



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DETEKCE OBJEKTŮ

OBJECT DETECTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Pavel Vítek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miloslav Richter, Ph.D.

BRNO 2022

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Pavel Vítek

ID: 221028

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Detekce objektů

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte prostředí pro sledování pohybujících se kontrastních objektů ve snímcích/videích a stanovení/vizualizaci odchylky aktuálního průběhu oproti jiným (poloha, natočení, mezní stavy, symetrie ...). Jako vzor použijte lidskou postavu.

- 1) Seznamte se s metodami vyhodnocení obrazu vhodnými pro řešení dané úlohy.
- 2) Stanovte podmínky pro správnou práci aplikace. Vytvořte databázi snímků, videí pro testování algoritmů.
- 3) Navrhněte algoritmy pro sledování pohybujících se objektů, archivaci změřených průběhů, stanovení a popis odchylek mezi průběhy.
- 4) Otestujte navržené algoritmy a zhodnoťte jejich kvalitu.

DOPORUČENÁ LITERATURA:

Gonzalez R.C., Woods R.E.: Digital Image Processing, 4th edition, Pearson, 2017, ISBN 978-0133356724

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Práce pojednává o detekci objektů, kde je lidská postava použita jako vzor. V úvodní kapitole jsou definovány pojmy detekce a objekt. V druhé kapitole jsou popsány základy teorie počítačového vidění, jako je např. zpracování obrazu a jeho součásti. Podstatná část je také věnována Cannyho detektoru hran, detekci horní části těla a detekci postoje. Třetí kapitola popisuje vývojové prostředí MATLAB, ve kterém jsou implementovány funkce pro zpracování obrazu. Poslední kapitola je praktickou ukázkou detekce objektů. Jsou zde srovnány různé metody detekce horní části těla. V poslední podkapitole jsou srovnány dva způsoby určení odchylek, které vycházejí z předlohy. Závěrem jsou komparovány veškeré metody s ohledem na dosažené výsledky.

Klíčová slova

počítačové vidění, detekce objektů, pixel, operátor, detekce horní části těla, Haarovy příznaky, detekce postoje, Viola-Jones, MATLAB

Abstract

This paper deals with object detection where a human figure is used as a model. The introductory chapter defines terms detection and object. The second chapter describes the basic theory of computer vision , such as image processing and its parts. A substantial part is also focused on the Canny edge detection, upper body detection and pose detection. The third chapter describes the MATLAB integrated development environment in which the image processing functions are implemented. The last chapter is a practical demonstration of object detection. Different upper body detection methods are compared there. In the last subsection, two methods of determining the deviations based on the template are compared. Finally, all methods are compared with respect to the obtained results.

Keywords

computer vision, object detection, pixel, operator, upper body detection, Haar features, pose detection, Viola-Jones, MATLAB

Bibliografická citace

VÍTEK, Pavel. *Detekce objektů*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/142505>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Miloslav Richter.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: Pavel Vítek
VUT ID studenta: 221028
Typ práce: *Bakalářská práce*
Akademický rok: 2021/22
Téma závěrečné práce: *Detekce objektů*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 23. května 2022

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Miloslavu Richterovi, Ph.D. za cenné rady a skvělé vedení bakalářské práce. Dále bych chtěl poděkovat Jakubu Petrášovi za demonstraci cviků.

V Brně dne: 23. května 2022

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
1. ÚVOD.....	10
2. TEORIE POČÍTAČOVÉHO VIDĚNÍ.....	11
2.1 OBRAZ.....	11
2.2 PŘEDZPRACOVÁNÍ OBRAZU	11
2.2.1 Bodové jasové transformace	11
2.2.2 Geometrické transformace	12
2.3 ZPRACOVÁNÍ OBRAZU	12
2.3.1 Osvětlení	12
2.3.2 Objektiv.....	13
2.3.3 Senzor.....	13
2.3.4 A/D převod.....	14
2.3.5 Předzpracování.....	14
2.3.6 Segmentace	14
2.3.7 Popis	14
2.3.8 Aplikace	14
2.4 DETEKCE HRAN	14
2.5 CANNYHO HRANOVÝ DETEKTOR	15
2.5.1 Aplikace Gaussova filtru.....	15
2.5.2 Určení gradientu.....	16
2.5.3 Prahování gradientu	16
2.5.4 Aplikace dvojitého prahování	17
2.5.5 Nalezení hran pomocí hystereze.....	17
2.6 DETEKCE HORNÍ ČÁSTI TĚLA	17
2.6.1 Detekce pomocí Haarových příznaků	18
2.6.2 Integrální obraz	19
2.6.3 Aplikace klasifikačního algoritmu AdaBoost	20
2.6.4 Využití kaskády klasifikátorů	20
2.7 DETEKCE POSTOJE.....	20
2.7.1 Detekce lidského těla	20
2.7.2 Detekce jednotlivých částí lidského těla a vykreslení teplotních map.....	22
2.7.3 Vizualizace klíčových bodů	23
3. VÝVOJOVÉ PROSTŘEDÍ.....	25
3.1 MATLAB R2020b.....	25
3.1.1 Image Processing Toolbox.....	25
3.1.2 Computer Vision Toolbox	25
3.1.3 Parallel Computing Toolbox.....	25
3.1.4 Deep Learning Toolbox	25
3.1.5 Statistics and Machine Learning Toolbox.....	25
3.1.6 MATLAB Support Package for USB Webcams	25
3.1.7 MATLAB Support Package for IP Cameras	26
3.1.8 Human Pose Estimation with Deep Learning	26

4. PRAKTICKÁ REALIZACE.....	27
4.1 VYTVORENÍ PŘEDLOHY	27
4.2 ZPRACOVÁNÍ PŘEDLOHY	27
4.2.1 Detekce hran metodou Canny a nastavení parametrů	31
4.2.2 Geometrická transformace.....	32
4.2.3 Detekce koncové polohy.....	33
4.3 TVORBA VIDEA.....	35
4.3.1 Detekce člověka v záběru	35
4.3.2 Určení vzdálenosti postavy od kamery.....	41
4.3.3 Detekce postoje	44
4.3.4 Určení odchylky	46
4.3.5 Archivace průběhů	50
5. ZÁVĚR.....	51
LITERATURA.....	53
SEZNAM PŘÍLOH.....	57

SEZNAM OBRÁZKŮ

2.1	Řetězec zpracování obrazu.....	12
2.2	Typy ohniskových vzdáleností a jejich využití [39].....	13
2.3	Masky Robertsova operátoru 2x2 [9].....	15
2.4	Masky operátoru Prewittové (vlevo horizontální, vpravo vertikální) [9].....	15
2.5	Maska 5x5 použitá pro Gausův filtr [6].....	16
2.6	Haarovy hranové příznaky.....	18
2.7	Haarovy čárové příznaky.....	18
2.8	Haarův středový příznak.....	18
2.9	Příklad porovnání Haarových příznaků s částmi lidského obličeje [20].....	19
2.10	Hodnota integrálního obrazu v bodě (u,v).....	19
2.11	Postup pro detekci člověka metodou HOG [26].....	21
2.12	Vstupní obraz (vlevo) a histogram orientovaných gradientů (vpravo)[28].....	21
2.13	Znázornění algoritmu metody ACF.....	22
2.14	Teplotní mapy – první řádek je výstup neuronové sítě, která má za úkol rozpoznat jednotlivé části těla. Druhý řádek je výstup druhé neuronové sítě po regresi [29].....	23
2.15	Architektura procesu detekce částí lidského těla.[29].....	23
2.16	Skelet lidského těla [21].....	24
4.1	Základní poloha.....	28
4.2	Koncová poloha.....	28
4.3	Převedený obrázek do stupňů šedi.....	29
4.4	Převedený obrázek do stupňů šedi po programové úpravě.....	30
4.5	Detekované hrany základní polohy, automaticky nastavený threshold.....	31
4.6	Zdetekované hrany, manuálně nastavený threshold.....	32
4.7	Ořezaná postava.....	33
4.8	Koncová poloha – zmenšený obrázek.....	34
4.9	Vizualizované teplotní mapy.....	34
4.10	Detekovaný postoj.....	35
4.11	Detekovaná horní část těla.....	36
4.12	Detekovaný nos.....	37
4.13	Detekovaná ústa.....	37
4.14	Detekované oči (velký pár).....	38
4.15	Detekované oči (malý pár).....	38
4.16	Nedetekování očí po nasazení brýlí.....	39
4.17	Detekovaný obličej.....	39
4.18	Falešná detekce – první typ.....	40
4.19	Falešná detekce (vpravo dole) – druhý typ.....	41
4.20	Člověk je daleko od kamery.....	42
4.21	Člověk je blízko kamery.....	43
4.22	Člověk je ve správné vzdálenosti.....	43
4.23	Indexování klíčových bodů.....	44
4.24	Převedené klíčové body na postavě.....	46
4.25	Vykreslené klíčové body předlohy (první způsob).....	47
4.26	Zbarvení klíčového bodu do zelena po dosažení patřičných mezí (první způsob).....	48
4.27	Vykreslené ideální body (červeně) vzhledem k druhé ruce (druhý způsob).....	49
4.28	Zbarvení bodů do zelena po dosažení mezí (druhý způsob).....	50

1. ÚVOD

Člověk má pět hlavních smyslů – zrak, hmat, sluch, čich a chuť. Jedním z cílů techniky je napodobení či překonání těchto schopností pomocí počítačů. Významným milníkem ve snaze napodobit zrak člověka je vynález fotoaparátu (1825) [1]. Zaznamenávání okolního světa se stalo každodenní záležitostí. S postupem času, díky novějším technologiím, bylo možné zaznamenaný obraz nahrát a uložit do počítače. Získávání a zpracování informací z těchto snímků lze nazývat počítačovým viděním.

Praktické využití se uplatnilo v mnoha oblastech techniky a průmyslu, jako je např. astronomie (detekce vesmírných objektů), vojenský průmysl (mapování terénu), medicína (snímky z RTG, mikroskopů) nebo i státní správa (detekce SPZ). Jako další z příkladů lze uvést automobilové odvětví, kdy je díky (nejen) počítačovému vidění umožněn autopilot – plně autonomní řízení se schopností zastavení na červenou a rozpoznáním značek jako stůj, dej přednost v jízdě [2].

Významnou aplikací počítačového vidění je detekce hran a s ní spojena detekce objektů. Slovní spojení „detekce objektů“ může být vyloženo mnoha způsoby, a proto je potřeba ho ujasnit.

Detekce – „zjišťování, odhalování, objevování neznámého, skrytého“ [3]. Pro člověka se na videu nebo obrázku můžou objekty (např. lidé, auta apod.) zdát zřetelné a jasné. Pro počítač ale tomu tak není, protože zmiňované video nebo obrázek „chápe“ pouze jako uskupení pixelů.

Objekt – v oblasti počítačového vidění je objektem myšlen jakýkoliv obraz, ať už 2D (plošný) nebo 3D (prostorový). Dále se práce bude zabývat 2D objekty (projekcí).

Tato práce se zabývá algoritmy detekce objektů a jejich následné implementace v praxi. Jako vzor je použita lidská postava. Ve druhé kapitole je popsán Cannyho hranový detektor, detekce horní části těla pomocí Haarových příznaků a detekce postoje s využitím metody HOG (Histogram orientovaných gradientů). Všechny uvedené metody jsou součástí vývojového prostředí MATLAB. Hlavním cílem práce je vytvořit aplikaci pro korektní provedení cviku natočeného na kameru. Aplikace by tedy měla uživateli znázornit, jak se dostat do korektní polohy určitého cviku. Snímky (cvičení) lze zaznamenat mnoha způsoby. Ideální by bylo využít 3D kameru (kinect), což ale není moc praktické. Proto se tato práce zaměřuje na „po domácku“ vytvořené snímky zaznamenané na běžnou web kameru vestavěnou v notebooku. Uvažuje se jednoduchá, rychle sestavitelná scéna bez jakéhokoli speciálního osvětlení. Postačí si stoupnout před bílou zeď za denního světla. Další cíle práce jsou zhodnotit a porovnat metody detekce objektů, archivovat průběhy a určit odchylky.

2. TEORIE POČÍTAČOVÉHO VIDĚNÍ

Kapitola se věnuje základům teorie počítačového vidění od definice pojmu pixel až po metody detekce postoje. Je zde vysvětlen i pojem konvoluce, integrální obraz a Haarovy příznaky. Bez veškeré níže popsané teorie by nebylo možné prakticky aplikovat detekci objektů, proto je nutné se jí zabývat.

2.1 Obraz

Základem zobrazování je převedení vstupních dat na pixely (jednotky obrazu). Podle nejjednodušší definice je pixel malý čtvereček. Nejvýstižnější definicí pro účely této práce je, že pixel je čtverec se třemi složkami. Každá složka reprezentuje jednu barvu. Červenou, zelenou a modrou. Tato skupina barev se obecně označuje RGB [4]. Hodnota intenzity jednotlivých složek lze zobrazit v rozsahu hodnot 0–255. Potom je možno hovořit o tzv. osmibitové reprezentaci jedné z barev (R, G nebo B). V dnešní době se však využívá True color (24-bit, spojení tří osmibitových složek), Deep color (30-bit), 36-bit nebo 48-bit (Adobe Photoshop) [5].

Pixely je potřeba také indexovat. Jeden ze způsobů je indexování řádků a sloupců tak, jak tomu je u matic. Dále se připouští indexování po celých nezáporných číslech, z čehož vyplývá, že poslední pixel bude mít index roven šířce vynásobené výškou. V dalších částech bude použita indexace po řádcích a sloupcích.

2.2 Předzpracování obrazu

Tato podkapitola vychází z přednáškových skript, viz [6].

Ještě před zpracováním obrazu je vhodné provést takové úpravy, aby následné zpracování bylo co nejjednodušší a nejefektivnější. K těmto úpravám patří např. optimalizace jasu a kontrastu, vyhlazení šumu nebo zvýraznění obrysů obrazu. Důležité je přizpůsobit úpravy dané aplikaci.

K metodám předzpracování obrazu patří bodové jasové transformace, geometrické transformace apod.

2.2.1 Bodové jasové transformace

Jas v bodě výstupního obrazu závisí na jasu bodu ve vstupním obrazu, z čehož vyplývá, že pro reprezentaci jasu pixelu na výstupu je potřeba upravit hodnoty vstupního pixelu. Špatný jas může být zapříčiněn např. špatným osvětlením nebo špatnými světlocitlivými prvky. Korekci jasu lze provést tak, že pořídíme snímek objektu, poté pouze pozadí, a nakonec pořídíme snímek za tmy. Následným korektním zpracováním se jas optimalizuje.

Praktická ukázka korekce jasů je znázorněna viz obrázek 4.4.

2.2.2 Geometrické transformace

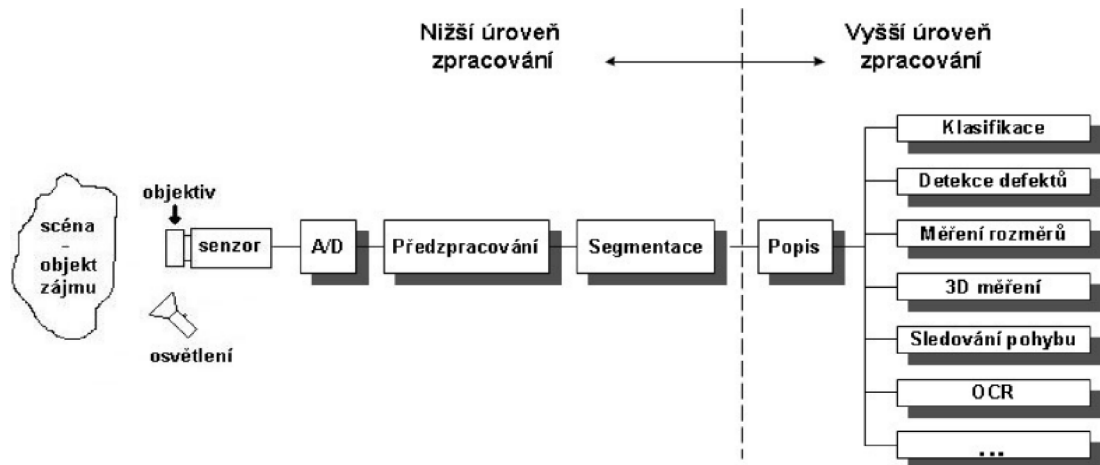
Při pořizování obrazu může dojít ke geometrické deformaci – zkosení vůči snímané ploše, pixely se namapují na stejné místo atd. K často užívaným geometrickým transformacím patří zkosení, otočení, zvětšení.

V práci lze geometrickou transformaci použít pro normalizaci velikosti postavy pro různou vzdálenost od kamery.

2.3 Zpracování obrazu

Tato podkapitola vychází z přednáškových skript, viz [6].

Zpracování obrazu je základním stavebním kamenem počítačového vidění. Tento postup lze graficky znázornit:



Obrázek 2.1 Řetězec zpracování obrazu

Jedná se o vhodnou posloupnost kroků. Tyto kroky budou efektivnější, pokud bude známo, k čemu bude obraz použit

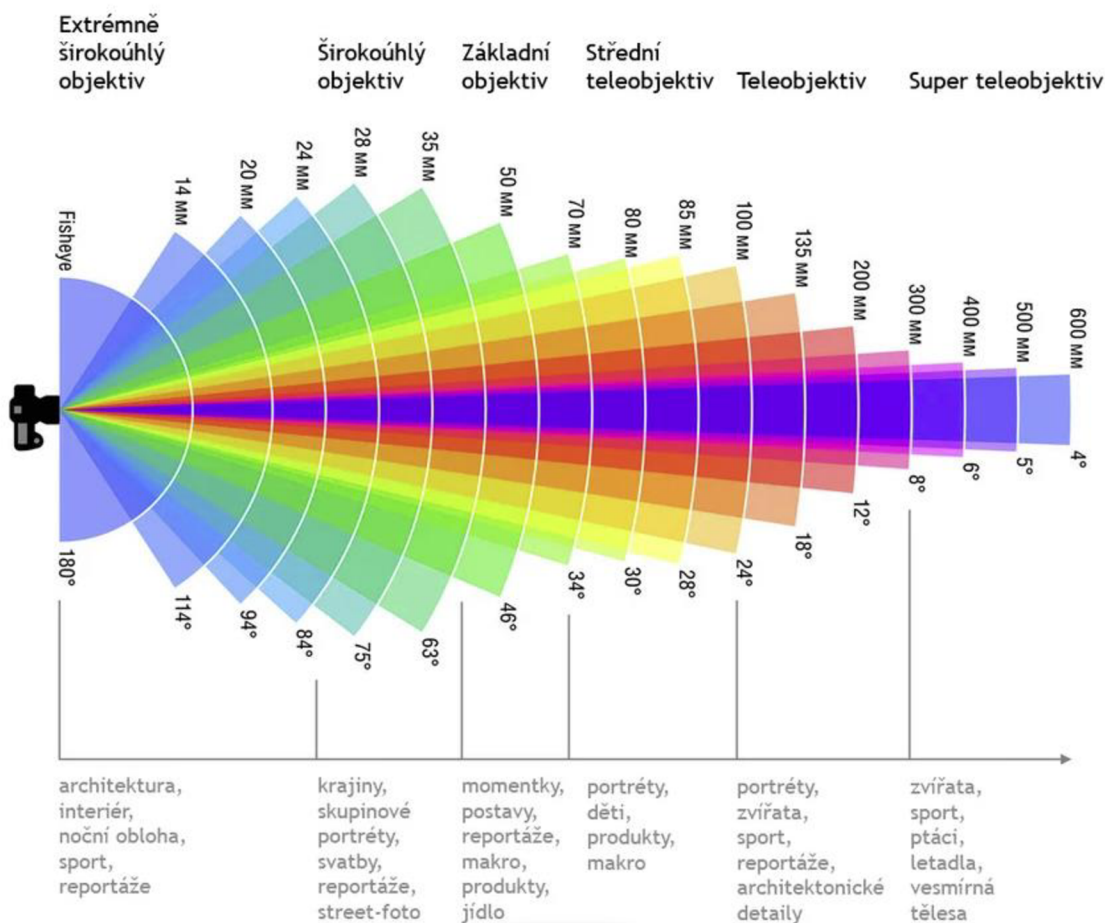
2.3.1 Osvětlení

Osvětlení je jeden z nejdůležitějších faktorů ovlivňující kvalitu zpracování obrazu. Bez kvalitního osvětlení není možné obraz kvalitně zachytit a tím pádem i zpracovat. Zdroje osvětlení můžeme rozdělit na přirozené (denní světlo) a umělé (zářivka, žárovka, LED světlo).

Pro tuto práci je vhodné, aby světlo nebylo bodové, ale všestranné a difuzní. Dalším důležitým faktorem je, aby se po osvětlení objektu nevytvořil stín, což by mohlo zapříčinit nekorektní detekci.

2.3.2 Objektiv

Objektiv je čočka nebo více poskládaných čoček za sebou [7]. Objektivy se liší podle ohniskových vzdáleností (např. 18 mm, 34 mm, 100 mm apod.). Každá vzdálenost je vhodná pro jinou scénu, viz obrázek 2.2. V práci se předpokládá využití web kamery s krátkou ohniskovou vzdáleností (tj. kromě člověka bude snímána i podstatná část pozadí).



Obrázek 2.2 Typy ohniskových vzdáleností a jejich využití [39]

2.3.3 Senzor

Senzor snímá přítomnost světla (zachycuje obraz), také určuje, zda bude obraz černobílý nebo barevný. Senzory se dělí na řádkový (lineární), plošný (maticový).

2.3.4 A/D převod

Převádí analogová(spojité) data na digitální (číslicová, diskrétní), která umožňují jednak kontrolu správnosti při přenosu a jednak další zpracování (např. na PC). Používají se různé typy A/D převodníků – paralelní, s postupnou integrací, sigma – delta apod.

2.3.5 Předzpracování

Jedná se o úpravu obrazu. Výstupem předzpracování by měl být obraz, který je co nejméně zašuměný, optimálně kontrastní a nezkreslený. Čím lepší budou tyto vlastnosti, tím lepší bude výsledný obraz a tím lepší bude eventuální detekce objektu.

2.3.6 Segmentace

Segmentace rozděluje obraz do segmentů(částí), redukuje data. Patří sem i oddělení popředí od pozadí.

2.3.7 Popis

Funkce popisu je popsat objekty v obrazu, vede k pochopení obrazu.

2.3.8 Aplikace

Zpracování obrazu se využívá v mnoha oborech. Některá využití byla probrána v úvodu této práce.

2.4 Detekce hran

Detekce hran je důležitou součástí detekce objektů. Existuje mnoho metod, jak hranu detekovat. Hranou (ve spojení s detekcí) se rozumí shluk pixelů, který má podstatně jinou intenzitu jasu, než okolí [8]. Mezi nejznámější hranové detektory patří Robertsův, Prewittové, Sobelův, Cannyho a jejich varianty. Každá z těchto metod se k problému staví jinak, ale v principu jsou stejné. Využívají masku, kterou postupně aplikují na každý prvek v matici pixelů a to tak, že střed masky se umístí na daný prvek, vynásobí se prvek i jeho okolí hodnotami masky, tyto vynásobené hodnoty se sečtou a výsledná hodnota je novou hodnotou prostředního prvku. Tomuto algoritmu se říká **konvoluce**. Maska je matice a často má rozměr 3x3 nebo 5x5. Výjimkou je Robertsova metoda detekce hran, která využívá masku 2x2.

V literatuře se může také vyskytnout v této souvislosti pojem *operátor*. Operátor je výsledek použití několika masek, přičemž maska je přímo vzor, který se aplikuje na obraz.

Dále bude probrán pouze Cannyho detektor, který je v práci využíváný.

-1	0
0	1

0	-1
1	0

Obrázek 2.3 Masky Robertsova operátoru 2x2 [9]

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Obrázek 2.4 Masky operátoru Prewittové (vlevo horizontální, vpravo vertikální) [9]

2.5 Cannyho hranový detektor

Jak již bylo zmíněno, za nejlepší se považuje Cannyho hranový detektor, který může využít jakýkoli operátor (např. Sobelův).

Postup tohoto algoritmu je následující, [10]

1. Aplikace Gaussova filtru
2. Určení gradientu
3. Prahování gradientu
4. Aplikace dvojitého prahování
5. Nalezení hran pomocí hystereze

2.5.1 Aplikace Gaussova filtru

Může se použít maska 3x3 nebo 5x5.

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Obrázek 2.5 Maska 5x5 použitá pro Gaussův filtr [6]

Dvourozměrný Gaussův filtr může být popsán následovně

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

Kde x a y jsou souřadnice obrazu a σ je směrodatná odchylka, která udává velikost okolí, ve kterém filtr pracuje [6].

Gaussův filtr se využije k odstranění šumu, který by jinak mohl zapříčinit nepřesnou nebo falešnou detekci hran.

2.5.2 Určení gradientu

Gradient může být definován jako směr růstu. Matematicky ho lze vyjádřit jako první derivaci. V tomto případě je to směr hran. Hrany jsou orientovány ve čtyřech směrech – horizontální, vertikální a dva směry diagonální. Proto se pro detekci hran ve všech směrech použijí čtyři filtry. Tyto filtry mohou využít některé z operátorů popsanych v kapitole 2.4 [10].

2.5.3 Prahování gradientu

Prahování (*thresholding*) – přiřadí všem pixelům hodnotu 0 (černá) nebo 1 (bílá) na základě jejich hodnoty. Pokud bude hodnota prahování např. 0,4, tak všechny pixely s hodnotou nižší než 0,4 budou klasifikovány jako černé (tmavé), všechny pixely s hodnotou vyšší než 0,4 budou klasifikovány jako bílé (světlé). Lze tak oddělit pozadí od popředí [11].

Porovnávají se hodnoty gradientu v okolí pixelu (ve směru hrany), kterým prochází hrana. Pokud mají vedlejší pixely nižší hodnoty gradientu, je pixel uznán jako hrana.

Pokud hodnoty jsou vyšší, není uznán jako hrana. Tímto postupem se zajistí ztenčení hran (thinning) a tím jejich přesnější detekce [10].

2.5.4 Aplikace dvojitého prahování

V tomto kroku již jsou definované všechny hrany čili i ty nejmenší (bezvýznamné). Proto je potřeba zajistit, aby se detekovaly pouze významné či opravdové hrany (ne ty způsobené šumem). Na rozdíl od ostatních algoritmů, u Cannyho hranového detektoru se využívá dvojitého thresholdu.

Nejdříve se empiricky zvolí horní a dolní práh (u různých obrazů bude různý). Poté se s těmito hodnotami porovnávají pixely. Mohou tedy nastat tři případy. Prvním případem je, že hodnota gradientu pixelu bude nižší než spodní hranice thresholdu. V tomto případě se nebude pixel brát v potaz. Druhým případem je, že pixel bude mít vyšší hodnotu, než je spodní hranice thresholdu a nižší hodnotu, než je horní hranice thresholdu. Tento pixel se označí jako slabá hrana. Poslední případ tedy je, že pixel bude mít větší hodnotu, než je horní hranice thresholdu. Takový pixel se označí jako silná hrana. Tato označení se využijí v dalším kroku [10].

2.5.5 Nalezení hran pomocí hystereze

V tomto kroku stále nejsou eliminovány falešné hrany způsobené šumem. Ve většině případů nejsou falešné hrany způsobené šumem spojeny se silnými hranami. Jejich odfiltrování se provede takzvanou analýzou blobů (*blob analysis*) [10]. Blob si lze představit jako velký objekt či nějaký region, ve kterém si jsou body podobné (když jsou např. spojeny či nějak jinak propojeny) [12]. Po této analýze má obraz finální podobu.

2.6 Detekce horní části těla

V práci je využito detekování horní části těla metodou Viola-Jones. Důvodem využití této detekce je ověření, zda je osoba v záběru.

Detektor Viola-Jones byl poprvé představen roku 2001 stejnojmennými autory. Pracuje s obrazy ve stupních šedi a využívá klasifikačního algoritmu AdaBoost [17]. Tento algoritmus může detekovat také obličej, nos, oči či ústa [18]. Je velmi robustní díky

nízké míře chyb typu I (falešná pozitivita). Také je velmi rychlý. Existují čtyři hlavní kroky, do kterých ho lze rozdělit. [19].

1. Detekce pomocí Haarových příznaků
2. Integrální obraz
3. Aplikace klasifikačního algoritmu AdaBoost
4. Využití kaskády klasifikátorů

Následující podkapitoly vycházejí převážně z diplomové práce viz [17].

2.6.1 Detekce pomocí Haarových příznaků

Haarovy příznaky jsou odvozeny od několika černobílých obdélníků. Ty se rozdělují podle typu informace a to na: hranové příznaky, čárové příznaky a středové příznaky. Bílá část obdélníku má váhu $w_0 = -1$. Černá část má váhu vypočtenou jako podíl černé a bílé plochy. Hodnota příznaku se vypočítá jako odezva na vstupní obraz dle rovnice:

$$f(x) = w_0 r_0 + w_1 r_1 \quad (2.2)$$

Kde $f(x)$ je odezva Haarova příznaku na vstupní snímek x , w_0 je váha bílé části obdélníku r_0 a w_1 je váha černé části r_1 .



Obrázek 2.6 Haarovy hranové příznaky

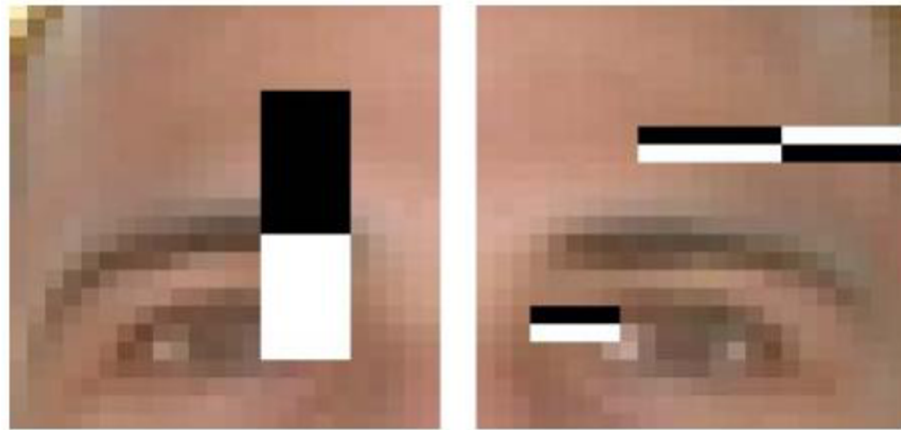


Obrázek 2.7 Haarovy čárové příznaky



Obrázek 2.8 Haarův středový příznak

Proces detekce probíhá porovnáním Haarových příznaků s různými rysy lidského těla (oblast očí je tmavší než kořen nosu apod.) [19].



Obrázek 2.9 Příklad porovnání Haarových příznaků s částmi lidského obličeje [20]

2.6.2 Integrální obraz

Integrální obraz je způsob, jak digitálně reprezentovat obraz. Princip spočívá v součtu hodnot předchozích pixelů směrem nahoru a doleva (poslední pixel, tj. vpravo dole, bude mít nejvyšší hodnotu). Využívá se pro zrychlení výpočtů.

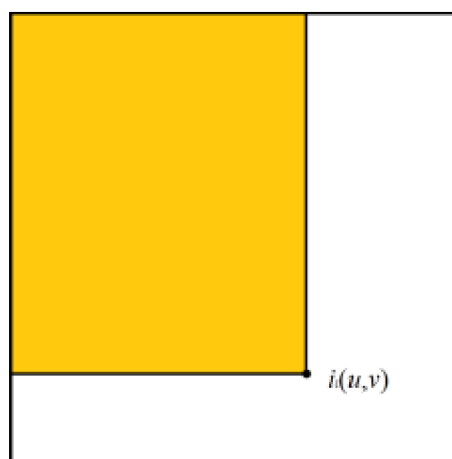
$$i_1(u, v) = \sum i(u' v'), \quad (2.3)$$

$$u' \leq u$$

$$v' \leq v$$

$i_1(u, v)$ – hodnoty integrálního obrazu

$i(u' v')$ - hodnoty vstupního obrazu



Obrázek 2.10 Hodnota integrálního obrazu v bodě (u,v)

2.6.3 Aplikace klasifikačního algoritmu AdaBoost

Před vysvětlením algoritmu AdaBoost je vhodné vědět, co je to lineární klasifikátor a jejich kaskáda.

Lineárním klasifikátorem se rozdělují data do dvou tříd (pozitivní a negativní). Tyto třídy jsou rozděleny libovolnou lineární funkcí.

AdaBoost (Adaptive Boosting) je metoda (algoritmus) pro trénování klasifikátorů. Jeho účelem je vybrat z množiny všech klasifikátorů H určitý počet slabých lineárních klasifikátorů $h(x)$ tak, aby jejich lineární kombinací vznikl silný nelineární klasifikátor $H(x)$ reprezentující stupeň kaskády.

Existují také další klasifikační algoritmy, např. SVM (Support vector machines).

2.6.4 Využití kaskády klasifikátorů

Kaskáda klasifikátorů má několik stupňů, kde každý stupeň je složen z několika slabých klasifikátorů. Jednotlivé stupně jsou silné monolitické nelineární klasifikátory. Mají určitou prahovou hodnotu P , podle které se klasifikuje do tříd (viz 2.6.3). Cílem je vytrénovat kaskádu tak, aby byl co nejmenší počet falešně pozitivních a falešně negativních dat. Kaskáda má za úkol zrychlit dobu detekce hledaného objektu.

2.7 Detekce postoje

Detekování či odhadování postoje (pozice, polohy) lidské postavy je jedním z odvětví počítačového vidění. Mezi aplikace patří např. snímání pohybu a rozpoznání gest (znaková řeč) [24]. Samotný proces detekce je často označován za náročný či těžko řešitelný, protože je potřeba se vypořádat s mnoha problémy. Mezi nejčastější patří úhel snímání kamery, různorodé tvary těl, špatné osvětlení nebo i pouhé oblečení [25]. Korektní detekce je docílena hlavně díky předtrénované neuronové síti. Architektura sítě využitá v práci vychází z modelu viz [22]. Tato neuronová síť je trénována datasetem COCO viz [23].

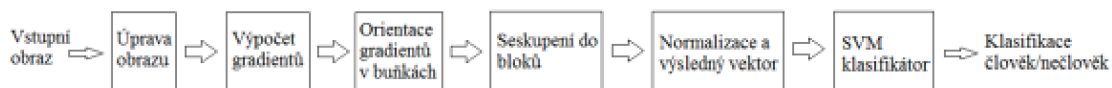
Protože práce využívá k detekci postoje [21], bude popisován postup detekce postoje uvedený v [21].

1. Detekce lidského těla
2. Detekce jednotlivých částí lidského těla a vykreslení teplotních map
3. Vizualizace klíčových bodů

2.7.1 Detekce lidského těla

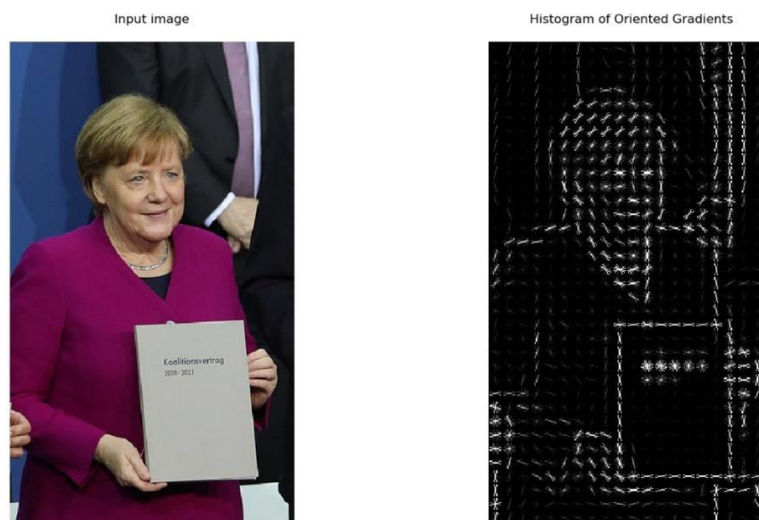
Aby bylo možné detekovat postoj, je nejprve nutné zjistit, zda a kde se člověk v záběru

nachází. Jedna z metod využívaných k detekci lidského těla (a zároveň implementovaných v MATLABu) je histogram orientovaných gradientů (angl. Histogram of Oriented Gradients, HOG). Metoda je založená, jak již název napovídá, na gradientu. Postup určení, jestli se jedná nebo nejedná o člověka, je následující.



Obrázek 2.11 Postup pro detekci člověka metodou HOG [26]

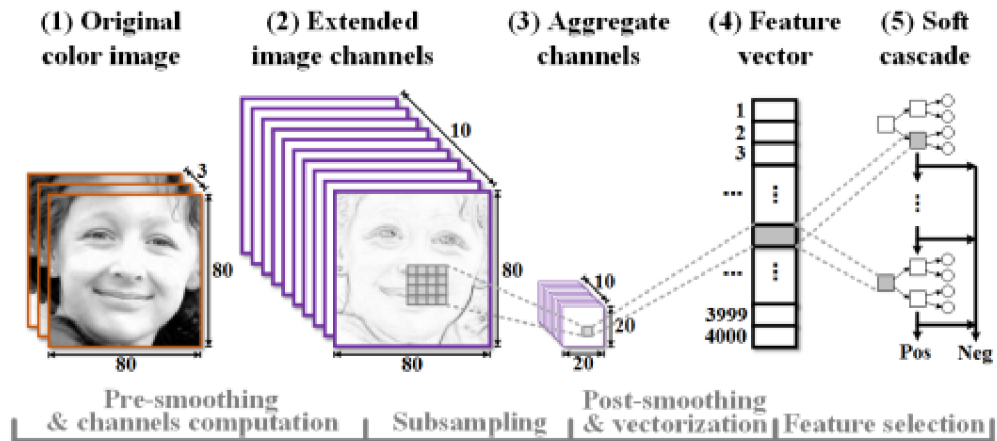
Nejdříve se provede úprava obrazu (úprava barev, gama korekce). Následně se vypočte gradient aplikací diskretní derivační masky $[-1, 0, 1]$. Poté se každý pixel rozdělí do jednoho z devíti košů podle toho, jakou má orientaci. Intenzita gradientu se liší v důsledku vlivů osvětlení a kontrastu. Proto je potřeba v kontrast normalizovat (předposlední blok). V poslední části se pomocí SVM klasifikátoru rozhodne, zda se jedná či nejedná o člověka a to tak, že klasifikátor rozdělí gradientní obraz na část, kde se člověk vyskytuje a kde nevyskytuje. Nakonec obraz vyhodnotí a rozhodne o přítomnosti postavy [26][27].



Obrázek 2.12 Vstupní obraz (vlevo) a histogram orientovaných gradientů (vpravo)[28]

Jedním z dalších způsobů detekce člověka je ACF (Aggregate Channel Features). Tato metoda využívá rozdělení RGB obrazu na barevné kanály. Následně se

v jednotlivých kanálech vyberou pixely, které jsou vektorizovány do vyhledávací tabulky (angl. Lookup Table, LTU). Ke klasifikaci se použije kaskáda klasifikátorů [35].



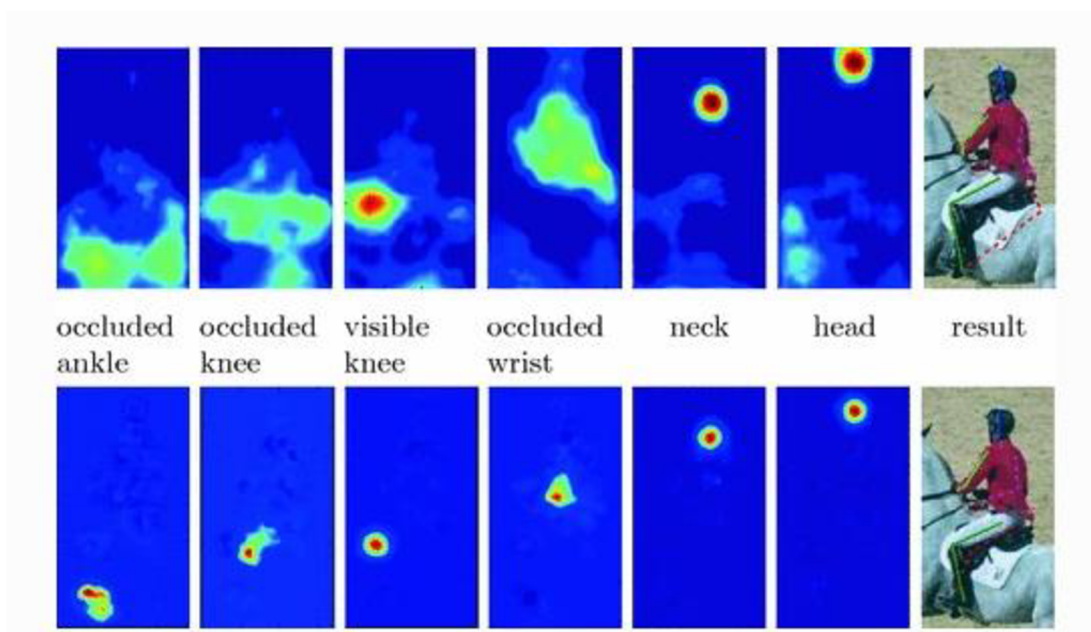
Obrázek 2.13 Znáornění algoritmu metody ACF

2.7.2 Detekce jednotlivých částí lidského těla a vykreslení teplotních map

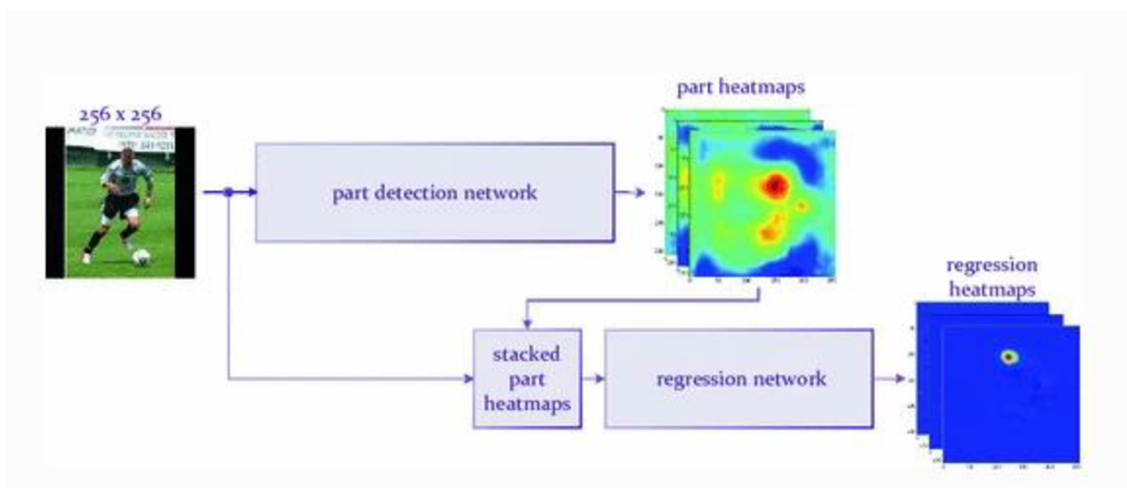
K detekci jednotlivých částí těla je zapotřebí neuronová síť. Princip této detekce je popsán v [29].

Nejdůležitější částí jsou dvě neuronové sítě. Jedna je trénována na rozpoznání jednotlivých částí těla (krk, hlava apod.). Jistota, s jakou si síť „myslí“, že se jedná o požadovanou část těla, je vynesena do tzv. teplotní mapy (viz 2.12). Druhá neuronová síť má za úkol regresi – předpovídání na základě vstupních dat (v tomto případě jsou

vstupní data teplotní mapy a původní obraz). Výstupem celého procesu jsou teplotní mapy. Architektura procesu je na obrázku 2.13[29].



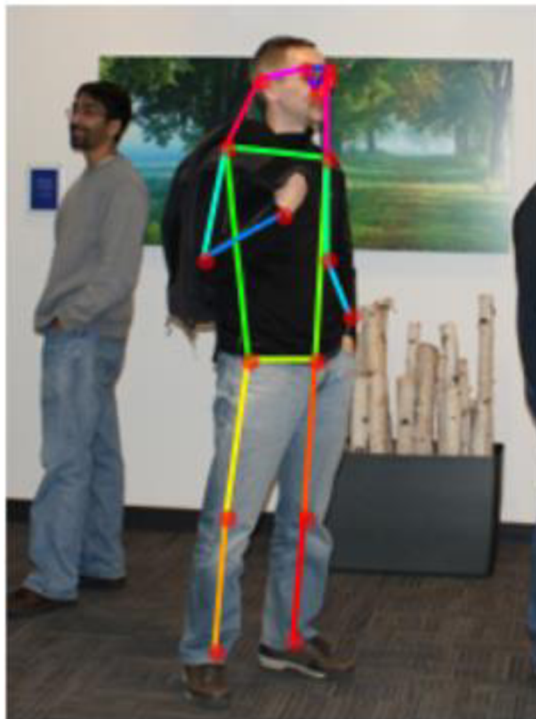
Obrázek 2.14 Teplotní mapy – první řádek je výstup neuronové sítě, která má za úkol rozpoznat jednotlivé části těla. Druhý řádek je výstup druhé neuronové sítě po regresi [29]



Obrázek 2.15 Architektura procesu detekce částí lidského těla.[29]

2.7.3 Vizualizace klíčových bodů

Zvýrazněním bodů z výstupních teplotních map na původním vstupním obrazu se vizualizují určité body těla, které se nazývají *klíčové body* (angl. keypoints). Patříčným spojením těchto bodů vznikne skelet lidského těla.



Obrázek 2.16 Skelet lidského těla [21]

3. VÝVOJOVÉ PROSTŘEDÍ

Kapitola se zabývá použitým vývojovým prostředím a jeho součástmi.

3.1 MATLAB R2020b

MATLAB je software využívaný pro vědeckotechnické výpočty, simulace, implementaci algoritmů, analýzu a prezentaci dat. Využívá stejnojmenného skriptovacího jazyka. Předností tohoto programu je to, že hlavní datovou strukturou jsou matice, pomocí kterých se provádí výpočty. Součástí MATLABu je mimo jiné také Simulink, který se využívá k simulaci dynamických systémů. Simulink nabízí intuitivní grafické rozhraní [13].

Následující podkapitoly krátce shrnují funkce jednotlivých knihoven využitých v práci.

3.1.1 Image Processing Toolbox

Image processing toolbox je knihovna MATLABu určená pro zpracování obrazu. Jsou v ní implementovány různé funkce, např. Cannyho hranový detektor. Umožňuje také segmentaci obrazu, redukci šumu, geometrickou transformaci atd. [14].

3.1.2 Computer Vision Toolbox

Computer vision toolbox je knihovnou počítačového vidění v MATLABu. Poskytuje algoritmy pro detekci objektů. Proto je jednou ze stěžejních knihoven v této práci [15].

3.1.3 Parallel Computing Toolbox

Jedná se o knihovnu, díky které lze využívat více jader procesoru a grafické karty. Hlavním účelem této knihovny je zvýšení výpočetní rychlosti [30].

3.1.4 Deep Learning Toolbox

Aby bylo možné implementovat neuronové sítě, je potřeba nainstalovat Deep Learning Toolbox. Umožňuje použít konvoluční neuronové sítě (CNN) a LSTM (long short-term memory) k regresi či klasifikaci [31].

dopsat [21]

3.1.5 Statistics and Machine Learning Toolbox

Obsahuje funkce na popsání, modelování a analýzu dat. Stěžejní funkcí pro tuto práci je *corr2* [32].

3.1.6 MATLAB Support Package for USB Webcams

Tento podpurný balíček umožňuje používat kamery jako vstupní zařízení. Může se jednat o kameru zabudovanou v notebooku nebo kameru připojenou přes USB port [33].

3.1.7 MATLAB Support Package for IP Cameras

Jediným rozdílem oproti 3.1.6 je fakt, že se nejedná o zabudované kamery v notebooku či připojené přes USB port, ale o IP kameru [34].

3.1.8 Human Pose Estimation with Deep Learning

Human Pose Estimation with Deep Learning je nejdůležitější knihovnou v této práci. Umožňuje vizualizovat klíčové body, vizualizovat teplotní mapy a také detekovat postoj [21].

4. PRAKTICKÁ REALIZACE

Jak již bylo zmíněno v úvodu, práce má za úkol vytvořit aplikaci, dle které se uživatel dozví, zda cvičí korektně. Následující podkapitoly popisují postup pro první cvik, který byl aplikován k dosažení výsledků. Realizace druhého cviku je obdobná, jako u toho prvního. Proto bude popisován pouze první cvik.

4.1 Vytvoření předlohy

Jako předloha se může použít snímek, video či video v reálném čase. Předlohou v práci jsou snímky, které byly vystříženy z videa. Natočením videa (např. s frekvencí 30 snímků/s) vznikne mnoho snímků, ze kterých lze vybrat. Tato skutečnost je výhodná pro testování, který snímek bude ideální předloha.

Nejdříve je nutné nahrát video, ve kterém se bude člověk detekovat. Pro co nejdokonalější zdetekování lidské postavy je potřeba kvalitní video. Kvalitním videem se rozumí video, které bude optimálně kontrastní (černý objekt na bílém pozadí nebo naopak), bez šumu, optimálně nasvětlené a obecně v co nejlepší kvalitě. Většinou jsou ale všechny tyto podmínky nereálné, protože požadují profesionální vybavení (stativ, fotolampy, kvalitní kameru apod.), které není běžně dostupné. Obstojných podmínek lze dosáhnout ale i tak, že člověk (např. v černém oblečení) si stoupne před bílou zeď.

Čím kvalitnější video bude, tím bude větší jeho velikost a tím delší bude čas nutný ke zpracování. Z těchto důvodů je tedy praktičtější využít kameru zabudovanou v notebooku nebo připojenou k počítači.

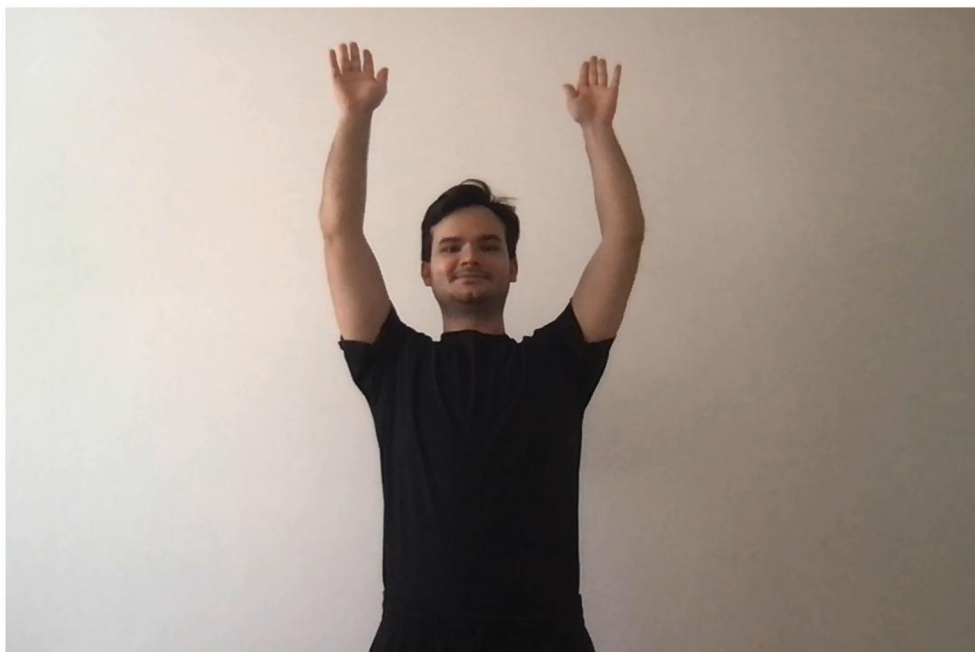
Video, které se následně využívá, bylo natočeno 720p kamerou zabudovanou v MacBook Air 2019 (Retina, 13 palců). Toto video dále slouží jako předloha pro určení základní a koncové polohy.

4.2 Zpracování předlohy

Aby bylo možné video zpracovat, musí se nahrát do MATLABu a do stejné složky, ve které je program. Nyní je potřeba video tzv. „přečíst“. Uživateli se zobrazí okno, které vyžaduje zadání čísla cviku, který chce provádět. Podle toho se přečte buďto video s prvním či s druhým cvikem. Přečtené video se přiřadí do proměnné *obj*. Protože je požadováno vystříhnout z videa snímky, musí se zjistit, z kolika snímků se video skládá. Počet snímků bude $snimky=obj.NumFrames$. Následně se zjistí, který snímek je základní poloha (postava stojí v klidu, nehybně) a koncová poloha (postava je v poslední či předposlední fázi cviku). Nakonec jsou snímky uloženy do proměnných k dalšímu zpracování.



Obrázek 4.1 Základní poloha



Obrázek 4.2 Koncová poloha

Postava na obou obrázcích stojí rovně. Bude se uvažovat, že postava na obou obrázcích je vertikálně symetrická a může být použita jako předloha. Předlohy pro druhý cvik viz příloha A.2.

Obrázek 4.1 je potřeba převést do stupňů šedi, protože v dalším kroku se v tomto obrázku budou detekovat hrany (funkce *edge* vyžaduje obrázky s jedním barevným kanálem, tj. černobílé).

Zakladni poloha ve stupnich sedi



Obrázek 4.3 Převedený obrázek do stupňů šedi

Programove upravena zakladni poloha



Obrázek 4.4 Převedený obrázek do stupňů šedi po programové úpravě

Lze si povšimnout, že na obrázku 4.4 je černá sytější a levá strana obrázku je světlejší. Byla zde totiž provedena úprava příkazem *imadjust*. Tento krok může být vynechán, ale

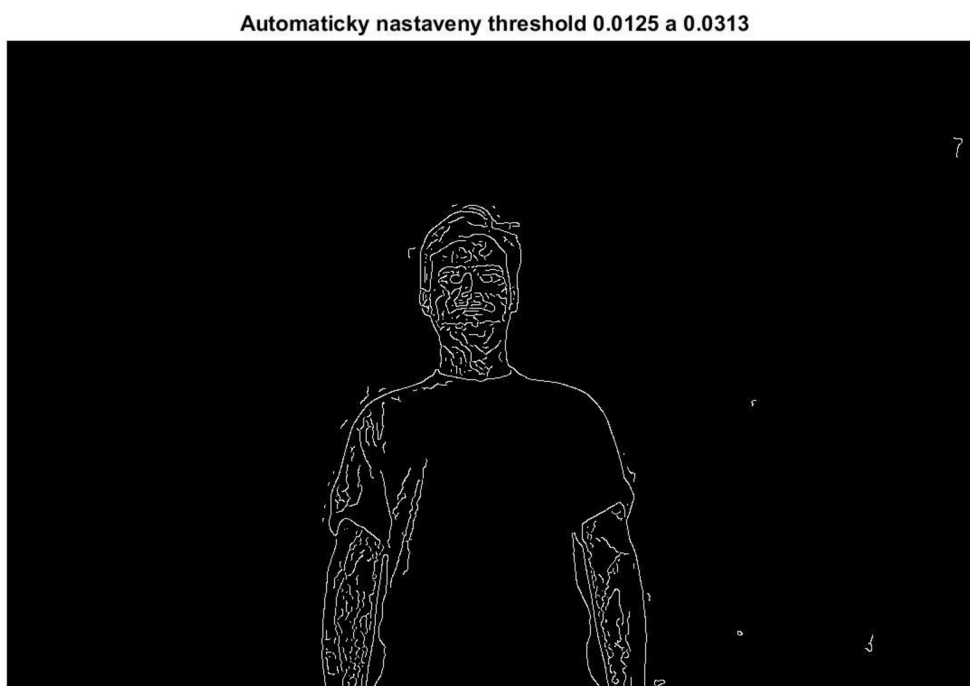
čím více kontrastní bude pozadí od popředí, tím lépe se budou detekovat hrany (je zajištěn lepší přechod z tmavé barvy na světlou a naopak).

Obrázky 4.1 a 4.2 budou využity v dalších krocích, jako je určování správné vzdálenosti postavy od kamery a korelace klíčových bodů.

4.2.1 Detekce hran metodou Canny a nastavení parametrů

Detekce hran se využije při určování korektní vzdálenosti postavy v základní poloze od kamery.

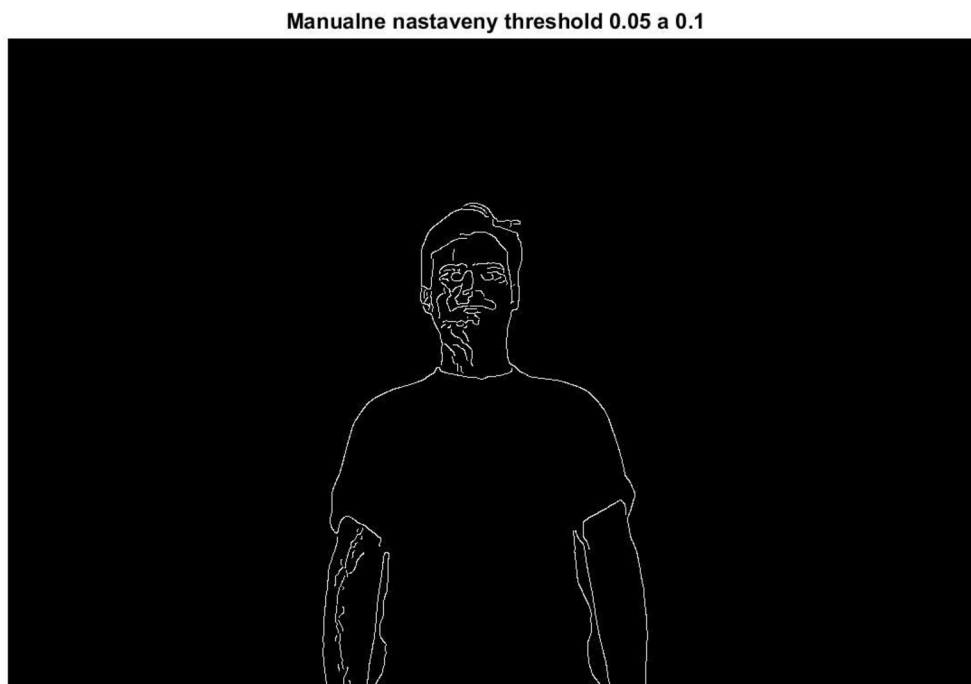
Jako první se zavolá funkce *edge* s povinnými parametry.



Obrázek 4.5 Detekované hrany základní polohy, automaticky nastavený threshold

Z obrázku 4.5 jde vidět, že automaticky nastavený threshold není vhodný, protože detekuje určité hrany navíc (vpravo dole a okolo postavy). Proto je potřeba nalézt vhodnější prahové hodnoty.

Jakmile nebyly detekovány hrany vně postavy a obrys postavy byl jasný a ničím nepřerušovaný, mohl být threshold považován za optimální.



Obrázek 4.6 Zdetekované hrany, manuálně nastavený threshold

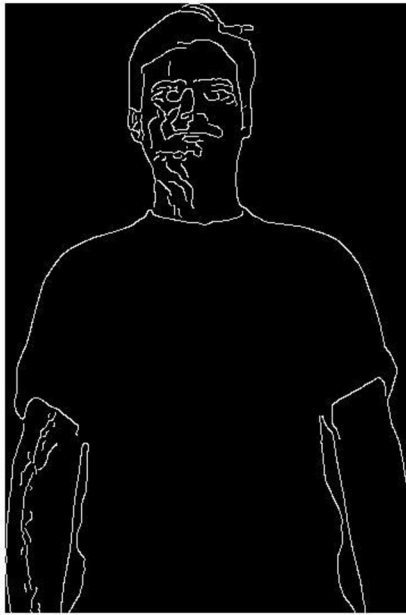
Pro účely této práce je stěžejní, aby byly detekované obrysové hrany. Proto je výše zmíněný nastavený threshold dostačující.

Aby bylo možné zjistit, jakou má postava šířku, je nutné nejdříve najít bílé pixely nejvíce vlevo a nejvíce vpravo a postavu oříznout.

4.2.2 Geometrická transformace

Funkcí *find* s návratovými hodnotami [*radky*, *sloupce*] se zjistí všechny souřadnice pixelů. Pro nalezení bílého pixelu, který je nejvíce vpravo, bylo využito funkce *min(sloupce)*, pro bílý pixel nejvíce vpravo *max(sloupce)*. Stejný postup se aplikuje pro řádky. V této fázi je vhodné postavu oříznout.

Oriznuta postava



Obrázek 4.7 Ořezaná postava

Šířka postavy v pixel potom bude $\max(\text{sloupce}) - \min(\text{sloupce})$. Postava má šířku 360 pixelů, což odpovídá 144 cm. Rovnice pro převod šířky postavy na vzdálenost od kamery je následující.

$$y = -0,5x + 324 \quad (4.1)$$

kde,

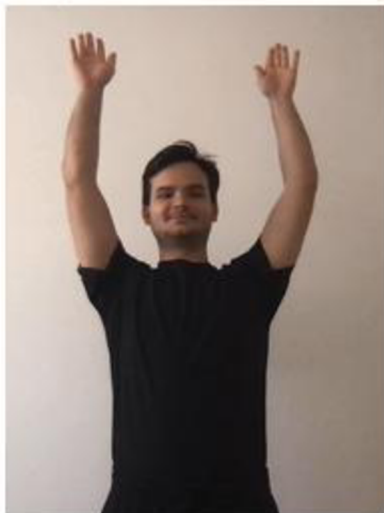
y – vzdálenost postavy od kamery [cm]

x – šířka postavy v pixelech

4.2.3 Detekce koncové polohy

Aby bylo možné ověřit, zda osoba cvičí korektně, je zapotřebí definovat správnou koncovou polohu. V tomto případě to bude obrázek 4.2, ve kterém se detekuje poloha a určí se souřadnice klíčových bodů. V první řadě je nutné změnit velikost obrázku na 256x192x3. Důvodem je skutečnost, že neuronová síť je natrénována na tuto velikost obrázků. V programu změnu velikosti zajišťuje příkaz *imresize*.

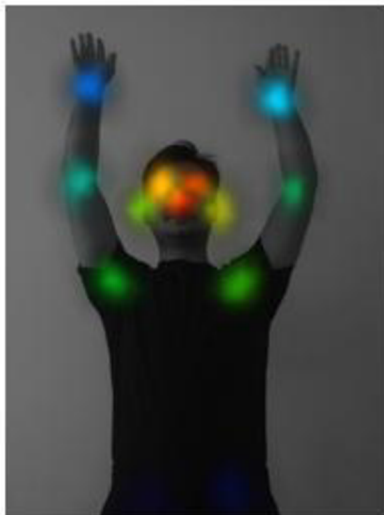
Zmenseny obrazek na 256x192x3



Obrázek 4.8 Koncová poloha – zmenšený obrázek

V tuto chvíli lze detekovat jednotlivé části těla. Vykreslit teplotní mapy není nutné, obrázek je zde spíše pro pochopení vnitřního fungování algoritmu.

Vizualizovane teplotni mapy



Obrázek 4.9 Vizualizované teplotní mapy

Nyní se na původní obrázek (4.8) vykreslí tyto body a následně se propojí.

Detekovaný postoj



Obrázek 4.10 Detekovaný postoj

Následně jsou veškeré předlohy připravené a může se začít zpracovávat video, které se bude s předlohou srovnávat.

4.3 Tvorba videa

Aby bylo dosaženo co největší efektivity, bude video zaznamenáváno v reálném čase a to tak, že z web kamery budou vyčítány jednotlivé snímky, které se ve *while* smyčce budou postupně aktualizovat.

Hlavní část programu běží ve *while* smyčce, kde se postupně přepínají stavy, které jsou reprezentovány proměnnou *x*.

Podmínkou pro provedení tohoto specifického cviku (viz obrázek 4.2) je, aby kamera zabírala postavu od pasu nahoru.

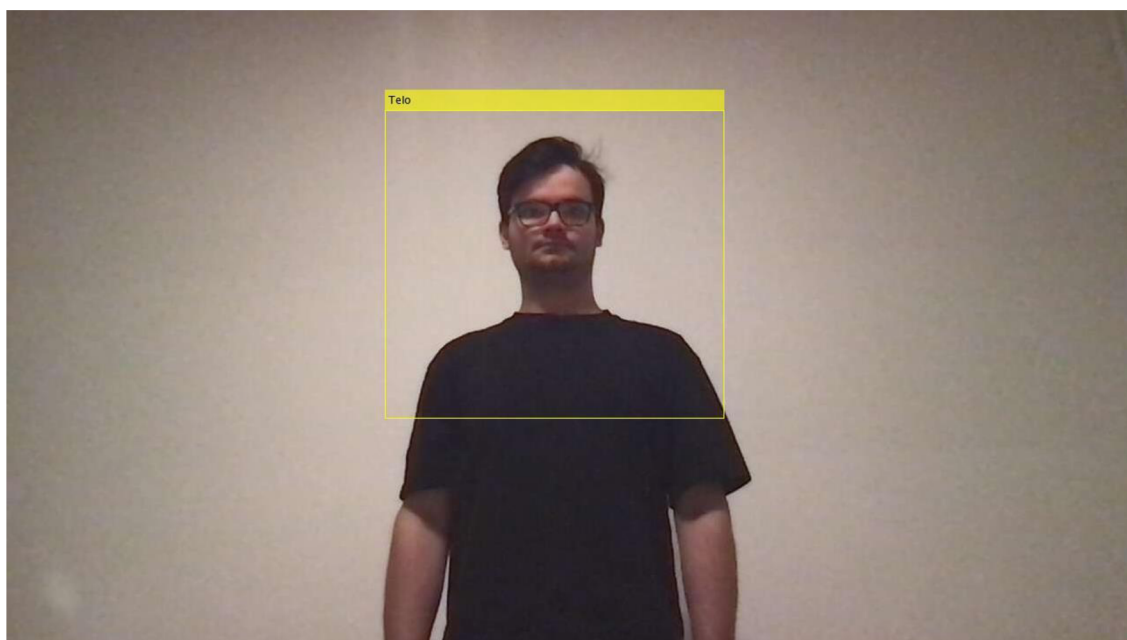
4.3.1 Detekce člověka v záběru

V první řadě je nutné zjistit, zda je člověk v záběru a zda tedy může cvik začít. První otestovanou metodou byl *vision.PeopleDetector* [36]. Tato metoda funguje na základě HOG (viz 2.7). Po vstoupení člověka do záběru nebyla metoda schopna detekce. I po upravení parametrů se detekování nezdařilo. Jedním z důvodů nefunkčnosti může být chybná implementace od tvůrce.

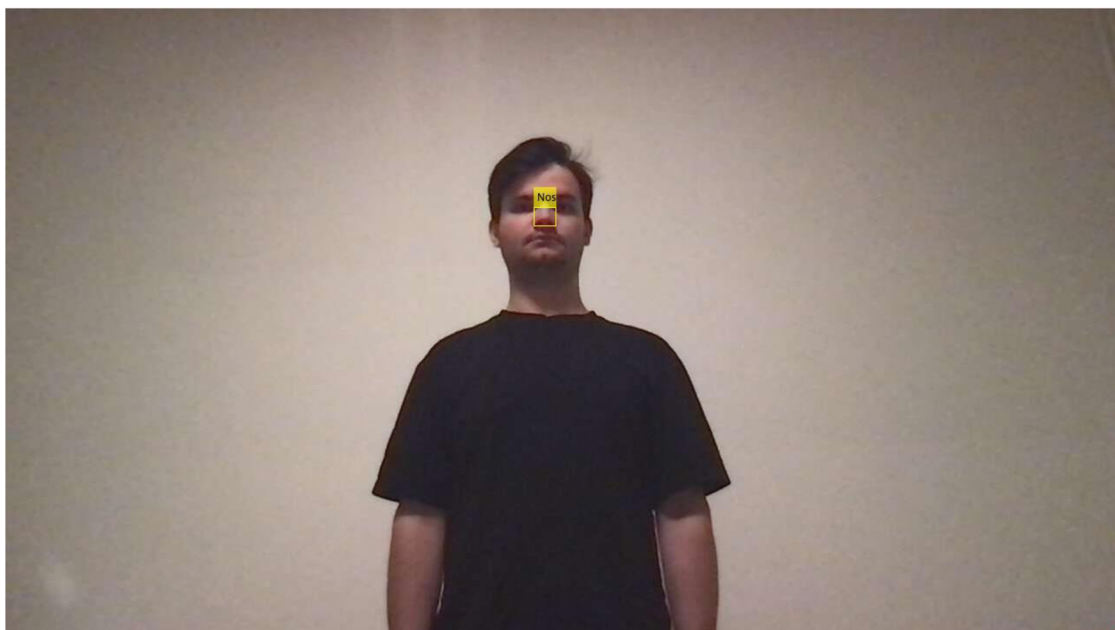
Další otestovanou metodou byl *peopleDetectorACF* [37]. Metoda založená na ACF (viz obrázek 2.12) taktéž nebyla schopna detekce i přes změnu parametrů. Opět je možné, že je chyba na straně tvůrce.

Obě tyto metody mají v popisku uvedeno „detect upright people“, což může být vyloženo mnoho způsoby. Pojem „upright“ znamená vzpřímený. V příkladech použití těchto metod jsou vždy detekované osoby bokem ke kameře. Z definice slova „upright“ ale není jisté, zda se jedná pouze o vzpřímeného člověka stojícího bokem ke kameře nebo obecně o vzpřímeného člověka.

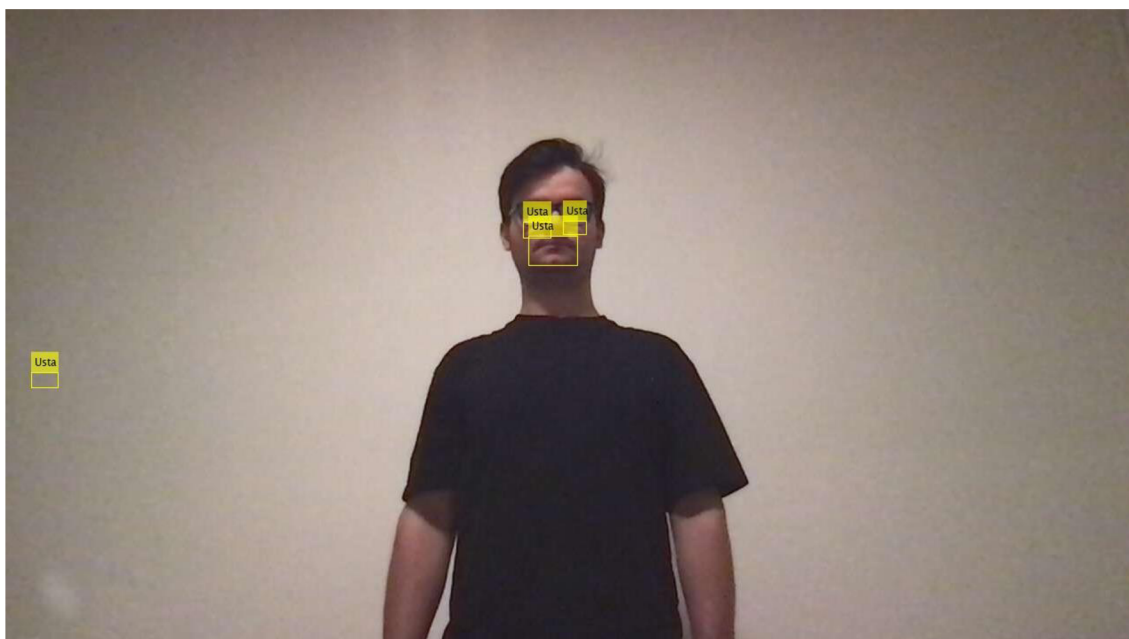
Poslední otestovanou metodou byl *vision.CascadeObjectDetector* využívající Viola-Jones algoritmus (viz 2.6). Jedním z parametrů, které je potřeba zvolit, je část, kterou je třeba detekovat. Může být detekována horní část těla, nos, ústa, oči nebo obličej. V této práci bylo otestováno detekování horní části těla, nosu, úst, obličej, velkého páru očí a malého páru očí.



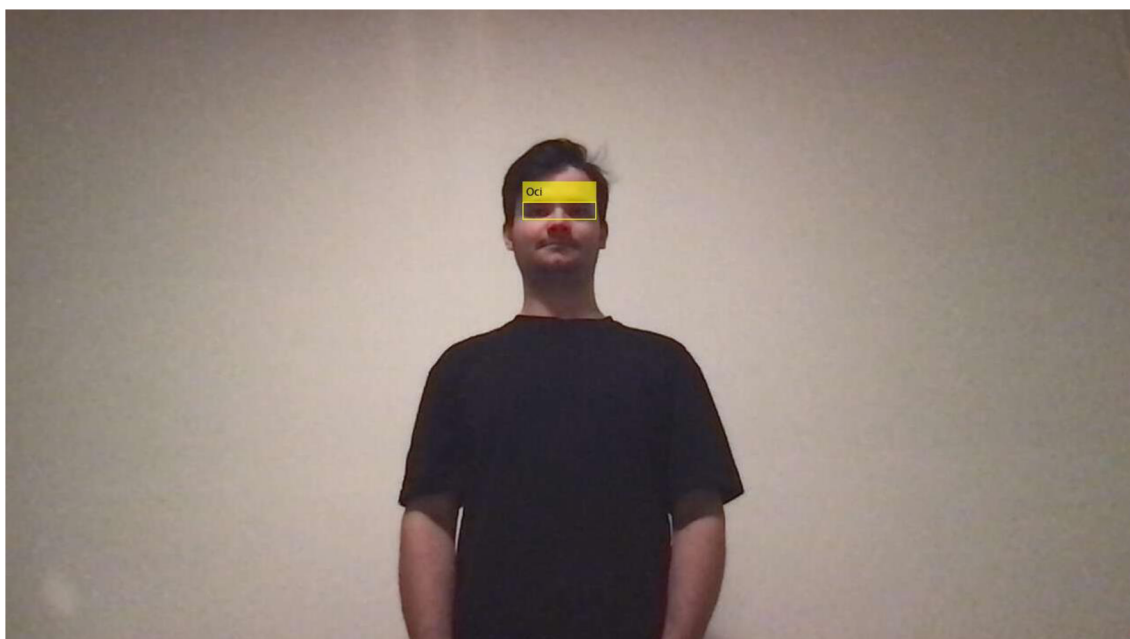
Obrázek 4.11 Detekovaná horní část těla



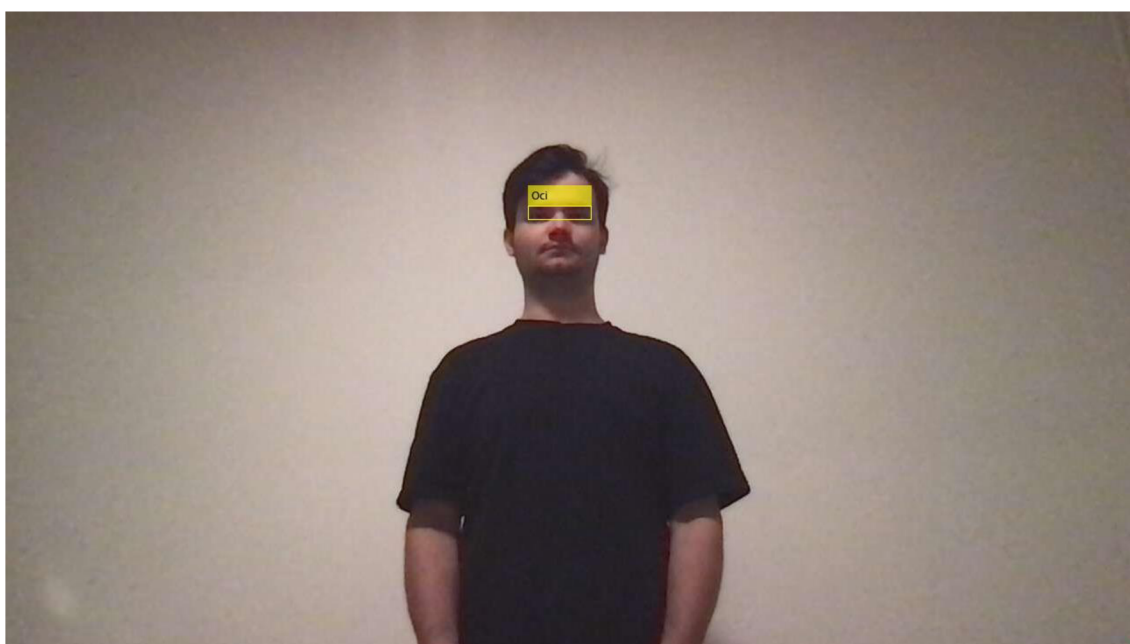
Obrázek 4.12 Detekovaný nos



Obrázek 4.13 Detekovaná ústa

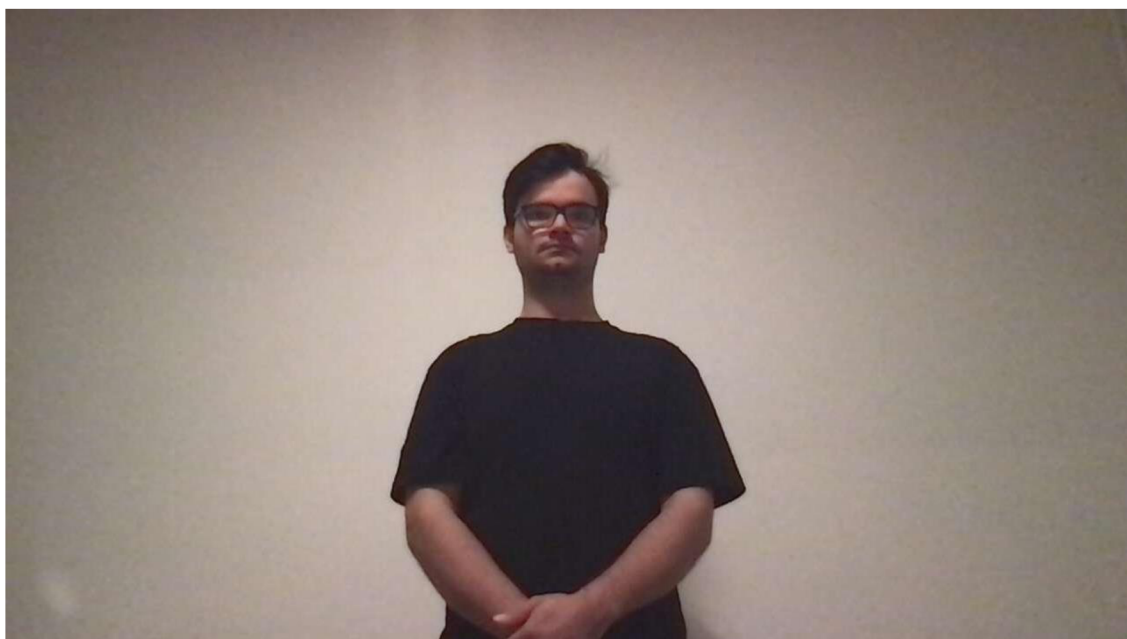


Obrázek 4.14 Detekované oči (velký pár)

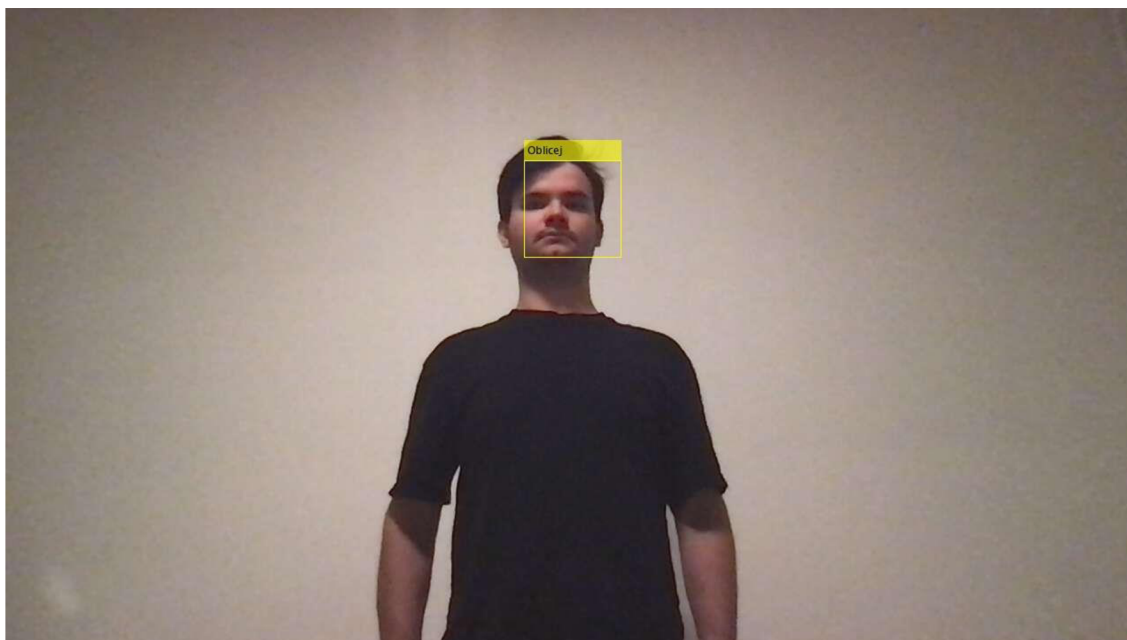


Obrázek 4.15 Detekované oči (malý pár)

Z obrázků 4.14 a 4.15 lze vidět, že detekce proběhla korektně. Problém ale nastává ve chvíli, kdy si osoba nasadí brýle. V tomto případě oči nebudou detekovány.



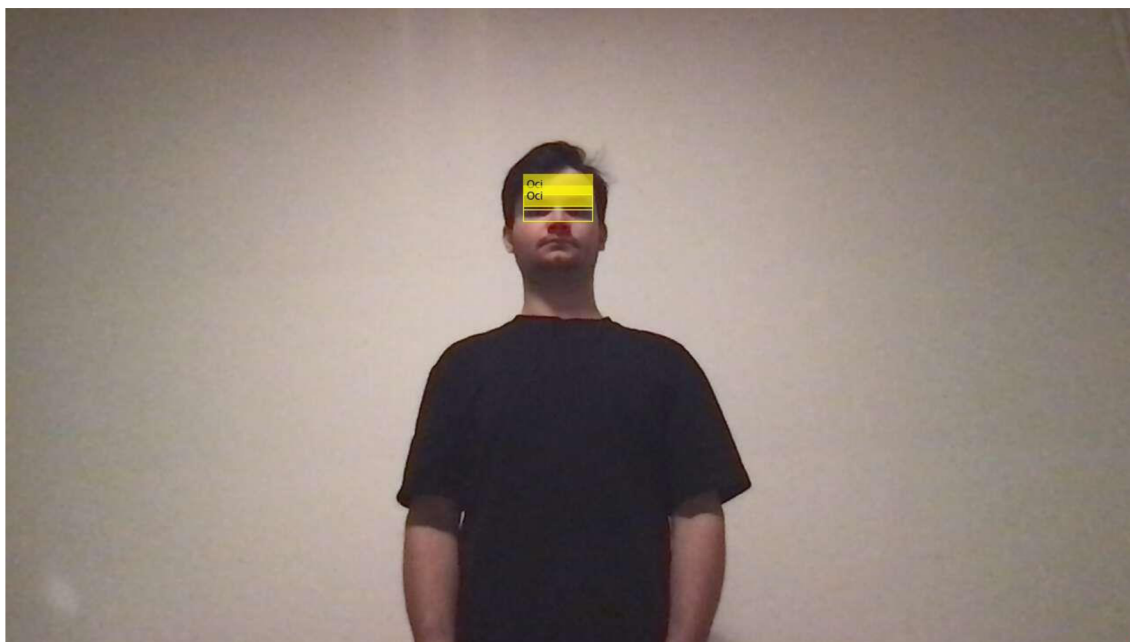
Obrázek 4.16 Nedetekování očí po nasazení brýlí



Obrázek 4.17 Detekovaný obličej

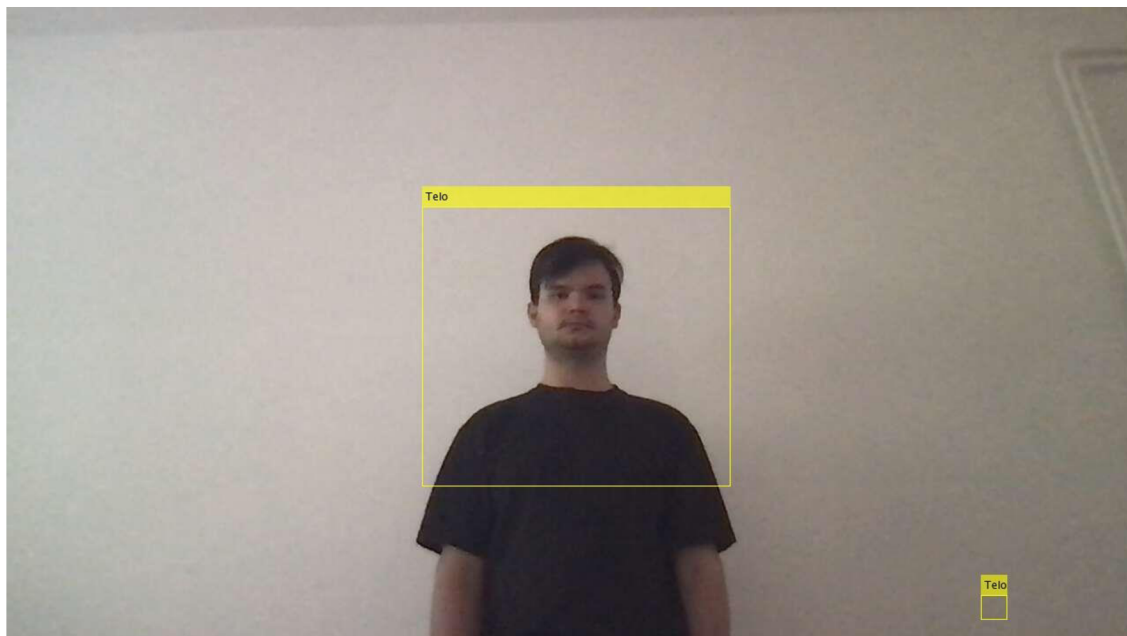
Všechny možnosti, až na detekci úst, korektně detekují dané části, podle kterých lze určit, zda je člověk v záběru.

Po důsledném testování bylo zjištěno, že v určitých případech proběhly 2 typy falešné detekce. První typ se vyskytl u člověka, který již byl v záběru. Kromě samotného člověka byla detekována např. část ramene, břicha či tváře (viz obrázek 4.13). Tato skutečnost může být zapříčiněna neoptimálním světlem a deformací oblečení. Protože je metoda založena na Haarových příznamech, může být patřičně nasvícená zdeformovaná část oblečení falešně klasifikována jako některá z horních částí těla.



Obrázek 4.18 Falešná detekce – první typ

Druhý typ falešné detekce je zapříčiněn šumem. Často se tento typ falešné detekce vyskytuje ve chvíli, kdy je zabíráno pouze pozadí (není podmínkou, tento jev lze pozorovat i s osobou v záběru).



Obrázek 4.19 Falešná detekce (vpravo dole) – druhý typ

Tomuto jevu lze předejít odečtením pozadí. Problémem je, že po odečtení pozadí se nedaří detekovat zvolené části horní poloviny těla. I přes občasnou falešnou detekci lze považovat metody za robustní a bez potřeby odečíst pozadí.

Nakonec byla vybrána detekce horní části těla. Prvním důvodem výběru byl fakt, že pro provedení cviku (viz obrázek 4.2) je nutno mít nastavenou kameru tak, aby byla vidět horní část těla. Nemůže tak vzniknout případ, že se spustí cvik i po detekování pouhého obličeje. Druhým důvodem byla nízká četnost falešné detekce obou typů. I přes to, že detekce horní části těla zabrala nejvíce času oproti ostatním metodám, tak nebyl rozdíl až tak markantní.

Srovnání průměrných časů potřebných k detekci lze nalézt v tabulce viz příloha A.1.

4.3.2 Určení vzdálenosti postavy od kamery

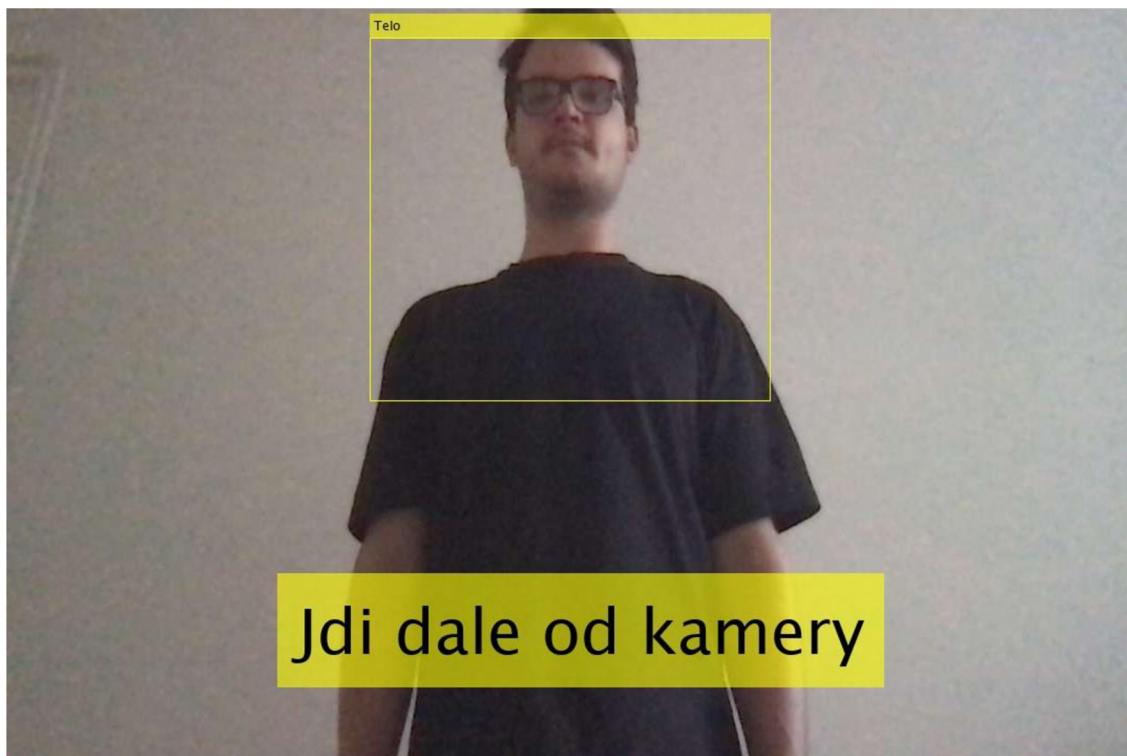
Aby bylo možné porovnat předlohu a aktuální snímek postavy, musí postava stát v určité vzdálenosti od kamery. Pokud se začnou snímky v reálném čase zpracovávat obdobně jako u Obrázku 4.7, lze porovnat šířku předlohy a aktuální šířku člověka ve videu. Vznikne tedy podmínka, aby člověk ve videu byl v takové vzdálenosti od kamery, aby byla šířka srovnatelná s předlohou, ideálně stejná. Ideálního stavu však nelze docílit, protože člověk se minimálně hýbe (mění tak svoji šířku) i když stojí „nehybně“. Proto je potřeba definovat meze. Pro tuto práci byly definovány meze pro první způsob určení odchylky jako max +1 a min -10 pixelů, což odpovídá +0.5 a -5 cm. Pro druhý způsob

jsou meze ± 10 pixelů. To odpovídá ± 5 cm. Bude-li šířka postavy ve stanovených mezích, bude považována za korektní.

Pokud je uživatel daleko od kamery, program to zaznamená a zobrazí hlášku „Jdi blize ke kamere“. Naopak, pokud je uživatel blízko kamery, zobrazí se hláška „Jdi dale od kamery“. V případě korektní vzdálenosti lze upozorovat „Spravna vzdalenost“. Jakmile je uživatel správně vzdálený od kamery po určitý časový úsek, přepne se program do dalšího stavu a začíná cvik.



Obrázek 4.20 Člověk je daleko od kamery



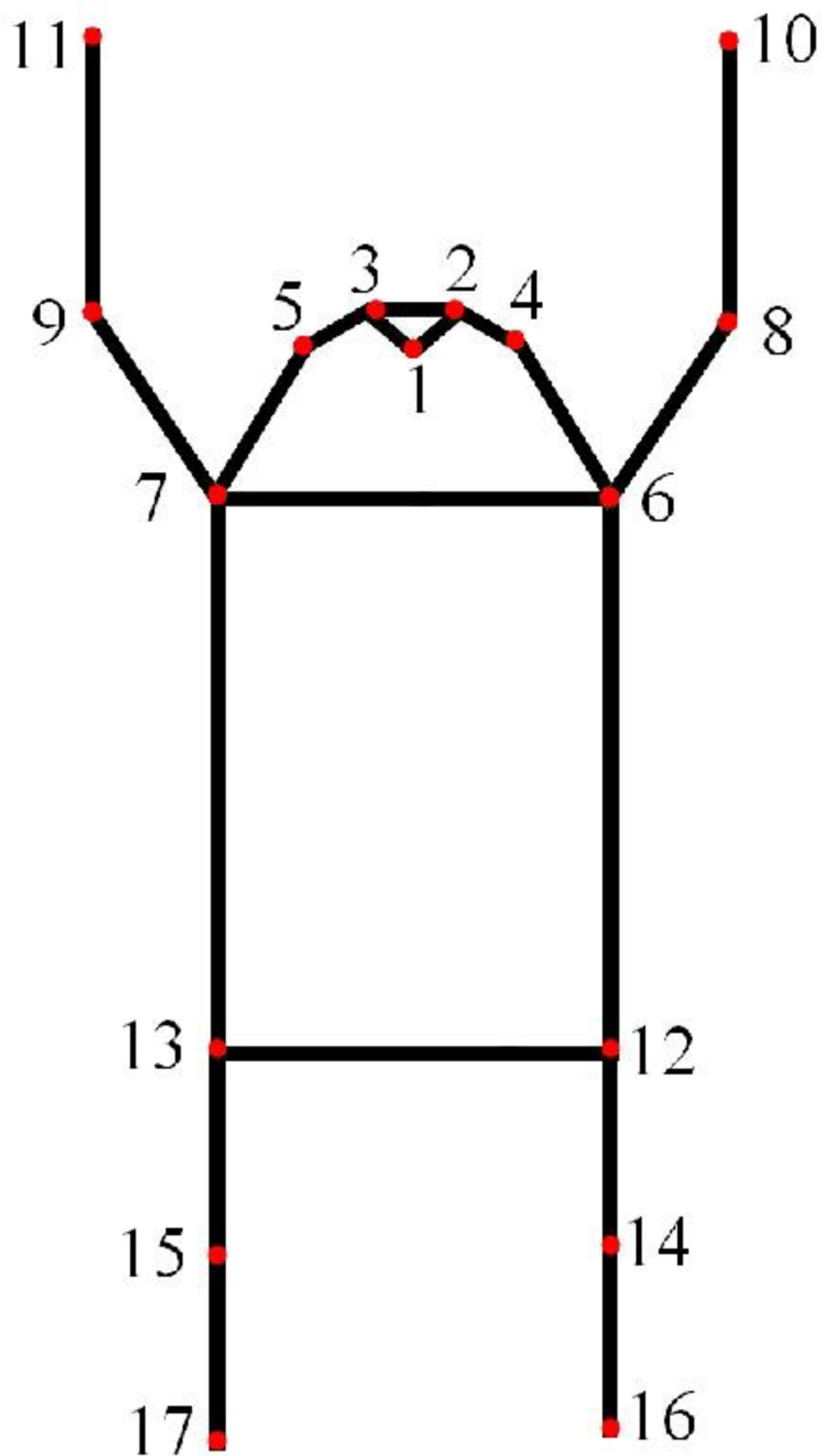
Obrázek 4.21 Člověk je blízko kamery



Obrázek 4.22 Člověk je ve správné vzdálenosti

4.3.3 Detekce postoje

Aplikuje se stejný postup jako v 4.2.3. Pro tento specifický cvik (viz Obrázek 4.2) bude potřeba pouze 6 klíčových bodů (6 až 11).



Obrázek 4.23 Indexování klíčových bodů

Aby bylo možné zobrazit klíčové body i na obraze větším, než je rozměr 256x192, je potřeba provést převod. Ten bude realizován vynásobením původních klíčových bodů z obrazu 256x192 převodními konstantami. Pro výšku a šířku budou převodní konstanty rovny:

$$kh = kph \cdot \frac{fh}{dih} + dih \quad (4.2)$$

kde,

kh – převedená výška klíčového bodu,

kph – výška původního klíčového bodu,

fh – výška snímku,

dih – výška výstupu neuronové sítě (256)

$$kw = kpw \cdot \frac{fw}{2 \cdot diw} \quad (4.3)$$

kde,

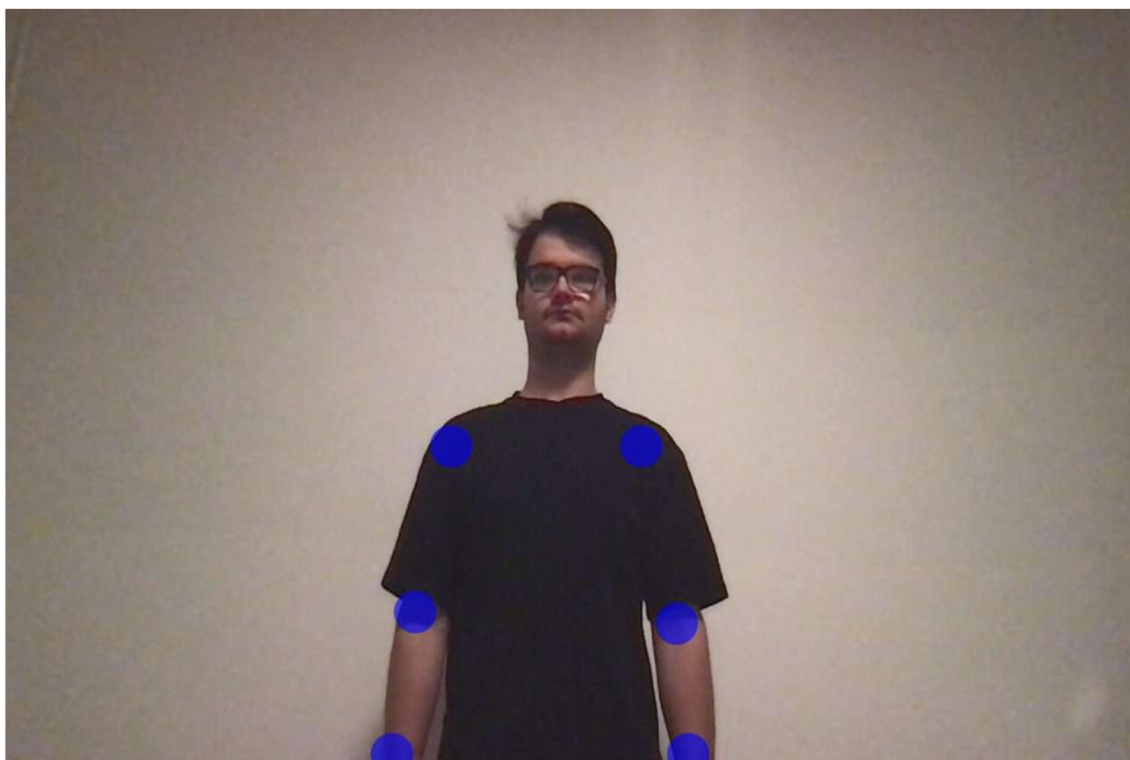
kw – převedená šířka klíčového bodu,

kpw – šířka původního klíčového bodu,

fw – šířka snímku,

$d iw$ – šířka výstupu neuronové sítě (192)

Ve *for* cyklu se pro každý potřebný klíčový bod budou vykreslovat kruhy s průměrem 20 pixelů.



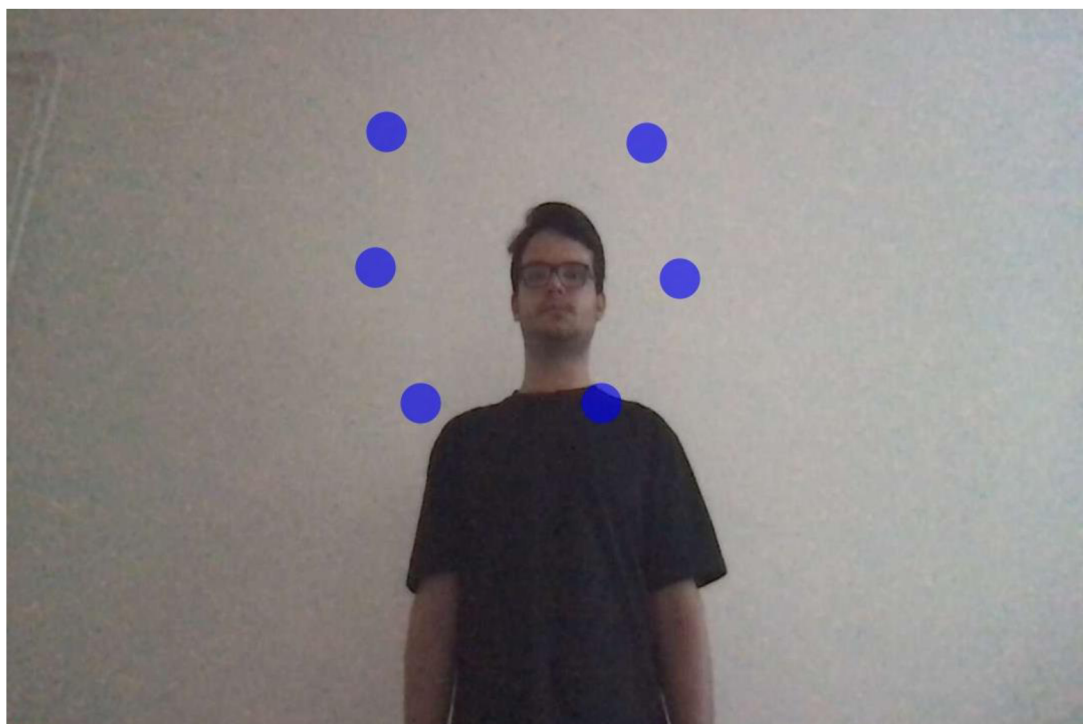
Obrázek 4.24 Převedené klíčové body na postavě

4.3.4 Určení odchylky

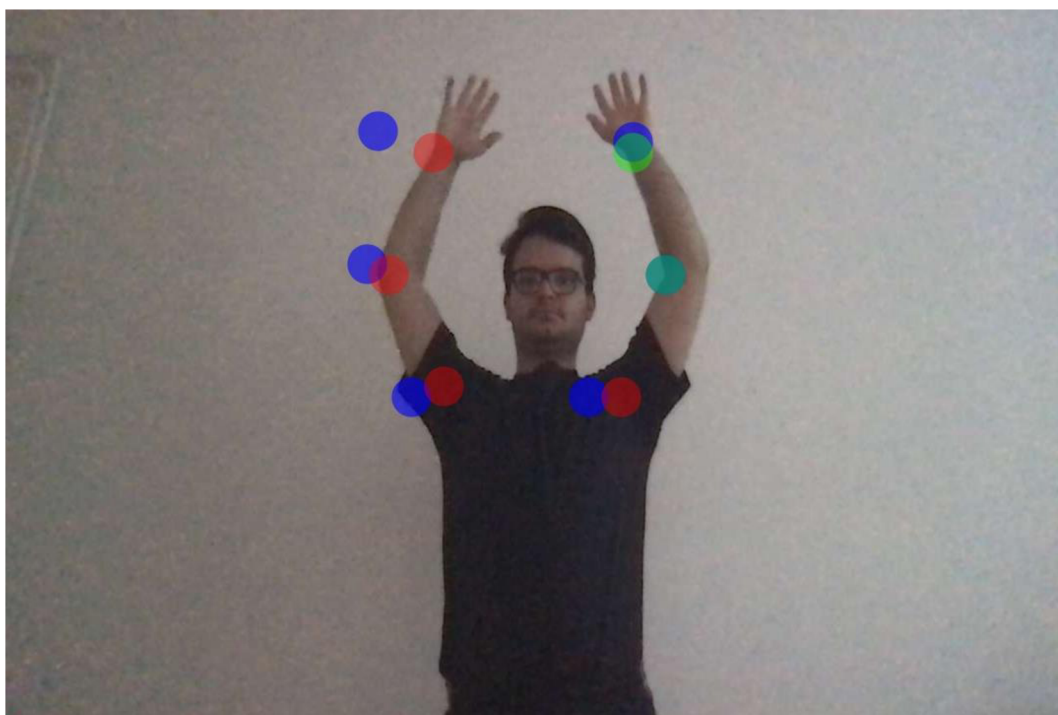
Byly otestovány dva způsoby určení odchylek. Oba tyto způsoby vycházely ze stejného základu. Klíčové body cvičícího člověka se korelační funkcí *corr2* [38] porovnaly s předlohou (viz obrázek 4.2). Výstupem byl korelační koeficient, dle kterého bylo možné poznat, zda člověk pouze stojí před kamerou, nebo zda cvičí (vzpažení). Z pouhého koeficientu uživatel nepozná, jestli je někde vychýlený (např. nakloněný na stranu).

První implementovaný způsob určení odchylky spočíval ve vizualizaci klíčových bodů předlohy. Tyto body se zobrazily červenou barvou a uživatel měl za úkol se na tyto klíčové body dostat. Jakmile klíčovým bodem dostal do určitých mezí (± 5 pixelů od

klíčového bodu předlohy), bod se zbarvil do zelena. Ve chvíli, kdy byly všechny body zelené, cvik byl považován za správně provedený a program byl ukončen.



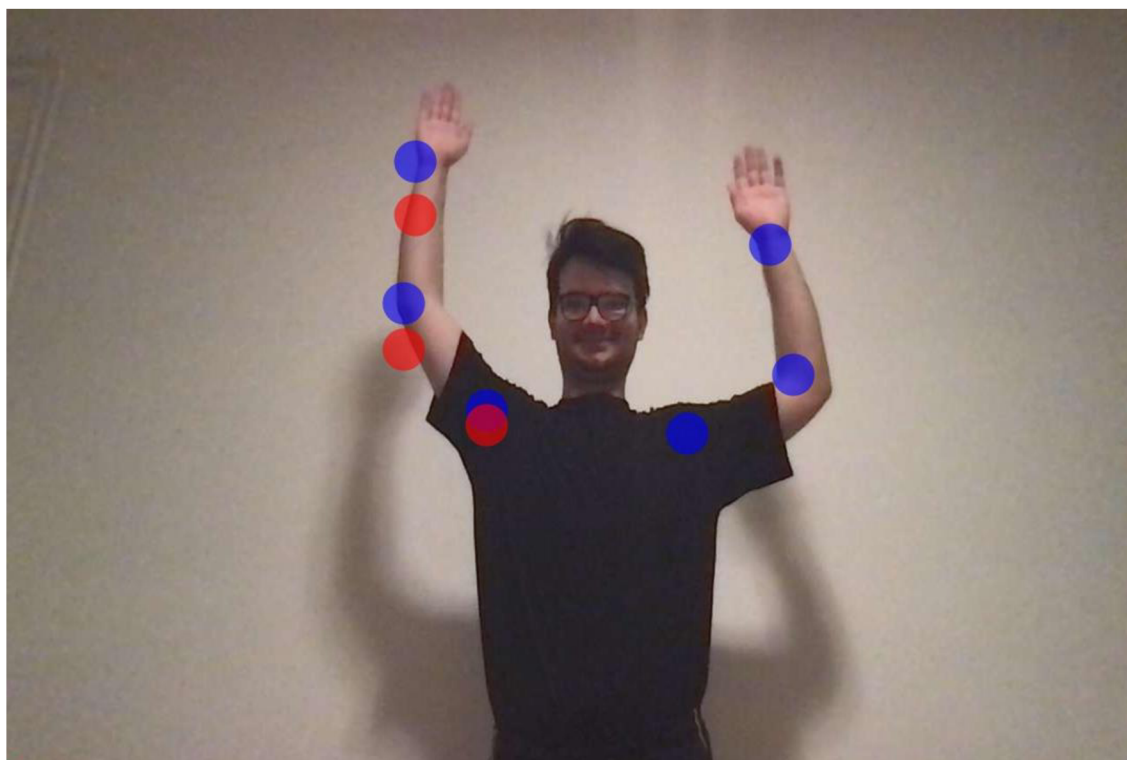
Obrázek 4.25 Vykreslené klíčové body předlohy (první způsob)



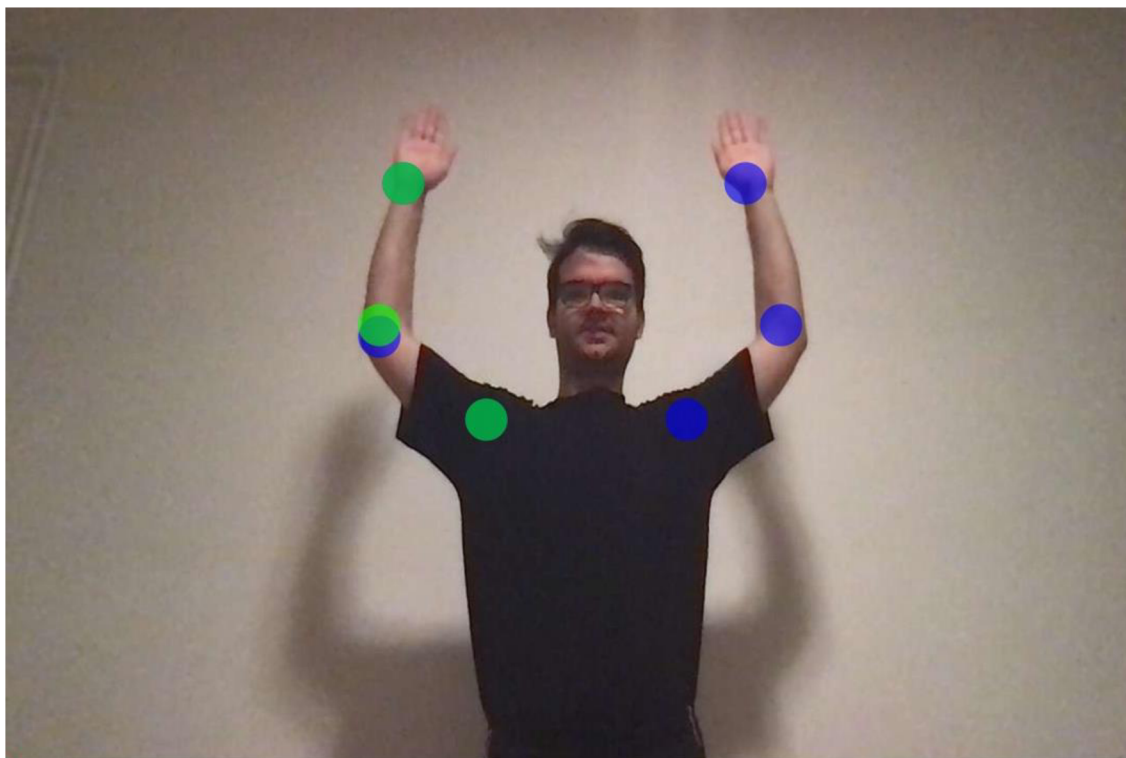
Obrázek 4.26 Zbarvení klíčového bodu do zelena po dosažení patřičných mezí (první způsob)

Druhý způsob se odvíjel od pozice jedné ruky cvičícího člověka. Pokud má být cvik proveden dobře, musí být klíčové body obou rukou ve stejné výšce. V tomto případě se tedy jedna ruka považovala za správnou a druhou se přizpůsobovalo. Jakmile člověk vzpažil, byly mu u jedné z rukou vykresleny ideální klíčové body vzhledem k ruce druhé.

Tyto body měly modrou barvu a uživatel měl opět za úkol se na tyto body dostat. Po dosažení bodu se bod zbarvil do zelena.



Obrázek 4.27 Vykreslené ideální body (červeně) vzhledem k druhé ruce (druhý způsob)



Obrázek 4.28 Zbarvení bodů do zelena po dosažení mezí (druhý způsob)

U obou metod po korektním cvičení program vypíše „Cvik končí“ a ukončí činnost.

4.3.5 Archivace průběhů

Program umožňuje archivovat průběhy, což je uživatelsky přívětivé a umožňuje se tak zpětná vazba uživateli (uživatel může zhlédnout video, kde uvidí eventuální nepřesnosti). Po zavření okna hlavního okna, ve kterém je zobrazován průběh, se objeví dialogové okno, ve kterém je výběr, zda si uživatel přeje video archivovat či nikoli. Pokud dosavadní průběh uložit nechce, program se ukončí. Pokud průběh uložit chce, bude vyžadováno zadání názvu. Je zde i ošetření zadání stejného jména, které se nachází v domovském adresáři (složka, kde se nachází program).

5. ZÁVĚR

Práce měla za cíl vytvořit aplikaci, která uživatele upozorní, zdali cvičí nebo necvičí korektně. Z praktických důvodů byla práce zasazena do domácího prostředí, kde i běžný uživatel dokáže sestrojít obstojnou scénu (bílé pozadí).

V první části byly vysvětleny základy teorie počítačového vidění, jako je zpracování a předzpracování obrazu. Z metod detekce hran byl zde popsán pouze Cannyho hranový detektor, který je také v práci dále využitý. Z metod detekce člověka byl popsán algoritmus Viola-Jones. Poslední podkapitola teoretické části byla věnována detekci postoje.

V druhé části bylo popsáno vývojové prostředí a různé typy knihoven využívaných v práci.

Poslední část práce se zabývá praktickou realizací. Nejdříve byla popsána tvorba předlohy, která zahrnovala programové úpravy jasu, geometrickou transformaci a detekci postoje. Následovala tvorba videa, ve kterém uživatel prováděl cvik. Aby byl program co nejefektivnější, video bylo nahráváno a zpracováno v reálném čase. Pro zpracování videa bylo otestováno několik metod, přičemž detekce horní části těla (ramena + hlava) se ukázala jako nejlepší. Důvod výběru byl popsán v podkapitole 4.3.1. Další krok spočíval v určení vzdálenosti postavy od kamery. Postup byl obdobný tomu, co je uvedeno v podkapitole 4.2.1. Jakmile byla vzdálenost optimální, mohl se začít detekovat postoj. Byly tak vypočítány klíčové body, které sloužily k porovnání s klíčovými body předlohy. Aby bylo možné určit míru podobnosti, byly funkcí *corr2* porovnány klíčové body. Nebylo vhodné porovnávat celé snímky, protože se do korelační funkce zaznamenával také například i vzhled či oblečení. Ve chvíli, kdy osoba vzpažila, začalo určování odchylky. Byly otestovány dvě metody určení odchylky. První metoda byla založena na vykreslení klíčových bodů předlohy. Uživatel měl za úkol se na tyto klíčové body (svými klíčovými body) dostat. V případě úspěchu se body intuitivně zbarvily do zelena. Druhá metoda se odvíjela od pozice jedné ruky. Pro úspěšné zvládnutí cviku se musely shodovat výšky, ve kterých měl uživatel klíčové body na rukou. Každá z těchto otestovaných metod má své klady i zápory. U první metody nejsou klíčové body předlohy ideální, takže může nastat případ, kdy je obtížné se na klíčové body dostat. Tento problém lze vyřešit nahrazením předlohy za anatomicky ideální postavu. Jedním z významných kladů je to, že pro tuto metodu nemusí být cvik symetrický podle vertikální osy. Druhá metoda se může potýkat s problémem, kdy ruka, která je považována za správnou, bude problematickou částí uživatele (např. pokud by uživatel byl po artroskopii ramene). Řešením tohoto problému by bylo implementování principiálně stejné metody pro druhou ruku. Následně by bylo na uživateli, zda by zvolil pravou či levou ruku jako správnou. Tato metoda uvažuje pouze symetrický cvik.

Celý program je koncipován tak, aby bylo možné vytvořit předlohy, jak bylo popsáno v kapitole 4.2, tyto předlohy programově načíst a následně cvičit (bez dalších úprav).

Znamená to, že pro libovolný symetrický cvik, ve kterém hrají hlavní roli ruce, je program funkční. Proto byl natočen i druhý cvik, který je možno provádět. Výsledné průběhy obou cviků jsou v příloze A.1

Problematika práce byla prakticky řešena v MATLABu. I když tento program nabízí prostředky k řešení problematiky, není však ideálním nástrojem. Některé operace, jako je nahrávání videa v reálném čase, významně zpomalují chod programu a podstatně tak záleží na výkonu počítače uživatele. Například u archivace snímků záleží na rychlosti, jakou jsou snímky aktualizovány. Vygenerované archivované video má nastavenou snímkovou frekvenci 30 snímků za sekundu, což odpovídá běžným videím. Pokud ale video v reálném čase má méně než 30 snímků (u slabších počítačů), vygenerované video bude zrychlené. Vznikne tak požadavek na uživatele, aby si video zpomalil. Množství videopřehrávačů tuto skutečnost umožňuje, nejedná se tak o neřešitelný problém. Řešením těchto problému by byla změna vývojového prostředí a tím i programovacího jazyku. C, C++ nebo i Python by byli vhodnějšími kandidáty na praktickou realizaci. Obzvláště s volně dostupnou knihovnou OpenCV.

Závěrem lze konstatovat, že cíle práce byly splněny s minimálními požadavky na uživatele.

LITERATURA

- [1] History of the camera. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-10-27]. Dostupné z: https://en.wikipedia.org/w/index.php?title=History_of_the_camera&oldid=1048526986
- [2] *Autopilot a Schopnost plně autonomního řízení* [online]. [cit. 2021-10-28]. Dostupné z: https://www.tesla.com/cs_CZ/support/autopilot-and-full-self-driving-capability
- [3] Pojem detekce. *Slovník cizích slov* [online]. [cit. 2021-10-29]. Dostupné z: <https://slovník-cizich-slov.abz.cz/web.php/slovo/detekce>
- [4] BLINN, J.F. What Is a Pixel? *IEEE Computer Graphics and Applications* [online]. 2005, **25**(5), 82-87 [cit. 2021-10-25]. ISSN 0272-1716. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/1510545/references#references>
- [5] Color depth. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-10-25]. Dostupné z: https://en.wikipedia.org/wiki/Color_depth
- [6] JANÁKOVÁ, Ilona. *Předzpracování obrazu* [online]. Brno [cit. 2021-11-12]. Dostupné z: http://vision.uamt.feec.vutbr.cz/POV/lectures/05_Predzpracovani%20obrazu.pdf
- [7] Camera lens. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-11-14]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Camera_lens&oldid=1044706289
- [8] TORRE, Vincent a Tomaso A. POGGIO. On Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 1986, **PAMI-8**(2), 147-163 [cit. 2021-12-22]. ISSN 0162-8828. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4767769&isnumber=4767763>
- [9] CHAPLE, Girish N., R. D. DARUWALA a Manoj S. GOFANE. Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA. In: *2015 International Conference on Technologies for Sustainable Development (ICTSD)* [online]. IEEE, 2015, 2015, s. 1-4 [cit. 2021-12-24]. ISBN 978-1-4799-8187-8. Dostupné z: [doi:10.1109/ICTSD.2015.7095920](https://doi.org/10.1109/ICTSD.2015.7095920)
- [10] Canny edge detector. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-12-25]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Canny_edge_detector&oldid=1061925758
- [11] WESZKA, Joan S. a Azriel ROSENFELD. Threshold Evaluation Techniques. *IEEE Transactions on Systems, Man, and Cybernetics* [online].

- 1978, **8**(8), 622-629 [cit. 2021-12-26]. ISSN 0018-9472. Dostupné z:
[doi:10.1109/TSMC.1978.4310038](https://doi.org/10.1109/TSMC.1978.4310038)
- [12] Blob detection. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-12-27]. Dostupné z:
https://en.wikipedia.org/w/index.php?title=Blob_detection&oldid=1018858472
- [13] MATLAB. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-12-28]. Dostupné z:
<https://cs.wikipedia.org/wiki/MATLAB>
- [14] Image Processing Toolbox. Mathworks [online]. Natick (MA): The MathWorks, 2021 [cit. 2021-12-28]. Dostupné z:
<https://www.mathworks.com/products/image.html>
- [15] Computer Vision Toolbox. Mathworks [online]. Natick (MA): The MathWorks, 2021 [cit. 2021-12-28]. Dostupné z:
<https://www.mathworks.com/products/computer-vision.html>
- [16] FISCHER, Robert, Simon PERKINS, Ashley WALKER a Erik WOLFRAT. Zero crossing detector. Image processing learning resources [online]. 2004 [cit. 2021-12-30]. Dostupné z: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/zeros.htm>
- [17] MAŠEK, Jan Detekce objektů v obraze s pomocí Haarových příznaků: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 72 s. Vedoucí práce byl Ing. Radim Burget, Ph.D. Dostupné z:
https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=51400
- [18] Vision.CascadeObjectDetector: Detect objects using the Viola-Jones algorithm. *Mathworks* [online]. Natick, Massachusetts: The MathWorks, 2012 [cit. 2022-04-12]. Dostupné z:
<https://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-system-object.html>
- [19] DANG, Kirti a Shanu SHARMA. Review and comparison of face detection algorithms. In: *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence* [online]. IEEE, 2017, 2017, s. 629-633 [cit. 2022-04-12]. ISBN 978-1-5090-3519-9. Dostupné z:
doi:10.1109/CONFLUENCE.2017.7943228
- [20] WHITEHILL, J. a C.W. OMLIN. Haar Features for FACS AU Recognition. In: *7th International Conference on Automatic Face and Gesture Recognition (FGRO6)* [online]. IEEE, 2006, s. 97-101 [cit. 2022-04-14]. ISBN 0-7695-2503-2. Dostupné z: doi:10.1109/FGR.2006.61
- [21] Human Pose Estimation with Deep Learning. *GitHub* [online]. Tohru Kikawada, 2021 [cit. 2022-05-02]. Dostupné z: <https://github.com/matlab-deep-learning/Human-Pose-Estimation-with-Deep-Learning/releases/tag/v1.0.3>
- [22] BIN, Xiao, Wu HAIPING a Wei YINCHEN. Simple Baselines for Human Pose Estimation and Tracking. *ArXiv* [online]. Čína: arXiv, 2018 [cit. 2022-04-28]. Dostupné z: <https://doi.org/10.48550/arXiv.1804.06208>

- [23] LIN, Tsung-Yi, Michael MAIRE, Serge BELONGIE, et al. Microsoft COCO: Common Objects in Context. *ArXiv* [online]. arXiv, 2014 [cit. 2022-04-28]. Dostupné z: <https://doi.org/10.48550/arXiv.1405.0312>
- [24] BELAGIANNIS, Vasileios a Andrew ZISSERMAN. Recurrent Human Pose Estimation. In: *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)* [online]. IEEE, 2017, 2017, s. 468-475 [cit. 2022-05-02]. ISBN 978-1-5090-4023-0. Dostupné z: [doi:10.1109/FG.2017.64](https://doi.org/10.1109/FG.2017.64)
- [25] THOMPSON, Jonathan, Arjun JAIN, Yann LECUN a Christoph BREGLER. Joint training of a convolutional network and a graphical model for human pose estimation. *jonathantompson.github.io* [online]. New York, 2014 [cit. 2022-05-02]. Dostupné z: <https://jonathantompson.github.io/others/joint-training-convolutional.pdf>
- [26] KŘIVÁNEK, Filip. *Detekce částí lidského těla v obraze* [online]. Brno, 2014 [cit. 2022-05-03]. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=87892. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Ondřej Šmirg.
- [27] DALAL, N. a B. TRIGGS. Histograms of Oriented Gradients for Human Detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* [online]. IEEE, 2005, s. 886-893 [cit. 2022-05-03]. ISBN 0-7695-2372-2. Dostupné z: [doi:10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177)
- [28] Histogram of oriented gradients. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-05-03]. Dostupné z: https://de.wikipedia.org/wiki/Histogram_of_oriented_gradients
- [29] BULAT, Adrian a Georgios TZIMIROPOULOS. Human Pose Estimation via Convolutional Part Heatmap Regression. *Link.springer* [online]. Computer Vision - ECCV 2016, 2016 [cit. 2022-05-04]. Dostupné z: https://doi.org/10.1007/978-3-319-46478-7_44
- [30] Parallel Computing Toolbox. Mathworks [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-04]. Dostupné z: <https://www.mathworks.com/products/parallel-computing.html>
- [31] Deep Learning Toolbox. Mathworks [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-05]. Dostupné z: <https://www.mathworks.com/products/deep-learning.html>
- [32] Statistic and Machine Learning Toolbox. Mathworks [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-05]. Dostupné z: <https://www.mathworks.com/products/statistics.html>
- [33] MATLAB Support Package for USB Webcams. Mathworks [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-05]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/45182-matlab-support-package-for-usb-webcams>

- [34] MATLAB Support Package for IP Cameras. Mathworks [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-05]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/49824-matlab-support-package-for-ip-cameras>
- [35] BIN YANG, Junjie YAN, Zhen LEI a Stan Z. LI. Aggregate channel features for multi-view face detection. In: *IEEE International Joint Conference on Biometrics* [online]. IEEE, 2014, 2014, s. 1-8 [cit. 2022-05-10]. ISBN 978-1-4799-3584-0. Dostupné z: doi:10.1109/BTAS.2014.6996284
- [36] vision.PeopleDetector [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-16]. Dostupné z: <https://www.mathworks.com/help/vision/ref/vision.peopledetector-system-object.html>
- [37] peopleDetectorACF [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-16]. Dostupné z: <https://www.mathworks.com/help/vision/ref/peopledetectoracf.html>
- [38] corr2 [online]. Natick (MA): The MathWorks, 2021 [cit. 2022-05-16]. Dostupné z: <https://www.mathworks.com/help/images/ref/corr2.html>
- [39] Jak fotit digitální zrcadlovkou (DSLR) a bezzrcadlovkou: 26. díl – OHNISKOVÁ VZDÁLENOST OBJEKTIVU. *Fotoporadna* [online]. 2022, (26) [cit. 2022-05-18]. Dostupné z: <https://www.megapixel.cz/jak-fotit-digitalni-zrcadlovkou-dslr-a-bezzrcadlovkou-26-dil-ohniskova-vzdalenost-objektivu>

SEZNAM PŘÍLOH

PŘÍLOHA A - VIDEO, SNÍMKY A HODNOTY	58
---	----

Příloha A - Video, snímky a hodnoty

A.1 Tabulka srovnání časů pro detekci jednoho snímku

Detekovaná část těla	Průměrný čas (z 60 snímků) [s]
Horní část těla (rameno + hlava)	0.3589
Ústa	0.1992
Oči (velký pár)	0.1619
Oči (malý pár)	0.1896
Obličej	0.2537
Nos	0.2912

Parametry PC, na kterých byl proveden test metod a měření času:

Processor – AMD Ryzen 5 3600 6-Core Processor

Grafická karta – NVIDIA GeForce GTX 1660 SUPER

Paměť – 16 GB DDR4

Disk – Samsung SSD 860 EVO 500 GB

A.2 Přiložený .zip soubor s videem, snímky a vývojovými diagramy