Osoba</b>	0,55	1,45
<b>Neznámý objekt</b>	1,46	1,54
<b>Osoba s kolem</b>	1,55	2,45
<b>Neznámý objekt</b>	2,46	2,54
<b>Osoba s kufrem</b>	2,55	3,45

Získané předpovědi zapsané v proměnné skóre nyní stačí vyhodnotit. Pokud je skóre rovno 1, případně je menší nebo rovno 1.45, jedná se o osobu atp. (viz Tab. 1). Mezi jednotlivými objekty je ponechána pomyslná mezera, aby v případě, že je hodnota skóre na hraně mezi dvěma objekty, se klasifikátor nerozhodoval špatně.

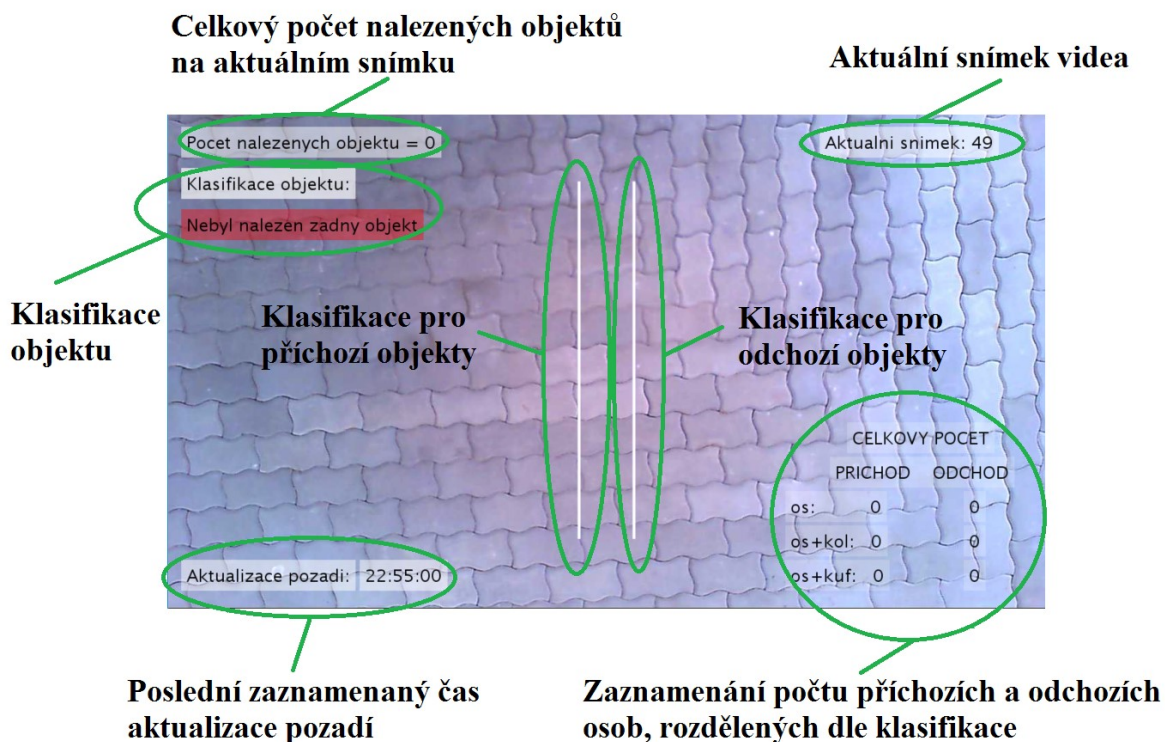
```
if score >= 0.55 && score <= 1.45
    text_Rozpoznani = sprintf('Osoba');
    pocet_osob = pocet_osob + 1;
elseif score >= 1.55 && score <= 2.45
    text_Rozpoznani = sprintf('Osoba s kolem');
    pocet_osob_kolo = pocet_osob_kolo + 1;
elseif score >= 2.55 && score <= 3.45
    text_Rozpoznani = sprintf('Osoba s kufrem');
    pocet_osob_kufr = pocet_osob_kufr + 1;
else
    text_Rozpoznani = sprintf('Neznamy objekt');
end
```

```
    % zobrazeni vysledku
resultFrame = insertText(resultFrame, [20 140],text_Rozpoznani,...
    'FontSize', 25, 'BoxColor', 'green','BoxOpacity', 0.5);
resultFrame = insertText(resultFrame, [20 200],num2str(score),...
    'FontSize', 25, 'BoxColor', 'white','BoxOpacity', 0.5);
```

Jak je ze zdrojového kódu zřejmé, každá podmínka `if`, případně `elseif`, popisuje jednu konkrétní klasifikaci. Po splnění podmínky se zobrazí ve video přehrávači text s názvem objektu. V případě, že není splněna žádná z mezí pro některou z klasifikací, se vypíše text – neznámý objekt.

Po zpuštění programu, vybrání a úspěšném načtení videa se zapnou dvě okna s video přehrávači. První okno zobrazuje aktuální snímky v barevném RGB modelu a druhé okno s video přehrávačem zobrazuje pouze černobílou mapu detekovaných objektů. V přehrávači jsou zobrazeny texty, které informují o stavu detekce, případně klasifikace objektů. Když je počet nalezených objektů roven nule, zobrazí se pouze informace o tom, že nebyl nalezen žádný objekt.

Popis k Obr. 51. V levém dolním rohu je napsáno: Aktualizace pozadí a čas, čas se zde mění v závislosti na poslední změně pozadí, například v případě změny světelných podmínek. Vpravo nahoře je uveden aktuální snímek videa. Vpravo dole je ukazatel celkového počtu osob, které buď přišly, nebo odešly. Svislé čáry uprostřed obrázku určují bod, kde je objekt vyhodnocován. Pokud se střed zeleného rámečku (objektu) dotkne levé svislé čáry, dojde ke klasifikaci o zobrazení výsledku klasifikace. První svislá čára je pro příchozí objekty, tedy pokud střed objektu projde zleva doprava, přičte se 1 k celkovému počtu příchozích, kde dále závisí na tom, jak byl objekt klasifikován. V opačném případě, kdy se střed objektu projde přes druhou svislou čáru zprava doleva, dojde ke klasifikaci a k odpovídajícímu odchozímu objektu se přičte 1.



Obr. 51. Popis aktuálně přehrávaného snímku v okně video přehrávače MATLABu.

Při úspěšném nalezení objektu je ohraničen zeleným rámečkem a po překročení první svislé čáry, jelikož objekt se pohybuje zleva doprava, se zobrazí klasifikace s odpovídajícím skóre. Hodnota skóre byla stanovena a vypsána dle klasifikátoru s hodnotou 1,1914. Takové skóre se dalo očekávat vzhledem k tomu, že osoba figuruje jako objekt na všech snímcích a je tady určitá podobnost pro osobu, která má hodnotu skóre o 0,19 nižší, tedy 1 (viz Obr. 52).





Obr. 52. Aktuálně detekovaný a rozpoznaný objekt.

Odovídající binární mapa k předešlému obrázku (Obr. 52). Z obrázku (Obr. 53) je zřetelně poznat obrys objektu a klasifikátor tak neměl velké problémy s jeho rozlišením.



Obr. 53. Binární mapa předcházejícího obrázku.

## 6 TESTOVÁNÍ A VYHODNOCENÍ VÝSLEDKŮ

V následující kapitole si otestujeme program napsaný v MATLABu a zhodnotíme navrženou metodu pro detekci a rozpoznávání osob.

Testování se odehrávalo ve stejných podmínkách, pouze bylo změněno osvětlení a byly měněny parametry kamery, jako je sytost barev, tón, hodnota expozice a vyvážení bílé. Testování proběhlo celkem na čtyřech osobách. Pro zjištění správnosti detekce byla použita i barva oblečení podobná pozadí.

Jak je vidět z obrázků níže, metoda odečítání pozadí pro získání popředí pracuje zcela správně a v binárních snímcích nejsou žádné dodatečné stopy po špatném odečtení snímků. Každý snímek znázorňuje detekci objektu ohraničeného zeleným rámečkem. Následně je zobrazeno skóre klasifikace a vypsáný název odpovídajícího objektu.

Ještě jednou bude připomenuto nastavení rozmezí skóre pro klasifikaci objektu:

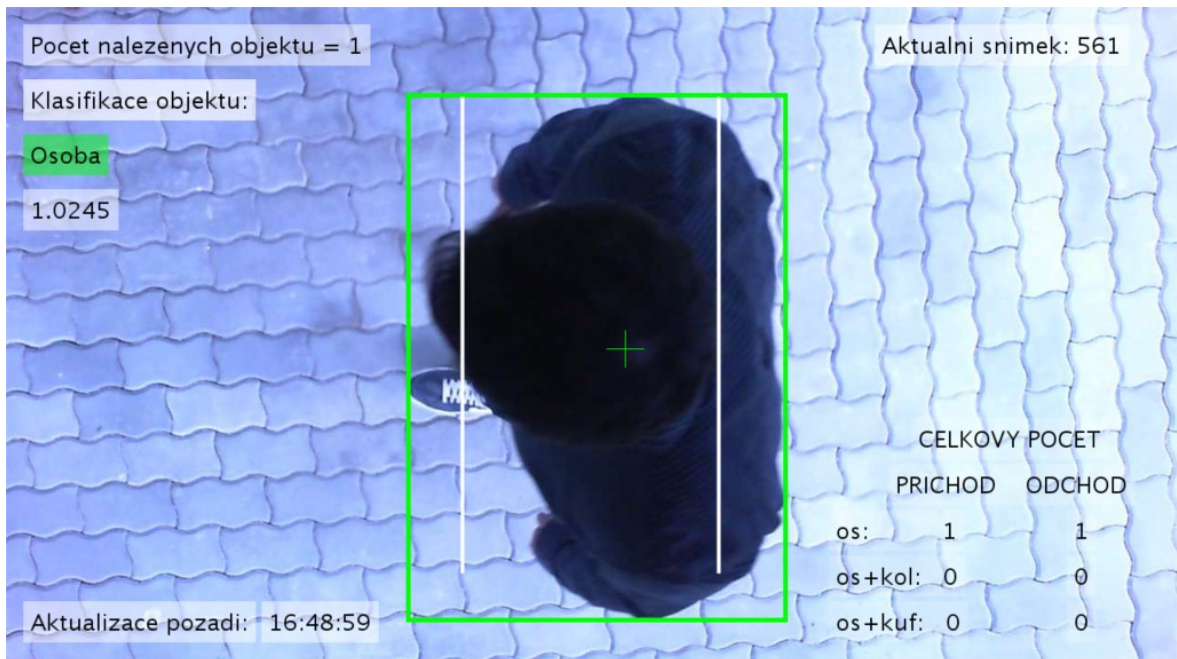
[1] Osoba:	$1 \pm 0.45$
[2] Osoba s kolem:	$2 \pm 0.45$
[3] Osoba s kufrem:	$3 \pm 0.45$

Všechny ostatní hodnoty skóre vypíší hlášení o neznámém objektu.

Níže bude převedeno testování všech možných detekcí objektu, popisy jednotlivých snímků a jejich odpovídajících binárních map. Snímky ukazují testování detekce a klasifikace objektu s pohybující se osobou na video záznamu.

### 6.1 Klasifikace osoby

Objekt byl správně detekován a následně klasifikován. Skóre vypsané pomocí SVM klasifikace jednoznačně rozhodlo, že se jedná o osobu (viz Obr. 54). Na binárním snímku (Obr. 55) není vidět žádný nežádoucí šum ani velký stín, který by mohl klasifikaci ovlivnit.



Obr. 54. Snímek s klasifikací - osoba

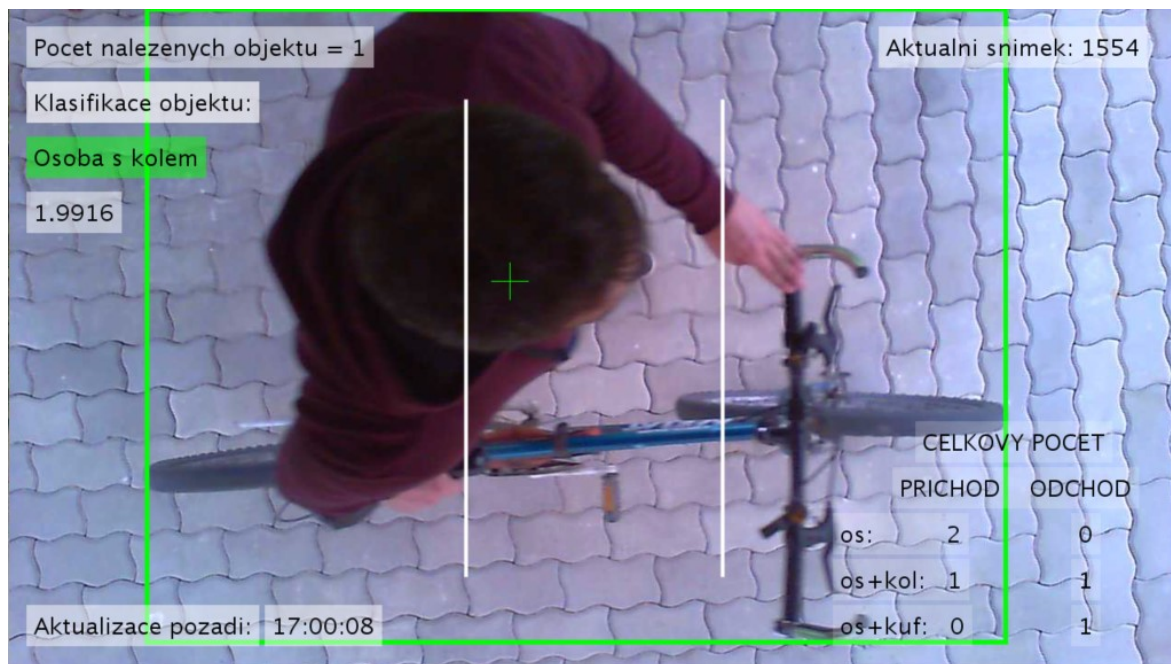
Binární mapa k obrázku rozhodujícím o klasifikaci osoba.



Obr. 55 Odpovídající binární mapa pro klasifikaci osoby.

## 6.2 Klasifikace osoby s kolem

Na obrázku (Obr. 56) níže je zobrazena scéna osoby s kolem, o které bylo správně rozhodnuto pomocí klasifikace, skóre bylo zjištěno 1,81.



Obr. 56. Snímek s klasifikací - osoba s kolem.

Binární mapa (Obr. 57) k obrázku rozhodující o klasifikaci osoba s kolem.



Obr. 57. Odpovídající binární mapa pro klasifikaci osoby s kolem.

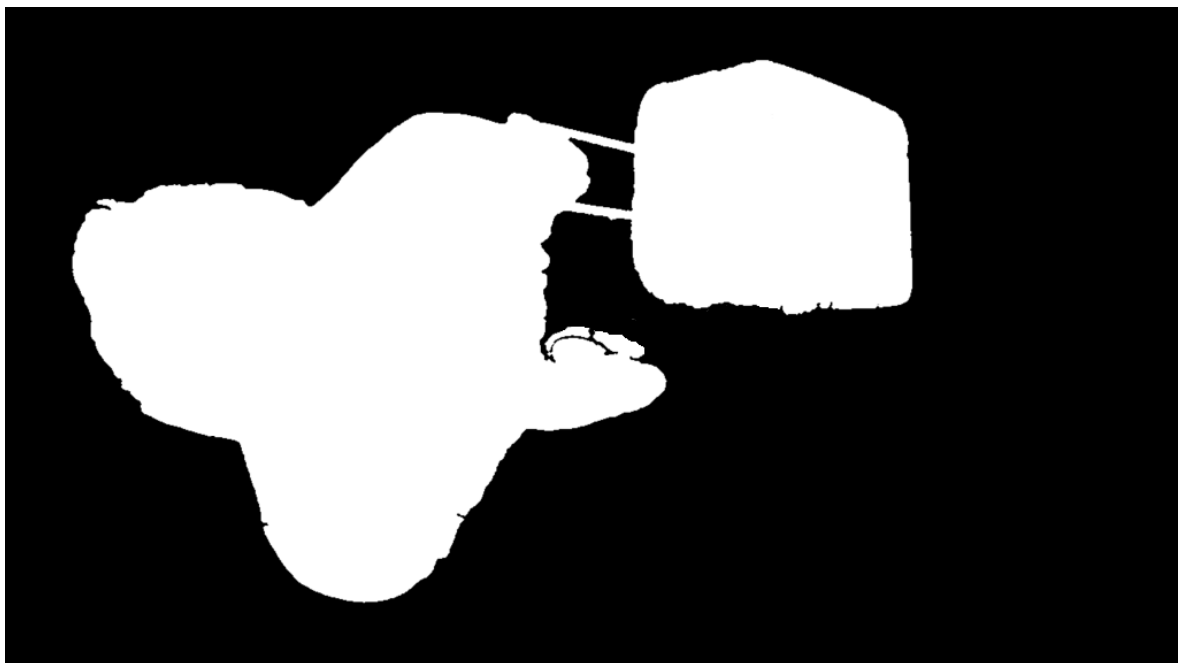
### 6.3 Klasifikace osoby s kufrem

Klasifikace osoby s kufrem je posední ukázkou programu, na jehož detekování byla naučena SVM metoda strojového učení. Objektem na obrázku (Obr. 58) je osoba s kufrem správně detekována rámečkem a rozpoznána.



Obr. 58. Snímek s klasifikací - osoba s kufrem.

U binární mapy (viz Obr. 59) je možné si všimnout ostrého a čistého obrysu postavy.

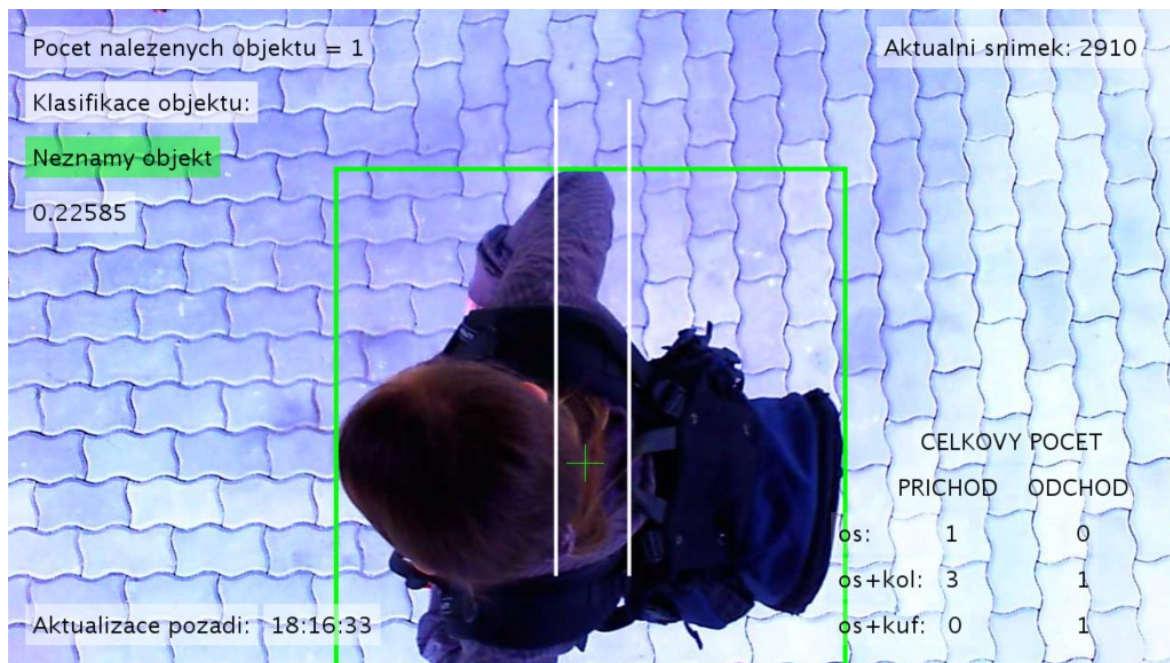


Obr. 59. Odpovídající binární mapa pro klasifikaci osoby s kufrem.



## 6.4 Detekce neznámého objektu

Pro případ, kdy klasifikátor nemohl s určitostí rozpoznat objekt, se vypíše text: Neznámý objekt. U následujícího obrázku (Obr. 60) klasifikátor nerozpoznal osobu s batohem, protože byl naučen pouze na samotnou osobu. Pro eliminaci takových případů, kdy objekt nebyl rozpoznán, by stačilo naučit klasifikátor na osobu s batohem a všech možných situací, které by se mohly vyskytnout při průchodu terminálem.



Obr. 60. Detekce neznámého objektu

U binární mapy (viz Obr. 61) je možné si všimnout objektu, který se nepochybně neliší od žádného z natrénovaných obrázků (viz kapitola 5.1), proto byl klasifikován jako neznámý objekt.



Obr. 61. Odpovídající binární mapa pro klasifikaci neznámého objektu

## 6.5 Klasifikace za zhoršených podmínek

Následuje testování klasifikace objektu za zhoršených podmínek. Na snímku níže (Obr. 62) je osoba s barvou oblečení velmi podobnou pozadí (podlaze). Nicméně díky správně nastavenému práhu (viz kapitola 3.2.1, kapitola 3.2.2) je na binární mapě obrázku pozorovatelné jen malé zhoršení odečtení pozadí a snímek je vyhodnocen správně, klasifikována je osoba s kolem. Nastavení nižšího práhu by znamenalo více stínů v objektu, což je nežádoucí, pokud klasifikace pracuje správně.



Obr. 62. Snímek s klasifikací – osoba, zhoršené podmínky.

Následující binární mapa (Obr. 63) ukazuje nedostatky metody odečítání pozadí, pokud je stejný odstín barev jako na pozadí, kdy může docházet ke zkreslování při substrakci popředí. Pokud se zaměříme na levou ruku objektu, která drží řídítka kola, můžeme si všimnout, že tato část není vykreslena, protože má stejnou barvu jako pozadí. Tento jev způsobilo velké množství dopadajícího světla, které zvýšilo jas objektu a zároveň pozadí. Obrys postavy ale zůstává stále rozeznatelný.



Obr. 63. Odpovídající binární mapa pro klasifikaci osoby s kolem, zhoršené podmínky.



## 6.6 Vyhodnocení výsledků

Pro vyhodnocení výsledků bylo využito čtyř osob, dále jen objektů, které byly natočeny kamerou při pěti různých situacích:

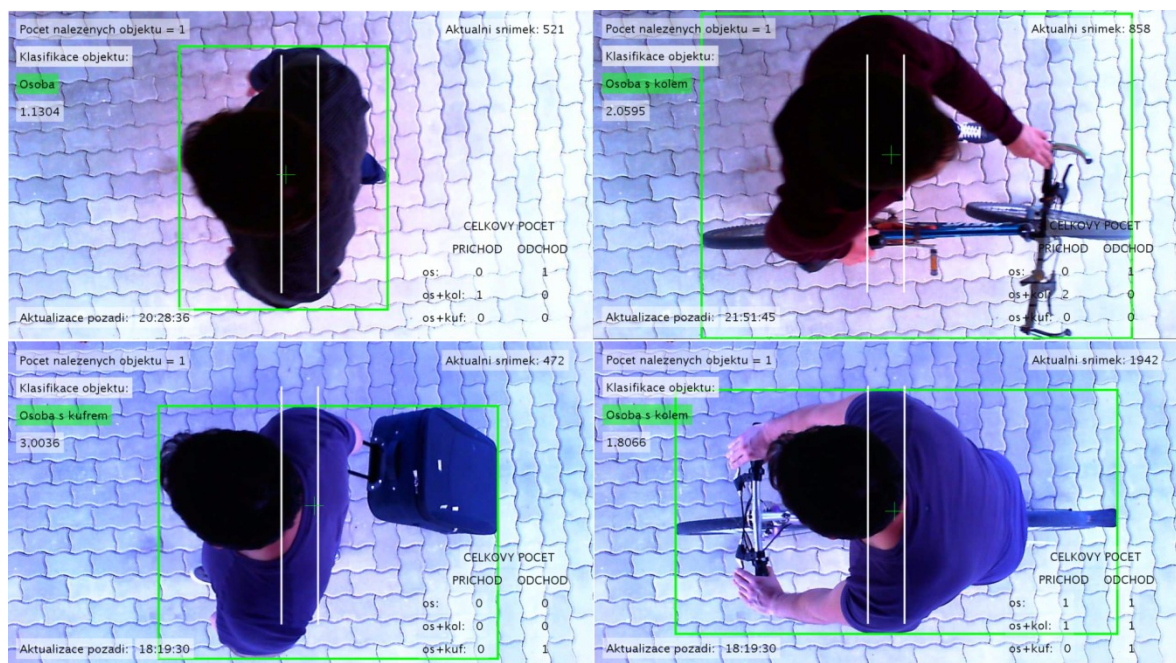
1. Osoba
2. Osoba s kolem
3. Osoba s kufrem
4. Osoba na kole
5. Osoba s batohem

První tři situace byly natrénovány na testovacím objektu číslo 1 (viz kapitola 6.6.1).

Pro další vyhodnocování v následujících kapitolách již nebudou uvedeny binární mapy obrázků.

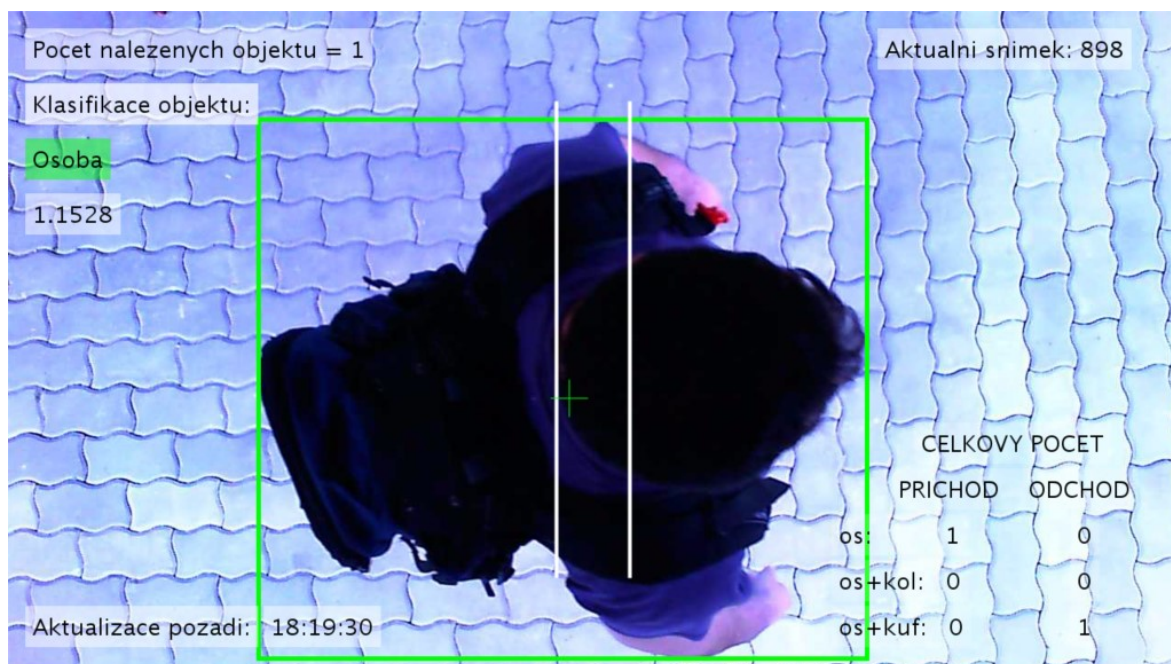
### 6.6.1 Testovaný objekt 1

První testovaný objekt 1 byl předmětem trénovací množiny, pouze zachycený při jiných podmínkách a s jiným oblečením (viz Obr. 64).



Obr. 64. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem (vlevo dole), osoba na kole (vpravo dole)

Jako další zvláštní případ, pro který nebyl klasifikátor naučen, byla vybrána osoba s batohem. Objekt byl vyhodnocen správně, jako osoba, viz následující obrázek (Obr. 65).



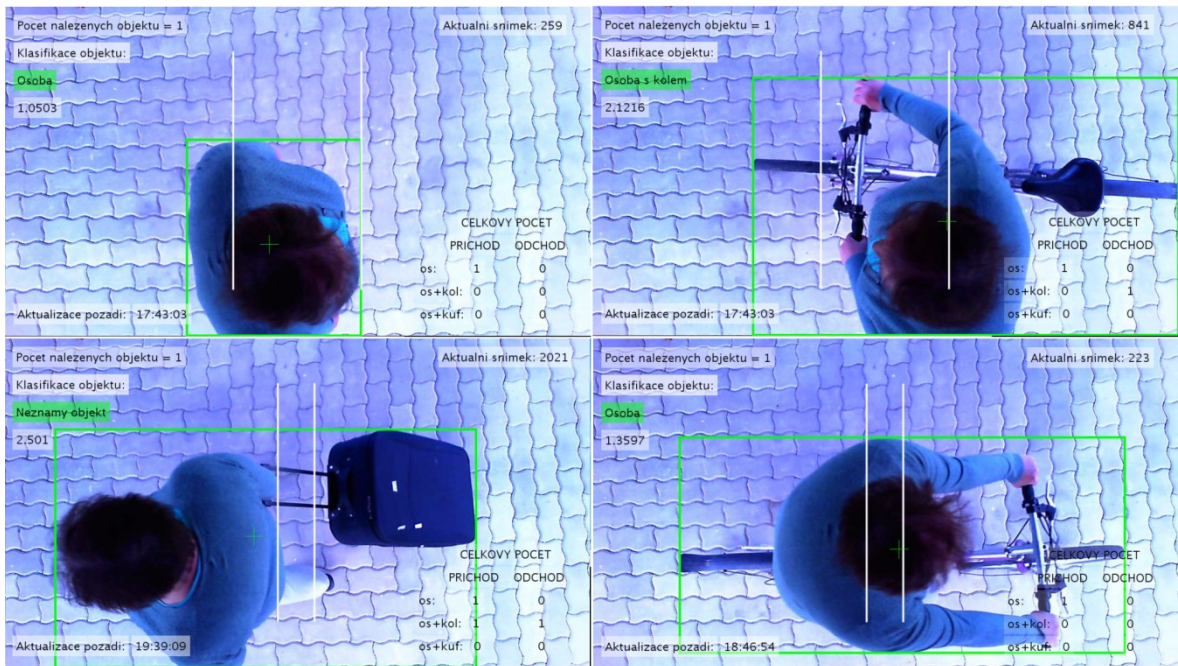
Obr. 65. Osoba - osoba s batohem

### 6.6.2 Testovaný objekt 2

Testovaný objekt 2 již nebyl předmětem trénovací množiny. Byl správně klasifikován jako osoba a osoba s kolem, ale při průchodu osoby s kufrem byl klasifikován jako neznámý objekt. Skóre se zastavilo na hodnotě 2,501, což je hranice přesně mezi osobou s kolem a osobou s kufrem.

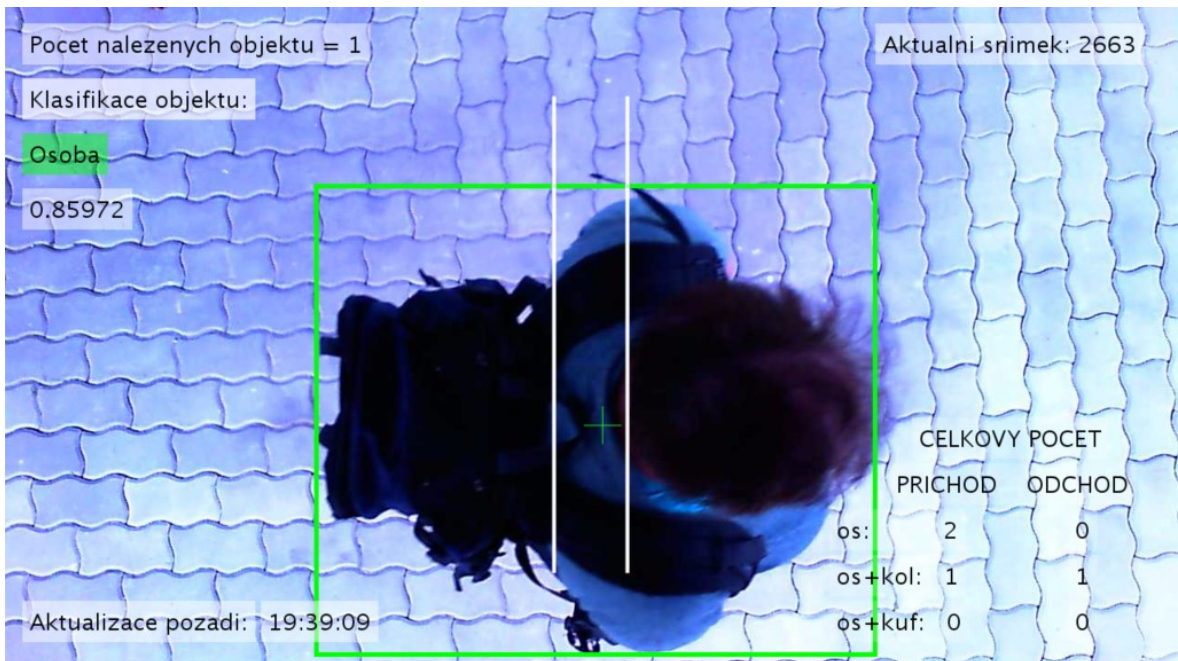
Klasifikace na obrázku (Obr. 66), vpravo dole byl snímek vyhodnocen jako osoba. Pro osobu na kole nebyl natrénován klasifikátor a tak se dalo předpokládat, že bude objekt vyhodnocen jako neznámý, případně jako osoba, nebo ještě osoba s kolem. Skóre se zastavilo na hodnotě 1,3597.





Obr. 66. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem – neznámý objekt (vlevo dole), osoba na kole (vpravo dole)

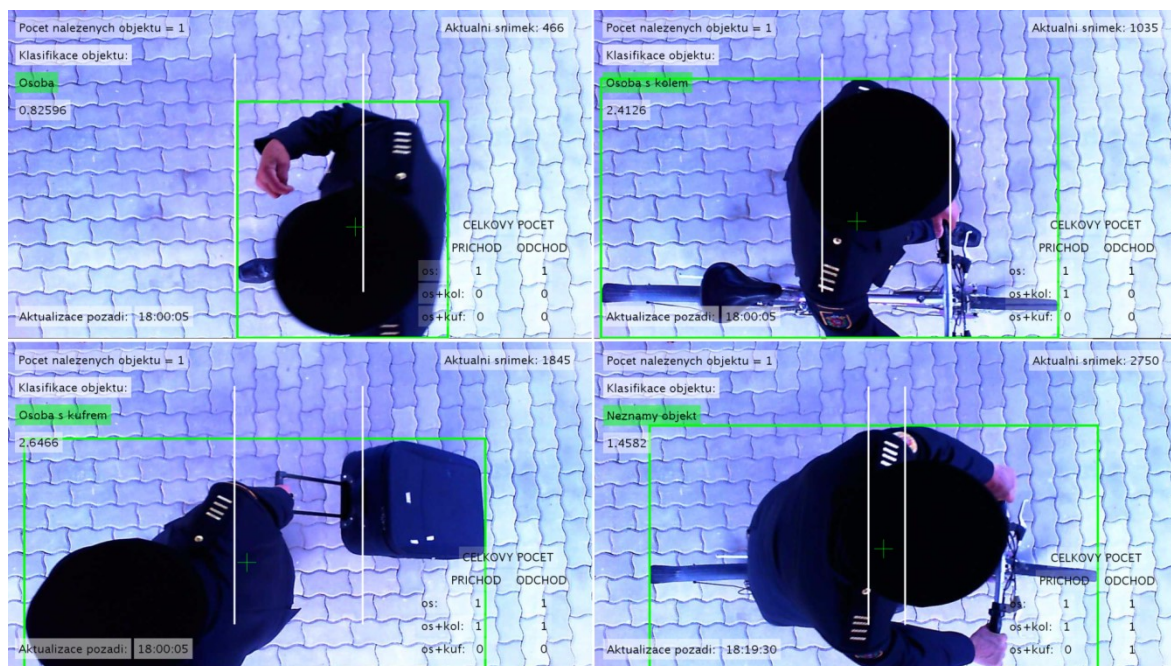
U následujícího obrázku (Obr. 67) je objekt vyhodnocen správně jako osoba, přestože osoba s batohem nebyla předmětem trénovací množiny klasifikátoru.



Obr. 67. Osoba – osoba s batohem

### 6.6.3 Testovaný objekt 3

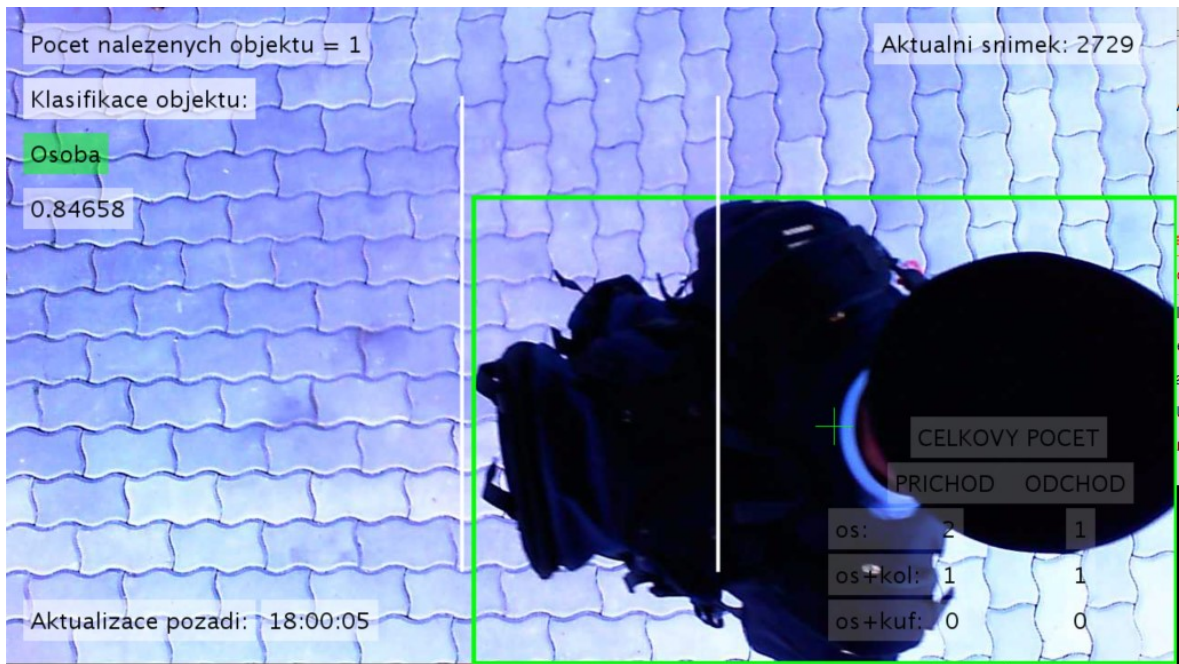
Testovaný objekt 3 také nebyl předmětem trénovací množiny. Byl správně klasifikován jako osoba, osoba s kolem, osoba s kufrem. V případě osoby na kole viz Obr. 68 byl objekt klasifikován jako neznámý, skóre se zastavilo na hodnotě 1,4582, tedy na hranici mezi osobou a osobou s kolem (viz Obr. 68).



Obr. 68. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem (vlevo dole), osoba na kole – neznámý objekt (vpravo dole)

U následujícího obrázku (Obr. 69) je objekt vyhodnocen správně jako osoba, přestože osoba s batohem nebyla předmětem trénovací množiny klasifikátoru.

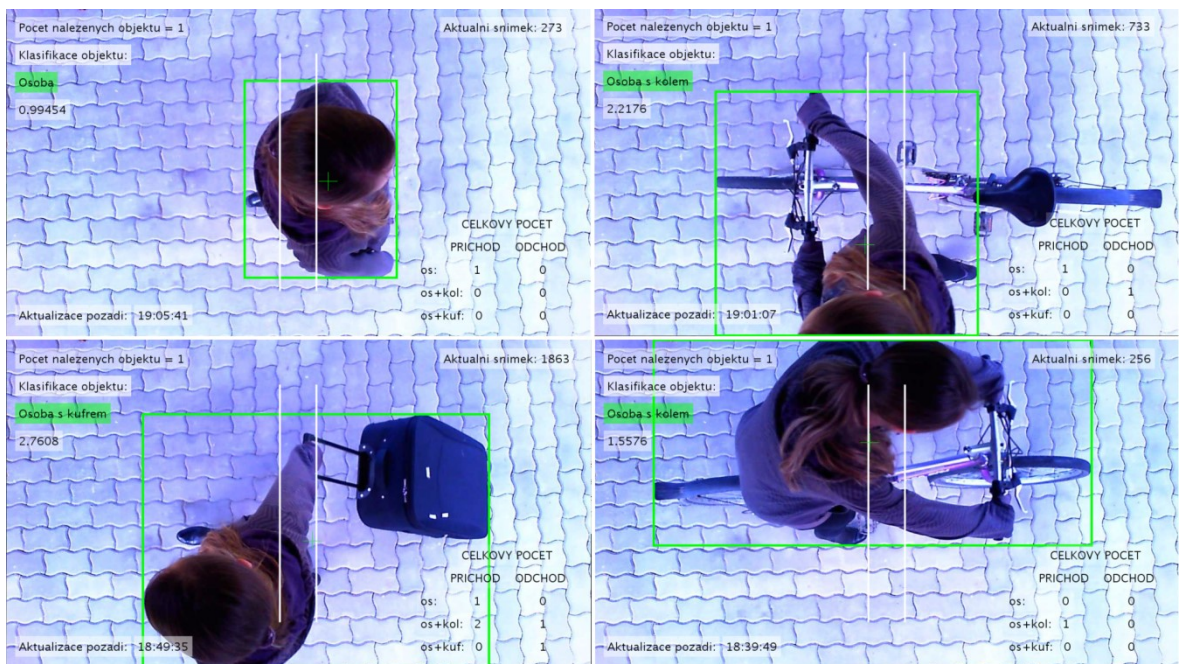




Obr. 69. Osoba – osoba s batohem

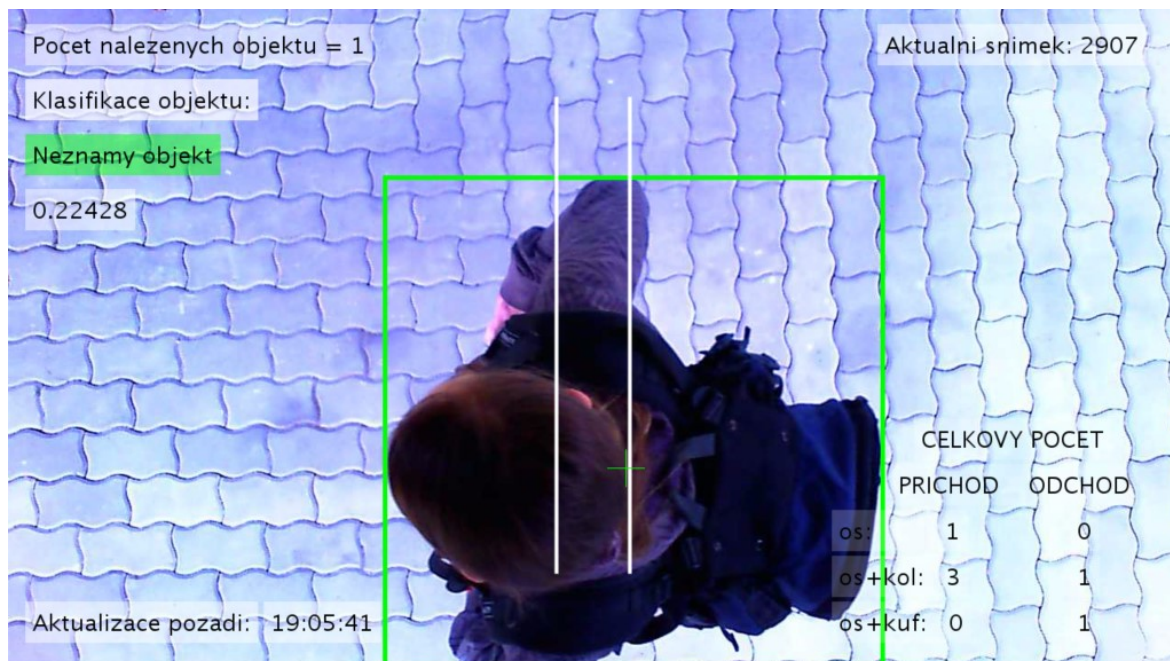
#### 6.6.4 Testovaný objekt 4

Testovaný objekt 4 rovněž nebyl předmětem trénovací množiny. Byl správně klasifikován jako osoba, osoba s kolem, osoba s kufrem i jako osoba na kole (s kolem), viz následující obrázek (Obr. 70).



Obr. 70. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem (vlevo dole), osoba na kole (vpravo dole)

U následujícího obrázku (Obr. 71) nebyl objekt klasifikován z toho důvodu, že klasifikátor nenašel podobnost mezi ani jedním z natrénovaných obrázků. Nejblíže je hodnota skóre přikloněna pro klasifikaci jako osoba.



Obr. 71. Osoba s batohem - neznámý objekt

V následující tabulce (Tab. 2) je vyobrazena úspěšnost rozpoznávání, kde zelená barva značí správně rozpoznáný objekt, oranžová barva značí nerozpoznaný objekt a červená barva značí špatně rozpoznáný objekt.

#### Vyhodnocení klasifikace:

- 1) **Předtrénované - na které byl klasifikátor naučen s objektem č.1 (Obj1, viz Tab. 2)**

Pouze jeden objekt byl klasifikován jako neznámý, skóre bylo zastaveno přesně na hranici mezi osobou s kolem a osobou s kufrem, které je přesně 2,5.

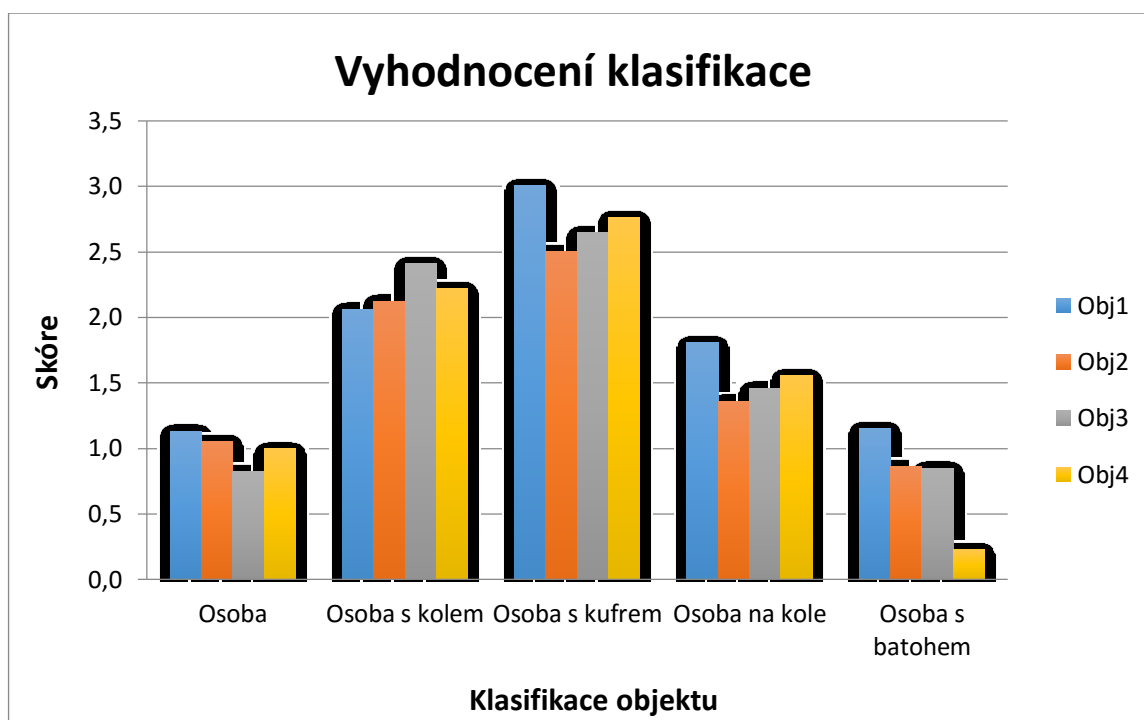
- 2) **Netrénované - na které nebyl klasifikátor naučen**

U objektu č.2 (Obj2, viz Tab. 2) byla osoba na kole klasifikována jako osoba. Vzhledem k tomu, že obrys osoby v binární mapě překrývá kolo, se tato situace dala očekávat. Předmětem trénování byla pouze osoba jdoucí vedle kola. Dále u objektu č.3 (Obj3, viz Tab. 2) nebyla rozpoznána osoba s batohem. A u objektu č.4 (Obj4, viz Tab. 2) nebyla rozpoznána osoba s batohem.

Tab. 2. Klasifikace objektu

Klasifikace objektu					
	Předtrénované			Netrénované	
	Osoba	Osoba s kolem	Osoba s kufrem	Osoba na kole	Osoba s batohem
Obj1	1,1304	2,0595	3,0036	1,8066	1,1528
Obj2	1,0503	2,1216	2,5010	1,3597	0,8597
Obj3	0,8259	2,4126	2,6466	1,4582	0,8466
Obj4	0,9945	2,2176	2,7608	1,5576	0,2243

Obrázek s grafem (Obr. 72) srovnává všechny objekty vedle sebe a můžeme si všimnout výkyvu skóre pro jednotlivé situace.



Obr. 72. Graf vyhodnocení klasifikace

Následující tabulka (Tab. 3) ukazuje vyhodnocení klasifikace a tabulka (Tab. 4) vyobrazuje celkovou procentuální úspěšnost. Osoba byla vždy klasifikována jako osoba, osoba s kolem byla taktéž vždy rozpoznána a osoba se zavazadlem byla jednou klasifikována jako neznámý objekt.

Tab. 3. Vyhodnocení klasifikace

Vyhodnocení		Skutečný objekt				
		Předtrénované			Netrénované	
		Osoba	Osoba s kolem	Osoba se zavazadlem	Osoba na kole	Osoba s batohem
Osoba	4	0	0	1	3	
Osoba s kolem	0	4	0	2	0	
Osoba se zavazadlem	0	0	3	0	0	
Neznámý objekt	0	0	1	1	1	

Tab. 4. Celková procentuální úspěšnost pro předtrénované objekty

	Osoba	Osoba s kolem	Osoba se zavazadlem
Úspěšnost	100%	100%	75%

## 6.7 Možnosti rozšíření

Úlohu by bylo možné do budoucna rozšířit o možnost klasifikace více tříd objektů, např. osoba s kočárkem, osoba s taškou v ruce, atp. Následně by bylo možné spustit testování v reálných podmínkách spolu s turnikety.

Dále by se dala úloha upravit pro použití v reálném čase, stačilo by snížit rozlišení webové kamery a optimalizovat časy jednotlivých operací. V našem případě byl použit obraz v rozlišení HD, ale pro klasifikaci založenou na učení klasifikátoru s binární mapou by stačilo mnohem menší rozlišení.

Úloha by se dala ještě doplnit o jiné metody klasifikace objektů, pro případ že by byl objekt klasifikován jako neznámý.



## ZÁVĚR

Cílem této práce bylo navrhnout vhodnou metodiku pro automatickou detekci a klasifikaci objektů v programovacím prostředí MATLAB. Předpokladem pro splnění tohoto cíle bylo nastudování literární rešerše a seznámení se s toolboxy MATLABu.

V samotném úvodu vlastní práce byla vybrána webová kamera nutná ke snímání a zpracování obrazu. V dalším kroku bylo nutné nastavit jednotlivé parametry kamery. Nejdůležitějším nastavením parametru kamery bylo pevné nastavení expozice, bez kterého by později zvolená metoda odečítání pozadí nebyla účinná ani použitelná.

Následně bylo nutné vytvořit binární obrázky s objekty v různých stádiích pohybu pro natrénování klasifikátoru SVM, který je součástí Statistics and Machine Learning Toolbox. Každá třída obsahovala 30 binárních obrázků. Takto natrénovaný klasifikátor byl uložen do souboru a připraven pro klasifikaci.

Pro segmentaci obrazu jsme použili metodu odečítání pozadí, která se jevila jako nejvhodnější pro tento typ úlohy se statickou kamerou. Segmentací obrazu nám vznikla maska popředí, u které bylo nutné redukovat stíny a dále na ni aplikovat morfologické operace. Vzniklá výsledná maska byla vstupem pro klasifikaci prediktoru.

Klasifikace probíhala pomocí metody strojového učení SVM, která byla naučena na klasifikaci binární mapy prvního z testovaných objektů. Následně byl detekovaný objekt ve stanoveném místě ihned pod kamerou klasifikován. Jednotlivé třídy objektů se rozřazovaly na základě skóre získaného z klasifikátoru. Takto rozpoznaný objekt byl následně započítán v tabulce příchodů nebo odchodů podle toho, ze kterého směru se objekt pohyboval.

Vyhodnocení výsledků proběhlo na celkem čtyřech objektech. Testovány byly i situace, na které nebyl klasifikátor trénován, konkrétně se jednalo o situaci: osoba na kole a osoba s batohem na zádech. Nicméně klasifikátor se rozhodoval správně ve většině případů a realizovaná metoda byla vyhodnocena jako úspěšná.

## SEZNAM POUŽITÉ LITERATURY

- [1] NOVOTNÝ, Stanislav. *Detekce mobilného robotu zpracováním obrazu*. Diplomová práce [online]. Brno, 2012 [cit. 2018-05-19]. Dostupné z: <https://dspace.vutbr.cz/xmlui/handle/11012/14063>
- [2] NOVOTNÝ, Martin. *Rozpoznávání objektů s využitím programu MATLAB*. Diplomová práce [online]. Brno, 2012 [cit. 2018-05-19]. Dostupné z: [http://is.mendelu.cz/zp/portal\\_zp.pl?prehled=vyhledavani;podrobnosti=48924;download\\_prace=1](http://is.mendelu.cz/zp/portal_zp.pl?prehled=vyhledavani;podrobnosti=48924;download_prace=1)
- [3] GONZALEZ, Rafael C., Richard E. WOODS a Steven L. EDDINS. *Digital Image processing using MATLAB*. 2nd ed. S.I.: Gatesmark Pub, 2009. ISBN 978-0982085400.
- [4] RUSS, John C. *The image processing handbook*. 6th ed. Boca Raton: CRC Press, 2011. ISBN 978-1439840450.
- [5] HORÁK, Karel. KALOVÁ, Ilona. PETYOVSKÝ, Petr. RICHTER, Miloslav. *Počítačové vidění* [online]. Brno, 2008 [cit. 2018-05-14]. Dostupné z: [http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove\\_videni\\_S.pdf](http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove_videni_S.pdf)
- [6] DAVIES, E. R. *Computer and machine vision: theory, algorithms, practicalities*. 4th ed. Boston: Elsevier, 2012. ISBN 9780123869081.
- [7] HLAVÁČ, Václav, SEDLÁČEK, Miloš. *Zpracování signálu a obrazu* [online]. Praha, 1999 [cit. 2018-05-14]. Dostupné z: [http://www.ugn.cas.cz/~kolcun/G/azro/txt/cvut\\_Hlavac\\_Sedlacek.pdf](http://www.ugn.cas.cz/~kolcun/G/azro/txt/cvut_Hlavac_Sedlacek.pdf)
- [8] HÁJOVSKÝ, R., PUSTKOVÁ R., KUTÁLEK F. *Zpracování obrazu v měřicí a řídicí technice: učební text* [online]. Ostrava, 2012 [cit. 2018-05-14]. Dostupné z: [http://www.person.vsb.cz/archivcd/FEI/ZOMRT/Zpracovani\\_obrazu\\_v\\_merici\\_a\\_ridici\\_technice.pdf](http://www.person.vsb.cz/archivcd/FEI/ZOMRT/Zpracovani_obrazu_v_merici_a_ridici_technice.pdf)
- [9] DUŠEK, František. *MATLAB a Simulink: úvod do používání* [online]. Pardubice, 2000 [cit. 2018-05-16]. Dostupné z: [http://bellman.zcu.cz/~mcech/simulink\\_navod.pdf](http://bellman.zcu.cz/~mcech/simulink_navod.pdf)
- [10] MARQUES, Oge. *Practical image and video processing using MATLAB* [online]. Florida Atlantic University, 2000 [cit. 2018-05-16]. ISBN 978111093467. Dostupné z:

- <https://www.researchgate.net/file.PostFileLoader.html?id=5767d7e496b7e4473f1419e1&assetKey=AS%3A375020493787139%401466423261773>
- [11] Image processing Toolbox. *Humusoft.cz* [online]. © HUMUSOFT 1991-2018 [cit. 2018-05-16]. Dostupné z: <http://www.humusoft.cz/matlab/image/>
- [12] Image acquisition Toolbox. *Humusoft.cz* [online]. © HUMUSOFT 1991-2018 [cit. 2018-05-16]. Dostupné z: <http://www.humusoft.cz/matlab/imaq/>
- [13] MACHOVÁ, Miluše. Metody rozpoznávání objektů v obrazu. *Anzdoc.com* [online]. © 2018 ANZDOC.COM [cit. 2018-05-16]. Dostupné z: <https://anzdoc.com/m-e-t-o-d-y-r-o-z-p-o-z-nani-ob-j-e-k-t-v-o-b-r-a-z-u.html>
- [14] DANEL, Roman. *Základy informatiky: počítačová grafika* [online]. Ostrava [cit. 2018-05-16]. Dostupné z: <http://slideplayer.cz/slide/2412023/>
- [15] VOLNER, Rudolf. *Počítačová grafika* [online]. Ružomberok, 2000 [cit. 2018-05-16]. Dostupné z: <http://docplayer.cz/2191485-Katolicka-univerzita-v-ruzomberku-pedagogicka-fakulta-katedra-informatiky-pocitacova-grafika-doc-ing-rudolf-volner-ph-d.html>
- [16] ŠVIHLÍK, Jan. *Metody rekonstrukce obrazu a odstranění šumu z obrazu* [online]. Praha [cit. 2018-05-16]. Dostupné z: <http://radio.feld.cvut.cz/courses/D37LBR/materialy.php?akce=dlf&zdroj=vpm&fkey=21&xtgt=2f686f6d652f53657276696365732f7777772f68746d6c2f6564755f6465706f742f2f5833374f4246>
- [17] HRŮZ, Marek. *LBP, HoG* [online]. Plzeň, 2015 [cit. 2018-05-17]. Dostupné z: <http://www.kky.zcu.cz/uploads/courses/mpv/05/materialy05.pdf>
- [18] *Support vector machines (SVM)*. [online]. [cit. 2018-05-17]. Dostupné z [https://is.muni.cz/el/1433/podzim2006/PA034/09\\_SVM.pdf](https://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf)
- [19] VOLNÁ, Eva. *Umělá inteligence* [online]. Ostrava, 2013 [cit. 2018-05-17]. Dostupné z: [http://www1.osu.cz/~volna/Umela\\_inteligence\\_skripta.pdf](http://www1.osu.cz/~volna/Umela_inteligence_skripta.pdf)
- [20] *Klasifikace pomocí hranic*. [online]. [cit. 2018-05-17]. Dostupné z: [https://is.muni.cz/www/98951/41610771/43823411/43823458/Analyza\\_a\\_hodnoc/44563155/56398089/Vicerozmerky\\_-\\_kap11.4\\_-\\_klasifikace\\_dle\\_hranic\\_-\\_FLDA\\_a\\_SVM.pdf](https://is.muni.cz/www/98951/41610771/43823411/43823458/Analyza_a_hodnoc/44563155/56398089/Vicerozmerky_-_kap11.4_-_klasifikace_dle_hranic_-_FLDA_a_SVM.pdf)
- [21] KOLÁČEK, Jan. KONEČNÁ, Kateřina. *Jak pracovat s MATLABem* [online]. [cit. 2018-05-19]. Dostupné z: <https://www.math.muni.cz/~kolacek/vyuka/vypsyst/navod.pdf>

- [22] HRÚZ, Marek. *SIFT, SURF, MSER* [online]. Plzeň, 2015 [cit. 2018-05-17].  
Dostupné z: <http://www.kky.zcu.cz/uploads/courses/mpv/04/materialy04.pdf>
- [23] NOVÁK, Jakub. *Rešerše principů a technologií snímání průchodu osob zabraňovacím zařízením*. Evropská unie – Evropský fond pro regionální rozvoj. OP Podnikání a inovace pro konkurenceschopnost. Název projektu: Modulární systém ENTER. Registrační číslo projektu: CZ.01.1.08/0.0/0.0/15\_019/0004581.
- [24] VONDRÁK, Ivo. *Neuronové sítě* [online]. Ostrava, 2009 [cit. 2018-05-18].  
Dostupné z: [http://vondrak.cs.vsb.cz/download/Neuronove\\_site.pdf](http://vondrak.cs.vsb.cz/download/Neuronove_site.pdf)

**SEZNAM POUŽITÝCH ZKRATEK**

MATLAB	MATrix LABoratory
HD	High Definition
RGB	Red Green Blue
HSV	Hue Saturation Value
SVM	Support Vector Machines
SURF	Speeded-Up Robust Features
HOG	Histogram orientací gradientů
Px	Pixel
3D	Troj rozměrný prostor
2D	Dvoj rozměrný prostor
Obr.	Obrázek
Obj.	Objekt

**SEZNAM OBRÁZKŮ**

Obr. 1. Řetězec snímání a zpracování obrazu [5] .....	12
Obr. 2. Typy vzorkovacích mřížek [5] .....	13
Obr. 3. Spojitý obraz ve vzorkovací mřížce (vlevo) a výsledný obraz po vzorkování a kvantování hodnot (vpravo) [8].....	14
Obr. 4. RGB model krychle [13].....	15
Obr. 5. Obrázek zobrazený v RGB modelu .....	15
Obr. 6. HSV Barevný model [13] .....	16
Obr. 7. Obrázky se zobrazenými jednotlivými složkami HSV (zleva: Hue, Saturation, Value) .....	16
Obr. 8. Šedo tónový obraz .....	17
Obr. 9. Binární maska.....	17
Obr. 10. Filtrace obrazu maskou 3x3 [16].....	18
Obr. 11. Příklad histogramu s osmi úrovní šedi [10] .....	20
Obr. 12. Původní obraz s histogramem (nahore) a ekvalizovaný obrázek s histogramem (dole) [5] .....	21
Obr. 13. Šedo tónový obraz před prahováním (vlevo) a po prahování (vpravo).....	21
Obr. 14. Schéma algoritmu odečítání pozadí.....	22
Obr. 15. Algoritmus pro získání masky popředí.....	23
Obr. 16. Model ideální a skutečné hrany [8] .....	24
Obr. 17. Detekce hran aplikovaná na šedo tónový obraz (vlevo) a na binární mapu (vpravo) .....	24
Obr. 18. Obraz z kamery a hloubková mapa [23].....	25
Obr. 19. Obraz barevné RGB aktivní kamery a odpovídající hloubková mapa [23].....	25
Obr. 20. Algoritmus detekce stínu .....	26
Obr. 21. Histogram orientací gradientů pro různé hodnoty intervalu $i$ a počtu binů $s$ [17].....	28
Obr. 22. Princip konstrukce HOG deskriptoru [17].....	29
Obr. 23. Vizualizace metody HOG při pohledu shora .....	29
Obr. 24. Princip vzniku možnosti lineárního oddělení dvou tříd s nelineárními hranicemi pomocí přidané dimenze. [18].....	30
Obr. 25. Ukázka dělení hranic trénovací množiny. [20] .....	31

Obr. 26. Hranice tolerančního pásma, hranice (černá přímka), tolerance (modrá přerušovaná přímka). [20] .....	32
Obr. 27. Součet hodno v bloku $\Sigma$ [22] .....	33
Obr. 28. Aproximace podle druhé derivace Gaussovy funkce podle x, y a dle xy [22].....	33
Obr. 29. Schématický model neuronu [24].....	34
Obr. 30. Dvou rozměrný případ rozpoznávání pomocí perceptronu.....	35
Obr. 31. Prostředí MATLAB R2017a .....	36
Obr. 32. Webová kamera Microsoft LifeCam HD-3000.....	41
Obr. 33. Model prostředí snímaného webovou kamerou .....	42
Obr. 34. Vlastnosti kamery zobrazené pomocí programu MATLAB .....	43
Obr. 35. Program pro záznam videa – náhled.....	44
Obr. 36. Blokové schéma programu pro klasifikaci objektu .....	46
Obr. 37. Blokové schéma trénování dat a učení objektů.....	47
Obr. 38. Binární mapa s obrysem postavy, určena k trénování SVM. ....	48
Obr. 39. Binární mapa s obrysem postavy a kolem, určena k trénování SVM.....	48
Obr. 40. Binární mapa s obrysem postavy a kufrem, určena k trénování SVM. ....	49
Obr. 41. Binární mapa po odečtení pozadí. ....	52
Obr. 42. Snímek převedený z RGB na HSV model a zobrazený v RGB.....	52
Obr. 43. Nalezený stín pro zadané hodnoty sytosti a jasu v programu. ....	53
Obr. 44. Odečtený stín od binární mapy.....	53
Obr. 45. Model pozadí v RGB (vlevo) a ve stupních šedi (vpravo). ....	54
Obr. 46. Model pozadí s objektem, obrázek v RGB (vlevo) a ve stupních šedi (vpravo). ....	54
Obr. 47. Binární maska vzniklá odečtem obrázku z pozadí a aktuálního snímku.....	55
Obr. 48. Upravená binární maska pomocí mediánového filtru.....	55
Obr. 49. Binární maska s odstraněnými objekty menšími než 1000px. ....	55
Obr. 50. Finální snímek po odfiltrování děr z obrázků. ....	56
Obr. 51. Popis aktuálně přehrávaného snímku v okně video přehrávače MATLABu.....	60
Obr. 52. Aktuálně detekovaný a rozpoznáný objekt. ....	61
Obr. 53. Binární mapa předcházejícího obrázku.....	61
Obr. 54. Snímek s klasifikací - osoba.....	63
Obr. 55. Odpovídající binární mapa pro klasifikaci osoby.....	63
Obr. 56. Snímek s klasifikací - osoba s kolem.....	64

Obr. 57. Odpovídající binární mapa pro klasifikaci osoby s kolem.....	64
Obr. 58. Snímek s klasifikací - osoba s kufrem. ....	65
Obr. 59. Odpovídající binární mapa pro klasifikaci osoby s kufrem. ....	65
Obr. 60. Detekce neznámého objektu .....	66
Obr. 61. Odpovídající binární mapa pro klasifikaci neznámého objektu.....	67
Obr. 62. Snímek s klasifikací – osoba, zhoršené podmínky. ....	68
Obr. 63. Odpovídající binární mapa pro klasifikaci osoby s kolem, zhoršené podmínky. ....	68
Obr. 64. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem (vlevo dole), osoba na kole (vpravo dole) .....	69
Obr. 65. Osoba - osoba s batohem .....	70
Obr. 66. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem – neznámý objekt (vlevo dole), osoba na kole (vpravo dole).....	71
Obr. 67. Osoba – osoba s batohem.....	71
Obr. 68. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem (vlevo dole), osoba na kole – neznámý objekt (vpravo dole).....	72
Obr. 69. Osoba – osoba s batohem.....	73
Obr. 70. Vyhodnocení výsledků klasifikace: osoba (vlevo nahoře), osoba s kolem (vpravo nahoře), osoba s kufrem (vlevo dole), osoba na kole (vpravo dole) .....	73
Obr. 71. Osoba s batohem - neznámý objekt.....	74
Obr. 72. Graf vyhodnocení klasifikace.....	75



**SEZNAM TABULEK**

<i>Tab. 1. Hranice klasifikovatelných kategorií.....</i>	58
<i>Tab. 2. Klasifikace objektu .....</i>	75
<i>Tab. 3. Vyhodnocení klasifikace .....</i>	76
<i>Tab. 4. Celková procentuální úspěšnost pro předtrénované objekty .....</i>	76