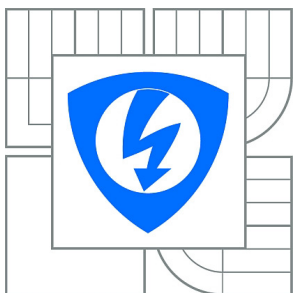


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DETEKTOR OBJEKTŮ V OBRAZECH ZALOŽENÝ NA METODĚ C4

IMAGE OBJECT DETECTOR BASED ON C4 ALGORITHM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

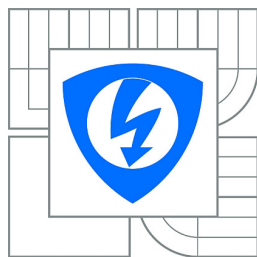
Bc. RADEK VYLÍČIL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN MAŠEK

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Radek Vylíčil

ID: 130685

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Detektor objektů v obrazech založený na metodě C4

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte obrazový detektor založený na nové metodě C4. V rámci detektoru bude vytvořen extraktor příznaků implementovaný v jazyce JAVA a celý algoritmus bude možné kombinovat s učícími se algoritmy umělé inteligence. Funkčnost detektoru ověřte na vybraných databázích snímků a výsledky zobrazte v grafech.

DOPORUČENÁ LITERATURA:

[1] J. Wu, C. Geyer, J.M. Rehg, "Real-time human detection using contour cues," In Robotics and Automation (ICRA), 2011 IEEE International Conference, ISSN 1050-4729, 2011.

[2] Ballard, D. H.; Brown, Ch. M. Computer Vision. Englewood Cliffs, N.J.: Prentice Hall, 1982.

Termín zadání: 9.2.2015

Termín odevzdání: 26.5.2015

Vedoucí práce: Ing. Jan Mašek

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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

Tato práce se zabývá detekcí v obraze s použitím metody C4. V textu jsou popsány metody trénování a testování navrženého objektového detektoru. Hlavní přínos spočívá ve vytvoření extraktoru příznaků pro vytvoření objektového detektoru v jazyce Java. Funkčnost objektového detektoru byla ověřena na databázích lékařských snímků.

KLÍČOVÁ SLOVA

Detekce objektu, zpracování obrazu, metoda C4, Extraktor příznaků, Java

ABSTRACT

This thesis deals with the image object detection using Contour cues. The text describes methods how to train and test object detector. The main contribution of this thesis consists in creation Feature extractor for creation object detector in Java programming. The functionality of object detector was demonstrated on medical images.

KEYWORDS

Object detection, image processing, Contour Cues, Feature extractor, Java

VYLÍČIL, Radek *Detektor objektů v obrazech založený na metodě C4*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 55 s. Vedoucí práce byl Ing. Jan Mašek

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Detektor objektů v obrazech založený na metodě C4“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Janu Maškovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych rád poděkoval své rodině za podporu během mého studia.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 Metody detekce objektů	12
1.1 Detekce obličeje	12
1.2 Rozpoznání úsměvu	13
1.3 Detektor očí	13
1.4 Detekce lidských postav	14
2 METODA C4	15
2.1 Integrální obraz	17
2.2 Sobelův operátor	18
2.3 Centrist deskriptor	19
2.4 Vektor vlastností	20
2.5 Detekce	21
2.6 Závěrečné zpracování	21
3 Praktická část	23
3.1 Extraktor příznaků	23
4 Řešené problémy	29
4.1 Detekce postav	29
4.1.1 Nastavení detektoru	31
4.1.2 Výsledky detekce postav	31
4.2 Detekce tepen	33
4.2.1 Trénování detektoru	33
4.2.2 Testování Detektoru	37
5 Zhodnocení výsledků	43
5.1 Zhodnocení výsledků detekce osob	43
5.2 Zhodnocení výsledků detektoru tepen	43
6 Závěr	44
Literatura	45
Seznam symbolů, veličin a zkratk	50
Seznam příloh	52

A	Obsah přiloženého CD	53
B	Ukázky detekce objektů	54
B.1	Správně detekované objekty	54
B.2	Falešně detekované objekty	55

SEZNAM OBRÁZKŮ

2.1	Průběh programu	16
2.2	Hodnota integrálního obrazu v bodě (u, v)	17
2.3	Výpočet integrálního obrazu (u, v)	18
2.4	Konvoluční jádra pro vertikální a horizontální přechod	18
2.5	Aplikace Sobel filtru	19
2.6	Příklad výpočtu hodnot CT	20
2.7	Vektor vlastností [20]	21
2.8	Označené oblasti	22
2.9	Závěrečné zpracování	22
3.1	Extrakce příznaků	24
3.2	Sobelův obraz	25
3.3	CT obraz	27
3.4	Extrahované příznaky	28
4.1	Detekce postav	30
4.2	Trénování modelu	34
4.3	Pozitivní trénovací snímky	35
4.4	Negativní trénovací snímky	35
4.5	Testování detektoru	37
4.6	Závislost detekce na velikosti detekčního okna	39
4.7	Závislost detekce na velikosti detekčního okna	40
4.8	Detekce databáze Ultrasonix	41
4.9	Detekce databáze Toshiba	41
B.1	Správně detekované tepny	54
B.2	Falešně detekované tepny	55

SEZNAM TABULEK

4.1	Porovnání detekce jednotlivých algoritmů pro pozitivní snímky	32
4.2	Porovnání detekce jednotlivých algoritmů pro negativní snímky	32
4.3	Velikost detekčního okna pro databázi Toshiba	38
4.4	Velikost detekčního okna pro databázi Ultrasonix	39

ÚVOD

Obrazový snímek v sobě ukrývá mnoho informací. Schopnost získat tyto informace je v dnešní době velice aktuálním tématem. Lidskou analýzou se sice tyto informace dají získat, ale tento proces je velice časově náročný, proto je velmi užitečné použít k analýze počítačové detektory. Snímek je analyzován a provede se na něm aplikace několika fází zpracování. V dnešní době se můžeme setkat s velkým množstvím různých druhů detektorů. Každá oblast má jiný důvod zpracování, nicméně cíl je vždy stejný. Vytvořit systém pro automatické rozpoznání daného objektu. Využití těchto systémů je velice rozsáhlé, od zabezpečovacích systémů[18], až po využití v lékařství[17]. Ve většině případů se ale jedná o detekci obličeje, očí, úsměvu nebo lidských postav.

Hlavním přínosem práce je vytvoření a otestování funkce extraktoru příznaků, pomocí kterého byl vytvořen klasifikační algoritmus pro rozpoznání příčných tepen. Dalším přínosem je vznik trénovací fáze, pomocí které je možné si natrénovat vlastní detektor pro vlastní data. Pro vytvořený algoritmus detekce příčných řezů tepen ze zašumělých ultrazvukových snímků byla přesnost testována dvakrát, jednou pro databázi z přístroje Toshiba a podruhé pro databázi z přístroje Ultrasonix. Přesnost pro Toshibu byla 18,01% a 21,94%. Pro databázi Ultrasonix byla přesnost 72,74% a falešně pozitivní míra 35,68%. Pro algoritmus detekce lidských postav byly dosažené procentuální výsledky úspěšnosti vytvořeného algoritmu navíc porovnány s ostatními nejčastěji používanými systémy detekce v současné době. Tento algoritmus postav pracuje v reálném čase. Přesnost systému byla na pozitivních snímcích obsahující lidské postavy 93,9%. Míra falešně pozitivních detekcí u pozitivních snímků byla 11,7%. U negativních snímků neobsahující lidské postavy byla přesnost 92% a 8% míra falešně pozitivních detekcí. Veškeré algoritmy jsou vytvořeny v jazyce Java a vycházejí z detekční metody C4.

Zbytek práce je dělen následovně: první kapitola popisuje různé metody detekce objektů v obraze. V druhé kapitole se nachází obecný popis metody C4 a popis jednotlivých částí zpracování obrazu. Ve třetí kapitole je popsána funkce vytvořeného extraktoru příznaků. Ve čtvrté kapitole se nachází popis vytvořených detekčních metod. Pátá kapitola popisuje dosažené výsledky.

1 METODY DETEKCE OBJEKTŮ

Detekce objektů v obraze je důležitou součástí velké škály aplikací, které se protínají v mnoha aspektech našeho života. Pro člověka je velice časově náročné analyzovat jednotlivé snímky, proto se pro analýzu využívá počítačové zpracování. V této kapitole budou uvedeny některé příklady využití detekce objektů v obraze.

1.1 Detekce obličeje

Velice rozsáhlou problematiku tvoří detekce obličeje. Jedná se o počítačovou technologii určující polohu a velikost lidské tváře v digitálním obraze.

V článku[8] je popsána hybridní metoda využívající Haarových příznaků založených na Viola-Jones[32] detektoru (V-J), který byl původně navržen pro snímky ve stupni šedi. Tento detektor je kombinován s filtrem barvy kůže, který poskytuje informace o obraze. Snímek je nejprve předzpracován pomocí Haarova detektoru, který je upraven tak, že pracuje v bodě, kde ROC křivka[10] má nízký počet správně detekovaných tváří, ale vysoký počet falešných detekcí. Aplikací filtru barvy kůže je mnoho z těchto falešných detekcí odstraněno. Dále se zde použil kompenzační algoritmus pro odstranění efektů osvětlení. Popsané testování bylo prováděno na BAO obličejové databázi. Výsledky ukazují, že navržená metoda je lepší než původní V-J algoritmus.

V[14] se návrh detekce obličeje realizuje pomocí ISM (Implicitní tvarový model) [19] podporující SVM (Support Vector Machine)[29] učení, kde na vybraných vzorcích pomocí Harrisova detektoru[23] byly spočítány HOG (Histogramy orientovaných gradientů)[6]. Z těchto vzorků byl prezentován obličejový obrázek. Dále došlo k natrénování SVM klasifikátoru založeném na extrahovaných vektorech. Nakonec se vytvořil jednoduchý obličejový detektor pomocí ISM. Velice úspěšných výsledků se dosáhlo pro detekci obličeje zepředu. V této metodě došlo k rozšíření o rozpoznání, zda se jedná o profil obličeje zleva nebo zprava. Tato metoda byla funkční pouze pokud se na snímku nacházel pouze jeden obličej, v případě více tváří je algoritmus nedokázal všechny detekovat.

V[39] se zabývají detekcí ve statických obrázcích pomocí kombinace dvou algoritmů Adaboost[15] a Neuronové sítě. V této metodě se využívají Haarovy příznaky pro rychlou extrakci obličeje. Klasifikátor Adaboost tuto rychlost ještě zvýší, ale výsledná detekce má velkou míru falešně detekovaných obrázků. Neuronová síť se používá pro konečné rozlišení, zda se jedná o obličej či nikoliv. Testování bylo prováděno na čtyřech různých databázích, ve kterých bylo obsaženo více než 1000 obrázků s velikostí 240x320 pixelů. Konečné výsledky ukazují, že metoda dosáhla úspěšnosti

93,34% a 0,34% míry falešné detekce v porovnání s původní metodou Adaboost, která dosahuje úspěšnosti 98,13% a 6,5% míry falešné detekce.

1.2 Rozpoznání úsměvu

Detekce úsměvu je velice zajímavý problém pro mnoho potencionálních aplikací. Tato detekce může být použita při měření pocitu spokojenosti nebo potěšení.

V článku [12] je popsána metoda detekce úsměvu pomocí LBP (Lokální binární vzory) obrazového deskriptoru [21]. Využívá se zde v kombinaci se dvěma metodami klasifikace k-NN (k-nejbližší sousedé) [4] a SVM. LBP mají za následek zdůraznění hran. Při detekci úsměvu se vychází z detekce obličeje, kdy je obraz obvykle rozdělen do bloků. Pomocí sběru informací z histogramů extrahovaných z těchto obdelníkových bloků se popisuje celý obličej. Tato metoda ale není pro detekci obrazu obličeje dostačující. V případě, že se oblast omezí pouze na ústa, odstraní se jiné textury, čímž dojde ke zvýšení výkonu a tato metoda se dá použít s velkou úspěšností pro nalezení úsměvu.

V [30] je popsán návrh detekční techniky pro úsměv, která je schopna detekovat úsměv v různých světelných podmínkách, pozici obličeje a velikosti obrázku s malou složitostí. Tato metoda nejprve ze snímku detekuje obličej, poté aplikuje detektor úst. Extrahuje rysy z této oblasti a klasifikuje, zda se jedná o usmívající nebo neusmívající stav. V této práci jsou analyzovány různé metody extrakce příznaků. Klasifikační techniky jsou analyzovány z výkonů těchto testů a výpočetní složitosti stanovisek. Nejlepší kompromis mezi výkonem a složitostí je reprezentován kombinovaným přístupem, který využívá jak tvar funkce, tak tvar textury. Využívá se klasifikátor založený na Mahalanobisově klasifikátoru [9]. Toto řešení dosáhlo velice dobrých výsledků s velice malou složitostí a je vhodná pro implementaci do mobilních telefonů.

1.3 Detektor očí

Oči jsou nejvýznamější součástí tváře. Každé oko má svůj jedinečný vzor. Využití této detekce se naskytá převážně v místech s vysokou oblastí zabezpečení.

Článek [7] popisuje jednu z mnoha metod detekce očí. Tento systém se skládá z automatického segmentačního systému, který je založen na vlnkové transformaci [1] a neuronové síti. Pro detekci se využívá biometrického systému, kdy se nejprve sejme digitální obraz oka. Tento obraz se poté pomocí matematických funkcí převede na biometrickou šablonu, která se uloží jako vzor do databáze. Pro získání těchto vzorů se používá speciální optická kamera. Identifikace se poté provádí na základě skenu

oka a jeho porovnání s uloženými vzory v databázích. Přesnost této metody byla 90% pro identifikaci zkušebních snímků s různou velikostí a za různých podmínek obsažených v tréninkové fázi. Popsaná metoda je navíc odolná vůči osvětlení, barvě pozadí nebo výrazu obličeje.

1.4 Detekce lidských postav

Detekce lidských postav je asi nejznámějším tématem detekce. Schopnost rozlišit, zda se na snímku nachází lidská postava má velké využití. Nejčastěji se jedná o různé zabezpečovací systémy [18].

V článku [22] se detekce lidí realizuje propojením HOG deskriptoru a SVM klasifikátoru. HOG+SVM je považován za jeden z nejúspěšnějších algoritmů detekce. Pro realizaci detekce je metoda založena na dvou možnostech zpracování. První možností je znovu použití rysů v bloku pro konstrukci HOG deskriptoru pro protínající se detekční okna. Druhou možností je využít sub-buněčné interpolace založené na efektivním výpočtu funkce HOG pro každý blok. Kombinací těchto dvou způsobů se dosáhlo podstatného zvýšení detekce lidí. Výsledky metody byly zjišťovány ze známé INRIA¹ databáze postav, pomocí které se prokázala účinnost a efektivita. Metoda je více než pětikrát účinnější.

Článek [3] pojednává o detekci postav sloučením Gaborových filtrů [11] a histogramu HOG. Sleduje použití Gáborova filtru pro předzpracování, zejména zlepšení zisku informací ze vstupního obrazu. Metoda dosáhla dobrých výsledků bez ohledu na obtížnosti obrazů. Dokázalo se, že použití Gáborova filtru pro předzpracování (velikost gradientu, orientaci, atd.) má za následek vylepšení funkcí metody. Je-li toto předzpracování také zavedeno i do ostatních metod extrakce příznaků, dosáhne lepších informací, čímž dojde k posílení lidského tvaru. Vzhledem k těmto dosaženým výsledkům se tato metoda velice hodí pro návrh inteligentních sledovacích systémů.

V [38] je představena metoda detekce lidí ze stacionárních sledovacích videí. Je založena na kaskádovém LogitBoost [13] klasifikátoru, který používá kovarianční matice [35] jako objektový deskriptor. Metoda se nejprve naučí vztah mezi vzhledem a popředím obrazu. Ukazuje, že informace lidského tvaru obsažené v popředí obrazu mohou dramaticky zvýšit výkon, při současném použití s vzhledovými rysy. Dále ukazuje, že využití kovarianční matice hraných podskupin, místo úplné sady, při zvyšování poskytuje podobný nebo lepší výkon a zároveň výrazně snižuje zatížení výpočtu. Dále obsahuje jednoduchý obrazový systém pro předzpracování pro využití integrálního obrazu. Testování se provádělo na INRIA databázi a na velké databázi obsahující video snímky. Metoda dosáhla velice dobrých výsledků.

¹<http://pascal.inrialpes.fr/data/human/>

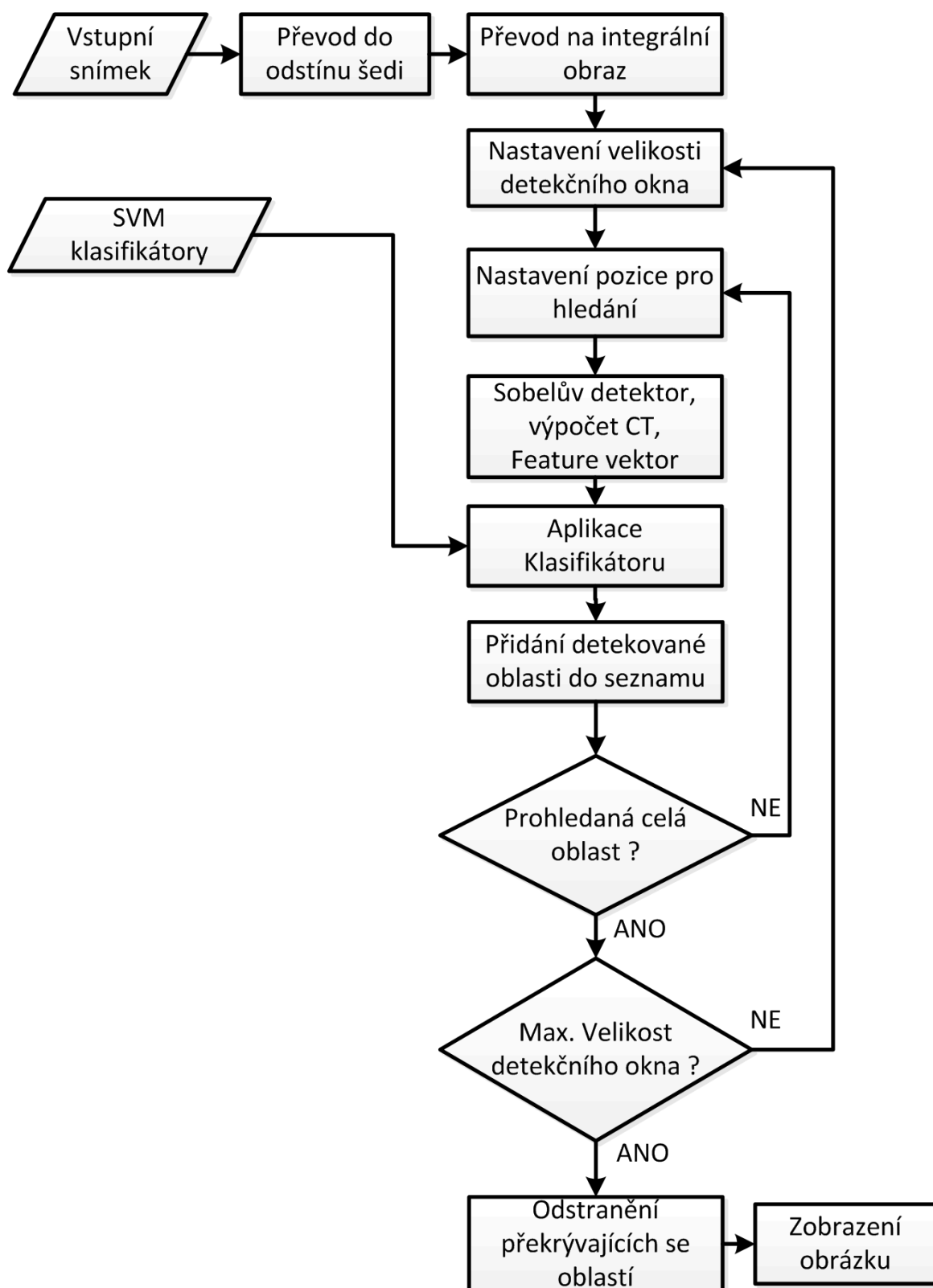
2 METODA C4

Metoda C4 je metoda pro rozpoznávání objektů v obraze. Tato metoda zpracovává vstupní obraz v reálném čase. Ke své funkci využívá obrysové kontury lidských postav. Původní verze pro testování detektoru využívala dvou kaskádových SVM klasifikátorů. Tato metoda detekce se dá upravit k natrénování vlastního klasifikátoru na vlastních datech a použitím libovolného učícího se algoritmu umělé inteligence. Řešení vlastního klasifikátoru bude popsáno v praktické části. Pro rychlou funkci detektoru slouží integrální obraz, Sobelův detektor hran a deskriptor CENTRIST. Metoda nepotřebuje žádné speciální hardwarové požadavky, ani obrazové předzpracování na vektorovou normalizaci. Potřebuje pouze jeden krok k otestování obrazového pod-okna [37]. Tato metoda je velice rychlá a velice jednoduchá pro zpracování. Velice zásadním rozdílem oproti existujícím metodám je ten, že postava člověka se u metody C4 detekuje přímo z obrázku vzniklém pomocí Sobelova detektoru hran.

Název metody je odvozen od použitých funkcí zpracování obrazu. Obrys(Contour), kaskádový klasifikátor(Cascade Classifier) a CENTRIST vizuální deskriptor, tudíž C4.

Na obrázku 2.1 je zachycen průběh a zpracování obrazu pomocí metody C4.

Celý proces začíná načtením vstupního snímku, který je určen k detekci, zároveň se načtou i dva SVM klasifikátory. Snímek se převede do odstínu šedi, ze kterého se vypočítá integrální obraz. Převod na integrální obraz je z důvodu rychlejší práce s obrazem, kdy se nemusí neustále dokola vypočítávat stav pixelů. Dalším krokem je aplikace Sobelova operátoru. V dalším kroku je nastaveno detekční okno pro vyhledání objektu. V detekčním okně je proveden výpočet Centrist hodnoty. Takto vypočtené hodnoty jsou uloženy do Feature vektoru vlastností. Na něj jsou aplikovány dva SVM klasifikátory, které slouží k určení potenciálního objektu. V případě, že je objekt detekován, jsou jeho souřadnice uloženy do seznamu. Poté se celá oblast detekčního okna posune na další souřadnice a celá detekce se opakuje. Takto detekce pokračuje, než je prohledán celý snímek. Poté se rozměry detekčního okna zvětší a proces se opakuje. Celý tento proces trvá tak dlouho, dokud nejsou rozměry detekčního okna stejné, jako je velikost snímku. Velice důležitým krokem je zpracování výsledného seznamu obsahující souřadnice nalezených objektů. Aby nedošlo k mnohonásobnému označení stejného objektu, je aplikována funkce pro odstranění těchto označení.



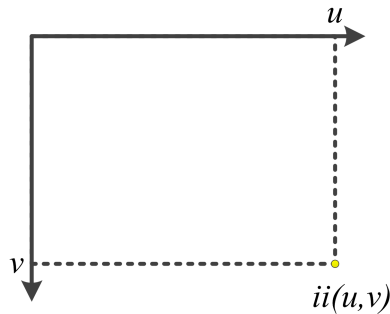
Obr. 2.1: Průběh programu

2.1 Integrovaný obraz

Integrovaný obraz je algoritmus sloužící k rychlému výpočtu hodnot jednotlivých příznaků ze vstupního obrazu. Algoritmus se aplikuje na vstupní obraz z toho důvodu, aby se pro každou tuto hodnotu nemusel počítat součet hodnot pixelů opakovaně. Tím vzniká integrovaný obraz. Hodnota na souřadnicích $ii(u, v)$ se v integrovaném obraze vypočítá jako:

$$ii(u, v) = \sum_{u' \leq u, v' \leq v} i(u', v'), \quad (2.1)$$

kde $ii(u, v)$ představují hodnoty pixelů integrovaného obrazu a $i(u, v)$ jsou hodnoty vstupního obrazu.



Obr. 2.2: Hodnota integrovaného obrazu v bodě (u, v)

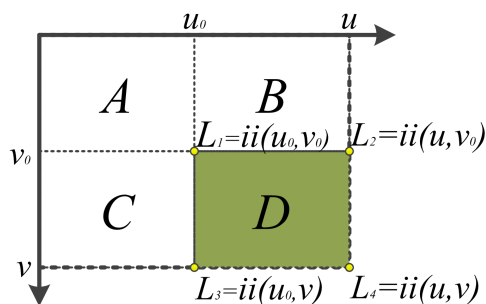
Každý bod představuje součet hodnot všech předcházejících bodů podle následujících rovnic [33]:

$$s(u, v) = s(u, v - 1) + i(u, v), \quad (2.2)$$

$$ii(u, v) = ii(u - 1, v) + s(u, v), \quad (2.3)$$

kde $s(u, v)$ odpovídá kumulovanému součtu všech hodnot v řádku obrazu, $ii(u, v)$ jsou hodnoty intenzit jednotlivých pixelů vstupního obrazu a $ii(u, v)$ jsou jednotlivé hodnoty integrovaného obrazu. Dále pak platí: $s(u, 0) = 0, ii(0, v) = 0$.

Na obrázku 2.3 jsou vidět hlavní oblasti (A, B, C, D) a body (L_1, L_2, L_3, L_4) . Hodnota L_1 je rovna sumě hodnot pixelů definovaných oblastí A . Hodnota L_2 je pak dána součtem oblastí A a B . Hodnota L_3 je součet oblastí A a C . Hodnota v bodě L_4 je dána součtem oblastí A, B, C, D . Výpočet pro hledanou oblast D je pak $L_4 + L_1 - L_2 - L_3$.



Obr. 2.3: Výpočet integrálního obrazu (u, v)

2.2 Sobelův operátor

Sobelův operátor, někdy také nazýván Sobelův filtr, je algoritmus pro detekci hran objektů v obraze. Aplikace se provádí na obraze ve stupních šedi. Tento operátor se skládá ze dvou konvolučních jader velikosti 3×3 , které jsou znázorněny na obrázku 2.4. Tato Jádra jsou navržena tak, aby reagovala na změnu jasu pixelů. Posouváním po vstupním obraze ve stupních šedi se zachytávají hrany objektů v horizontálním (G_x) a vertikálním (G_y) směru vzhledem k mřížce obrazových bodů. Pro každý pixel je poté vypočítána absolutní velikost gradientu a orientace tohoto přechodu [31].

-1	0	1
-2	0	2
-1	0	1

G_x

1	2	1
0	0	0
-1	-2	-1

G_y

Obr. 2.4: Konvoluční jádra pro vertikální a horizontální přechod

Obě jádra jsou téměř totožná, jedno jádro je jednoduše to druhé, otočené o 90° . V případě potřeby zjištění hodnot gradientu pouze v jednom směru (G_x nebo G_y) se mohou jádra využít samostatně. Po získání gradientů v jednotlivých směrech mohou být tyto hodnoty s využitím vzorce 2.4 sečteny dohromady, a tím se zjistí celková absolutní velikost gradientu v každém bodě.

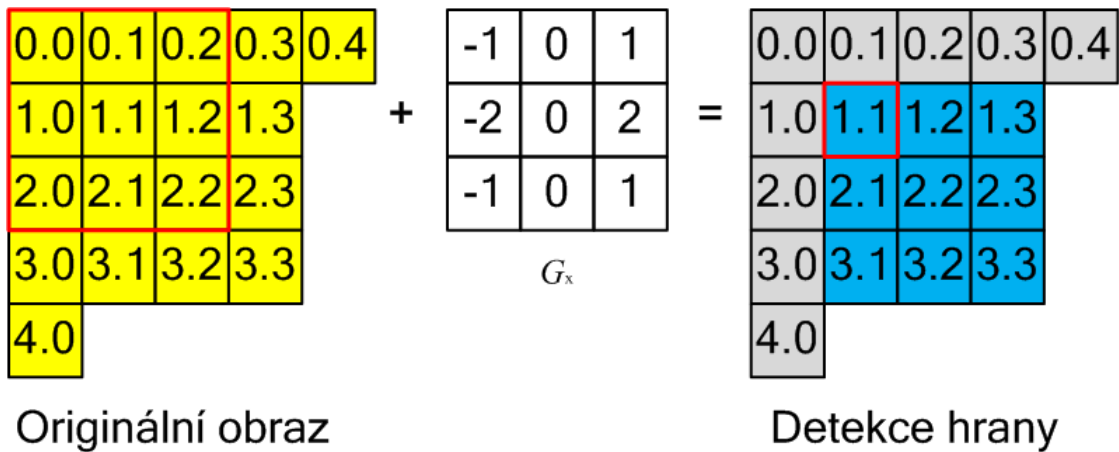
$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

Přibližná velikost Gradientu lze vypočítat pomocí

$$|G| = |G_x| + |G_y| \quad (2.5)$$

Na hodnoty v prvním a posledním řádku, stejně tak v prvním a posledním sloupci, nemůže být Sobelův operátor aplikován. Při umístění středu masky jádra přes pixel, například v první řadě, maska bude mimo hranice obrázku.

Na obrázku 2.5 je zobrazeno, jak se tento filtr aplikuje. Tento filtr se postupně posunuje v horizontální ose, až projde celým řádkem obrazu. Přesune se na řádek další a pokračuje v horizontálním směru. Pro výpočet výstupní hodnoty na souřadnicích [1.1] se využije vztahu 2.6. Podobný postup se dodržuje i v případě vertikální osy, zde se ale postupuje po sloupcích a namísto G_x se použije konvoluční jádro G_y .



Obr. 2.5: Aplikace Sobel filtru

$$\begin{aligned}
 [1.1] = & ([0.0][-1]) + ([0.1][0]) + ([0.2][1]) + ([1.0][-2]) + \\
 & + ([1.1][0]) + ([1.2][2]) + ([2.0][-1]) + ([2.1][0]) + ([2.2][1])
 \end{aligned} \quad (2.6)$$

Takto přetvořené obrázky se spojí do jednoho Sobel obrázku, na kterém je poté proveden výpočet hodnoty Centrist.

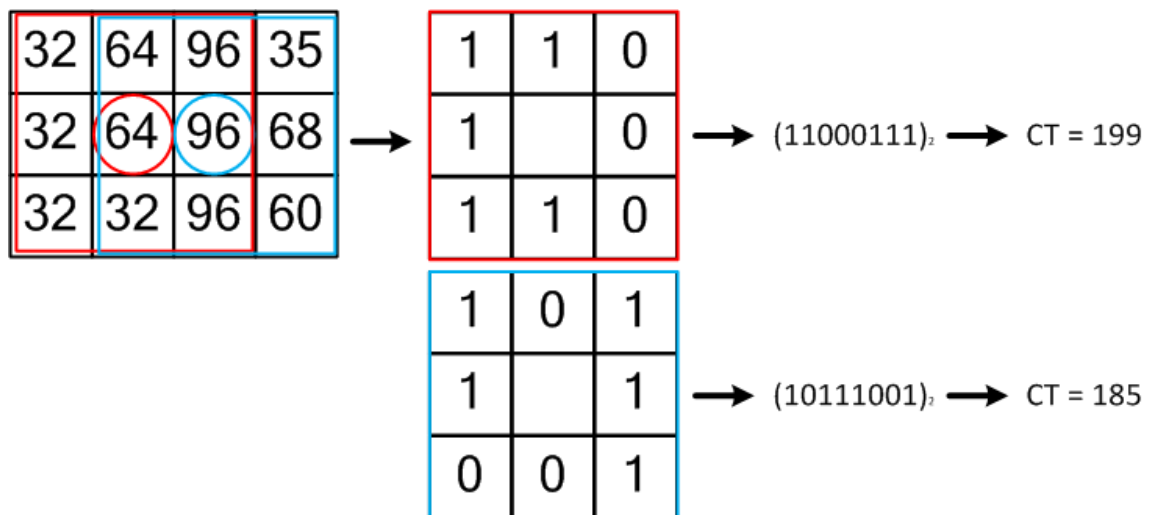
2.3 Centrist deskriptor

Vizuální deskriptor Centrist (CENSus TRansform hISTogram) je určen pro rozpoznání objektů. Deskriptor je aplikován na Sobelův obraz a převádí obrázky pomocí porovnání hodnoty intenzity ze sousedících pixelů [16]. Vytváří transformační okno

3x3 a uspořádává pixely do 8-bitového řetězce pro každý pixel. Tím vznikne nová hodnota daného pixelu. Pokud je prostřední pixel větší než jeho soused, je nastavena bitová hodnota na 1, v opačném případě je tato hodnota nastavena na 0 viz. 2.7.

$$p_{xy} = \begin{cases} 0 & \text{je-li } p_{\text{center}} > p_{xy} \\ 1 & \text{jinak } p_{\text{center}} \leq p_{xy} \end{cases} \quad (2.7)$$

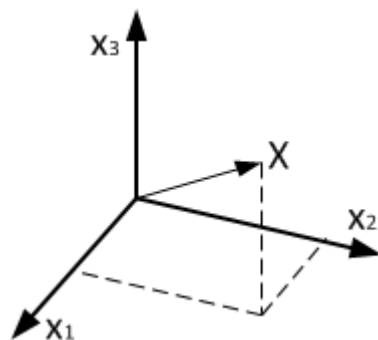
Obrázek 2.6 ukazuje příklad postupu a výpočtu CT hodnoty, kdy se po výpočtu hodnoty CT pro daný pixel celé transformační okno posune na následující pixel a výpočet se opakuje. Vzniklých 8 bitů může být poskládáno jakýmkoliv způsobem, v tomhle případě zleva doprava a shora dolů. Ze vzniklého řetězce se převedením čísla do základní dekadické podoby určí hodnota CT pro prostřední pixel. Takto vypočítaná hodnota se uloží do f vektoru. Centrist deskriptor je vlastně histogram těchto hodnot [36]. Nevýhoda Centristu je ta, že nedokáže zachytit globální strukturu a obrys větší než je okno 3x3 pixelů.



Obr. 2.6: Příklad výpočtu hodnot CT

2.4 Vektor vlastností

Vektor vlastností neboli Feature vector obsahuje vlastnosti vstupního objektu. Nejčastěji se jedná o řadu čísel reprezentující testovaný objekt. Obsahuje velké množství hodnot. Nejčastěji se používá ve spojitosti se strojovým učením. Používá se jako vstup pro aplikaci klasifikačního algoritmu [20]. Obrázek 2.7 zobrazuje, jak může vektor vlastností vypadat.



Obr. 2.7: Vektor vlastností [20]

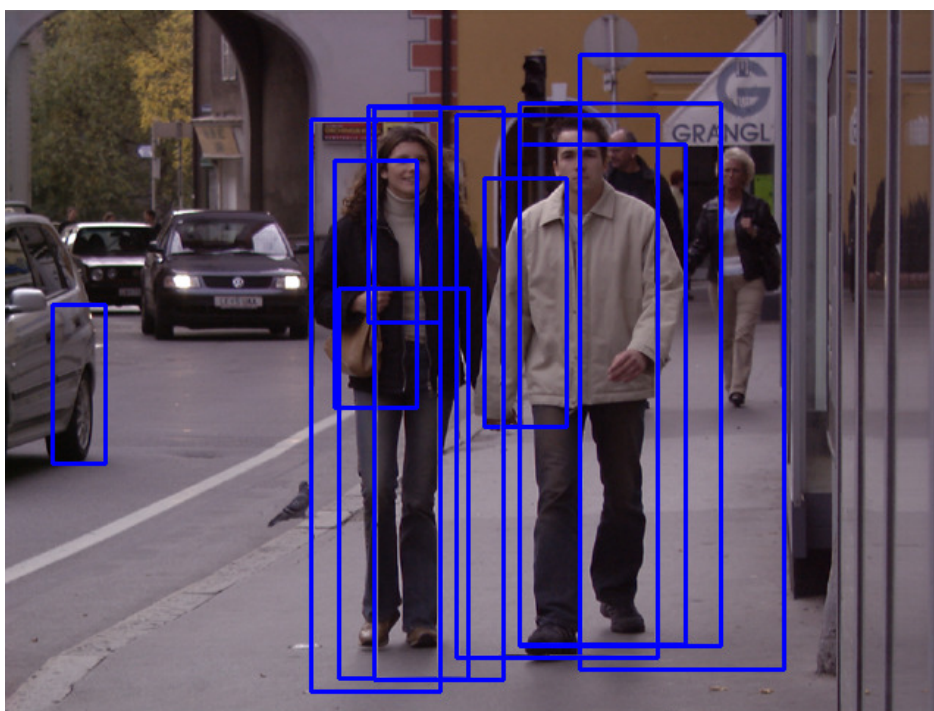
2.5 Detekce

Vzhledem ke své účinnosti je k detekování objektů použito Centrist vizuálního deskriptoru. Nejprve je vytvořeno detekční okno. Toto okno je poté rozděleno do určitého počtu bloků. V detekčním okně jsou provedeny výpočty feature vektoru. Na takto vypočtený feature vektor se aplikují natrénované SVM klasifikátory [37]. Tyto klasifikátory se porovnají s vypočtenými hodnotami feature vektoru vlastností a určí se, zda se zde nachází hledaný objekt. Detekční okno se poté přesune na další oblast a proces je proveden znovu. Tento proces pokračuje tak dlouho, dokud detekční okno nedosáhne velikosti vstupního obrazu. V tom momentu je celý detekční proces ukončen. Výsledně označené oblasti jsou poté určeny k post-procesu, aby se odstranilo mnohonásobné označení jednoho objektu.

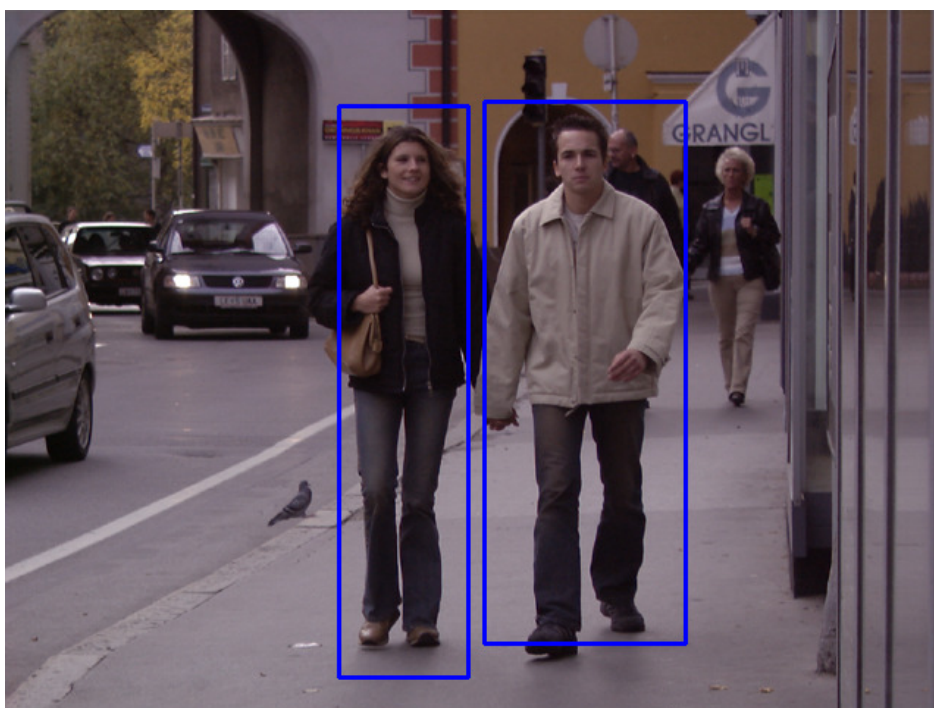
2.6 Závěrečné zpracování

Po zpracování detekce vzniknou ve snímcích oblasti, ve kterých se podle klasifikátorů nachází hledaný objekt. Z těchto oblastí jsou vybrány detekce, které si jsou velice podobné. Buď se nachází jedna uvnitř druhé, nebo se překrývají. Na takto nalezených oblastech je aplikována funkce pro zprůměrování. Tím dojde ke sloučení více detekcí stejného objektu do jednoho označení. Takto sloučené oblasti jsou nakonec převedeny a zobrazeny ve vstupním obraze. Na obrázku 2.8 můžete vidět, jak vypadají označené oblasti před závěrečným zpracováním. Obrázek 2.9 zobrazuje výsledný snímek po aplikování funkce sloučení. Takto označený snímek se uloží do souboru. Obrázky slouží jako příklad a jsou převzaty z databáze INRIA¹

¹<http://pascal.inrialpes.fr/data/human/>



Obr. 2.8: Označené oblasti



Obr. 2.9: Závěrečné zpracování

3 PRAKTICKÁ ČÁST

V rámci praktické částí diplomové práce bylo úkolem vytvořit program pro detekci objektů v obraze. Tento program vychází ze zdrojového kódu, který uveřejnili autoři J.Wu, C. Geyer a J.M. Rehg ve své práci [37]. Modifikací těchto kódů byl vytvořen algoritmus v jazyce Java využívající metodu C4 pro detekci postav. Pro realizaci funkcí programu byla použita knihovna OpenCV¹. Pro otestování tohoto algoritmu byla použita databáze postav INRIA², ze které bylo náhodně vybráno 100 snímků obsahující lidské postavy a 100 snímků obsahující pozadí. Tyto snímky obsahují různý počet osob a mají různou velikost. Na snímcích s postavami bylo celkem 264 osob.

Dalším úkolem bylo vytvoření extraktoru příznaků, který dokáže uložit převedený obraz do číselných souřadnic. Pomocí tohoto extraktoru byl vytvořen tréninkový set. Takto vytvořený set byl použit jako vstupní data pro algoritmus rozhodovacího stromu. Díky rozhodovacímu stromu vznikl klasifikační model pro nalezení příčných tepen v obraze. Natrénování a testování vytvořeného modelu bylo prováděno na databázích tepen vytvořených ultrazvukovými přístroji Ultrasonix a Toshiba³ [2], [24], [25], [26], [27]. Databáze pro přístroj Toshiba obsahovala celkem 433 snímků určených pouze k testování detektoru. Databáze Ultrasonix obsahovala 283 pozitivních a 284 negativních snímků určených pro natrénování modelu. Testovací set obsahoval 538 snímků.

3.1 Extraktor příznaků

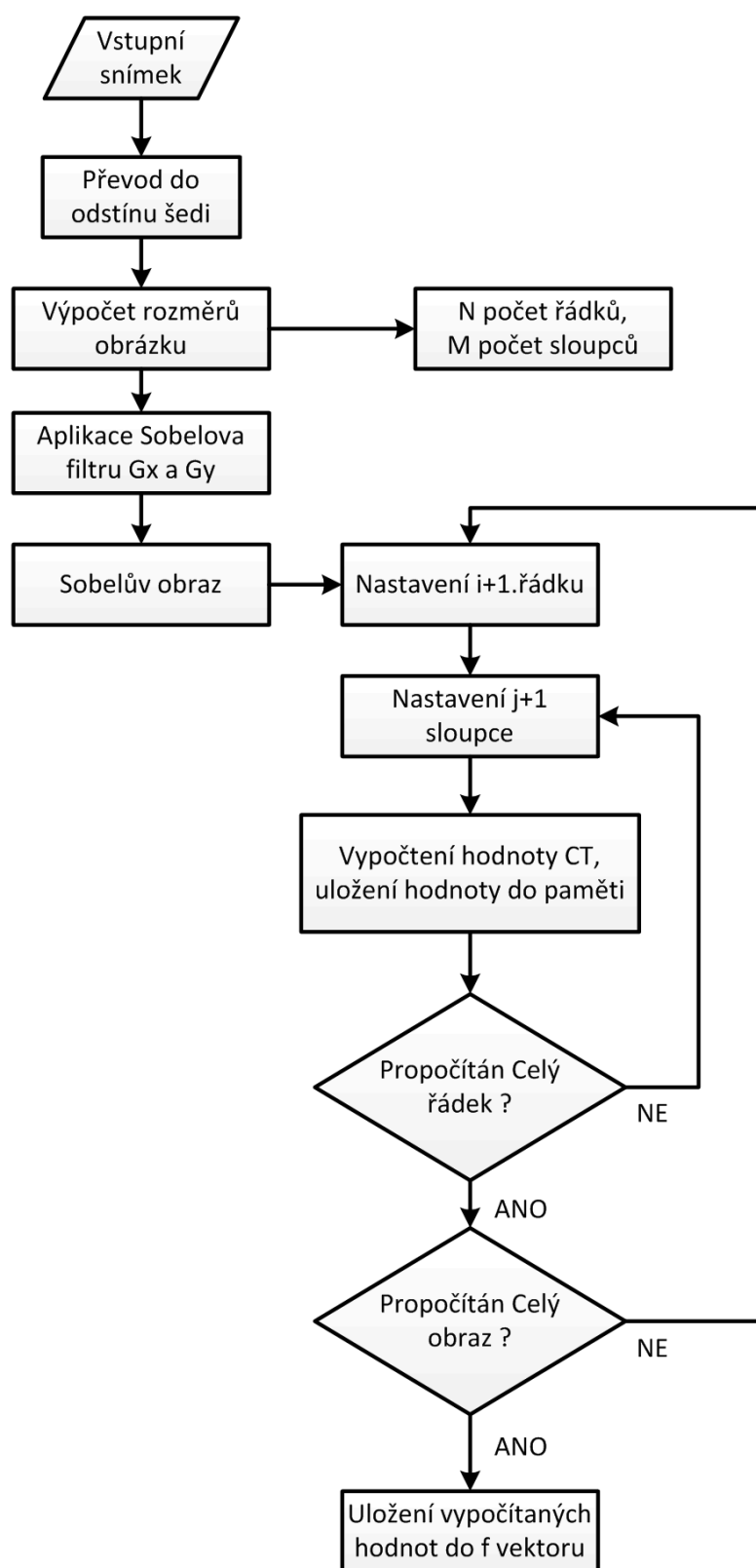
Při vzniku vlastního detekčního modelu bylo nutné nejprve vytvořit extraktor příznaků. Každý obrázek má určité vlastnosti, pro jejich získání musí obrázek projít několika fázemi zpracování. Na obrázku 3.1 je názorně zobrazeno, jak tento extraktor pracuje.

Na začátku celého procesu je načten snímek určený k extrakci příznaků. Dále je nastavena velikost pod-okna na 5x5 pixelů. Vstupní snímek je převeden do odstínu šedi (obrázek 3.2a). Pro obraz jsou nejprve zjištěny jeho rozměry $N \times M$ pixelů. S takto zjištěnými rozměry se poté pracuje jako s hranicí obrazu. Dále jsou zjištěny velikostní hodnoty jednotlivých pixelů. Tyto hodnoty jsou uloženy do paměti. V obraze jsou poté nastaveny souřadnice pro aplikaci G_x a G_y Sobelových filtrů. Jedná se o kaskádové matice 3x3. Jako prvotní souřadnice je nastaveno $i=2$ a $j=2$, kde i značí číslo řádku a j číslo sloupce. Nastaví se tedy druhý řádek a druhý sloupec z toho,

¹<http://opencv.org/>

²<http://pascal.inrialpes.fr/data/human/>

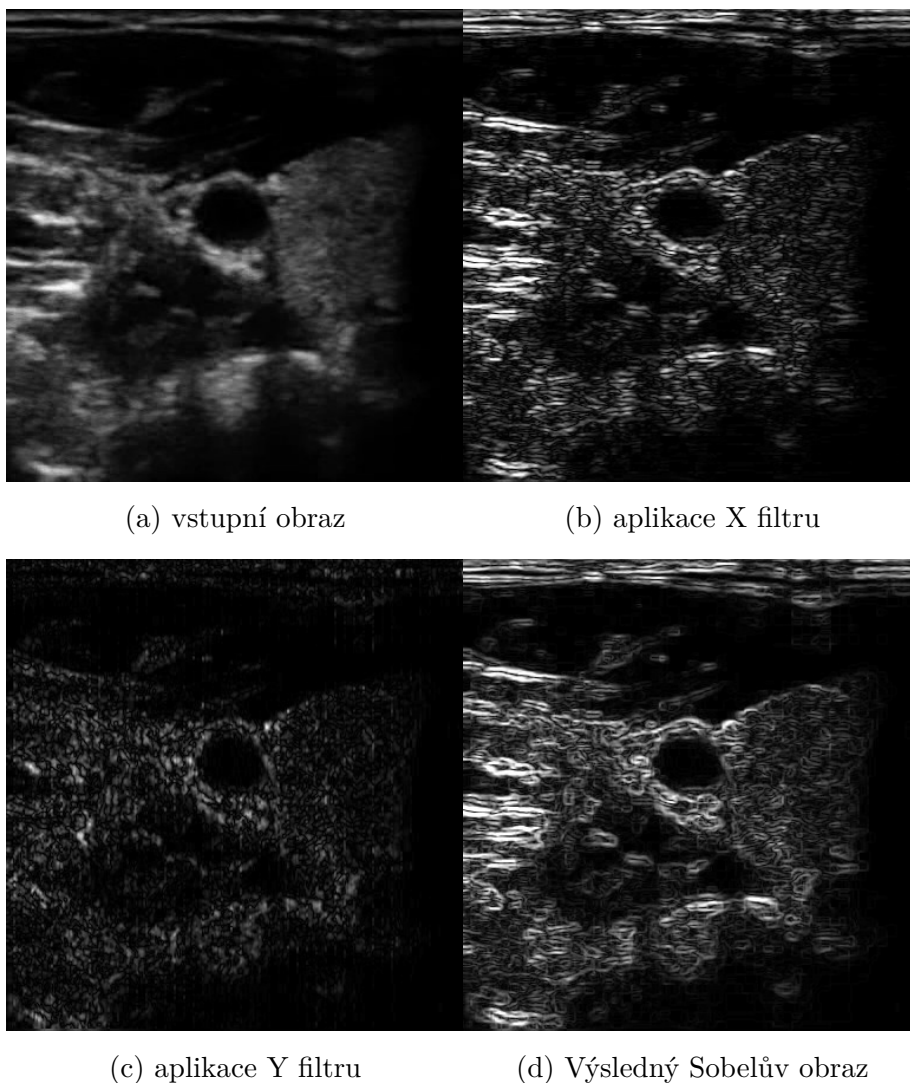
³<http://splab.cz/en/research/zpracovani-medicinskyh-signalu/databaze/artery>



Obr. 3.1: Extrakce příznaků

že pro okrajové pixely by převod vycházel mimo obraz, a to možné není. Na této souřadnici je poté aplikován G_x filtr, který přepočítá pixely. Současně je aplikován také G_y , který provádí totéž, ale pro vertikální osu. Na daných souřadnicích vznikne výřez obrazu 3×3 a na něj se aplikuje matice filtrů tak, že se mezi sebou jednotlivé pixely vynásobí, tím vznikne výsledná hodnota G_x (obrázek 3.2b) a G_y (obrázek 3.2c). Výpočet celkového gradientu daného pixelu se provádí pomocí vzorce 2.4.

Takto se postupuje po celém řádku dokud j nedosáhne velikosti hranice snímku. Po dosažení této hranice se celý převod přenastaví na následující řádek na souřadnice $i=3$ $j=2$ a celý proces se opakuje. Výpočty gradientů se provádí dokud se neprojde celý snímek. Pro každý pixel vznikne tedy nová hodnota. Sloučením těchto filtrací vznikne Sobelův obrázek (obrázek 3.2d).



Obr. 3.2: Sobelův obraz

Pro lepší pochopení funkce převodu je zde uveden pseudokód Sobelova filtru.

```
1 Start
SobelImage{
3   obraz N,M; \\ rozmery snimku
   for (i = 2 to N - 2) {
5     hodnoty vseh pixelu z pameti;
     for (j = 2 to M - 2){
7       for (vybrana oblast 3x3){
         aplikace Gx a Gy filtru;
9         Vypocet histogramu G;
         Ulozeni;
11        }
      }
13   }
Konec
```

Na výsledný snímek je poté aplikován CT deskriptor. Stejně jako u Sobel obrázku se hodnota CT nedá vypočítat pro první a poslední řádek, a stejně tak pro první a poslední sloupec daného snímku, jelikož jsou zadefinovány jako okraj obrazu a výpočet by nebyl možný. Proto je prvotní pozice pro výpočet opět nastavena na souřadnice 2. řádek 2. sloupec. Na této souřadnici je spočtena hodnota CT, kdy se aplikuje výřez 3x3. Z tohoto výřezu je pro prostřední pixel zjištěna hodnota, za pomoci osmi okolních pixelů, kdy se porovnává, zda je hodnota větší či menší než prostřední pixel. Z takto propočítaných hodnot je seskládána bitová posloupnost pro daný pixel. Tato hodnota se uloží do paměti. Celý proces se poté posunuje po celém řádku tedy na 3.sloupec, 4. atd. Poté co dorazí deskriptor na konec řádku, je jeho pozice přenastavena na následující řádek. Celý proces výpočtů se opět opakuje, dokud nejsou propočítány všechny pixely v celém snímku. Jak vypadá výsledný snímek je vidět na obrázku 3.3 Stejně jako u Sobel obrázku je i zde uveden pseudokód pro lepší pochopení výpočtu hodnot CT.

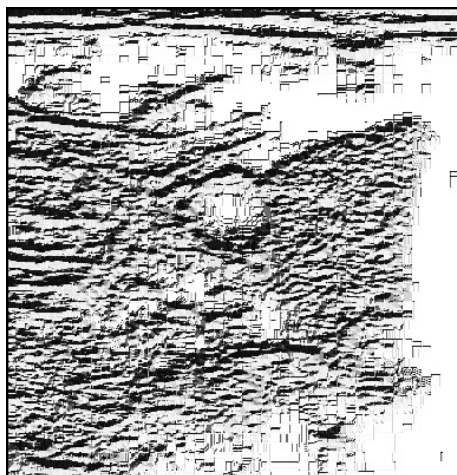
Listing 3.1: CT pseudokód

```
Start
2 computeCT{
   obraz N,M; \\ rozmery snimku
4   for (i = 2 to N - 2){
     //vypocet pixelu v okoli stredu mrizky
6     for (j = 2 to M - 2){
       // n = 1.- 8. soused
```

```

8         if (stredova hodnota <= n-soused) {
           hodnota = 1};
10        else {hodnota = 0};
           Vypocet ct (0-255);
12           ulozeni hodnot;
           }
14     }
}
16 Konec

```



Obr. 3.3: CT obraz

Pro uložení těchto souřadnic dochází k přepočtu do vektoru vlastností. Velikost vektoru není závislá na počtu pixelů obrázku, ale na nastavení a počtu pod-oblastí. Celkový počet příznaků je dán pomocí výpočtu 3.1, tento výpočet se skládá z několika parametrů. Nastavení velikosti pod-okna extrakce v x-ové a y-ové ose, konstanta EXT a základní délka, která má z důvodu výpočtu histogramu pro dvě sousední oblasti velikost 256.

$$\begin{aligned}
 FLength &= (XDIV - EXT) * (YDIV - EXT) * BASELENGTH = \\
 &= (5 - 1) * (5 - 1) * 256 = 4 * 4 * 256 = 4096, \quad (3.1)
 \end{aligned}$$

kde XDIV je nastavená velikost podoblasti v ose x, YDIV je nastavená velikost podoblasti v ose y, EXT je konstanta pixelu a BASELENGTH je základní délka histogramu.

Jak už bylo zmíněno výše, pro nastavení 5x5 je celkový počet příznaků 4096. Takto vypočítané souřadnice jsou uloženy do f vektoru vlastností a celý proces extrakce se ukončí. Jak tyto vektory příznaků vypadají, je vidět na obrázku 3.4, kdy jeden celý řádek odpovídá jednomu obrázku.

```

0.3462603878116344 0.00554016620498615 0.01662049861495845 0.01662049861495845 0.01662049861495845 0.002770083102493075 0.0221606648191
0.3518005540166205 0.008310249307479225 0.0110803324099723 0.0110803324099723 0.00554016620498615 0.0 0.030470914127423823 0.0415512461
0.38271604938271603 0.015432098765432098 0.009259259259259259 0.015432098765432098 0.015432098765432098 0.0030864197530864196 0.0246913
0.3611111111111111 0.021604938271604937 0.0030864197530864196 0.021604938271604937 0.009259259259259259 0.0 0.018518518518518517 0.033
0.3549382716049383 0.018518518518518517 0.021604938271604937 0.009259259259259259 0.009259259259259259 0.0030864197530864196 0.0308641
0.33641975308641975 0.009259259259259259 0.030864197530864196 0.02777777777777777 0.018518518518518517 0.006172839506172839 0.0246913
0.3487654320987654 0.018518518518518517 0.021604938271604937 0.012345679012345678 0.015432098765432098 0.0 0.030864197530864196 0.0401
0.3175 0.005 0.005 0.02 0.015 0.0 0.0225 0.0725 0.14 0.025 0.0025 0.0675 0.01 0.0025 0.01 0.2075 0.18 0.01 0.0025 0.0125 0.035 0.0025 0
0.2875 0.015 0.0225 0.015 0.0125 0.0025 0.045 0.075 0.085 0.04 0.01 0.1025 0.0125 0.0025 0.01 0.185 0.0875 0.0025 0.005 0.0025 0.0475 0
0.2880886426592798 0.00554016620498615 0.019390581717451522 0.0110803324099723 0.0110803324099723 0.0 0.030470914127423823 0.0526315781
0.29085872576177285 0.0110803324099723 0.027700831024930747 0.00554016620498615 0.019390581717451522 0.0 0.019390581717451522 0.047091
0.2631578947368421 0.013850415512465374 0.038781163434903045 0.0221606648199446 0.0110803324099723 0.0 0.030470914127423823 0.036011081
0.2742382271468144 0.008310249307479225 0.0332409972299169 0.0110803324099723 0.024930747922437674 0.0 0.01662049861495845 0.0526315781
0.275 0.015 0.0275 0.0225 0.0175 0.0025 0.0275 0.0375 0.0425 0.0175 0.0075 0.0575 0.005 0.0025 0.0075 0.1025 0.05 0.005 0.005 0.0025 0
0.285 0.015 0.045 0.0125 0.01 0.0 0.015 0.045 0.0375 0.015 0.005 0.0525 0.0 0.005 0.005 0.105 0.03 0.0025 0.0025 0.0025 0.02 0.0025 0
0.25 0.01 0.0225 0.02 0.01 0.0 0.0325 0.025 0.045 0.0175 0.01 0.07 0.0025 0.0 0.005 0.1275 0.06 0.01 0.0025 0.0075 0.04 0.0 0.05 0.095
0.245 0.0125 0.025 0.025 0.015 0.0 0.0425 0.0375 0.045 0.03 0.005 0.0775 0.0025 0.0 0.0075 0.07 0.05 0.0 0.0025 0.0025 0.03 0.0025 0.0
0.2475 0.0075 0.0075 0.0075 0.01 0.0 0.02 0.04 0.0675 0.025 0.005 0.07 0.0075 0.0025 0.0025 0.1475 0.0475 0.0025 0.0025 0.0025 0.02 0.0
0.27 0.01 0.0325 0.0375 0.0125 0.0 0.025 0.045 0.03 0.03 0.005 0.0675 0.0025 0.0 0.0075 0.1525 0.065 0.0 0.0025 0.0 0.035 0.0 0.03 0.0
0.27977839335180055 0.0 0.01662049861495845 0.01662049861495845 0.01662049861495845 0.0 0.0221606648199446 0.030470914127423823 0.0554
0.2725 0.005 0.03 0.015 0.01 0.0025 0.025 0.035 0.0425 0.0275 0.0 0.0725 0.0075 0.0025 0.0025 0.105 0.0475 0.0025 0.0025 0.0 0.045 0.0 0
0.27146814404432135 0.008310249307479225 0.05263157894736842 0.01662049861495845 0.008310249307479225 0.0 0.013850415512465374 0.03601
0.29916897506925205 0.0110803324099723 0.019390581717451522 0.008310249307479225 0.0110803324099723 0.0 0.036011080332409975 0.0470914
0.2853185595567867 0.002770083102493075 0.024930747922437674 0.030470914127423823 0.00554016620498615 0.0 0.04155124653739612 0.022160
0.31301939058171746 0.00554016620498615 0.027700831024930747 0.0221606648199446 0.0221606648199446 0.0 0.036011080332409975 0.060941821
0.3047091412742382 0.0 0.04709141274238227 0.027700831024930747 0.013850415512465374 0.002770083102493075 0.0443213296398992 0.0249307
0.3047091412742382 0.002770083102493075 0.019390581717451522 0.008310249307479225 0.0110803324099723 0.0 0.019390581717451522 0.055401
0.31301939058171746 0.002770083102493075 0.019390581717451522 0.024930747922437674 0.008310249307479225 0.0 0.024930747922437674 0.0471
0.31024930747922436 0.008310249307479225 0.01662049861495845 0.019390581717451522 0.0110803324099723 0.0 0.036011080332409975 0.0692521
0.31024930747922436 0.00554016620498615 0.038781163434903045 0.01662049861495845 0.0221606648199446 0.0 0.038781163434903045 0.0387811
0.32409972299168976 0.008310249307479225 0.0221606648199446 0.019390581717451522 0.008310249307479225 0.0 0.0221606648199446 0.0470914
0.29916897506925205 0.0110803324099723 0.036011080332409975 0.01662049861495845 0.013850415512465374 0.002770083102493075 0.0193905817
0.296398891966759 0.0110803324099723 0.0443213296398992 0.01662049861495845 0.01662049861495845 0.0 0.0443213296398992 0.0443213296398
0.2770083102493075 0.008310249307479225 0.038781163434903045 0.019390581717451522 0.00554016620498615 0.0 0.027700831024930747 0.03878
0.25 0.0075 0.0175 0.01 0.02 0.0025 0.0275 0.05 0.08 0.0075 0.005 0.08 0.0025 0.0 0.01 0.1975 0.03 0.0025 0.0 0.005 0.0275 0.0025 0.05
0.2525 0.0025 0.015 0.0075 0.01 0.0075 0.035 0.0425 0.04 0.025 0.0125 0.115 0.005 0.0 0.005 0.1475 0.04 0.005 0.0 0.0025 0.0375 0.005 0
0.2575 0.01 0.035 0.025 0.0175 0.005 0.0525 0.0375 0.06 0.02 0.0 0.1075 0.01 0.0025 0.0 0.15 0.055 0.0025 0.0 0.005 0.0225 0.0025 0.03
0.2631578947368421 0.00554016620498615 0.019390581717451522 0.030470914127423823 0.0221606648199446 0.0 0.038781163434903045 0.0193905
0.2825 0.01 0.03 0.0175 0.0175 0.0 0.0325 0.035 0.06 0.035 0.005 0.1 0.0025 0.0025 0.01 0.1575 0.0575 0.0075 0.005 0.005 0.0325 0.0 0
0.33240997229916897 0.013850415512465374 0.06925207756232687 0.027700831024930747 0.0110803324099723 0.002770083102493075 0.0138504155
0.3185595567867036 0.008310249307479225 0.030470914127423823 0.019390581717451522 0.019390581717451522 0.0 0.030470914127423823 0.0526
0.3185595567867036 0.00554016620498615 0.036011080332409975 0.027700831024930747 0.008310249307479225 0.0 0.013850415512465374 0.03601
0.32132963988919666 0.019390581717451522 0.0332409972299169 0.0221606648199446 0.0110803324099723 0.0 0.036011080332409975 0.033240997
0.3185595567867036 0.008310249307479225 0.0221606648199446 0.01662049861495845 0.0221606648199446 0.00554016620498615 0.01662049861495
0.3157894736842105 0.002770083102493075 0.0221606648199446 0.01662049861495845 0.0110803324099723 0.00554016620498615 0.02216066481994
0.31024930747922436 0.008310249307479225 0.036011080332409975 0.0221606648199446 0.0 0.00554016620498615 0.024930747922437674 0.049861
0.3047091412742382 0.013850415512465374 0.04155124653739612 0.027700831024930747 0.008310249307479225 0.002770083102493075 0.027700831
0.29085872576177285 0.008310249307479225 0.036011080332409975 0.01662049861495845 0.027700831024930747 0.008310249307479225 0.02216066
0.3268698060941828 0.013850415512465374 0.038781163434903045 0.01662049861495845 0.019390581717451522 0.00554016620498615 0.0443213296
0.3 0.0175 0.0325 0.0075 0.0325 0.005 0.01 0.045 0.0675 0.0125 0.005 0.06 0.0025 0.0 0.015 0.245 0.09 0.005 0.0075 0.0125 0.0225 0.002
0.27977839335180055 0.008310249307479225 0.019390581717451522 0.01662049861495845 0.024930747922437674 0.002770083102493075 0.01939058
0.29085872576177285 0.019390581717451522 0.024930747922437674 0.0221606648199446 0.013850415512465374 0.0 0.01662049861495845 0.030470

```

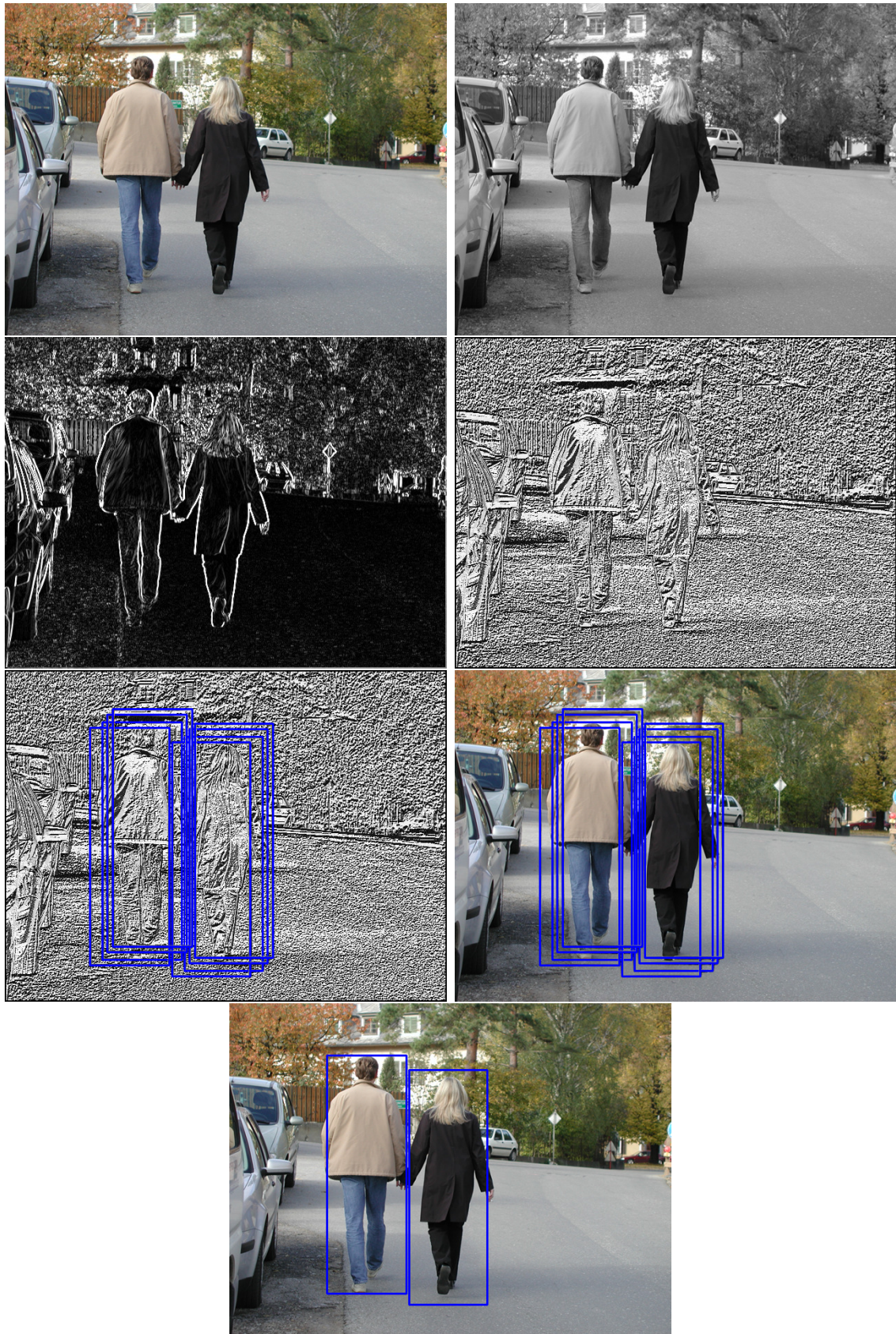
Obr. 3.4: Extrahované příznaky

4 ŘEŠENÉ PROBLÉMY

Na základě výše popsaného algoritmu byly vytvořeny a natrénovány detektory pro detekci postav a detekci příčných tepen v obraze. Oba detektory byly otestovány na testovacích datech. V podkapitolách uvedených níže jsou popsány tyto detektory a jejich úspěšnost.

4.1 Detekce postav

Celý proces začíná načtením vstupního obrazu, ve kterém se bude hledat lidská postava. Tento obraz je převeden do odstínu šedi. Na obraz v odstínech šedi je aplikována funkce pro převedení do integrálního obrazu pro zjednodušenou aplikaci dalších funkcí. V tomto okně jsou pomocí Sobelova operátoru určeny obrysové hrany člověka. Z nich jsou poté pomocí výpočtu určeny hodnoty Census transformace (CT). CT porovnává, zda jsou sousední pixely menší nebo větší a vytvoří bitovou posloupnost. Poté již následuje samotný proces detekce postav v obraze. Nejprve je nastavena velikost detekčního okna na 108×36 pixelů. Toto okno je rozděleno do bloků o velikosti 9×4 . Přilehlé bloky 2×2 se nazývají super-bloky. Celkově se v detekčním okně tedy nachází $8 \times 3 = 24$ super-bloků. V detekčním okně snímku je provedena aplikace dvou SVM klasifikátorů. Tyto klasifikátory porovnají uložené hodnoty ve vektoru vlastností s vypočítanými vlastnostmi obrazu. Na základě toho určí, zda-li se na obrázku nachází lidská postava. Oba dva klasifikátory a jejich hodnoty byly převzaty z původního programu v jazyce C++ [37]. Poté se oblast detekčního okna celá posune na další souřadnice a celý proces je opakován. To trvá do doby, dokud není pro daný rozměr detekčního okna prozkoumán celý obraz. V případě, že je nalezena oblast s potencionální postavou, jsou souřadnice této postavy uloženy do seznamu oblastí. Poté jsou rozměry detekčního okna zvětšeny a celý proces je opakován, dokud velikost detekčního okna nedosáhne velikosti vstupního obrázku. Výsledkem je poté seznam oblastí, ve kterých by se mohly postavy nacházet. Protože se tyto oblasti mohou překrývat nebo mírně lišit pro jednu a tu samou osobu, je provedeno ještě závěrečné zpracování těchto výsledků. Tím je odstraněno mnohonásobné označení oblastí, které nejsou lokálním maximem, nebo jsou překryté jinou oblastí. Takto zredukované označení se uloží do obrázku a ten se uloží a zobrazí. Na obrázcích 4.1 je graficky znázorněn celý proces detekce.



Obr. 4.1: Detekce postav

4.1.1 Nastavení detektoru

Pro co nejlepší výsledky je nutné přesně nastavit detektor. Detektor je nastaven na velikost detekčního okna 108x36 pixelů, tato oblast se dále dělí na pod-oblasti o velikosti 9x4 pixelů. Takto nastavené hodnoty se po celou dobu detekce nemění. Při takto nastaveném detektoru má rychlost zpracování detekce postav průměrnou rychlost 275,5 ms. Celková doba zpracování pro 100 snímků obsahující postavy a 100 snímků obsahující pozadí byla 55,11 sekund.

4.1.2 Výsledky detekce postav

Databáze snímků postav byly testovány následujícím způsobem. V databázi obsahující pozitivní snímky se spočítal počet detekovaných, nedetekovaných a falešně detekovaných postav. Z databáze obsahující negativní snímky se spočítal pouze počet falešně pozitivních detekcí. Z těchto hodnot se vypočítala přesnost a falešně pozitivní míra. Pro výpočet přesnosti se využilo vzorce 4.1, kde je detekční přesnost spočtena jako podíl mezi správně detekovanými osobami v obraze a celkovým počtem osob. Pro falešně pozitivní míru se využil vzorec 4.2, kde je výpočet proveden jako podíl mezi falešně detekovanými osobami a všemi osobami. Obdobně se přesnost a falešně pozitivní míra počítá i pro negativní snímky, zde se ale pro výpočet používá počet obrázků, viz vzorce 4.3 a 4.4. Tyto úspěšnosti byly porovnány s dalšími algoritmy detekce, které se dají v současné době použít. Výsledky jsou uvedeny v tabulkách 4.1 a 4.2. Hodnoty algoritmů v těchto tabulkách pro srovnání s metodou C4 poskytl vedoucí práce.

$$R_{\text{TP}} = \frac{N_{\text{TP}}}{N_{\text{celk}}} = \frac{248}{264} = 0,9393 = 93,94\%, \quad (4.1)$$

$$R_{\text{FP}} = \frac{N_{\text{FP}}}{N_{\text{celk}}} = \frac{31}{264} = 0,1174 = 11,74\%, \quad (4.2)$$

kde N_{TP} je počet správně detekovaných osob, N_{FP} je počet falešně detekovaných osob a N_{celk} je počet celkových osob v obraze.

$$R_{\text{TP}} = \frac{N_{\text{TPN}}}{N_{\text{celk}}} = \frac{92}{100} = 0,92 = 92\%, \quad (4.3)$$

$$R_{\text{FP}} = \frac{N_{\text{FPN}}}{N_{\text{celko}}} = \frac{8}{100} = 0,08 = 8\%, \quad (4.4)$$

kde N_{TPN} je počet správně nedetekovaných osob, N_{FPN} je počet falešně detekovaných osob a N_{celko} je celkový počet obrázků.

Tab. 4.1: Porovnání detekce jednotlivých algoritmů pro pozitivní snímky

	Viola-Jones	HOG	HOG+SVM	Metoda C4
Správně detekované	149 osob	234 osob	229 osob	248 osob
Nedetekované	115 osob	30 osob	35 osob	16 osob
Falešně detekované	45 detekcí	98 detekcí	18 detekcí	31 detekcí
Přesnost detekce	56,4%	88,6%	86,7%	93,94%
Falešně pozitivní míra	17%	37,1%	6,7%	11,7%

Tab. 4.2: Porovnání detekce jednotlivých algoritmů pro negativní snímky

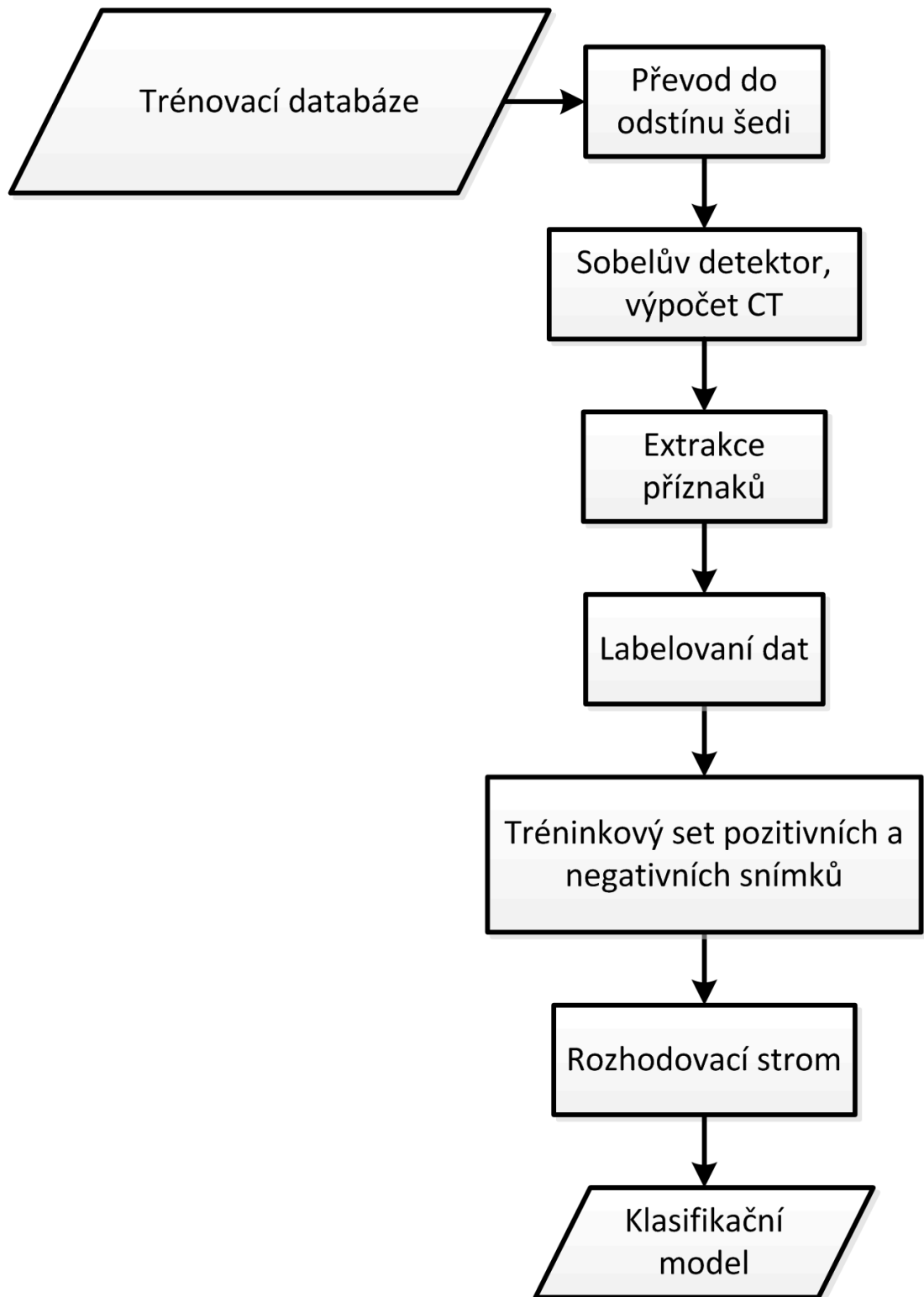
	Viola-Jones	HOG	HOG+SVM	Metoda C4
Správně nedetekované	86	74	96	92
Falešně detekované	25	51	7	8
Přesnost detekce	86%	74%	96,7%	92%
Falešně pozitivní míra	25%	51%	7%	8%

4.2 Detekce tepen

Pro detektor příčných tepen musel být nejprve vytvořen algoritmus. Tento algoritmus vzniknul modifikací metody C4. Celý proces vzniku algoritmu se skládá z několika částí zpracování.

4.2.1 Trénování detektoru

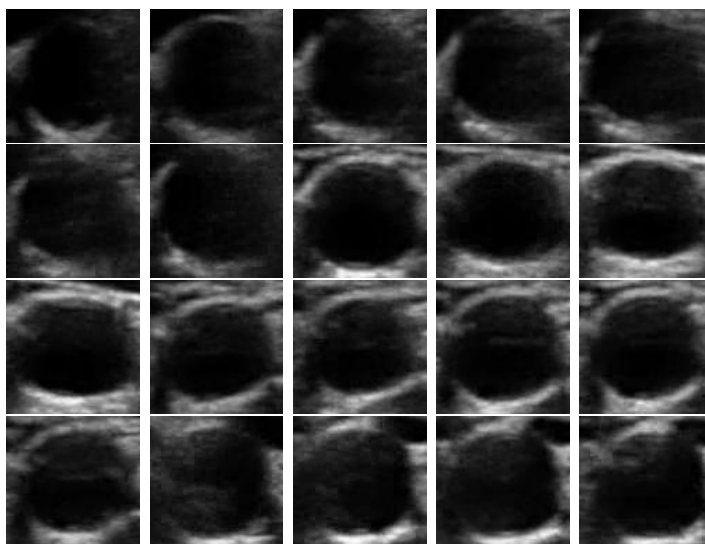
Tato kapitola se zabývá popisem metody natrénování vlastního klasifikátoru pro detekci příčných tepen. Jako hodnotící kritérium pro tento detektor byl použit rozhodovací strom. Na obrázku 4.2 je názorně ukázáno, jak tento celý proces probíhá. Začíná načtením trénovací databáze pozitivních a negativních snímků. Z těchto snímků se aplikací Sobelových filtrů vytvoří Sobelův obraz. Na takto přetvořeném obrazu se vypočítá hodnota Centrist. Takto vypočítané hodnoty se pomocí extraktoru příznaku uloží jako soustava číselných souřadnic. Těmto datům je nutné přiřadit label hodnotu pro určení pozitivního (obsahuje tepnu) či negativního (neobsahuje tepnu) snímku. Tím vznikne olabelovaná trénovací databáze. Tento trénovací set se použije jako vstupní data rozhodovacího stromu, který z těchto dat spočítá dělicí kritérium. Na základě zjištěného dělicího kritéria natrénuje klasifikační model. Model je poté použit jako hodnotící klasifikátor k určení, zda se v obraze nachází tepna či nikoliv.



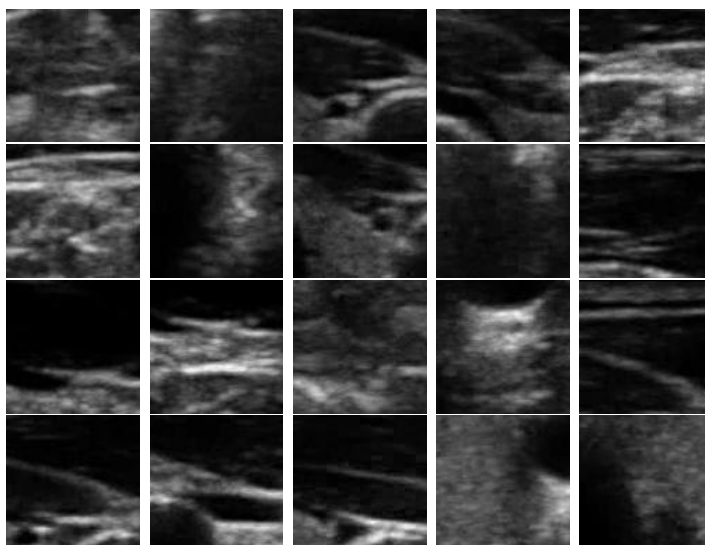
Obr. 4.2: Trénování modelu

Trénovací databáze

Pro natrénování vlastního klasifikátoru jsou nutné trénovací obrázky. K natrénování detektoru byla použita databáze Ultrasonix. Tato databáze obsahuje 284 pozitivních (obrázek 4.3) a 285 negativních snímků (obrázek 4.4). Celkem tedy 569 snímků určených k trénování. Velikost těchto snímků byla upravena na shodnou velikost 72x72pixelů. Takto převedené snímky byly načteny do extraktoru příznaků.



Obr. 4.3: Pozitivní trénovací snímky



Obr. 4.4: Negativní trénovací snímky

Extraktor příznaků

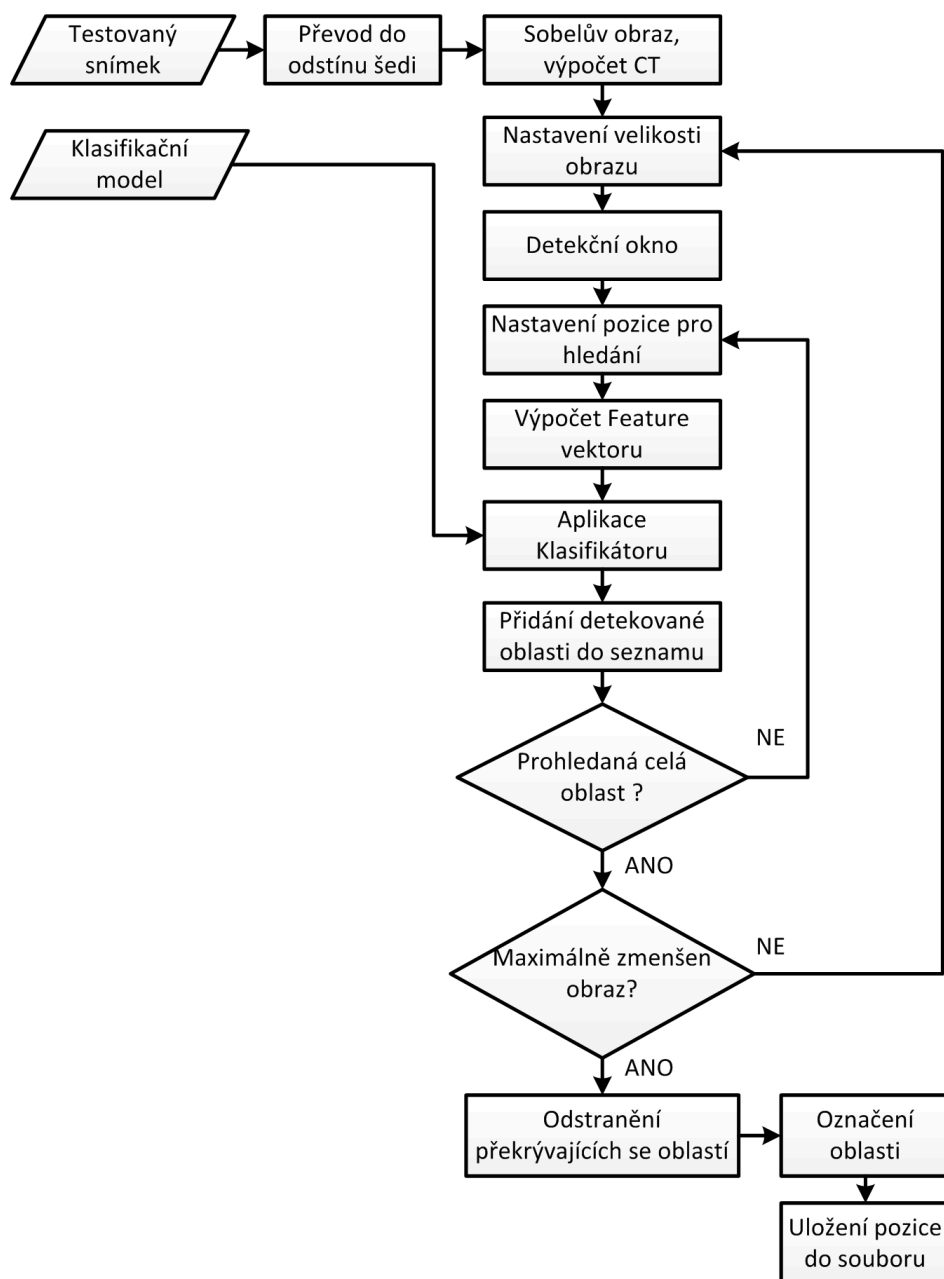
Úkolem extraktoru příznaků je získat informace ze vstupních snímků. Snímky z trénovací databáze jsou převedeny na Sobel obrázek. Na těchto snímcích je poté vypočítána hodnota CT. Takto vypočítané hodnoty se uloží a použijí se jako vstupní data pro klasifikační model.

Klasifikační model

Než se přistoupí k samotnému modelu, je nutné vyextrahovaným souřadnicím přidat label. Tento label určuje, zda se jedná o pozitivní (obsahující objekt), či negativní (neobsahující objekt) souřadnice. Výsledkem tohoto zpracování je trénovací databáze. Tato databáze je určena k natrénování klasifikačního modelu. Tyto hodnoty byly načteny do rozhodovacího stromu, který vznikl v rámci bakalářské práce na téma „Rozpoznávání emocí z textu pomocí umělé inteligence“ [34], kde se olabelovaná trénovací data rozdělí do dvou větví, pozitivní a negativní. Takto natrénovaný model je použit jako klasifikační vzor, podle kterého dochází k rozhodnutí, zda se ve snímku nachází objekt či nikoliv. Místo algoritmu rozhodovacího stromu lze použít i jiné rozhodovací algoritmy umělé inteligence, jakými jsou například: neuronové sítě, k-nejbližších sousedů, systémy podpůrných vektorů(SVM), atd.

4.2.2 Testování Detektoru

Navržený detektor bylo nutné otestovat. Celkový proces testování obsahuje několik částí. Tato kapitola popisuje, jak se tento test detekce provádí. Graficky je znázorněn na obrázku 4.5.



Obr. 4.5: Testování detektoru

Testovací data

Pro otestování detektoru byla použita testovací data ze dvou databází příčných tepen. Databáze z přístroje Ultrasonix obsahující 538 testovacích snímků a databáze z přístroje Toshiba obsahující 344 snímků. Tyto snímky mají různou velikost a kvalitu. Obsahují různě velké a tvarově rozdílné tepny. Proto je nutné uzpůsobit nastavení detektoru dané databázi. Ke každému snímku je připojen i textový soubor obsahující pozici tepny v obraze, pro případné porovnání výsledné detekce s původními souřadnicemi.

Nastavení detektoru

Pro správnou funkci detektoru je nutné nastavit jeho parametry. Detektor se skládá ze čtyř čísel. První dvě určují velikost detekčního okna. Toto okno se poté rozděljuje na pod-oblasti o velikost 5x5 pixelů. Třetí číslo určuje počet kroku, po kterých pod-oblast postupuje. Je to z důvodu, aby se pod-oblasti neposunovaly po všech pixelech, ale aby poskočily po n pixelech. Tím je zajištěna lepší rychlost detekce. Poslední číslo udává kolikrát se má daný snímek po průchodu algoritmem zmenšit.

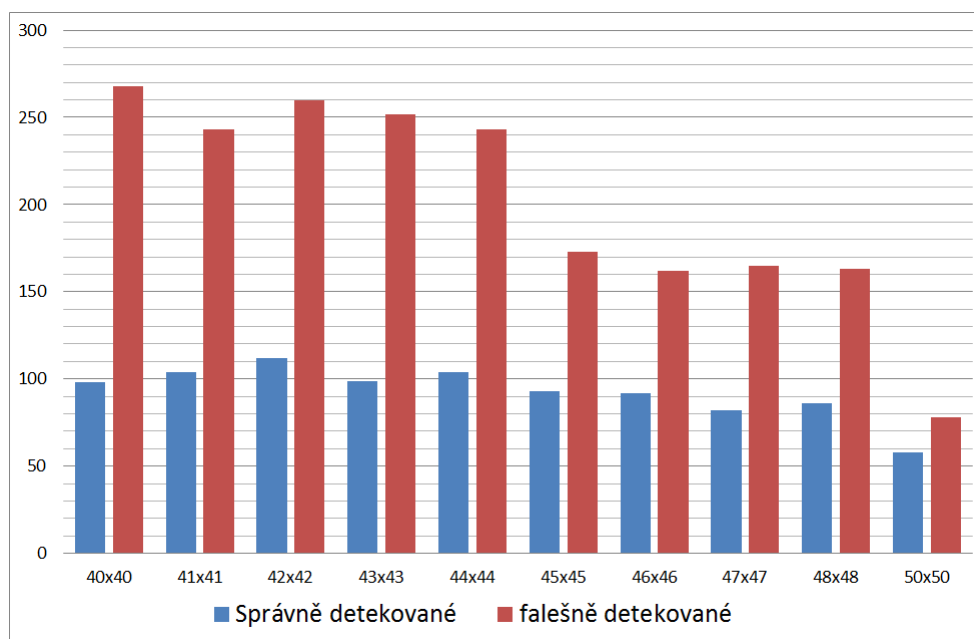
Nastavení detektoru bylo zjištěno experimentálně, aby byl zaručen nejlepší poměr přesnost/falešně pozitivní míra/čas detekce. Pro databázi Toshiba bylo provedeno celkově deset testování. Výsledky jsou uvedeny v tabulce 4.3. Stejně tomu bylo i pro databázi Ultrasonix, zde bylo celkem provedeno pět testování. Výsledky jsou uvedeny v tabulce 4.4. Z těchto tabulek se vytvořily grafy pro lepší přehlednost 4.6 a 4.7.

Tab. 4.3: Velikost detekčního okna pro databázi Toshiba

Testovací detekční okno	správně detekované	falešně detekované	čas detekce
40x40p	98	268	5,006s
41x41p	104	243	4,864s
42x42p	112	260	5,035s
43x43p	99	252	5,018s
44x44p	104	243	4,960s
45x45p	93	173	4,927s
46x46p	92	162	2,071s
47x47p	82	165	5,308s
48x48p	86	163	2,164s
50x50p	58	78	2,195

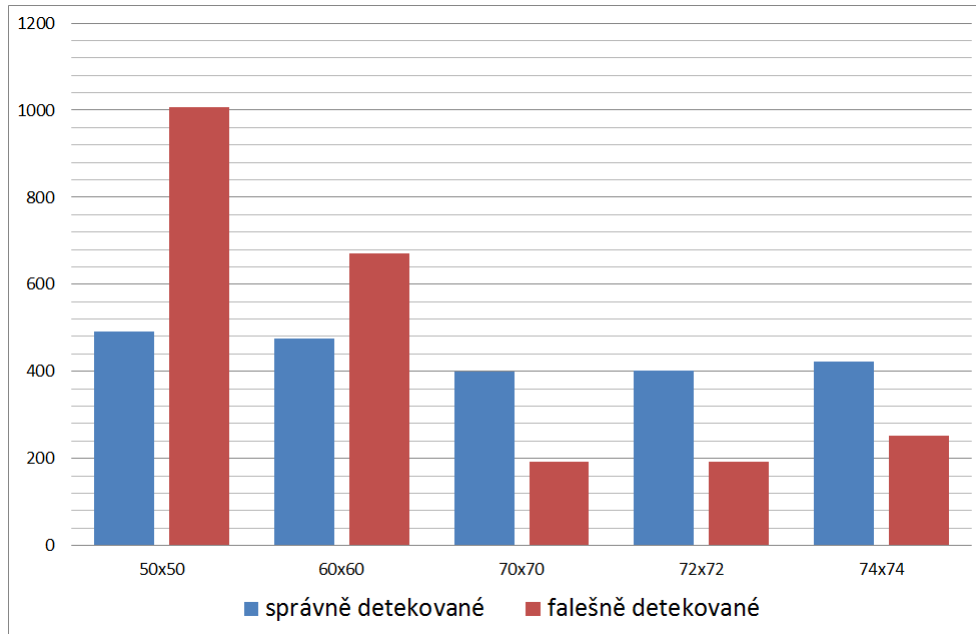
Tab. 4.4: Velikost detekčního okna pro databázi Ultrasonix

Testovací detekční okno	správně detekované	falešně detekované	čas detekce
50x50p	491	1007	5,411s
60x60p	475	671	5,247s
70x70p	399	192	5,184s
72x72p	402	192	4,792s
74x74p	422	252	5,653s



Obr. 4.6: Závislost detekce na velikosti detekčního okna

Pro databázi Ultrasonix byla po provedení několika detekčních testů zvolena velikost detekčního okna 72x72 pixelů. Při tomto nastavení se rychlost detekce měnila v závislosti na velikosti snímku. Průměrná rychlost detekce byla 4,792s na obrázek. U databáze Toshiba musela být velikost detekčního okna menší, jelikož databáze obsahuje obrázky s menším průměrem tepen. Po testování byla zvolena velikost 46x46 pixelů. Rychlost detekce byla v průměru 2,071s na obrázek.



Obr. 4.7: Závislost detekce na velikosti detekčního okna

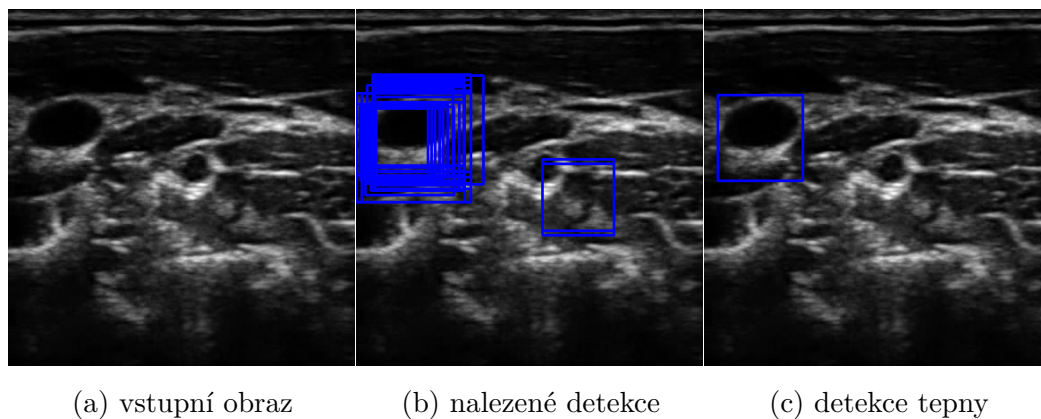
Detekce

Na začátku celého procesu je načten obrázek (obr.4.8a), ve kterém se bude hledat objekt. Jako další je načten klasifikační model. Snímek je poté převeden do odstínu šedi, z něj se pak přetvoří na Sobelův obraz a vypočítá se CT hodnota. Tím je obraz připraven. Dále se nastaví velikost detekčního okna a pozice tohoto okna ve snímku, což má za následek vzniknutí pod-okna. V tomto pod-okně se vypočítá f vektor a dle klasifikátoru se rozhodne, jestli se zde nachází hledaný objekt či nikoliv. V případě, že klasifikátor v oblasti zaznamená objekt, uloží jeho souřadnice do seznamu detekovaných oblastí. Detekční okno je poté ve snímku posunuto na další souřadnice, kde se proces opakuje, dokud takto není prohledán celý snímek. Velikost vstupního obrazu je poté zmenšena. Tento detekční proces se opakuje tak dlouho, dokud není velikost vstupního obrazu menší, než velikost detekčního okna. V takovém případě se detekce ukončí. Výsledkem tohoto procesu je seznam obsahující oblasti, kde se dle klasifikátoru nachází tepna (obr.4.8b).

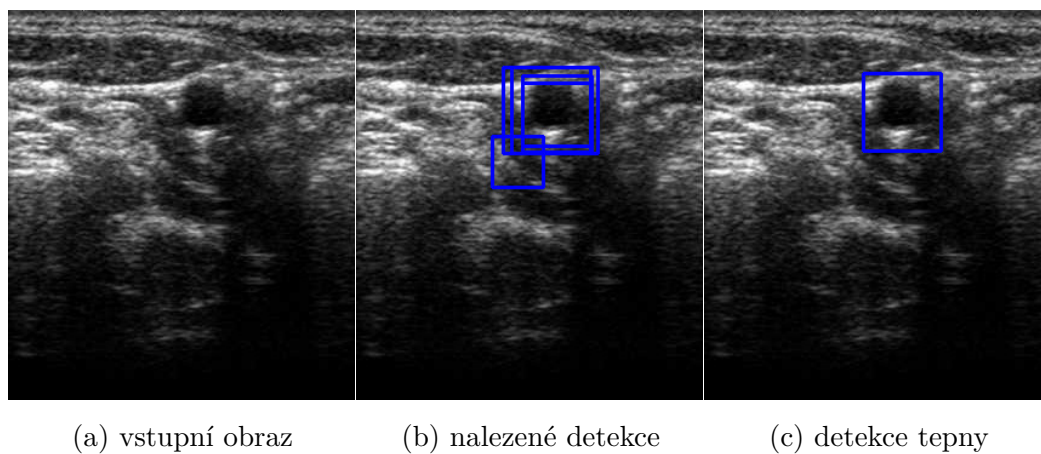
Zpracování detekce

Po procesu detekce vznikne seznam oblastí, kde by se mohly objekty vyskytovat. Jelikož je klasifikační model aplikován na snímek několikrát, dojde k tomu, že je určitý objekt označen v seznamu mnohonásobně. Aby se ve výsledném zobrazení tyto označení nenacházely, je nutné na tento seznam aplikovat funkci sloučení. Sloučení je

provedeno vypočtením průměrné hodnoty dané oblasti. Tato oblast je poté uložena do snímku a zobrazena modrým čtvercem (obr.4.8c). Na obrázcích 4.9a,4.9b a 4.9c je ukázána detekce tepen z přístroje Toshiba.



Obr. 4.8: Detekce databáze Ultrasonix



Obr. 4.9: Detekce databáze Toshiba

Výsledky detektoru tepen

Pro vytvořený algoritmus bylo taktéž nutné zjistit jeho procentuální přesnost. Testování se provádělo pro dvě databáze tepen: Databázi Toshiba a databázi Ultrasonix. Databáze Toshiba obsahovala 433 a databáze Ultrasonix 538 snímků. Stejně jako u algoritmu postav se i zde spočítal počet správně detekovaných, nedetekovaných a falešně detekovaných tepen pro obě databáze. Pro určení procentuální přesnosti byl použit vzorec 4.5. Tato hodnota je vypočítána jako poměr mezi správně detekovanými tepnami a celkovým počtem tepen ve snímcích. Falešně pozitivní míra 4.6 se vypočítá jako podíl mezi falešně detekovanými tepnami a celkovým počtem tepen. Obdobně byly tyto výpočty provedeny i pro databázi Toshiba.

Pro databázi Ultrasonix se využilo naměřených hodnot pro velikost detekčního okna 72x72 pixelů.

$$R_{\text{TP}} = \frac{N_{\text{TP}}}{N_{\text{celk}}} = \frac{402}{538} = 0,7472 = 74,72\%, \quad (4.5)$$

$$R_{\text{FP}} = \frac{N_{\text{FP}}}{N_{\text{celk}}} = \frac{192}{538} = 0,3568 = 35,68\%, \quad (4.6)$$

Pro databázi Toshiba se použily naměřené hodnoty pro detekční okno o velikosti 46x46 pixelů.

$$R_{\text{TP}} = \frac{N_{\text{TP}}}{N_{\text{celk}}} = \frac{92}{433} = 0,2124 = 21,24\%, \quad (4.7)$$

$$R_{\text{FP}} = \frac{N_{\text{FP}}}{N_{\text{celk}}} = \frac{162}{433} = 0,3741 = 37,41\%, \quad (4.8)$$

kde N_{TP} je počet správně detekovaných tepen, N_{FP} je počet falešně detekovaných tepen a N_{celk} je celkový počet tepen v obrazech.

5 ZHODNOCENÍ VÝSLEDKŮ

Pro zjištění účinnosti vytvořených algoritmů bylo nutné určit jejich procentuální přesnost a jejich chybovost. Výsledky byly zjišťovány na dvou-jádrovém procesoru Intel Core i5-3337U s frekvencí 1.8Ghz.

5.1 Zhodnocení výsledků detekce osob

Účelem těchto výpočtů bylo zjištění procentuální úspěšnosti algoritmu pro detekci postav. Úspěšnosti byly porovnány s jinými metodami detekce. Z výše umístěných tabulek je patrné, že vytvořená metoda C4 má velice dobré výsledky v porovnání s ostatními metodami. Vytvořený algoritmus má relativně vysokou úspěšnost. Na pozitivních snímcích je přesnost detekce 93,94% a 11,7% míra falešně pozitivních detekcí. Na negativních snímcích byla přesnost 92% a 8% míra falešně pozitivních detekcí. Velikou výhodou oproti ostatním metodám je ta, že metoda C4 je velice rychlá a nenáročná pro využití CPU. Pro 200 snímků byla průměrná detekční rychlost 275,5 ms.

5.2 Zhodnocení výsledků detektoru tepen

Účelem bylo zjistit, jak je navržený detektor úspěšný. Výpočty byly prováděny pro obě databáze. U databáze Ultrasonix byla úspěšnost detekce 74,21% a 35,68% falešně pozitivní míra. Průměrná rychlost detekce ve snímku pro tento vybraný rozměr byla 4,792s. Pro databázi Toshiba byla vybrána velikost 46x46p. Úspěšnost byla vypočítána na 21,24% a 37,41% falešně pozitivní míra. Detekční rychlost byla v průměru 2,071 sekundy pro jeden snímek.

Pro zlepšení těchto rychlostí může být využito převodu na integrální obraz při předzpracování. Dalším zlepšením může být použití více klasifikačních modelů.

6 ZÁVĚR

Cílem této práce bylo prostudovat a otestovat obrazový detektor C4. Na základě získaných znalostí bylo dalším úkolem vytvoření extraktoru příznaků pro jazyk Java.

Přínosem této práce je vytvoření extraktoru příznaků, pomocí kterého je možné natrénovat si svůj vlastní model. Takovýto model vznikl a byl otestován na dvou databázích tepen. Model využívá jako hodnotící kritérium algoritmus rozhodovacího stromu. Namísto rozhodovacího stromu lze použít libovolný učící algoritmus umělé inteligence a natrénovat si tak vlastní detektor. Výsledné hodnoty vytvořeného algoritmu byly určovány pro dvě databáze příčných tepen. Databázi Toshiba a databázi Ultrasonix. U databáze Ultrasonix byly výpočty prováděny pro detekční okno 72x72 pixelů. Přesnost detekce byla 72,74% a 35% falešně pozitivní míra s průměrnou rychlostí 5,7s na obrázek. U databáze Toshiba se výpočty prováděly pro detekční okno 48x48 pixelů. Hodnota přesnosti byla 18,01% a falešně pozitivní míra byla 21,94%. Průměrná rychlost detekce byla 4,7s na obrázek. Dalším přínosem je vytvoření algoritmu pro detekci lidských postav v obraze. Program využívá real-time metodu C4 a je vytvořen v jazyce Java. Ke své realizaci využívá knihovnu OpenCV. Pomocí výpočtů pro přesnost a falešnou míru pozitivních detekcí byla na vytvořeném programu zjištěna jeho procentuální úspěšnost. Ta byla porovnána s jinými druhy současných metod detekcí. Metoda C4 má velice vysokou úspěšnost. U pozitivních snímků je přesnost detekce 93,9% a 11,7% míra falešně pozitivních detekcí. U negativních snímků je přesnost 92% a 8% míra falešně pozitivní míra detekce. Rychlost detekce byla 275,5ms na snímek. Velikou výhodou této metody je to, že je oproti ostatním metodám velice rychlá a nenáročná na využití CPU. Veškeré testování bylo prováděno na dvou-jádrovém procesoru Intel Core i5-3337U s frekvencí 1.8Ghz.

Na tuto práci lze do budoucna navázat například vytvořením lepšího klasifikačního modelu, při použití jiného algoritmu umělé inteligence, pro zlepšení rychlosti a účinnosti detekce tepen.

LITERATURA

- [1] ANTONINI, M.; BARLAUD, M.; MATHIEU, P.; DAUBECHIES, I. *Image coding using wavelet transform*, Image Processing, IEEE Transactions on, vol.1, no.2, pp.205,220, Apr 1992 DOI: 10.1109/83.136597 Dostupné z URL:<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=136597&isnumber=3730>>
- [2] BENEŠ, R.; BURGET, R.; KARÁSEK, J.; ŘÍHA, K. *Automatically designed machine vision system for the localization of CCA transverse section in ultrasound images*. COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE. 2013. 109(3). p. 92 – 103. ISSN 0169-2607.
- [3] CONDE, C.; MOCTEZUMA,D.; MARTÍN DE DIEGO, I.; CABELLO, E. *HoGG: Gabor and HoG-based human detection for surveillance in non-controlled environments*. Neurocomputing. IEEE, 2013, vol. 100, issue 10, s. 19-30. DOI: 10.1016/j.neucom.2011.12.037. Dostupné z URL:<<http://linkinghub.elsevier.com/retrieve/pii/S092523121200313X>>
- [4] COVER, T.; HART, P. *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory. IEEE, 1967, vol. 13, issue 1, s. 21-27. DOI: 10.1109/TIT.1967.1053964. Dostupné z URL:<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1053964>>
- [5] CROW, Franklin C. *Summed-area tables for texture mapping*. Proceedings of the 11th annual conference on Computer graphics and interactive techniques - SIGGRAPH '84. New York, New York, USA: ACM Press, 1984, s. 207-212. DOI: 10.1145/800031.808600. Dostupné z URL:<[Dostupné z:http://portal.acm.org/citation.cfm?doid=800031.808600](http://portal.acm.org/citation.cfm?doid=800031.808600)>
- [6] DALAL, N.; TRIGGS, B. *Histograms of Oriented Gradients for Human Detection*. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, s. 886-893. DOI: 10.1109/CVPR.2005.177. Dostupné z URL:<<http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>>
- [7] EL-SAYED, MOHAMED A.; KHAFAGY, MOHAMED A. *An Identification System Using Eye Detection Based On Wavelets And Neural Networks* International Journal Of Computer And Information Technology. Vol. 1, No. 2, pp. 43-48, 2012. arXiv admin note: text overlap with arXiv:1205.5097 by other authors without attribution Dostupné z URL:<<http://arxiv.org/ftp/arxiv/papers/1401/1401.5108.pdf>>

- [8] ERDEM, C. E.; ULUKAYA, S.; KARAALI, A.; a ERDEM, A. T. *Combining Haar Feature and skin color based classifiers for face detection*. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2011, s. 1497-1500. DOI: 10.1109/ICASSP.2011.5946777. Dostupné z URL:<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5946777>>
- [9] ESDRAS, J.; GALEANO, P.; LILLO, R. E. *The Mahalanobis distance for functional data with applications to classification* Technometrics. 2014-09-10, s. 00-00. DOI: 10.1080/00401706.2014.902774. Dostupné z URL:<<http://www.tandfonline.com/doi/abs/10.1080/00401706.2014.902774>>
- [10] FAWCETT, T. *An introduction to ROC analysis*. Pattern Recognition Letters. 2006, vol. 27, issue 8, s. 861-874. DOI: 10.1016/j.patrec.2005.10.010. Dostupné z URL:<<http://linkinghub.elsevier.com/retrieve/pii/S016786550500303X>>
- [11] FEICHTINGER, H. G; STROHMER, T. *Gabor Analysis and Algorithms*, Birkhäuser, 1998; ISBN 0-8176-3959-4
- [12] FREIRE, D.; SANTANA, M.C.; DÉNIZ-SUARÉZ, O. *Smile Detection using Local Binary Patterns and Support Vector Machines*, in Proc. VISSAPP (1), 2009, pp.398-401. Dostupé z URL:<http://www.dis.ulpgc.es/contenido/investigacion/trabajos_publicados/VISAPP09.pdf>
- [13] FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. *Additive logistic regression: a statistical view of boosting*. Annals of Statistics 28(2), 2000. 337–407. Dostupé z URL:<<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.9525>>
- [14] HUANG, ZHIKAI; HOU, LINGYING; ZHANG, XINGWANG; LIAN, JING-JING *Using Implicit Shape Model and SVM for Face Detection*. International Journal of Advancements in Computing Technology. 2011, vol. 3, issue 11, s. 130-137. DOI: 10.4156/ijact.vol3.issue11.17. Dostupné z URL:<http://www.aicit.org/ijact/pp1/017_vol3.num11.pdf>
- [15] HUMANSCAN, *BioID-Technology Research*, 2009 Dostupné z URL:<<http://www.bioid.com/About/BioID-Face-Database>>

- [16] HWANG, Y.S.; KWAK, J.C.; LEE, K.Y. *Implementation of a Pedestrian Detection Device based on CENTRIST for an Embedded Environment*, *Advanced Science and Technology Letters Embedded Ubiquitous*, SERSC, vol. 46, (2014), pp. 123-127 Dostupné z URL:<http://onlinepresent.org/proceedings/vol46_2014/29.pdf>
- [17] CHUNG-MING CHEN,; YI HONG CHOU,; NORIO TAGAWA,; YOUNGHAEE DO, *Computer-Aided Detection and Diagnosis in Medical Imaging* *Computational and Mathematical Methods in Medicine*, vol.2013, Article ID 790608, 2 pages, 2013. doi:10.1155/2013/790608 Dostupné z URL:<<http://www.hindawi.com/journals/cmmm/2013/790608/>>
- [18] KRAUSZ, B.; BAUCKHAGE, C. *Automatic detection of dangerous motion behavior in human crowds* *Advanced Video and Signal-Based Surveillance (AVSS)*, 2011 8th IEEE International Conference on , vol., no., pp.224,229, Aug. 30 2011-Sept. 2 2011 Dostupné z URL:<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6027326&isnumber=6027273>>
- [19] LEIBE, B.; LEONARDIS, A.; SCHIELE, B. *An Implicit Shape Model for Combined Object Categorization and Segmentation*. *Pattern Recognition Letters*. 2006, vol. 27, issue 8, s. 508. Dostupné z URL:<http://link.springer.com/10.1007/11957959_26>
- [20] LIU, H.; MOTODA, H.; *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers 1998., Norwell, MA, USA.
- [21] Mäenpää, T.; Pietikäinen, M. *Texture analysis with local binary patterns*. 2005 Chen CH and Wang PSP (eds) *Handbook of Pattern Recognition and Computer Vision*, 3rd ed, World Scientific, 197-216 Dostupné z URL:<http://www.ee.oulu.fi/research/mvmp/mvg/files/pdf/pdf_524.pdf>
- [22] PANG, Y. Y.; LI, Y. X.; PAN, J. *Efficient HOG human detection*. *Signal Processing*. 2011, vol. 91, issue 4, s. 773-781. DOI: 10.1016/j.sigpro.2010.08.010. Dostupné z URL:<<http://linkinghub.elsevier.com/retrieve/pii/S0165168410003476>>
- [23] RYU, J. B.; PARK, H. H.; PARK, J. *Corner classification using Harris algorithm*. *Electronics Letters*. 2011, vol. 47, issue 9, s. 536-. DOI: 10.1049/el.2011.0594. Dostupné z URL:<<http://digital-library.theiet.org/content/journals/10.1049/el.2011.0594>>

- [24] ŘÍHA, K.; BENEŠ, R. *Circle Detection in Pulsative Medical Video Sequence*. In Proceedings of International Conference on Signal Processing, vol. I. Beijing, IEEE Press. 2010. p. 674 – 677. ISBN 978-1-4244-5898-1.
- [25] ŘÍHA, K.; CHEN, P.; FU, D. *Detection of Artery Section Area Using Artificial Immune System Algorithm Using Artificial Immune System Algorithm*. In Proceedings of The 7th WSEAS International Conference on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING. Puerto De La Cruz, Spain. 2008. p. 46 – 52. ISBN 978-960-474-035-2.
- [26] ŘÍHA, K.; MAŠEK, J.; BURGET, R.; BENEŠ, R.; ZÁVODNÁ, E. *Novel Method for Localization of Common Carotid Artery Transverse Section in Ultrasound Images Using Modified Viola-Jones Detector*. ULTRASOUND IN MEDICINE AND BIOLOGY. 2013. 39(10). p. 1887 – 1902. ISSN 0301-5629.
- [27] ŘÍHA, K.; POTÚČEK, I. *The Sequential Detection of Artery Sectional Area Using Optical Flow Technique*. In Proceedings of The 8th WSEAS International Conference on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING, WSEAS Applied Informatics & Communications. Puerto De La Cruz, Spain. 2009. p. 222 – 226. ISBN 978-960-474-139-7, ISSN 1790-5117.
- [28] SOBEL, I. *History and Definition of the Sobel Operator*, 2014,
- [29] STEINWART, I.; CHRISTMANN, A. *Support Vector Machines*, Springer, New York, 2008. 602 s. ISBN 978-0-387-77241-7
- [30] TOMASELLI, V.; GUARNERA, M.; MARCHISIO, C. D.; MORO, S.; BINGHAM, P. R.; LAM, E. Y. *Low complexity smile detection technique for mobile devices*. 866100-. DOI: 10.1117/12.2002449. Dostupné z URL:<<http://dx.doi.org/10.1117/12.2002449>>
- [31] VAIRALKAR, MANOJ K.; NIMBHORKAR, S.U. *Edge Detection of Images Using Sobel Operator*. International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459, Volume 2, Issue 1, January 2012 Dostupné z URL:<http://www.ijetae.com/files/Volume2Issue1/IJETAE_0112_51.pdf>
- [32] VIOLA, P.; JONES, M. *Rapid Object Detection using a Boosted Cascade of Simple Features*. [online], IEEE CVPR, 2001 Dostupné z URL:<<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>>

- [33] VIOLA, P.; JONES, M. *Robust Real-time Object Detection*, 2001. Vancouver, Canada.
- [34] VYLÍČIL, R.; *Rozpoznávání emocí z textu pomocí umělé inteligence: bakalářská práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 42 s. Vedoucí práce byl Ing. Jan Mašek
- [35] WASSERMAN, L. *All of Statistics: A Concise Course in Statistical Inference*. 2004 ISBN 0-387-40272-1.
- [36] WU, J; REHG, J.M. *CENTRIST: A Visual Descriptor for Scene Categorization*. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011, vol. 33, issue 8, s. 1489-1501. DOI: 10.1109/TPAMI.2010.224. Dostupné z URL:<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5674051>>
- [37] WU, J.; GEYER, C.; REHG, J.M. *Real-time human detection using contour cues* in Proc. ICRA, 2011, pp.860-867.
- [38] YAO, J.; ODOBEZ, J.-M.; HARWOOD D.; DAVIS, L. S.; *Fast human detection from joint appearance and foreground feature subset covariances*. Computer Vision and Image Understanding. 2011, vol. 115, issue 10, s. 1414-1426. DOI: 10.1016/j.cviu.2011.06.002. Dostupné z URL:<<http://linkinghub.elsevier.com/retrieve/pii/S1077314211001391>>
- [39] ZAKARIA, Z.; SUANDI, A. *Face detection using combination of Neural Network and Adaboost*. TENCON 2011 - 2011 IEEE Region 10 Conference. IEEE, 2011, s. 335-338. DOI: 10.1109/TENCON.2011.6129120. Dostupné z URL:<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6129120&isnumber=6128995>>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AdaBoost	Adaptive Boosting - klasifikační algoritmus dat
BaseLength	Základní velikost Histogramu
C4	Contour Cues - Klasifikační metoda využívající Obrys, Kaskádový klasifikátor a Centrist
CENTRIST	Census Transform Histogram - Census transformační histogram
CPU	Central Processing Unit - Centrální procesorová jednotka
CT	Census Transform - Census transformace
EXT	konstanta pixelu
f vektor	Feature vektor - vektor vlastností
G_x	konvoluční jádro pro horizontální směr
G_y	konvoluční jádro pro vertikální směr
HOG	Histogram of Oriented Gradients – histogramy orientovaných gradientů
i	nastavený řádek
ii	Integral image - integralní obraz
ISM	Implicit Shape Model – Implicitní tvarový model
j	nastavený sloupec
k-NN	k-Nearest Neighbours - algoritmus k-nejbližších sousedů
LBP	Local Binary Patterns – Lokální binární vzory
LogitBoost	Algoritmus strojového učení
M	celkový počet sloupců vstupního obrazu
N	celkový počet řádků vstupního obrazu
N_{celk}	celkový počet osob v obraze
N_{celko}	celkový počet obrázků
N_{FP}	počet falešně detekovaných osob

N_{FPN}	počet falešně detekovaných osob
N_{TP}	počet správně detekovaných osob
N_{TPN}	počet správně nedetekovaných osob
OpenCV	Open Source Computer Vision
R_{TP}	Přesnost
R_{FP}	míra falešně pozitivních detekcí
SVM	Support Vector Machine - algoritmus podpůrných vektorů
Toshiba	Databáze příčných tepen
Ultrasonix	Databáze příčných tepen
V-J	Viola Jones detektor
XDIV	velikost X-ové pod-oblasti detektoru
YDIV	velikost Y-ové pod-oblasti detektoru

SEZNAM PŘÍLOH

A	Obsah přiloženého CD	53
B	Ukázky detekce objektů	54
	B.1 Správně detekované objekty	54
	B.2 Falešně detekované objekty	55

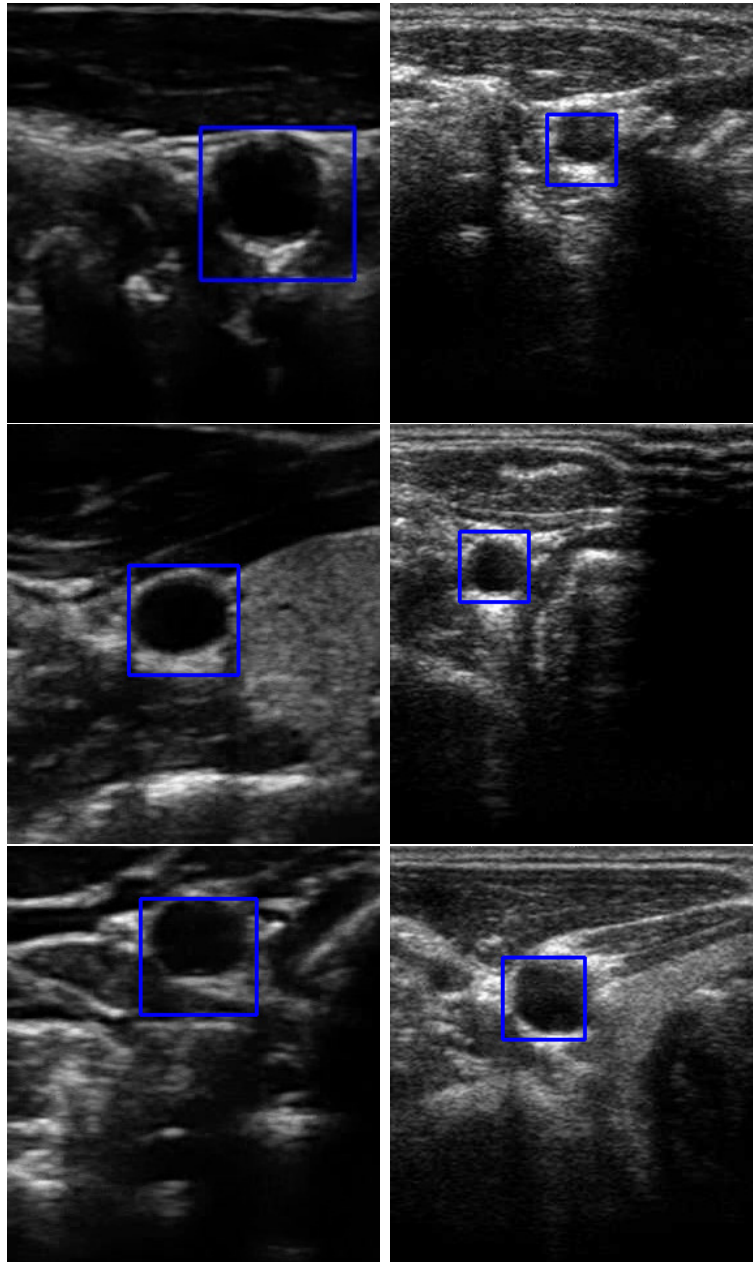
A OBSAH PŘILOŽENÉHO CD

- xvylic00-DP.pdf - Elektronická verze této práce ve formátu PDF
- Struktura adresáře DP-RadekVylicil.zip

```
|_ bin
|_ data
|   |_ ARTERY TRANSVERSAL - Trénovací a testovací snímky, Toshiba a Ultrasonix
|   |_ images - snímky pro detekci postav
|   |_ model - klasifikační modely
|   |_ results - Výsledné zobrazení označených snímků
|_ lib
|_ opencv - Knihovny opencv
|_ src
|   |_ cz
|       |_ vutbr
|           |_ feec
|               |_ utko
|                   |_ xvylic00 - Algoritmus detekce postav
|                   |   |_ Extractor - Algoritmus Extraktoru příznaků
|                   |   |_ Scanner - Algoritmus Navrženého detektoru tepen
|                   |_ xvylic00BP - Algoritmus rozhodovacího stromu k natrénování
|                       klasifikačního modelu
```

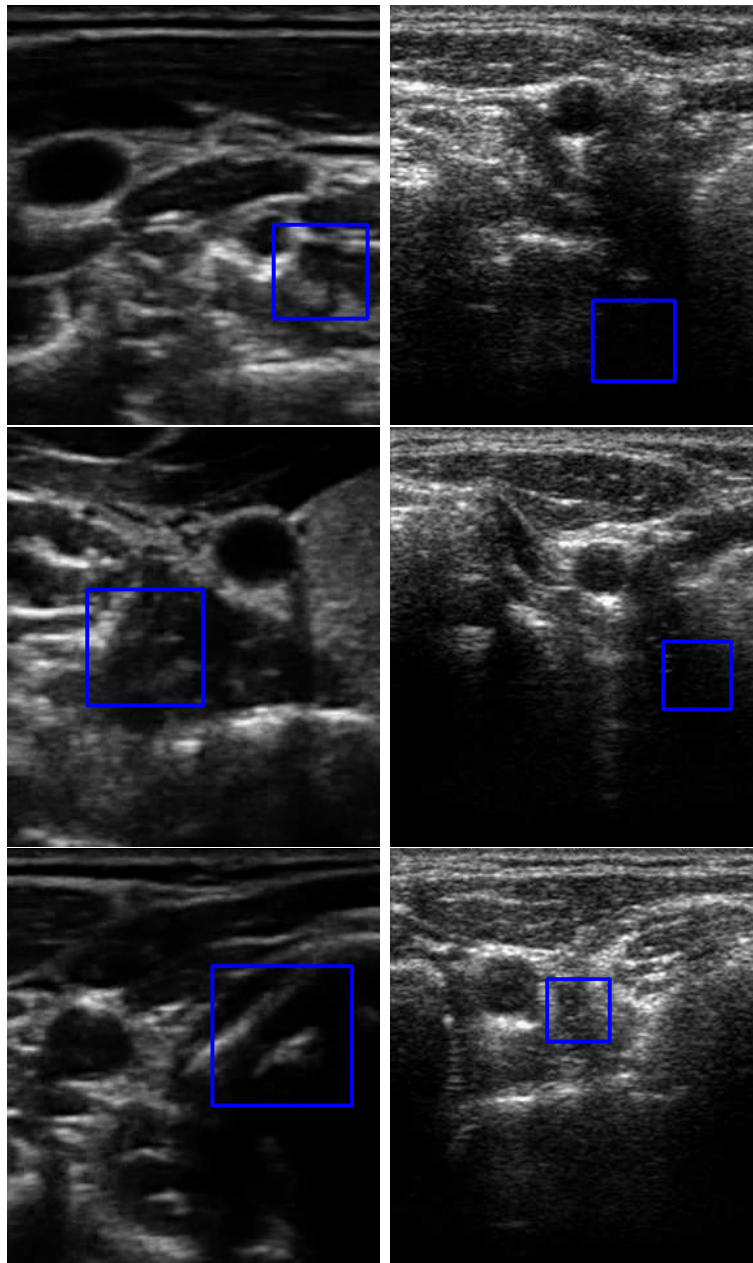
B UKÁZKY DETEKCE OBJEKTŮ

B.1 Správně detekované objekty



Obr. B.1: Správně detekované tepny

B.2 Falešně detekované objekty



Obr. B.2: Falešně detekované tepny