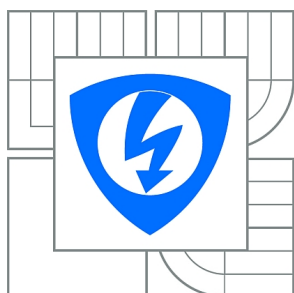




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE OBJEKTŮ V OBRAZE S POMOCÍ HAAROVÝCH PŘÍZNAKŮ

IMAGE OBJECT DETECTION USING HAAR-LIKE FEATURES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN MAŠEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADIM BURGET, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jan Mašek

ID: 106624

Ročník: 2

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Detekce objektů v obraze s pomocí Haarových příznaků

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodou Viola-Jones, Haarovými příznaky a algoritmem Ada-Boost. S jejich pomocí vytvořte v prostředí jazyka JAVA detektor objektů v obraze a ověřte jeho přesnost na vybraných příkladech ze zdravotnictví. Získané výsledky vynesete do grafů a zhodnotíte.

DOPORUČENÁ LITERATURA:

- [1] Paul Viola and Michael J. Jones. 2004. Robust Real-Time Face Detection. Int. J. Comput. Vision 57, 2 (May 2004)
- [2] D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall; 2 edition (November 7, 2011)

Termín zadání: 6.2.2012

Termín odevzdání: 24.5.2012

Vedoucí práce: Ing. Radim Burget, Ph.D.

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, 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 objektů v obraze s použitím Haarových příznaků a algoritmu AdaBoost. V textu jsou popsány metody trénování a testování objektového detektoru. Hlavní přínos práce spočívá ve vytvoření objektového detektoru v programovacím jazyce Java. Vytvořené algoritmy byly integrovány jako rozšíření programu RapidMiner, který patří celosvětově mezi nejznámější a nejpoužívanější programy pro dolování informací z báze dat. Dále byly vytvořeny návody pro použití vytvořených operátorů a několik vzorových příkladů spustitelných v programu RapidMiner. Funkčnost objektového detektoru byla ověřena na vybraných zdravotnických snímcích.

KLÍČOVÁ SLOVA

Zpracování obrazu, detekce objektů, Viola–Jones, Haarovy příznaky, AdaBoost, RapidMiner, Java

ABSTRACT

This thesis deals with the image object detection using Haar-like features and AdaBoost algorithm. The text describes methods how to train and test an object detector. The main contribution of this thesis consists in creation image object detector in Java programming language. Created algorithms were integrated as plugin into the RapidMiner tool, which is widely used and known worldwide as tool for data mining. The thesis contains the instructions for created operators and few examples for executing in RapidMiner tool. The functionality of image object detector was demonstrated on selected medical images.

KEYWORDS

Image processing, object detection, Viola–Jones, Haar-like features, AdaBoost, RapidMiner, Java

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.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Detekce objektů v obraze s pomocí Haarových příznaků“ 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)

Zde bych rád poděkoval vedoucímu diplomové práce Ing. Radimovi Burgetovi, Ph.D za pomoc, cenné rady, jeho trpělivost a čas při zpracování mé diplomové práce. Dále bych rád poděkoval Ing. Kamilovi Říhovi, Ph.D za pomoc při psaní diplomové práce.

V neposlední řadě děkuji své rodině za podporu během mého studia.

Výzkum popsáný 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.

OBSAH

Úvod	12
1 Současné metody detekce objektů v obraze	13
1.1 Detekce natočených objektů v obraze	13
1.2 Detekce zvířat	14
1.3 Metody trénování klasifikátorů a kaskády	14
2 Viola–Jones detektor	15
2.1 Haarovy příznaky	15
2.2 Integrální obraz	17
2.3 Lineární klasifikátor	18
2.4 Kaskáda klasifikátorů	19
2.5 AdaBoost	20
3 Trénování detektoru	21
3.1 Určení prahových hodnot klasifikátorů	21
3.1.1 Matthewsův korelační koeficient	21
3.2 Proces trénování	23
4 Implementace genetických algoritmů pro trénování detektoru	25
4.1 Evoluční skládání kaskády klasifikátorů	25
4.2 První evoluční funkce	27
4.3 Druhá evoluční funkce	27
4.4 Hodnoticí funkce	28
5 Testování detektoru	29
5.1 Proces detekce	29
5.2 Zpracování detekcí	29
5.3 Nastavení parametrů detektoru	31
6 Realizace operátorů v RapidMineru	32
6.1 Operátor pro načítání sady obrázků	32
6.2 Operátor pro vytváření trénovacích dat	32
6.3 Operátor pro trénování kaskády klasifikátorů	33
6.4 Operátor pro testování kaskády klasifikátorů	34
6.5 Operátor pro hodnocení kaskády klasifikátorů	34
6.6 Operátor pro načtení kaskády klasifikátorů	35
6.7 Operátor pro detekci objektů v obraze	35

6.8	Operátor pro úpravu kaskády klasifikátorů	36
6.9	Aplikace pro uložení pozice objektu do textového souboru	37
7	Použití operátorů v RapidMineru	39
7.1	Vytváření trénovacích dat	39
7.2	Trénování kaskády klasifikátorů	39
7.3	Trénování metodou křížové validace	40
7.4	Úprava natrénované kaskády klasifikátorů	41
7.5	Příklad použití objektového detektoru	42
8	Konkrétní řešení problémy a jejich zhodnocení	45
8.1	Detekce příčných tepen	45
8.1.1	Použité databáze snímků	46
8.1.2	Výsledky testování – příčné tepny	47
8.1.3	Vlastnosti zvolené kaskády klasifikátorů	50
8.2	Detekce podélných tepen	54
8.2.1	Použité databáze snímků	54
8.2.2	Výsledky testování – podélné tepny	55
8.3	Detekce čelistních kloubů	57
8.3.1	Použité databáze snímků	58
8.3.2	Výsledky testování – čelistní klouby	59
8.4	Zhodnocení výsledků	61
9	Závěr	62
	Literatura	63
	Seznam symbolů, veličin a zkratk	65
	Seznam příloh	67
A	Obsah CD	68
B	Ukázky detekovaných objektů	69
B.1	Správně detekované objekty	69
B.2	Falešně pozitivní detekce	72

SEZNAM OBRÁZKŮ

2.1	Hranové Haarovy příznaky.	15
2.2	Čárové Haarovy příznaky.	15
2.3	Středový Haarův příznak.	16
2.4	Schéma generování Haarových příznaků.	16
2.5	Haarův příznak aplikovaný na vstupní obraz.	16
2.6	Zobrazení hodnoty integrálního obrazu v bodě (u, v)	17
2.7	Ukázka výpočtu součtu pixelů v obdélníku v integrálním obraze. . . .	19
2.8	Schéma kaskádového zapojení klasifikátorů.	20
3.1	Schéma určení prahových hodnot klasifikátorů.	22
3.2	Schéma trénování kaskády klasifikátorů.	24
4.1	Schéma evolučního skládání kaskády klasifikátorů.	26
5.1	Schéma detekčního procesu.	30
6.1	Způsob zápisu kaskády klasifikátorů v textovém souboru.	33
6.2	Operátor pro trénování kaskády klasifikátorů a jeho parametry. . . .	34
6.3	Operátor pro detekci objektů v obraze a jeho parametry.	36
6.4	Okno aplikace.	38
7.1	Zapojení operátorů při generování trénovacích dat.	39
7.2	Ukázka vygenerovaných trénovacích snímků.	39
7.3	Příklad trénování kaskády klasifikátorů.	40
7.4	Příklad trénování kaskády klasifikátorů metodou křížové validace. . .	40
7.5	Ukázka nastavení vnitřních procesů operátoru pro křížovou validaci. .	41
7.6	Příklad zapojení operátorů pro modifikaci kaskády klasifikátorů. . . .	41
7.7	Příklad zapojení operátorů pro detekci a zobrazení objektů v obraze. .	42
7.8	Způsob vykreslení detekce pomocí obdélníků.	43
7.9	Způsob vykreslení detekce pomocí linek.	44
7.10	Ukázka detekce obličejů v RapidMineru.	44
8.1	Ukázka detekce příčné tepny z přístroje Ultrasonix.	45
8.2	Ukázka detekce příčné tepny z přístroje Toshiba.	46
8.3	Ukázka trénovacích snímků příčných řezů tepen.	47
8.4	Počty zamítnutých pod-oken v jednotlivých stupních kaskády.	51
8.5	Počty propuštěných pod-oken v jednotlivých stupních kaskády. . . .	52
8.6	Ukázka několika prvních vybraných příznaků v příčném řezu tepny. .	52
8.7	Graf závislosti průměru tepny na čase – příčný řez.	53
8.8	Ukázka detekce podélného řezu tepny.	54
8.9	Ukázka trénovacích snímků podélných řezů tepen.	55
8.10	Ukázka několika vybraných příznaků ve snímku podélného řezu tepnou. .	56
8.11	Graf závislosti průměru tepny na čase – podélný řez.	57

8.12	Ukázka detekce čelistního kloubu.	58
8.13	Ukázka trénovacích snímků příčných řezů tepen.	59
8.14	Ukázka několika vybraných příznaků ve snímku čelistního kloubu. . .	60
B.1	Ukázka správně detekovaných příčných řezů tepen.	69
B.2	Ukázka správně detekovaných podélných řezů tepen.	70
B.3	Ukázka správně detekovaných čelistních kloubů.	71
B.4	Tepny příčné – falešně pozitivní detekce.	72
B.5	Tepny podélné – falešně pozitivní detekce.	72
B.6	Čelistní klouby – falešně pozitivní detekce.	72

SEZNAM TABULEK

2.1	Typy a počty Haarových příznaků pro rozměr okna 24x24 pixelů. . .	17
8.1	Přesnost detekce různě trénovaných kaskád klasifikátorů (bez omezení počtu výskytů tepen v obraze).	49
8.2	Přesnost detekce různě trénovaných kaskád klasifikátorů (omezené předpokladem výskytu pouze jedné tepny v obraze).	49
8.3	Přesnost detekce kaskády klasifikátorů (pro detekci podélných řezů tepen).	56
8.4	Přesnost detekce kaskády klasifikátorů (pro detekci čelistních kloubů). .	60

ÚVOD

Tato práce se zabývá detekováním objektů v obrazech. V dnešní době se metody detekování objektů využívají v celé řadě oblastí, kde se běžný člověk nejčastěji setká například s detekcí obličejů, nebo lidských postav v digitálních fotoaparátech nebo kamerách. Mezi další možnosti využití patří detekce objektů z medicínských snímků. Příkladem může být detekce tepny z ultrazvukového snímku, nebo detekce čelistního kloubu ze snímku magnetické rezonance.

Pro detekci objektů v obraze se používá detektor, který byl navržen pány Violou a Jonesem [20]. Detektor je relativně přesný a spolehlivý a byl navržen, aby umožňoval detekci objektů v obrazech v reálném čase. Trénování detektoru probíhá na pozitivních a negativní trénovacích snímcích.

V rámci této práce byl vytvořen Viola–Jones objektový detektor. Pro realizaci detektoru bylo třeba implementovat trénovací algoritmus AdaBoost a generátor Haarových příznaků. Všechny tyto metody a algoritmy byly vytvořeny v programovacím jazyce Java a byly integrovány jako operátory do programu RapidMiner, který patří mezi nejpoužívanější programy pro dolování informací z báze dat. Dále byly předpřipraveny trénovací databáze pro RapidMiner. Jedná se o trénovací databáze ultrazvukových snímků podélných a příčných řezů tepen a MRI snímků čelistních kloubů. S použitím navržených operátorů byly vytvořeny demonstrační příklady pro trénování a testování detektoru spustitelné v programu RapidMiner. Pro implementované operátory byly vytvořeny návody k použití.

Hlavním přínosem této práce je implementace genetických algoritmů pro trénování stávajícího Viola–Jones objektového detektoru. Metoda trénování kaskády klasifikátorů za pomoci genetických algoritmů je prvně publikována a implementována v rámci této práce. Tato metoda byla zrealizována pro trénování kaskády klasifikátorů na menším množství trénovacích dat (asi 500 snímků) tak, aby byla dosažena co nejlepší přesnost detekce. S využitím této metody se podařilo vytvořit detektor příčných řezů tepen (arterií) s přesností 97 %, dále detektor podélných řezů tepen s přesností 96 % a detektor čelistních kloubů s přesností 67 %. Tyto detektory jsou schopné na běžném počítači pracovat v reálném čase.

Zbytek této práce je členěn následovně: první kapitola popisuje nejnovější metody detekce objektů v obrazech. V druhé kapitole je uveden Viola–Jones detektor a popsány jeho části. Třetí kapitola popisuje metody trénování detektoru. Čtvrtá kapitola popisuje modifikaci klasického způsobu trénování genetickými algoritmy. Pátá kapitola popisuje testování detektoru. V šesté kapitole jsou popsány vytvořené operátory pro RapidMiner a v sedmé kapitole je ukázáno, jak se tyto operátory dají použít. Osmá kapitola popisuje dosažené výsledky.

1 SOUČASNÉ METODY DETEKCE OBJEKTŮ V OBRAZE

V současné době jsou stále vyvíjeny metody pro detekci objektů v obraze, při tom ty nejúspěšnější vychází z původního přístupu, který prvně publikovali Viola a Jones. [20] V následujících podkapitolách jsou zmíněny metody detekce natočených objektů v obraze, dále detekce zvířat v obraze a nové metody trénování klasifikátorů a metody sestavování kaskád klasifikátorů.

1.1 Detekce natočených objektů v obraze

Tato podkapitola se zabývá popisem metod pro detekci obličejů v obrazech. Protože běžné fotky obsahují obličeje nejen z pohledu zepředu, ale i obličeje natočené nebo zobrazené z boku, byl hledán detektor pro takováto zobrazení obličejů. Takový detektor musel být trénován zvlášť pro každou kategorii objektů, kterou měl detekovat. Jeho trénování probíhalo na snímcích obličejů, které byly natočené o určité úhly. Vzniklý detektor se skládá z tolika kaskád klasifikátorů, kolik kategorií objektů má detekovat.

V článku [9] je popsána metoda trénování detektoru pro detekci obličejů s různým natočením. Detektor je pak schopný odlišit od sebe obličeje při pohledu zepředu i obličeje natočené úplně doleva, mírně doleva, mírně doprava, zcela doprava. Dále detektor rozlišuje naklonění nebo otočení obličeje. Další změnou oproti původnímu přístupu Violy a Jonese je za prvé použití algoritmu Real AdaBoost k trénování a za druhé se jiným způsobem skládá kaskáda klasifikátorů: z prvního stupně kaskády jsou všechny klasifikátory vloženy do jednoho reprezentativního klasifikátoru, který je poté zařazen do druhého stupně. Tento postup se aplikuje stejným způsobem i na zbylé stupně kaskády.

V jednom z dalších článků [19] Violy a Jonese je popsána detekce natočených obličejů za použití nových Haarových příznaků pro trénování detektoru. Podle [24] zobrazuje výsledný detektor kromě detekovaného objektu i hodnotu úhlu, o kterou byl daný obličej natočen.

V [11] je řešena detekce natočených obličejů za použití evolučních algoritmů (EA). Cílem implementovaných evolučních algoritmů je snížit počty klasifikátorů, které jsou trénovány algoritmem AdaBoost. Výsledkem je tedy menší počet klasifikátorů v jednotlivých stupních kaskády a s tím související rychlejší funkce detektoru objektů.

1.2 Detekce zvířat

V současné době je detekce zvířat z obrazů stále problém. Následující text obsahuje přehled, co je možné v současnosti detekovat, a jaké metody jsou při tom použity.

Jedním z nových přístupů je detekce hlav zvířat. V článku [22] je popsán způsob detekce hlavy lva ve video snímcích. Pro detekci v reálném čase je použit klasický Viola–Jones detektor, který je natrénován na detekování hlavy lva. Součástí detektoru je i funkce, která sleduje již detekovaný objekt. Tím je snížena výpočtová náročnost detekce.

V [2] jsou popsány metody detekce hlav zvířat ve statických obrazech. Detektor je natrénován na detekci hlavy kočky, tygra, pandy a geparda. Pro trénování detektoru byla zvolena nová metoda pro vytváření příznaků. Jedná se o Haarovy příznaky založené na metodě histogramů orientovaných gradientů (HOOG – Haar of Oriented Gradients). Tato metoda se zaměřuje jak na detekci tvarů, tak i na detekci textury. Pro detekci textury byly použity Haarovy příznaky a pro detekci tvarů byly použity histogramy orientovaných gradientů (HOG – Histogram of Oriented Gradients) [3]. Nejlepší výsledky detekce byly dosaženy při spojení metod pro detekci tvarů i detekci textury (HOOG). O něco horší výsledky měla metoda s Haarovými příznaky pro detekci textury a nejméně přesné výsledky měla metoda pro detekci tvarů (HOG).

1.3 Metody trénování klasifikátorů a kaskády

Původní metody trénování klasifikátorů a kaskády, které publikovali Viola a Jones byly postupem času zlepšovány.

V [16] jsou popsány nové trénovací metody pro trénování lineárních asymetrických klasifikátorů (LAC – Linear Asymmetric Classifier) [23]. Algoritmy, které dosahují lepších výsledků při trénování než algoritmus AdaBoost, se nazývají Fisher-Boost a LACBoost.

Pro obrovské trénovací sady, které mohou čítat až sta tisíce trénovacích snímků, je vhodné použít metodu maticové struktury trénování (MSL – Matrix–Structural Learning) [18]. Detektor natrénovaný touto metodou je díky dostatečnému počtu trénovacích snímků velmi přesný a má nižší počet falešně pozitivních detekcí, než Viola–Jones detektor. Jednotlivé stupně kaskády jsou v případě užití této metody (MSL) trénovány vždy na falešně pozitivních a zároveň falešně negativních detekcích snímků z předcházejícího stupně kaskády.

V [13] je popsáno, jakým způsobem lze snížit počet generovaných Haarových příznaků pro trénování. Metoda zajišťuje generování pouze Haarových příznaků důležitých pro trénování. V článku je popsáno trénování detektoru, který by měl být schopen detekovat obličeje, auta a motorky.

2 VIOLA–JONES DETEKTOR

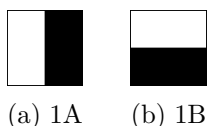
Detektor objektů v obraze, který byl prvně představen P. Violou a M. Jonesem v roce 2001, se vyznačuje především svou rychlostí a přesností detekce. Detektor pracuje s obrazy ve stupních šedi. Pro natrénování detektoru využívá klasifikační algoritmus AdaBoost, který skládá jednoduché klasifikátory do kaskády na základě vstupní množiny negativních a pozitivních vzorků (snímků). Jako jednoduchý klasifikátor jsou použity Haarovy příznaky s prahovou hodnotou. Pro rychlou funkci detektoru slouží integrální obraz a kaskáda klasifikátorů. [20] [14]

2.1 Haarovy příznaky

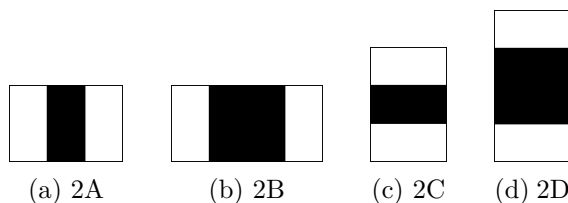
Na začátku trénovacího procesu jsou vygenerovány Haarovy příznaky. Tyto příznaky jsou odvozeny od několika obdélníků, které se dělí podle typu informace, která má být s jejich pomocí detekována na: hranové příznaky (viz obrázek 2.1), čárové příznaky, (viz obrázek 2.2) a na příznaky středové (viz obrázek 2.3). Bílá obdélníková oblast má stanovenou váhu $w_0 = -1$ a váha černé obdélníkové oblasti je vypočtena jako podíl ploch bílé a černé oblasti. Hodnota příznaku je rovna jeho odezvě na vstupní obraz a je vypočtena ze vztahu:

$$f(x) = w_0 r_0 + w_1 r_1, \quad (2.1)$$

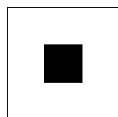
kde $f(x)$ je odezva Haarova příznaku na vstupní snímek x , w_0 je váha bílé obdélníkové oblasti r_0 , w_1 je váha černé obdélníkové oblasti r_1 . [12]



Obr. 2.1: Hranové Haarovy příznaky.

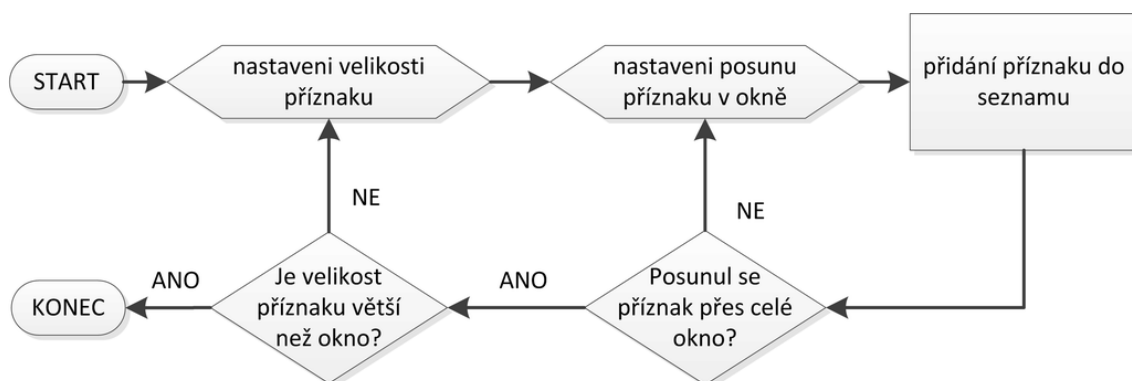


Obr. 2.2: Čárové Haarovy příznaky.

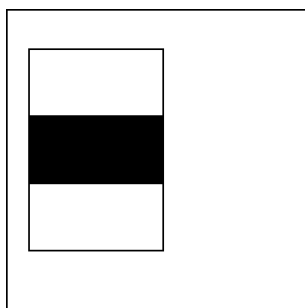


Obr. 2.3: Středový Haarův příznak.

Většinou jsou příznaky generovány pro velikost detekčního okna 24x24 pixelů. Při procesu generování příznaků jsou nejprve nastaveny nejmenší možné rozměry příznaku a poté je příznak iterativně posouván oknem vždy po jednom pixelu buď ve vertikálním, nebo v horizontálním směru. Při každém posuvu je příznak přidán do seznamu všech příznaků. Pokud je příznak posunut přes celé okno, je zvětšena jeho velikost a opět je posouván oknem. Tento proces se opakuje, dokud není velikost příznaku větší než velikost okna. Tento způsob generování je znázorněn na obrázku 2.4. Počty jednotlivých příznaků jsou vyobrazeny v tabulce 2.1. Celkový počet všech vygenerovaných příznaků je v tomto případě 189 664. Obrázek 2.5 zobrazuje Haarův příznak aplikovaný na vstupní obraz.



Obr. 2.4: Schéma generování Haarových příznaků.



Obr. 2.5: Haarův příznak aplikovaný na vstupní obraz.

Tab. 2.1: Typy a počty Haarových příznaků pro rozměr okna 24x24 pixelů.

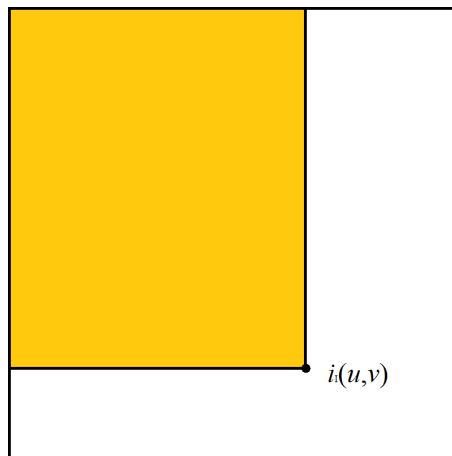
Typ příznaku	Počet příznaků
hranový – 1A	43 200
hranový – 1B	43 200
čárový – 2A	27 600
čárový – 2B	19 800
čárový – 2C	27 600
čárový – 2D	19 800
středový – 3A	8 464

2.2 Integrální obraz

Aby se při výpočtu odezvy Haarova příznaku na vstupní obraz nemusel pokaždé počítat součet hodnot pixelů v definovaných obdélnících, je tento vstupní obraz převeden na obraz integrální. Tím je dosažena mnohem vyšší rychlost výpočtu odezvy příznaku. Hodnota na souřadnicích $i_1(u, v)$ v integrálním obrazu, viz obrázek 2.6, se vypočítá dle vzorce [1]:

$$i_1(u, v) = \sum_{u' \leq u, v' \leq v} i(u', v'), \quad (2.2)$$

kde $i_1(u, v)$ jsou hodnoty integrálního obrazu a $i(u, v)$ jsou hodnoty pixelů vstupního obrazu. [21]



Obr. 2.6: Zobrazení hodnoty integrálního obrazu v bodě (u, v) .

Převod obrazu na integrální obraz lze realizovat podle následujících vzorců:

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

$$i_1(u, v) = i_1(u - 1, v) + s(u, v), \quad (2.4)$$

kde $s(u, v)$ je kumulovaný součet hodnot pixelů v řádku, za podmínek $s(u, -1) = 0$ a $i_1(-1, v) = 0$. Výpočet obdélníků v integrálním obrazu je pak jednoduchý. Na obrázku 2.7 jsou zobrazeny čtyři oblasti (A,B,C,D) a čtyři body. Hodnota v bodě 1 je rovna sumě hodnot pixelů, které jsou definovány oblastí A. Hodnota v bodě 2 je dána součtem oblastí A a B, hodnota v bodě 3 je součtem oblastí A a C a hodnota v bodě 4 je definována součtem oblastí A, B, C a D. Hledaná oblast D se vypočítá jako $4 - 3 - 2 + 1$. [21]

Dále je kromě integrálního obrazu také používán kvadrát hodnot integrálního obrazu. Takový obraz má označení i_{ISqr} a je vypočten podle předchozích dvou rovnic pouze s tím rozdílem, že místo $i(u, v)$ je použito $i^2(u, v)$. Protože každý snímek (okno o velikosti 24x24 pixelů) má jinou intenzitu osvětlení, je nutné odezvu Haarova příznaku na snímek normalizovat. Normalizace je provedena podle vzorce:

$$f(x)' = \frac{f(x)}{wh \sqrt{\frac{i_{\text{ISqr}}(w, h) - \frac{i_1^2(w, h)}{wh}}{wh}}}, \quad (2.5)$$

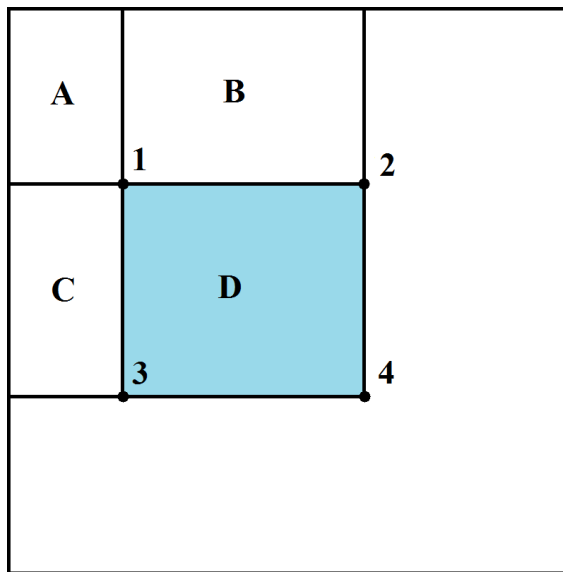
kde $f(x)$ je odezva Haarova příznaku na snímek x , $f(x)'$ je normalizovaná odezva, w je šířka detekčního okna, h je výška detekčního okna, $i_1(w, h)$ je suma hodnot pixelů vstupního snímku v detekčním okně o velikosti wh , $i_{\text{ISqr}}(w, h)$ je suma kvadrátů hodnot pixelů snímku v detekčním okně.

2.3 Lineární klasifikátor

Funkcí klasifikátoru je zařadit správně data do klasifikačních tříd (v našem případě do dvou – pozitivní či negativní) s úspěšností větší než 50 %. Lineární slabý klasifikátor se skládá z příznaku, polarity a prahové hodnoty (threshold). Z rovnice 2.6 je patrné, jakým způsobem klasifikátor stanovuje, do které třídy snímek x patří:

$$h(x, f, p, \Theta) = \begin{cases} 1 & pf(x) < p\Theta \\ 0 & \text{jinak,} \end{cases} \quad (2.6)$$

kde f je hodnota daného Haarova příznaku, p je polarita a Θ je prahová hodnota lineárního klasifikátoru. Při aplikování klasifikátoru na vstupní obraz x dochází nejprve k vypočítání odezvy příznaku, dále je podle polarity p určeno, jestli se pozitivní



Obr. 2.7: Ukázka výpočtu součtu pixelů v obdélníku v integrálním obraze.

klasifikační třída nachází nad prahovou hodnotou ($p = 1$) nebo pod prahovou hodnotou ($p = -1$). Nakonec je odezva příznaku porovnána s prahovou hodnotou a je rozhodnuto, zda snímek x patří do pozitivní $h(x) = 1$ nebo do negativní $h(x) = 0$ klasifikační třídy. [14] [4]

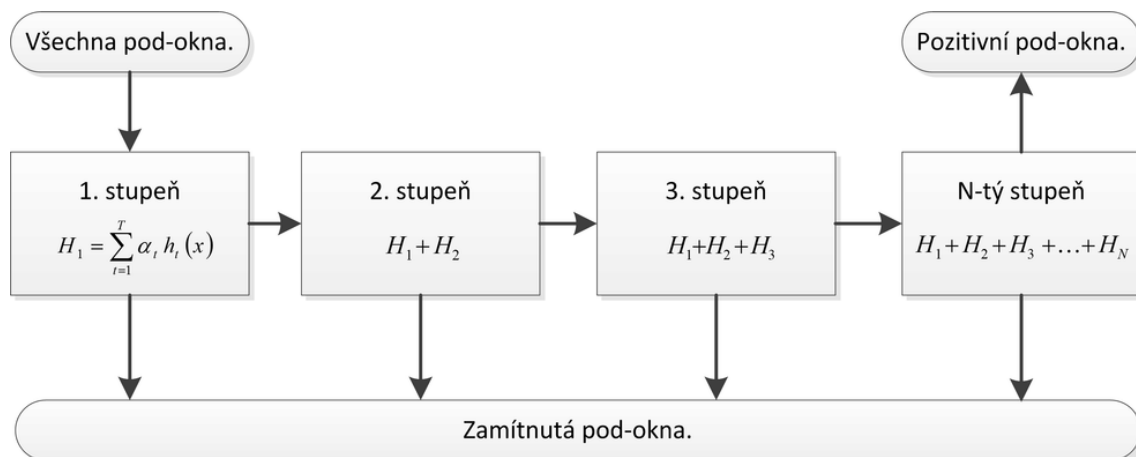
2.4 Kaskáda klasifikátorů

Kaskáda klasifikátorů je sestavena z určitého počtu stupňů, kde každý stupeň kaskády obsahuje určitý počet slabých klasifikátorů. Každý stupeň kaskády je vlastně silný monolitický nelineární klasifikátor $H(x)$, který má svou prahovou hodnotu P , podle které je rozhodnuto, zda aktuální pod-okno je pozitivním nebo negativním snímkem. Následující rovnice popisuje funkci silného klasifikátoru:

$$H(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq P \\ 0 & \text{jinak,} \end{cases} \quad (2.7)$$

kde α_t je váha slabého klasifikátoru h_t . Účelem kaskády je výrazně urychlit dobu detekce hledaného objektu. Při detekci objektů ve vstupním obraze je zpracováváno velké množství pod-oken. Každý stupeň kaskády je navržen tak, aby poslal k dalšímu zpracování pozitivní pod-okna na následující stupeň kaskády a zároveň, aby zamítl nějaké množství negativních pod-oken. Obrázek 2.8 zobrazuje zapojení silných klasifikátorů do kaskády. První stupeň kaskády by měl obsahovat jen několik klasifikátorů a měl by zamítnout co nejvíce pod-oken. Následující stupeň kaskády

vždy obsahuje všechny slabé klasifikátory ze stupně předcházejícího. Kaskáda by měla být natrénována tak, aby dosahovala vysoké úspěšnosti detekce a zároveň nízký počet falešně pozitivních detekcí. Trénování kaskády může probíhat podle metody Violy a Jonese. Každý stupeň je trénován tak, aby dosahoval určité procentuální hodnoty pozitivních detekcí i hodnoty falešně pozitivních detekcí. [21]



Obr. 2.8: Schéma kaskádového zapojení klasifikátorů.

2.5 AdaBoost

Základní metodou pro trénování kaskády klasifikátorů je klasifikační algoritmus Adaboost (Adaptive Boosting), který byl publikován v roce 1995 Yoavem Freundem a Robertem Schapirem. Účelem algoritmu je vybrat určité množství slabých lineárních klasifikátorů $h(x)$ z množiny všech klasifikátorů \mathbf{H} tak, aby jejich lineární kombinací vznikl silný nelineární klasifikátor $H(x)$, který reprezentuje stupeň kaskády. Takto vzniklých silných klasifikátorů bude více, v závislosti na počtu stupňů kaskády. Vstupem algoritmu jsou pozitivní a negativní trénovací data. V tomto případě tvoří trénovací data obrázky obsahující hledaný objekt (pozitivní snímky) a obrázky obsahující pozadí (negativní snímky). Funkce algoritmu AdaBoost bude podrobně vysvětlena v kapitole 3. [21] [5] [6]

3 TRÉNOVÁNÍ DETEKTORU

V této kapitole jsou popsány metody trénování kaskády klasifikátorů. Metody trénování vycházejí z přístupu Violy a Jonese. Trénovací proces tedy začíná načtením všech trénovacích dat. Dále jsou všechny snímky převedeny do formátu integrálního obrazu z důvodu výrazně rychlejší doby výpočtu odezvy Haarova příznaku na vstupní obraz. Další etapou je vygenerování všech Haarových příznaků pro velikost vyhledávacího okna 24x24 pixelů viz obrázek 2.4.

3.1 Určení prahových hodnot klasifikátorů

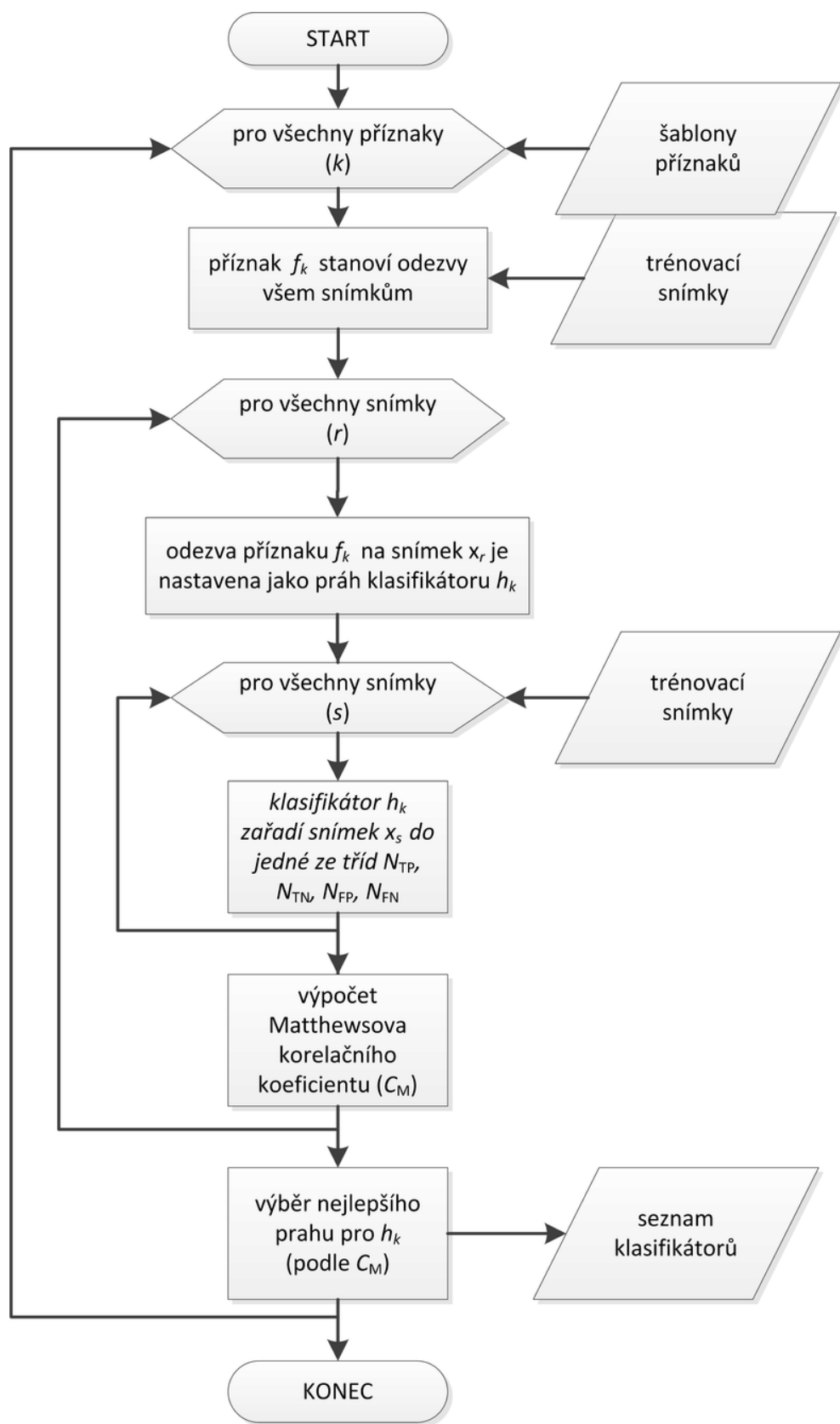
Proces, který má za úkol určit prahové hodnoty a polaritu všem klasifikátorům je výpočetně velmi náročný. Jeho výpočetní složitost je $O(kn^2)$, kde k je počet příznaků, což činí v tomto případě 189 664, a n je počet pozitivních a negativních trénovacích snímků. Pokud je zvoleno za $n = 500$, tak je celkový počet vykonaných operací během procesu tedy asi $4,7 \cdot 10^{10}$ a je zřejmé, že při použití většího množství trénovacích snímků se doba trénování kaskády klasifikátorů výrazně zvětší. Schéma procesu určení prahových hodnot klasifikátorů je patrné z obrázku 3.1. Během procesu se z každého Haarova příznaku, po určení prahové hodnoty a polaritu, stane lineární klasifikátor. Cílem procesu je nalézt takovou prahovou hodnotu a polaritu pro daný klasifikátor, aby byl tento klasifikátor schopný rozdělit trénovací snímky na pozitivní a negativní s co nejmenší chybou zařazení.

3.1.1 Matthewsův korelační koeficient

Pro určení, jak vhodně je zvolena prahová hodnota klasifikátoru, jsou všechny trénovací snímky klasifikovány do čtyř tříd: správně klasifikované pozitivní snímky (N_{TP}), správně klasifikované negativní snímky (N_{TN}), falešně pozitivní snímky (N_{FP}) a falešně negativní snímky (N_{FN}). Na základě těchto hodnot je vypočten Matthewsův korelační koeficient:

$$C_M = \frac{N_{TP} N_{TN} - N_{FP} N_{FN}}{\sqrt{(N_{TP} + N_{FP})(N_{TP} + N_{FN})(N_{TN} + N_{FP})(N_{TN} + N_{FN})}}. \quad (3.1)$$

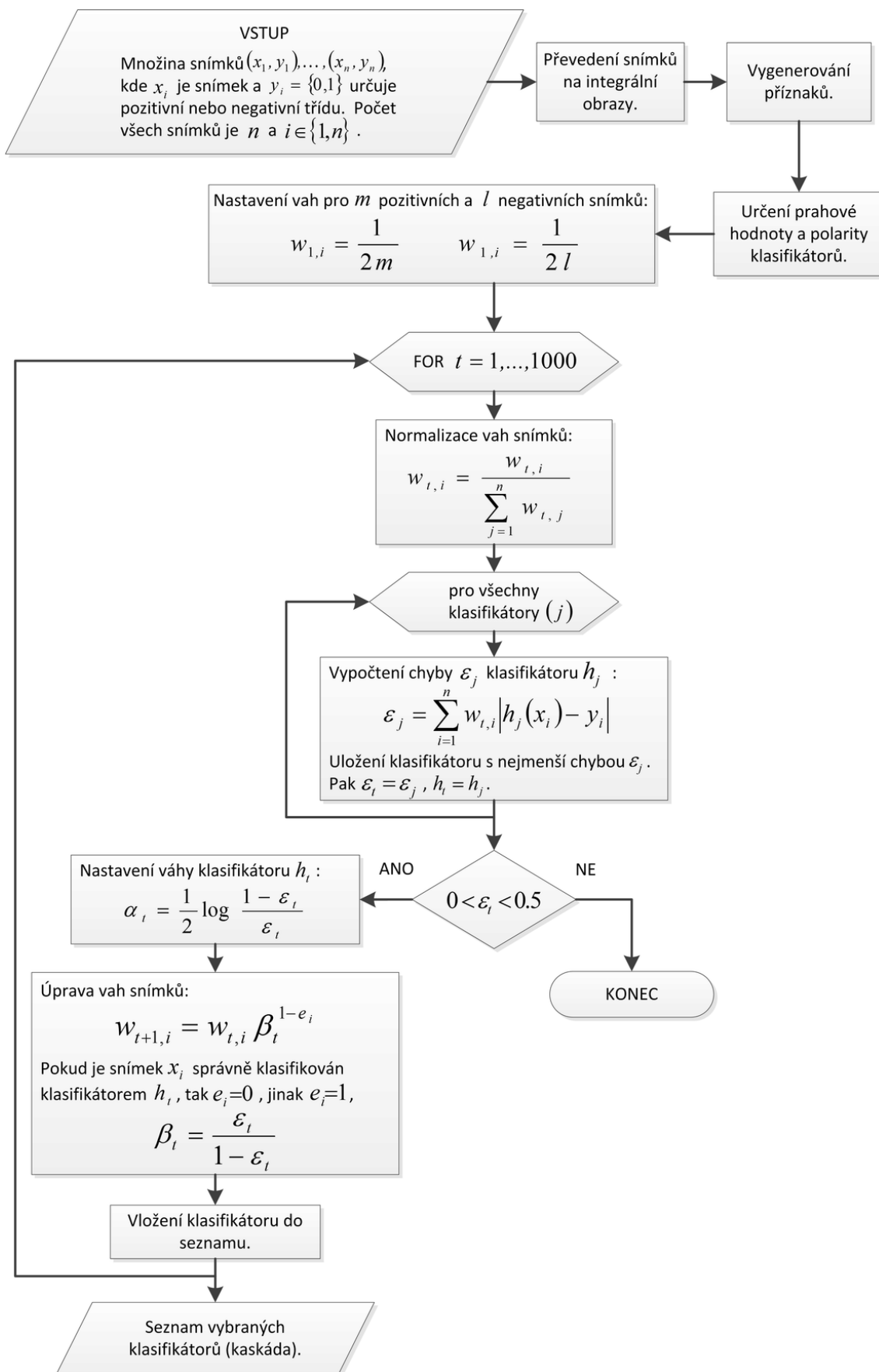
Prahová hodnota klasifikátoru s největší absolutní hodnotou C_M je určena jako nejlepší pro daný klasifikátor. Podle znaménka hodnoty C_M je určena polarita klasifikátoru. Po nalezení nejlepší prahové hodnoty a polaritu je klasifikátor uložen a celý cyklus se opakuje pro všechny klasifikátory.



Obr. 3.1: Schéma určení prahových hodnot klasifikátorů.

3.2 Proces trénování

Proces trénování kaskády algoritmem AdaBoost je znázorněn na obrázku 3.2. Nejprve jsou nastaveny startovní váhy pozitivním a negativním snímkům. Poté jsou tyto váhy normalizovány. Následně je vybrán jeden ze 189 664 klasifikátorů, který má na pozitivních a negativních snímcích nejvyšší přesnost klasifikace. Dále je zjištěno, zda chyba klasifikátoru ϵ_t splňuje podmínku $0 < \epsilon_t < 0,5$. Pokud není podmínka splněna, tak je celý proces trénování ukončen. Při splnění podmínky, jsou nastaveny vybranému lineárnímu klasifikátoru váhy α_t . Dále dochází k úpravě vah trénovacích pozitivních a negativních snímků, kde jsou váhy špatně klasifikovaných snímků zvětšeny. Váhy správně klasifikovaných snímků se nezmění. Nakonec je slabý klasifikátor vložen do seznamu vybraných klasifikátorů. Tento cyklus je opakován 1000 krát. Výsledkem trénování je tedy jeden monolitický nelineární klasifikátor obsahující 1000 slabých klasifikátorů. Jedná se tedy o kaskádu obsahující jeden stupeň.



Obr. 3.2: Schéma trénování kaskády klasifikátorů.

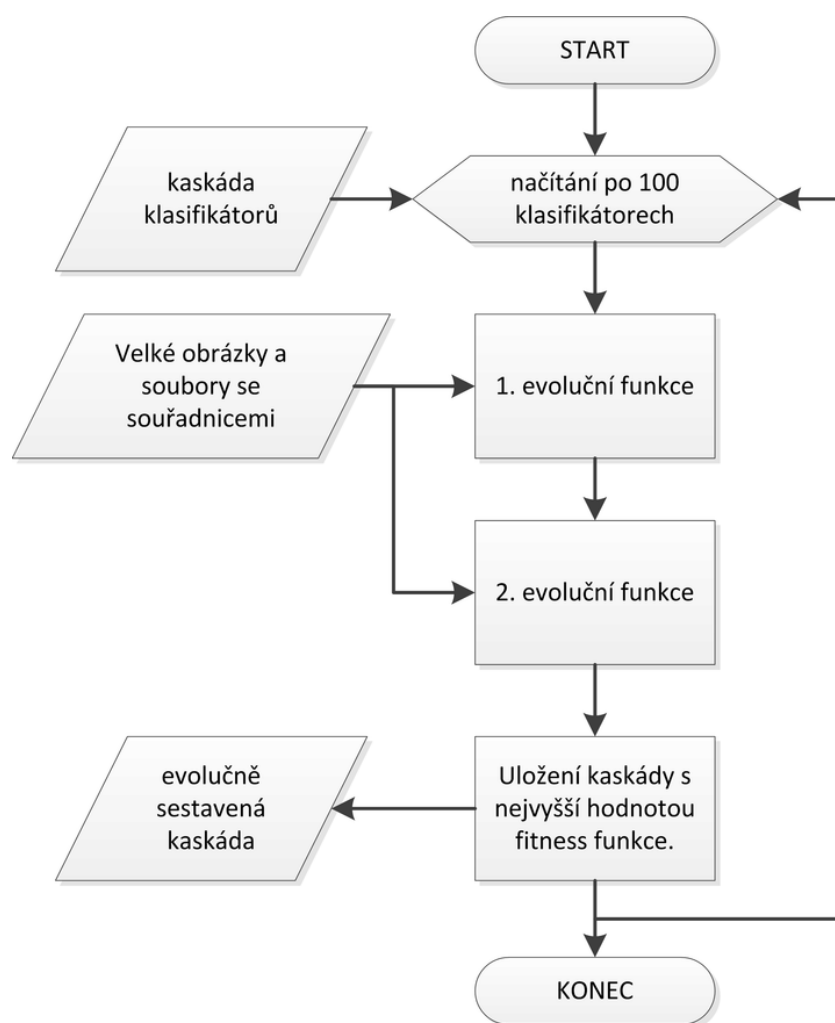
4 IMPLEMENTACE GENETICKÝCH ALGORITMŮ PRO TRÉNOVÁNÍ DETEKTORU

Původně byla kaskáda klasifikátorů trénována podle metody Violy a Jonese. Zřejmě kvůli nízkému počtu trénovacích snímků (asi 500) se trénování detektoru touto metodou neosvědčilo. Další možností je sestavit kaskádu klasifikátorů ručně. To znamená, že musí být vhodně zvoleny parametry: celkový počet klasifikátorů v kaskádě, počty klasifikátorů v jednotlivých stupních kaskády a prahové hodnoty jednotlivých stupňů kaskády. Určit tyto parametry je celkem složité a vyžaduje to určité zkušenosti. Při nízkém počtu vybraných klasifikátorů dochází k nárůstu počtu falešně pozitivních detekcí. Naopak při vysokém počtu vybraných klasifikátorů dochází k přetrénování kaskády a tím je snížena přesnost detekce. Proto byly pro trénování detektoru implementovány genetické algoritmy, které rozřazují klasifikátory do stupňů, aby měla kaskáda jako celek co nejlepší přesnost detekce na trénovacích velkých snímcích. Pro zrealizování funkcí byla použita knihovna JGAP¹ (Java Genetic Algorithms Package).

4.1 Evoluční skládání kaskády klasifikátorů

Proces skládání kaskády, viz obrázek 4.1, začíná načtením kaskády klasifikátorů, která byla trénována algoritmem AdaBoost. Dalšími vstupními soubory jsou velké trénovací snímky s příslušnými textovými soubory, které obsahují ručně nalezené souřadnice hledaných oblastí. Evoluční skládání kaskády probíhá iteračně po 100 klasifikátorech. To znamená, že v prvním cyklu bude kaskáda sestavena z celkem 100 slabých klasifikátorů, v druhém z 200 klasifikátorů a v posledním cyklu budou zpracovány všechny klasifikátory. V každém cyklu je tedy vytvořena nová kaskáda. Vždy je uložena kaskáda s nejvyšší hodnotou fitness funkce, která byla dosažena během evolučního trénování. Kaskáda s nejlepší úspěšností je uložena na disk a dále použita při testování. Nyní bude popsáno, co se děje v jednom cyklu evolučního skládání kaskády. K tomuto účelu byly navrženy dvě evoluční funkce. První evoluční funkce sestaví kaskádu a následně druhá funkce mění prahové hodnoty stupňů této kaskády. Tyto evoluční funkce využívají společnou hodnotící fitness funkci pro ohodnocení úspěšnosti trénované kaskády a jejich parametry jsou: 30 evolučních cyklů, velikost populace je 10 jedinců.

¹Dostupný z URL: <http://jgap.sourceforge.net/>



Obr. 4.1: Schéma evolučního skládání kaskády klasifikátorů.

4.2 První evoluční funkce

První evoluční funkce se snaží sestavit kaskádu tak, že zkouší měnit počty stupňů a počty klasifikátorů obsažené ve stupních. Před startem samotné evoluce musí být nastaveno několik parametrů: prvním je hodnota maximálního počtu stupňů kaskády a druhým je maximální počet klasifikátorů v jednom stupni. První parametr je vypočten podle vzorce:

$$N_{cs} = N_s = \sqrt[2]{\frac{3}{2}N_c}, \quad (4.1)$$

kde N_{cs} je maximální počet klasifikátorů v jednom stupni, N_s je maximální počet stupňů kaskády a proměnná N_c určuje počet všech lineárních klasifikátorů, které jsou určeny pro skládání kaskády. Dále je vytvořen chromozom, který obsahuje geny, jejichž počet je roven hodnotě parametru N_s . Hodnoty alel vytvořených genů jsou rovny hodnotě parametru N_{cs} . Pouze hodnota alely prvního genu (stupně) je odlišná. V našem případě je rovna číslu 3. Nyní následuje 30 evolučních cyklů, při kterých bude kaskáda skládána. V průběhu skládání se v kaskádě mění počty klasifikátorů v jednotlivých stupních, přičemž první stupeň kaskády může obsahovat pouze 1 až 3 klasifikátory. A to z důvodu, že na prvním stupni je odmítnuta většina negativních pod-oken, a kdyby první stupeň obsahoval více klasifikátorů, docházelo by ke snížení rychlosti detekce. Další stupně mohou obsahovat hodnoty od 0 až do maximálního počtu klasifikátorů N_c . Pokud po skončení evoluce některý ze stupňů neobsahuje žádný klasifikátor, tak je tento stupeň zrušen. Při evolučním sestavení kaskády musí být určeny prahové hodnoty jednotlivých stupňů kaskády. Prahová hodnota P stupně je:

$$P = \frac{1}{2} \sum_{t=1}^T \alpha_t, \quad (4.2)$$

kde α_t je váha lineárního klasifikátoru h_t .

4.3 Druhá evoluční funkce

Po skončení první evoluční funkce je natrénovaná kaskáda zpracovávána druhou evoluční funkcí. Účelem této operace je u dané kaskády měnit prahové hodnoty jednotlivých stupňů. Počet genů chromozomu je roven maximálnímu počtu stupňů kaskády. Hodnoty alel genů obsahují změnu (v %) původní prahové hodnoty P každého stupně kaskády. Změna prahové hodnoty stupně je pro všechny stupně $\pm 25\%$.

4.4 Hodnoticí funkce

Obě evoluční funkce používají stejnou hodnoticí (fitness) funkci, která ohodnocuje úspěšnost natrénované kaskády přímo na souboru velkých trénovacích obrazů. Ke každému snímku přísluší textový soubor, ve kterém jsou uloženy informace o umístění (souřadnice středu a rozměry oblasti) hledaných objektů. Daná kaskáda je otestována za pomoci objektového detektoru, který bude popsán v kapitole 5. Hlavním účelem detektoru je za pomoci natrénované kaskády klasifikátorů detekovat ve velkých snímcích objekty. Při testování úspěšnosti této detekce jsou souřadnice detekovaných objektů porovnány se souřadnicemi v příslušných textových souborech. Při srovnání souřadnic je uvažována jistá tolerance přesnosti v intervalu $\pm \frac{w}{2}$ pixelů, kde w je hodnota šířky oblasti v pixelech. Stejným způsobem lze určit i toleranci pro výšku oblasti (v pixelech). Tato tolerance přesnosti je nutná pro správné určení detekovaných objektů a je zavedena, protože detekovaný střed objektu může být trochu posunut oproti středu definovaném v textovém souboru. Hodnota této tolerance je relativně vysoká, protože při vytváření těchto souřadnic (obsažených v textových souborech) dochází k drobným odchylkám. Na základě tohoto srovnání souřadnic jsou pro každý obrázek určeny počty detekovaných objektů a počty falešně pozitivních detekcí. Podle počtu správných a špatných detekcí je hodnoticí fitness funkce C_F dána vztahem:

$$C_F = \frac{1}{aN_{FN} + bN_{FP} + 0,1}, \quad (4.3)$$

kde N_{FN} značí počet nedetekovaných tepen a N_{FP} značí počet falešně pozitivních detekcí tepen. Hodnoty koeficientů a a b určují váhy, které jsou vztaženy k proměnným N_{FN} a N_{FP} . Při trénování genetickými algoritmy byly natrénovány kaskády klasifikátorů, kde každá kaskáda měla pro trénování odlišně zvoleny koeficienty a a b . Podle výsledků testování lze říci, že čím je větší hodnota podílu $\frac{a}{b}$, tím má detektor větší přesnost detekce, ale zároveň narůstá celkový počet falešně pozitivních detekcí. Číslo 0,1 ve jmenovateli zlomku zaručuje, že čitatel nebude nikdy dělen nulou.

5 TESTOVÁNÍ DETEKTORU

K otestování natrénované kaskády klasifikátorů na testovacích snímcích slouží objektový detektor. Následující text popisuje funkci detektoru a jeho vstupní parametry.

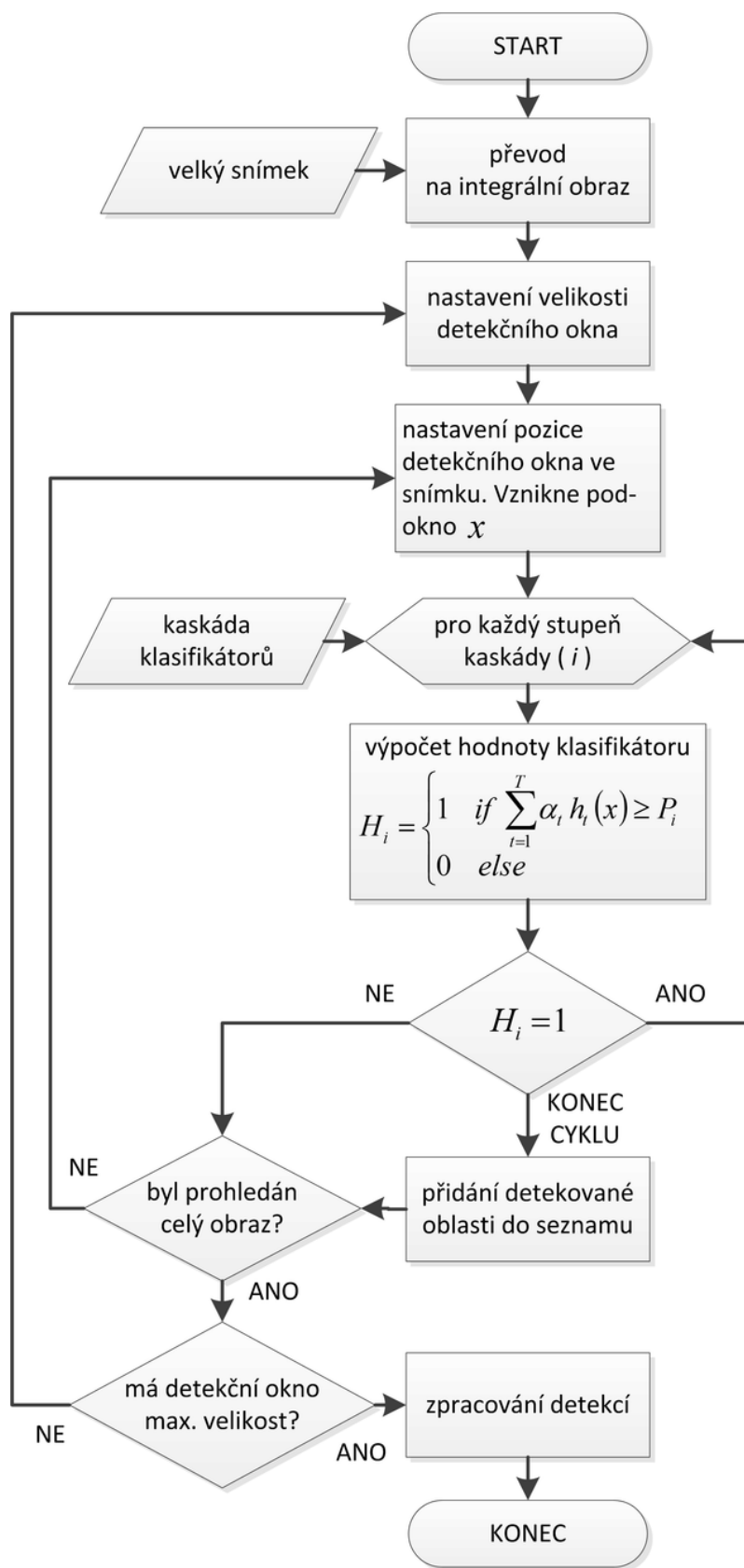
5.1 Proces detekce

Princip funkce detektoru je zobrazen na obrázku 5.1. Na začátku detekčního procesu je načten velký snímek, ve kterém bude detektor hledat objekty. Dále je na vstupu načtena natrénovaná kaskáda klasifikátorů. Vstupní snímek je v dalším kroku převeden na integrální obraz, poté už následuje samotný proces detekce objektů v obraze. Nejprve je nastavena velikost detekčního okna, což je v našem případě 24x24 pixelů. Poté je nastavena pozice detekčního okna ve vstupním snímku, čímž vznikne pod-okno (výřez části obrazu) a následně se bude pomocí kaskády klasifikátorů rozhodovat, jestli se v tomto pod-okně nachází nebo nenachází hledaný objekt. Pokud toto pod-okno projde všemi stupni kaskády, tak jsou souřadnice pod-okna přidány do seznamu detekovaných oblastí. Dále je detekční okno ve vstupním snímku posunuto a vzniká tím další pod-okno, na které bude opět aplikována kaskáda klasifikátorů. Pokud je některým stupněm kaskády rozhodnuto, že pod-okno neobsahuje hledaný objekt, tak je prohledávání pod-okna následujícími stupni kaskády přerušeno a proces pokračuje vygenerováním následujícího pod-okna. Tento cyklus se opakuje do té doby, dokud není pro daný rozměr detekčního okna prohledán celý obraz. Poté jsou rozměry detekčního okna zvětšeny a vše se opakuje až do doby, kdy velikost detekčního okna dosáhne velikosti vstupního velkého snímku. Výsledkem procesu je seznam obsahující detekované oblasti, které mohou být následně dále zpracovány.

5.2 Zpracování detekcí

Po skončení procesu detekce mohou být výsledky detekce dodatečně zpracovány. Zpracovávat detekce lze dvěma způsoby. Buď je na výsledky detekce aplikováno sloučení vícenásobných detekcí stejného objektu do jedné detekce, která je nejvíce vypovídající (týká se detekce příčných tepen, čelistních kloubů) nebo v druhém způsobu jsou nalezené detekce zpracovány tak, že jako výsledek procesu jsou zobrazeny dvě přímky detekující objekt (použito při detekci podélných řezů tepen).

1. Proces zpracování detekcí u prvního způsobu probíhá následovně: nejprve jsou vyhledány navzájem podobné detekce jedné oblasti. To znamená, že jedna detekovaná oblast se nachází uvnitř druhé, nebo se tyto oblasti překrývají (tolerance překrytí detekovaných oblastí je 20 % z šířky jedné z těchto oblastí).



Obr. 5.1: Schéma detekčního procesu.

Poté je určen počet podobných detekcí v rámci jedné oblasti. V rámci tohoto způsobu zpracování detekcí byly použity dvě metody:

- Při zpracování první metodou jsou odstraněny všechny vícenásobné detekce jednoho objektu, jejichž počet je menší než je definovaná hodnota. Poté jsou všechny zbylé detekce v rámci jedné oblasti zprůměrovány v jednu detekci.
- Při zpracování druhou metodou je vybrána oblast s nejvyšším počtem podobných detekcí. Zbylé detekce jsou odstraněny. Následuje zprůměrování detekcí této jedné oblasti do jedné detekce.

Nakonec jsou detekované oblasti zobrazeny ve vstupním snímku.

2. Při použití druhého způsobu jsou nalezené detekce nejprve rozděleny na dvě poloviny podle toho, v které části snímku se nacházejí (pravé nebo levé). Poté jsou z těchto detekcí odstraněny ty, které se nacházejí mimo linii, kde je většina detekcí. Dalším krokem je zprůměrování souřadnic a rozměrů detekcí zvlášť pro levou i pro pravou část obrazu. Tím vzniknou dvě detekce a za použití souřadnic a rozměrů těchto detekcí je možné vykreslit dvě přímky.

5.3 Nastavení parametrů detektoru

U detekčního procesu je při jeho startu nastaveno několik parametrů. Prvním parametrem je hodnota, která udává, kolikrát bude zvětšena velikost detekčního okna. Původní velikost okna je 24x24 pixelů, pokud je hodnota parametru 3, tak startovní velikost detekčního okna bude 72x72 pixelů. Druhým parametrem je hodnota, která udává, kolikrát bude v každém kroku zvětšena velikost detekčního okna. Dále je třeba určit, o jak velký krok bude detekční okno posouváno. Tento krok je definován jako součin velikosti detekčního okna s hodnotou tohoto třetího parametru. Dále je třeba nastavit hodnotu parametru pro zpracování detekcí. Pokud je počet detekcí jednoho objektu menší než hodnota parametru, tak nebudou tyto detekce zobrazeny. Hodnoty těchto parametrů by měly být zvoleny tak, aby bylo dosaženo kvalitních výsledků při velké výpočetní rychlosti.

6 REALIZACE OPERÁTORŮ V RAPIDMINERU

Cílem práce bylo vytvořit operátory pro aplikaci RapidMiner¹ (RM) verze 5.0. Operátor je v prostředí programu RapidMiner zobrazen jako objekt (obdélník), který může mít vstupy a výstupy a vykonává definované úkony. Tento operátor je možné pomocí vstupů a výstupů propojit s dalšími jinými operátory. Operátory jsou vytvořeny v programovacím jazyce Java a jsou součástí knihovny „Image Processing Extension²“, která je vyvíjena na Ústavu telekomunikací výzkumnou skupinou Signal Processing Laboratory. Celkem bylo v rámci této práce vytvořeno 8 operátorů a jejich podrobný popis a funkce jsou uvedeny níže.

6.1 Operátor pro načítání sady obrázků

Operátor, jehož název v RM je *Read Image Set*, slouží k načítání obrázků z disku počítače do prostředí programu RM. Vstupními parametry operátoru jsou cesty ke složkám s obrázky. Prvním typem dat, která operátor načítá, jsou trénovací obrázky. Zde se zvolí složky s pozitivními a s negativními snímky. Pozitivní snímky jsou snímky, na kterých se nachází objekt zájmu. Negativní snímky obsahují různé části okolí kolem objektu zájmu. Dalším typem dat, která mohou být načtena, jsou obrázky a k nim příslušné textové soubory. Tyto textové soubory obsahují souřadnice detekovaných oblastí v obraze. Tyto snímky spolu s textovými soubory mohou být použity buď pro validaci, nebo spolu s již zmíněnými trénovacími pozitivními a negativními snímky slouží k trénování detektoru genetickými algoritmy. Načítat lze tedy buď jen trénovací data, nebo jen obrázky spolu s textovými soubory, nebo vše najednou. Na výstupní port jsou odeslána data (image set) ve formě tabulky, která zobrazuje cesty k souborům a značí, ke které klasifikační třídě náleží trénovací snímky.

6.2 Operátor pro vytváření trénovacích dat

Pro vytvoření trénovacích dat slouží operátor v RM označovaný jako *Create Training Data*. Operátor funguje pouze ve spojení s operátorem pro načítání sady obrázků a proto očekává na svém vstupním portu tabulku (image set). Vstupními parametry operátoru jsou cesty ke složkám, kam budou ukládány pozitivní a negativní snímky. Dále je možné zvolit, zda bude proces vykonán paralelně. Vytváření pozitivních snímků pak probíhá následovně: nejprve je načten původní velký obrázek

¹Dostupný z URL: <http://rapid-i.com/>.

²Dostupný z URL: <http://spl.utko.feec.vutbr.cz/>.

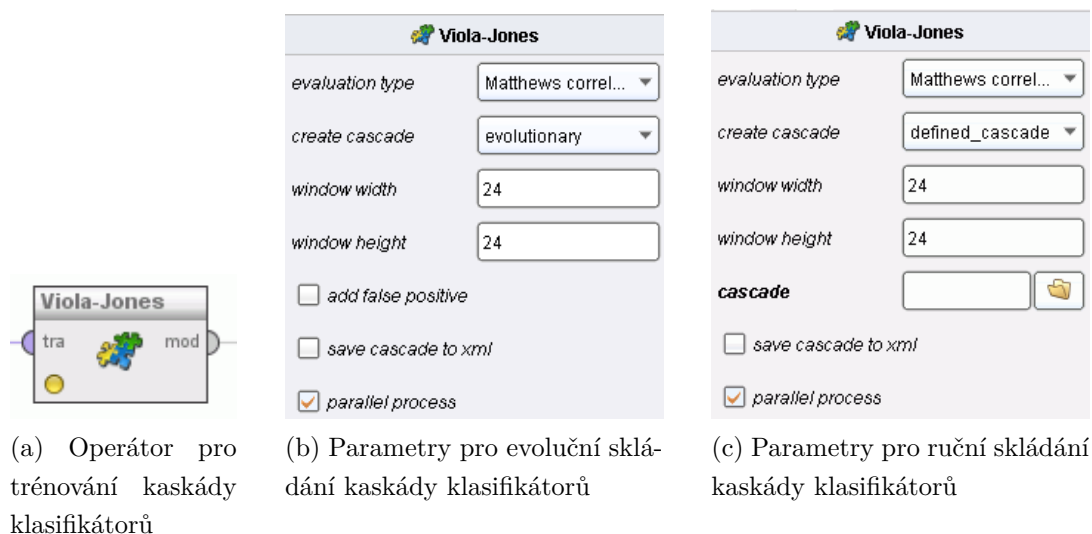
a k němu příslušný textový soubor, který obsahuje souřadnice a velikosti oblastí zájmu, dále jsou na základě těchto parametrů z částí velkých obrázků vytvořeny pozitivní snímky a uloženy ve formátu *jpg* do určené složky na disk. Při vytváření negativních snímků je postup obdobný, ale s tím rozdílem, že vytvářený negativní snímek se musí nacházet kdekoli jinde v původním velkém obrázku, než kde se nacházejí oblasti zájmu. Volba, z které části původního obrázku bude negativní snímek vytvořen, je náhodná.

6.3 Operátor pro trénování kaskády klasifikátorů

Operátor (obrázek 6.2a) se v RM nazývá *Viola-Jones* a jeho funkcí je natrénovat kaskádu klasifikátorů pro detekci objektů v obraze. Na vstup operátoru jsou přivedena trénovací data z operátoru pro načítání sady obrázků. Dále musí být zvolen způsob trénování kaskády klasifikátorů. Prvním způsobem je trénování pomocí genetických algoritmů (obrázek 6.2b). Vstupními daty pro trénování jsou pozitivní a negativní snímky i velké obrázky s příslušnými textovými soubory. Při zvolení druhého způsobu (obrázek 6.2c) je kaskáda klasifikátorů skládána za pomoci vstupního textového souboru, který definuje její strukturu. Obsah textového souboru je zobrazen na obrázku 6.1. První číslo brané v textovém souboru zleva identifikuje stupeň kaskády. Druhé číslo určuje počet příznaků, které budou jako nové přidány do stupně kaskády. Poslední číslo udává velikost změny prahové hodnoty (threshold) stupně kaskády. Hodnota je v procentech. Vstupními daty pro trénování tímto způsobem jsou pouze pozitivní a negativní snímky. Pro oba způsoby trénování jsou společné vstupní parametry: paralelní běh procesu, nastavení velikosti trénovacího okna (standardně 24x24 pixelů) a možnost převedení a uložení natrénované kaskády klasifikátorů do souboru typu *xml*. Dále následuje podle zvolených parametrů trénovací proces. Výstupem operátoru je natrénovaný model (kaskáda klasifikátorů), který může být uložen na disk operátorem *Write Model* nebo zpětně načten z disku operátorem *Read Model*.

1; 3; 5
2; 13; -3
3; 5; -7
4; 2; 12
5; 4; 0

Obr. 6.1: Způsob zápisu kaskády klasifikátorů v textovém souboru.



Obr. 6.2: Operátor pro trénování kaskády klasifikátorů a jeho parametry.

6.4 Operátor pro testování kaskády klasifikátorů

Tento operátor testuje natrénovaný model na testovacích datech. Operátor se v RM nazývá *Apply Model Viola-Jones* a má dva vstupní porty a jeden výstupní port. Na prvním vstupu operátor očekává natrénovaný model (kaskádu klasifikátorů) a na druhém vstupu jsou přijímána data z operátoru pro načítání sady obrázků. Tento operátor vychází z funkce operátoru *Detector* (kapitola 6.7) a jeho účelem je zpracovat větší množství snímků a určit u nich počty detekovaných objektů a falešně pozitivních detekcí. Operátor *Detector* zpracovává vždy pouze jeden snímek, ve kterém detekuje objekty. Vstupní parametry tohoto operátoru jsou stejné jako u operátoru *Detector* a budou rozebrány v kapitole 6.7. Na vstupní obrázky je aplikována kaskáda klasifikátorů, která detekuje v těchto obrazech objekty. Souřadnice těchto detekovaných objektů jsou pak srovnány se souřadnicemi obsaženými v textových souborech. Při srovnání souřadnic je uvažována jistá tolerance přesnosti. Na základě tohoto srovnání jsou pro každý obrázek určeny počty detekovaných objektů a počty falešně pozitivních detekcí. Na výstupu operátoru se nachází výstupní data (example set), která obsahují informace o cestě k obrázku, dále obsahují počet detekovaných objektů, počet všech hledaných objektů a počet falešně pozitivních detekcí.

6.5 Operátor pro hodnocení kaskády klasifikátorů

V RM nese operátor název *Performance Viola-Jones* a jeho funkcí je zhodnotit úspěšnost natrénovaného modelu (kaskády klasifikátorů) na testovacích datech. Na

vstup operátoru jsou přivedena data (example set) z výstupu operátoru pro testování natrénovaného modelu. Vstupní data jsou zpracována a odeslána na výstupy operátoru. Hodnotou prvního výstupu je výkonnost (performance), což je číslo vyovídající o úspěšnosti natrénovaného modelu. Čím je hodnota čísla větší, tím větší úspěšnost má model při detekci objektů v obraze. Data (result) z druhého výstupu jsou zobrazena v prostředí RM jako textový výpis, který udává počet detekovaných objektů, počet všech objektů obsažených v obrázcích, celkovou přesnost detekce a počet falešně pozitivních detekcí. Posledním výstupem operátoru je tabulka (example set) obsahující totožná data s tabulkou na vstupu operátoru.

6.6 Operátor pro načtení kaskády klasifikátorů

Operátor se v RM nazývá *Create Model From Xml*. Jeho funkcí je načíst kaskádu klasifikátorů uloženou v souboru typu *xml* a následně ji převést na model, který již spolupracuje s operátorem pro detekci objektů v obraze, dále s operátorem pro testování kaskády klasifikátorů a s operátorem pro úpravu kaskády klasifikátorů. Jako vstupní *xml* soubor může být načtena kaskáda klasifikátorů trénovaná operátorem *Viola-Jones*. Dále můžou být načteny kaskády klasifikátorů, které jsou součástí projektu OpenCV³ (Open Source Computer Vision). Vstupním parametrem operátoru je pouze cesta k souboru s kaskádou klasifikátorů. Na výstupu operátoru je model.

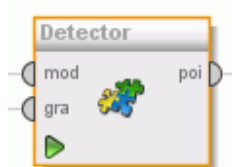
6.7 Operátor pro detekci objektů v obraze

Operátor (obrázek 6.3a), který je v RM pojmenován jako *Detector* detekuje s pomocí natrénovaného modelu oblasti ve vstupním snímku. Na prvním vstupu přijímá operátor natrénovaný model a druhým vstupem je přijímán obrázek ve stupních šedi (grayscale image plus). Výstupem operátoru jsou body zájmu (points of interest), které obsahují informace, kde se detekované objekty nachází a jaké mají rozměry. Detekční proces funguje tak, že nejprve je stanovena startovní velikost detekčního okna, poté je okno posouváno tak, aby prošlo celým obrazem. V dalším cyklu je okno zvětšeno a znovu prochází celý obraz a detekuje objekty. Tento cyklus se opakuje pokud jsou rozměry detekčního okna menší než rozměry obrázku. Operátor má tyto vstupní parametry (obrázek 6.3b):

- Parametr *base scale* – tato hodnota udává, kolikrát bude zvětšena startovní velikost detekčního okna. Původní velikost trénovacího okna je zvolena při trénování kaskády klasifikátorů. Hodnota parametru je rovna poměru mezi velikostí startovního detekčního okna a velikostí trénovacího okna.

³Dostupný z URL: <http://opencv.willowgarage.com>

- Parametr *scale increment* – velikost detekčního okna je zvětšena v každém kroku hodnotou tohoto parametru.
- Parametr *increment* – detekční okno je v obraze posouváno o určitý krok, který je definován součinem velikosti okna detektoru a tohoto parametru.
- Parametr *min neighbors* – pokud je zvolena funkce pro zpracování detekovaných oblastí, tak se vícenásobné detekce jedné oblasti sloučí v jednu detekci. Funkce funguje tak, že pokud je počet detekcí jednoho objektu menší než hodnota parametru, tak nebudou tyto detekce zobrazeny.
- Parametr *best one detection* – aktivováním tohoto parametru dojde k zobrazení pouze jedné nejvýznamnější oblasti, která byla ze všech oblastí nejvíce krát detekována.
- Parametr *rectangles to lines* – aktivováním tohoto parametru jsou všechny nalezené oblasti zpracovány tak, že jako výsledek procesu jsou zobrazeny dvě čáry detekující objekt (použito při detekci podélných řezů tepen).
- Parametr *parallel process* – slouží k paralelnímu běhu procesu detekce.



(a) Operátor detektor

Detector	
base scale	3.0
scale increment	1.1
increment	0.1
min neighbors	2.0
<input checked="" type="checkbox"/> use min neighbors	
<input type="checkbox"/> best one detection	
<input type="checkbox"/> rectangles to lines	
<input checked="" type="checkbox"/> parallel process	

(b) Parametry operátoru

Obr. 6.3: Operátor pro detekci objektů v obraze a jeho parametry.

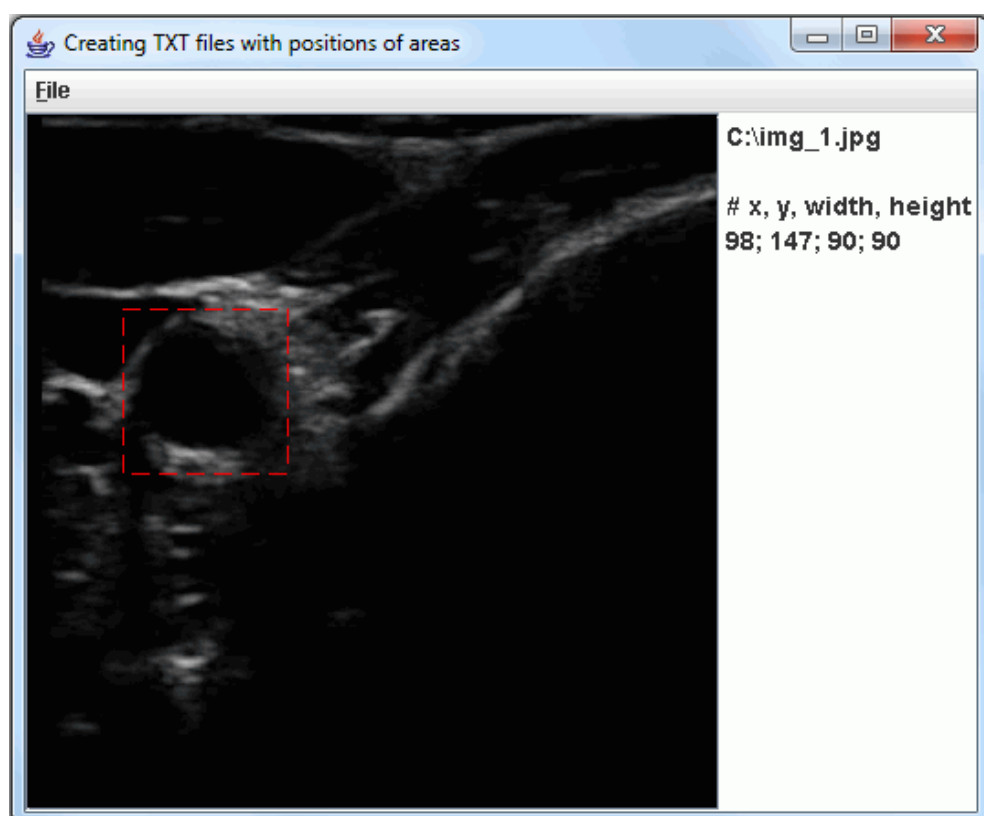
6.8 Operátor pro úpravu kaskády klasifikátorů

Funkcí operátoru (v RM je operátor nazýván *Change Cascade*) je přijmout na vstupu natrénovaný model a podle vstupních parametrů tento model upravit a poslat jej na výstup nebo uložit jako kaskádu klasifikátorů do souboru *xml*. Jako vstupní data jsou kromě natrénovaného modelu ještě použita data z výstupu operátoru pro načítání

sady obrázků (pro evoluční skládání kaskády klasifikátorů). Poté co operátor obdrží na vstupu model (kaskádu klasifikátorů), použije klasifikátory ze všech stupňů již sestavené vstupní kaskády a podle definovaných pravidel z nich začne skládat novou kaskádu. Ve vstupních parametrech operátoru se určí, jakým způsobem bude kaskáda sestavena. Výběr parametrů je stejný jako u operátoru, který složí k trénování modelu. Sestavení kaskády je tedy možné za pomoci genetických algoritmů nebo za pomoci textového souboru s definovanou kaskádou. Tento proces lze opět spustit paralelně.

6.9 Aplikace pro uložení pozice objektu do textového souboru

Protože je nutné pro trénování a testování kaskády klasifikátorů používat snímky a k nim příslušné textové soubory obsahující rozměry hledaných oblastí v daných snímcích, tak byla vytvořena aplikace, pomocí které lze jednoduše a rychle přesně získat souřadnice a rozměry hledaných objektů a následně tyto informace uložit do textového souboru. Okno aplikace je zobrazeno na obrázku 6.4. Použití aplikace je následovné: nejprve musí být vybrána složka obsahující snímky. Pak je zobrazen první snímek ze složky. Oblast zájmu se nachází uvnitř obdélníku (zobrazen červenou přerušovanou čarou), který lze zvětšit nebo zmenšit pomocí kolečka myši nebo lze změnit jeho tvar kliknutím pravého tlačítka myši a následným posunutím kurzoru. Pokud je vybrána hledaná oblast, tak jsou po kliknutí levého tlačítka myši souřadnice a rozměry této oblasti zaznamenány. Takto lze v rámci jednoho snímku zaznamenat libovolný počet oblastí. Pokud jsou v aktuálním snímku označeny již všechny hledané oblasti, tak se lze přepnout na snímek následující pomocí klávesy „V“. Vrácení se na předcházející snímek je provedeno stiskem klávesy „C“. Při každém posunu z jednoho snímku na další dojde k uložení souřadnic daného snímku do textového souboru, který je uložen ve stejné složce jako daný snímek. Název textového souboru je stejný jako název snímku (mimo typu souboru).



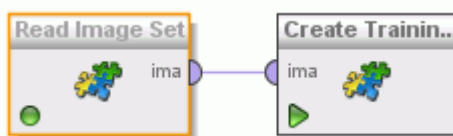
Obr. 6.4: Okno aplikace.

7 POUŽITÍ OPERÁTORŮ V RAPIDMINERU

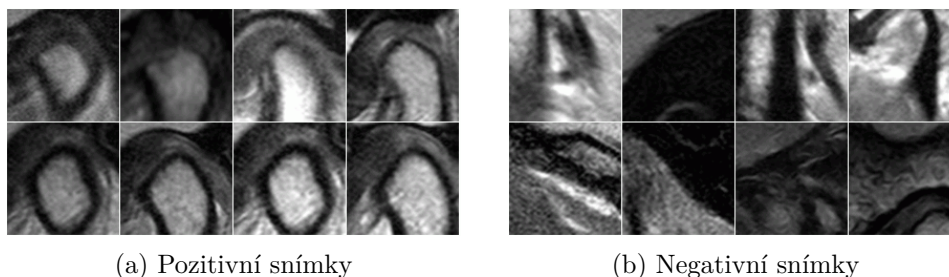
V této kapitole je popsáno pět příkladů, které ukazují, jak lze vytvořené operátory použít v prostředí programu RapidMiner. V rámci práce bylo vytvořeno několik demonstračních příkladů, které jsou uloženy na přiloženém DVD.

7.1 Vytváření trénovacích dat

Pro vytváření trénovacích dat je nutné propojit operátor pro načítání sady obrázků s operátorem, který generuje trénovací snímky. Propojení operátorů je patrné z obrázku 7.1. Vstupní operátor nahraje obrázky a k nim příslušné textové soubory a výstupní operátor uloží do definovaných složek nově vzniklé pozitivní a negativní snímky. Na obrázku 7.2a jsou zobrazeny vygenerované pozitivní snímky. Vygenerované negativní snímky jsou zobrazeny na obrázku 7.2b. Takto vygenerované snímky jsou určeny pro natrénování kaskády klasifikátorů.



Obr. 7.1: Zapojení operátorů při generování trénovacích dat.

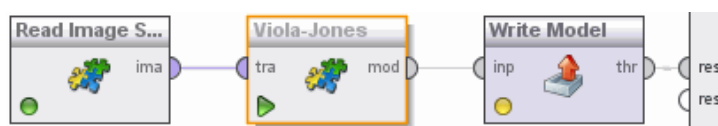


Obr. 7.2: Ukázka vygenerovaných trénovacích snímků.

7.2 Trénování kaskády klasifikátorů

Pro trénování kaskády klasifikátorů je třeba zapojit operátory viz obrázek 7.3. Operátorem pro načtení sady obrázků jsou zvoleny snímky určené k trénování. Dále jsou zvoleny parametry operátoru *Viola–Jones* pro trénování kaskády klasifikátorů

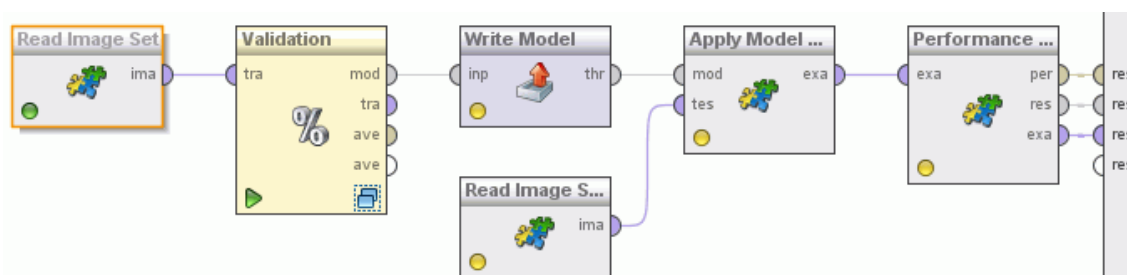
a nakonec je tato kaskáda uložena do souboru typu *xml* nebo pomocí operátoru pro uložení modelu.



Obr. 7.3: Příklad trénování kaskády klasifikátorů.

7.3 Trénování metodou křížové validace

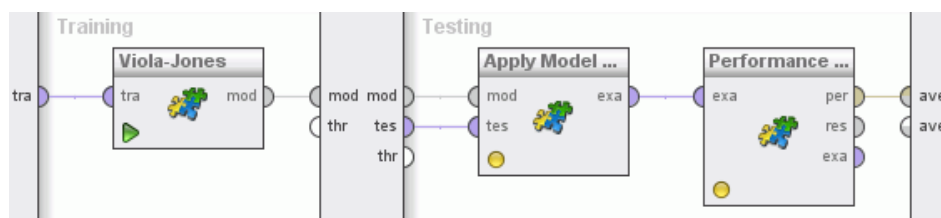
Při použití křížové validace (Cross-Validation) pro natrénování modelu jsou trénovací data rozdělena na definovaný počet stejných částí. Často jsou trénovací data rozdělena na deset stejných částí. Potom devět částí slouží pro trénování a zbylá část slouží k testování natrénovaného modelu. Toto se opakuje dokud se neprotestují všechny části tak, aby každá část sloužila pro testování právě jednou. Cílem křížové validace je vytvořit co nejreálnější model, který bude mít na validačních datech co nejlepší přesnost. [8] Na obrázku 7.4 je zobrazeno zapojení operátorů. Na vstup operátoru pro křížovou validaci jsou přivedena trénovací data. Po natrénování je na výstupní port odeslán výsledný model, který je uložen na disk a poté je model otestován na validačních datech, která jsou nezávislá na datech trénovacích. Poslední zapojený operátor hodnotí úspěšnost natrénovaného modelu a data z jeho výstupů jsou zobrazena jako výsledky.



Obr. 7.4: Příklad trénování kaskády klasifikátorů metodou křížové validace.

Vnitřní procesy operátoru křížové validace jsou zobrazeny na obrázku 7.5. Zde je patrné, jaké procesy probíhají v režimu trénování a jaké v režimu testování. V procesu trénování je pomocí operátoru pro natrénování modelu natrénován na trénovacích datech model a v procesu testování je tento model otestován na testovacích datech. Operátor pro zhodnocení úspěšnosti natrénovaného modelu (*Performance*

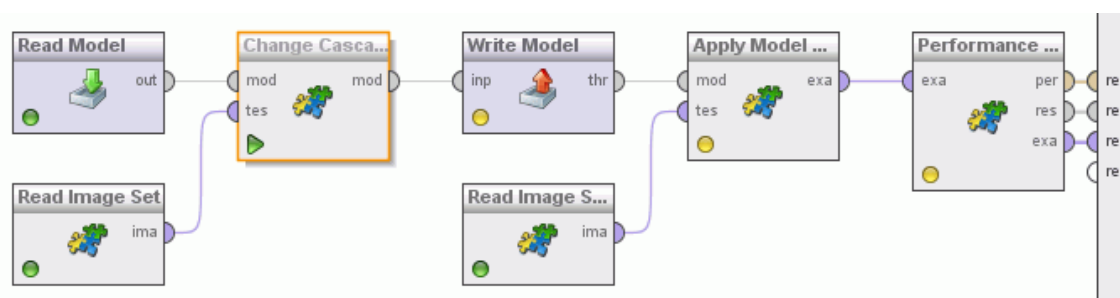
Viola-Jones) předává na vnitřní vstup operátoru křížové validace číselnou hodnotu, která vypovídá o úspěšnosti natrénovaného modelu. Vstupní parametr „počet validací“ u operátoru křížové validace určuje kolikrát se bude opakovat proces trénování a proces testování.



Obr. 7.5: Ukázka nastavení vnitřních procesů operátoru pro křížovou validaci.

7.4 Úprava natrénované kaskády klasifikátorů

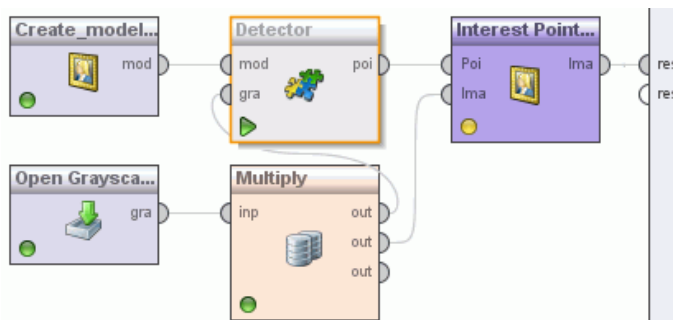
Pokud není původní natrénovaný model (kaskáda klasifikátorů) přesný nebo dostatečně rychlý, může se s pomocí operátoru pro úpravu kaskády klasifikátorů (*Change Cascade*) změnit tak, aby se jeho přesnost nebo rychlost detekce zlepšila. Z obrázku 7.6 je patrné zapojení jednotlivých operátorů. Na první vstup operátoru pro úpravu modelu je přiveden model, který má být modifikován. Na druhý vstup operátoru je přivedena sada snímků. Druhý vstup se zapojuje pouze pokud je zvolena u operátoru pro úpravu natrénovaného modelu jako vstupní parametr evoluční metoda. Výstupem operátoru je modifikovaný model, který je uložen na disk. Dále je zjištěna jeho přesnost detekce ověřením na testovacích datech.



Obr. 7.6: Příklad zapojení operátorů pro modifikaci kaskády klasifikátorů.

7.5 Příklad použití objektového detektoru

Pro detekci objektů ve vstupním obraze a následné zobrazení detekovaných oblastí v RM, je nutné zapojit operátory podle obrázku 7.7. V tomto příkladě je kaskáda klasifikátorů nahrána z *xml* souboru, ale lze použít pro načtení modelu i operátor *Read Model*. Výstupem detektoru jsou body zájmu (points of interest), které jsou po zpracování operátorem pro vizualizaci bodů zájmu v obraze (v RM je nazýván *Interest Points Visualizer*) zobrazeny jako detekované oblasti ve vstupním snímku.

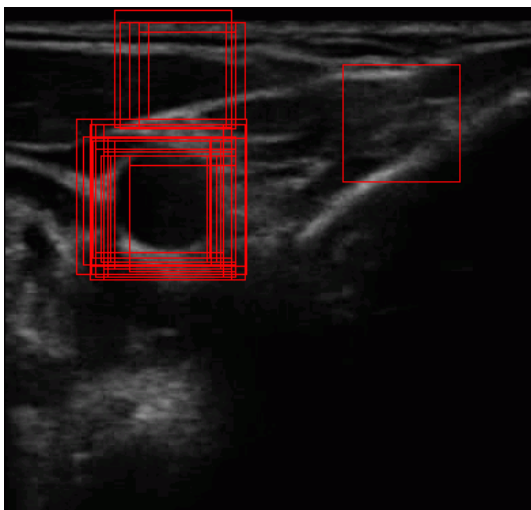


Obr. 7.7: Příklad zapojení operátorů pro detekci a zobrazení objektů v obraze.

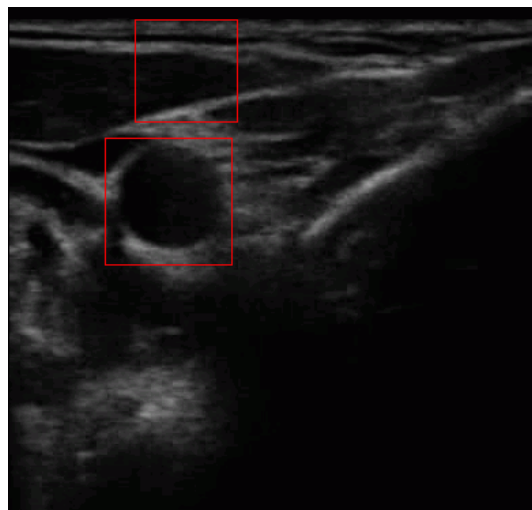
Pro detekci příčného řezu tepny je vhodné nastavit parametry detektoru viz obrázek 6.3b. Pokud nebude zapnuta funkce pro sloučení vícenásobných detekcí (parametr *use min neighbors*), tak budou zobrazeny všechny detekované oblasti (viz obrázek 7.8a). Pokud je funkce pro sloučení vícenásobných detekcí zapnuta a hodnota jejího parametru (*min neighbors*) bude nastavena na 2, tak jsou všechny detekce oblastí, které byly detekovány méně než dvakrát, smazány. Ostatní vícenásobné detekce objektu jsou sloučeny v jednu detekci (obrázek 7.8b). Pokud má být zobrazena pouze jedna nejvýznamnější detekce v celém snímku, tak musí být zapnuta příslušná funkce (parametr *best one detection*). Jako nejlepší detekovaná oblast je zobrazena ta, která byla ze všech oblastí detekována nejvíce krát (obrázek 7.8c).

Pro detekci podélných řezů tepen musí být ve vstupních parametrech detektoru vypnuta funkce pro sloučení vícenásobných detekcí a zároveň musí být zvolena funkce pro vykreslení linek (*rectanges to lines*). Výsledek detekce s těmito nastavenými parametry je patrný z obrázku 7.9a. Pokud je funkce pro vykreslení linek vypnuta, tak jsou zobrazeny všechny detekované objekty (obrázek 7.9b).

Dalším příkladem použití operátoru je detekce obličejů v obraze. Výsledek detekce je patrný z obrázku 7.10 (převzat z [10]). Kaskáda klasifikátorů byla převzata z projektu OpenCV. V tomto případě byla pro detekci zvolena kaskáda klasifikátorů *haarcascade_frontalface_alt2.xml*.



(a) Zobrazení všech nalezených detekcí

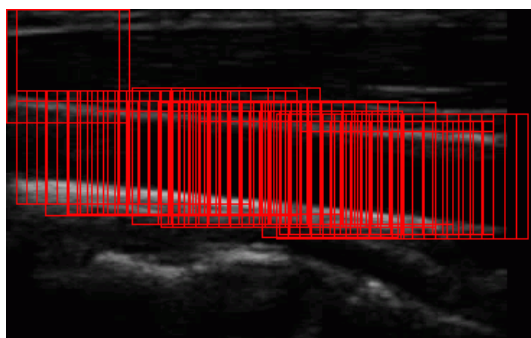


(b) Detekce po zpracování funkcí pro sloučení vícenásobných detekcí

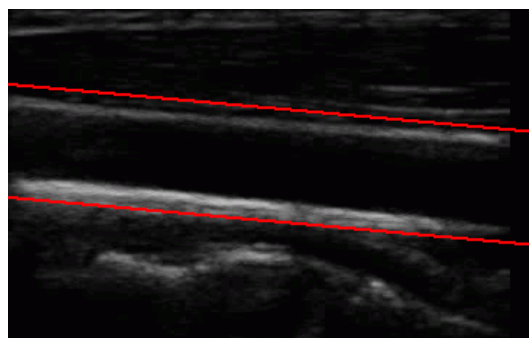


(c) Zobrazení nejvýznamnější detekce

Obr. 7.8: Způsob vykreslení detekce pomocí obdélníků.



(a) Zobrazení všech nalezených detekcí



(b) Detekce po zpracování funkcí pro vykreslení linek

Obr. 7.9: Způsob vykreslení detekce pomocí linek.



Obr. 7.10: Ukázka detekce obličejů v RapidMineru.

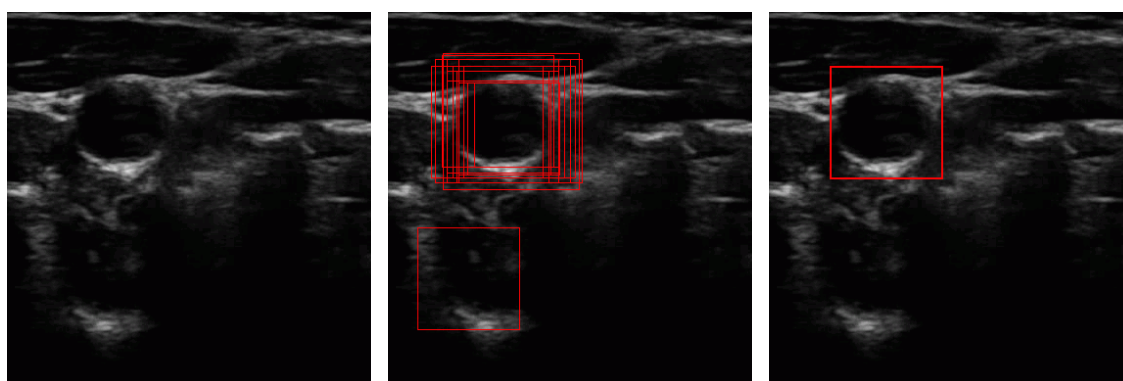
8 KONKRÉTNÍ ŘEŠENÉ PROBLÉMY A JEJICH ZHODNOCENÍ

Na základě popsaných algoritmů byly natrénovány detektory pro detekci příčných a podélných řezů tepen z ultrazvukových snímků a dále pro detekci čelistní kloubů ze snímků magnetické rezonance. Tyto detektory (kaskády klasifikátorů) byly otestovány na testovacích datech. V následující podkapitolách jsou popsány řešené problémy a jejich vyhodnocení.

8.1 Detekce příčných tepen

Prvním z řešených problémů byla detekce příčných řezů tepen z ultrazvukových snímků. Na základě detekce tepny z videosekvence je možné určit tepovou frekvenci pacienta nebo změřit průměr tepny při systolickém i diastolickém tlaku. Určením geometrických parametrů tepny může být diagnostikována nemoc jakou je ateroskleróza.

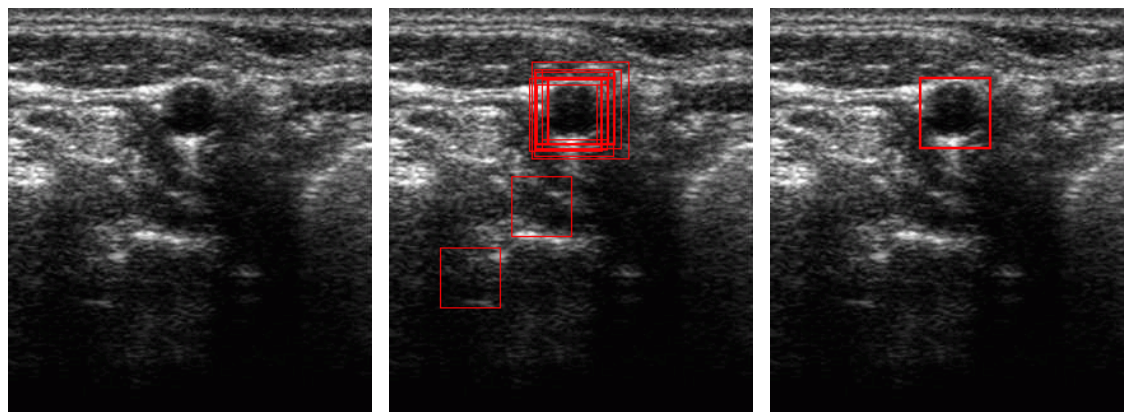
K natrénování a otestování detektoru posloužily videosekvence tepen získané z ultrazvukových přístrojů Ultrasonix OP (s lineární sondou L 14-5/38) a Toshiba Nemio XG (s lineární sondou na frekvenci 7,5 MHz). Na obrázku 8.1a je zobrazen originální snímek dospělého pacienta pořízený přístrojem Ultrasonix. V tomto snímku (obrázek 8.1b) jsou červeným obdélníkem zobrazeny všechny detektorem detekované oblasti. Posledním krokem je vybrání nejvýznamnějších detekcí a jejich následné sloučení v jednu nejvíce vypovídající detekci. Tato detekce je prezentována červeným čtvercem (obrázek 8.1c).



(a) Originální snímek tepny dospělého člověka (b) Zobrazení všech nalezených detekcí (c) Zobrazení detekované tepny

Obr. 8.1: Ukázka detekce příčné tepny z přístroje Ultrasonix.

Obrázek 8.2a zobrazuje originální mírně zašumělý vstupní snímek dětského pacienta, který byl pořízen na přístroji Toshiba. Všechny nalezené detekce jsou patrné z obrázku 8.2b a konečný výsledek je zobrazen na obrázku 8.2c.



(a) Originální snímek tepny dítěte (b) Zobrazení všech nalezených detekcí (c) Zobrazení detekované tepny

Obr. 8.2: Ukázka detekce příčné tepny z přístroje Toshiba.

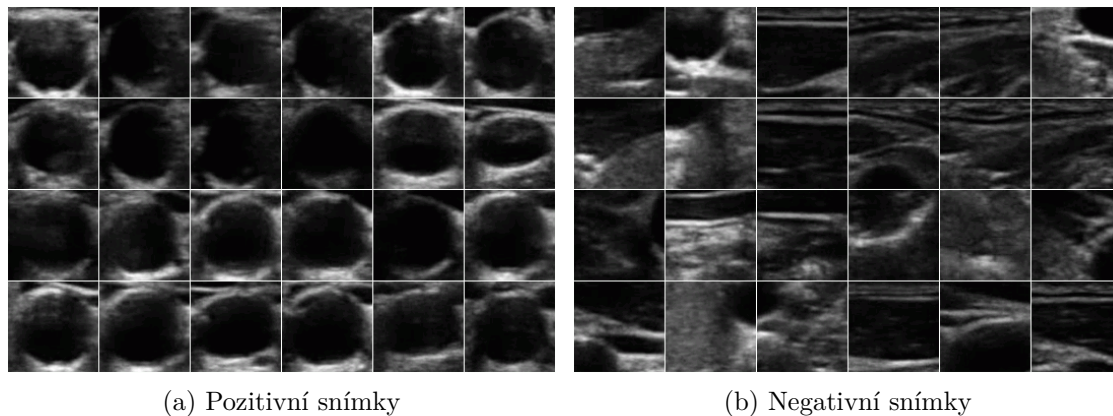
8.1.1 Použité databáze snímků

Na přístroji Ultrasonix byly vytvořeny dvě databáze, jedna pro trénování a druhá pro testování. Na přístroji Toshiba byla vytvořena databáze pouze pro testování. Všechna data byla nasnímana na 18-ti dospělých a jednom dětském pacientovi (velikost tepny výrazně menší). Pro natrénování a dostatečné otestování detektoru nebyl důležitý počet pacientů, ale zejména různorodost nasnímaných videosekvencí, která byla dosažena změnou nastavení parametrů snímacího zařízení (frekvence, fokus, TGC křivka) a změnou způsobu snímání samotného (přítlak sondy a její pootočení vzhledem ke snímané tepně).

Trénovací snímky

Pro vytvoření trénovacích snímků posloužily tři videosekvence, které byly nasnímány na jednom dospělém pacientovi. Tyto videosekvence záměrně obsahovaly i soubor nežádoucích jevů, vyskytujících se v praxi. V trénovací sadě se tedy vyskytovaly také zašumělé obrazy a tepny, které jsou zdeformované díky přítlaku a natočení sondy vzhledem k analyzované oblasti. Pro trénování byly použity všechny snímky obsažené ve zmíněných videosekvencích. Celkem bylo pro trénování použito 283 snímků, ke kterým byly ručně vytvořeny doprovodné textové soubory obsahující pozice a velikosti hledaných tepen. Tato data byla použita pro evoluční skládání kaskády a

dále při vytváření trénovacích snímků o rozměrech 24x24 pixelů. To znamená, že bylo vytvořeno 283 pozitivních (obrázek 8.3a), a stejný počet negativních snímků (obrázek 8.3b), celkem tedy trénování probíhalo na 566 snímcích.



Obr. 8.3: Ukázka trénovacích snímků příčných řezů tepen.

Testovací snímky

Pro otestování natrénovaného detektoru byla použita data nasnímaná na obou ultrazvukových přístrojích. Na přístroji Ultrasonix bylo zaznamenáno celkem 50 videosekvencí od 15-ti dospělých pacientů, kde některé videosekvence byly pořízeny při nádechu nebo polykání. Z každé této videosekvence byl každý dvacátý snímek vybrán do souboru testovacích dat. Vzniklo tak 538 testovacích snímků. Na přístroji Toshiba bylo zaznamenáno 22 videosekvencí od dvou dospělých a jednoho dětského pacienta. Z každé videosekvence byl vybrán každý desátý snímek v řadě a přidán do souboru testovacích dat. Celkem vzniklo 433 snímků, které jsou zašumělé a obsahují i částečně zdeformované tepny. Ke všem testovacím snímkům byly vytvořeny referenční textové soubory obsahující souřadnice středů a velikostí hledaných tepen.

8.1.2 Výsledky testování – příčné tepny

Následující text popisuje nastavení parametrů detekčního procesu, při kterém byly otestovány natrénované kaskády klasifikátorů. Dále jsou popsány dosažené výsledky. V příloze jsou zobrazeny správné detekce příčných řezů tepen (viz obrázek B.1). Dále jsou na obrázku B.4 zobrazeny falešně pozitivní detekce ve vybraných snímcích.

Parametry detekčního procesu

Detekční proces probíhal s využitím operátoru *Detector*, který měl nastaveny tyto vstupní parametry: startovní velikost detekčního okna byla 72x72 pixelů, parametr

zvětšení detekčního okna v každém kroku byl 1,1. Hodnota parametru posunu detekčního okna v obraze byla 0,1. Startovní velikost detekčního okna byla zvolena na základě experimentů tak, aby byla minimalizována výpočetní náročnost pro většinu zpracovávaných obrazů (dospělých jedinců). Analyzované snímky mohou však obsahovat i například tepny dětských pacientů, které pro danou velikost okna nemohou být detekovány, protože detekované tepny mají menší rozměr než je startovní rozměr detekčního okna. Z tohoto důvodu musel být detekční proces modifikován tak, aby realizoval i detekci takovýchto objektů. Detekční proces probíhá klasickým způsobem s velikostí startovního okna 72x72 pixelů a až v momentě, kdy není detekována žádná tepna, je startovní velikost detekčního okna zmenšena na rozměry 48x48 pixelů a celý detekční proces proběhne znovu.

Testování kaskád klasifikátorů

Celkem byly s pomocí genetických algoritmů natrénovány 4 typy kaskád klasifikátorů pro různé nastavení parametrů fitness funkce. Při evolučním skládání kaskád byla použita rovnice 4.3 pro výpočet hodnotící funkce, kde byly pro každou kaskádu klasifikátorů dosazovány jiné hodnoty koeficientů a a b . Přesnost detekce byla vypočtena jako podíl mezi sumou správně detekovaných tepen a sumou všech hledaných tepen. Každá kaskáda byla testována dvěma způsoby:

1. Se započítáním všech možných detekcí, takže ve výsledcích je kromě hodnoty přesnosti detekce $R_{TP} = \frac{N_{TP}}{N_I}$ také informace o míře falešně pozitivních detekcí $R_{FP} = \frac{N_{FP}}{N_I}$, kde N_{TP} určuje počet správně detekovaných objektů, N_{FP} určuje celkový počet falešně pozitivních detekcí a N_I určuje celkový počet snímků, na kterých probíhalo testování. Výsledky jsou uvedeny v % a zobrazeny v tabulce 8.1.
2. Se započítáním pouze nejvýznamnější detekce díky předpokladu, že se v celém snímku nachází pouze jedna tepna – viz tabulka 8.2.

Testování probíhalo na jedné sadě trénovacích a dvou sadách testovacích dat (podle daného přístroje). Z tabulky 8.1 jsou patrné výsledky přesnosti detekce pro jednotlivé kaskády klasifikátorů se započítáním falešně pozitivních detekcí. Testování bylo provedeno pro různé varianty natrénovaných kaskád s různými hodnotami koeficientů a a b z rovnice 4.3. Z výsledků vyplývá, že čím je větší hodnota podílu mezi a a b , tím má kaskáda klasifikátorů větší přesnost detekce, ale zároveň narůstá celkový počet falešně pozitivních detekcí. Tabulka 8.2 zobrazuje přesnosti detekce pro natrénované kaskády pro zpracování všech detekovaných oblastí tak, aby algoritmus označil vždy pouze jednu nejvýznamnější detekci. Z výsledků je patrné, že nejlepší přesnost detekce má kaskáda s parametry trénování $a = 1$ a $b = 0$, která dosáhla přesnosti 97,21 % na testovacích datech z přístroje Ultrasonix a 96,99 % na

Tab. 8.1: Přesnost detekce různě trénovaných kaskád klasifikátorů (bez omezení počtu výskytů tepen v obraze).

	Trénovací data Ultrasonix (283)		Testovací data Ultrasonix (538)		Testovací data Toshiba (433)	
Kaskády kl.	R_{TP} [%]	R_{FP} [%]	R_{TP} [%]	R_{FP} [%]	R_{TP} [%]	R_{FP} [%]
$a = 1, b = 0$	99,64	67,14	100	84,01	96,99	28,18
$a = 10, b = 1$	99,29	19,43	98,51	24,16	95,61	14,09
$a = 2, b = 1$	97,17	11,66	94,84	12,83	86,14	4,16
$a = 1, b = 1$	95,75	11,31	93,49	11,9	92,14	4,39

Tab. 8.2: Přesnost detekce různě trénovaných kaskád klasifikátorů (omezené předpokladem výskytu pouze jedné tepny v obraze).

	Trénovací data Ultrasonix (283)	Testovací data Ultrasonix (538)	Testovací data Toshiba (433)
Kaskády kl.	R_{TP} [%]	R_{TP} [%]	R_{TP} [%]
$a = 1, b = 0$	98,93	97,21	96,99
$a = 10, b = 1$	98,58	95	96,3
$a = 2, b = 1$	94,34	93,68	91,68
$a = 1, b = 1$	92,22	88,84	93,99

datech z přístroje Toshiba. Při porovnání výsledků obou způsobů detekce je zřejmé, že detekční proces bude nejpřesnější při zapnutí funkce pro jednu možnou detekci. Dále je patrné, že kaskády klasifikátorů natrénované na přístroji Ultrasonix lze použít i pro detekci tepen pořízených z jiného přístroje (Toshiba), který má nastaveny jiné parametry.

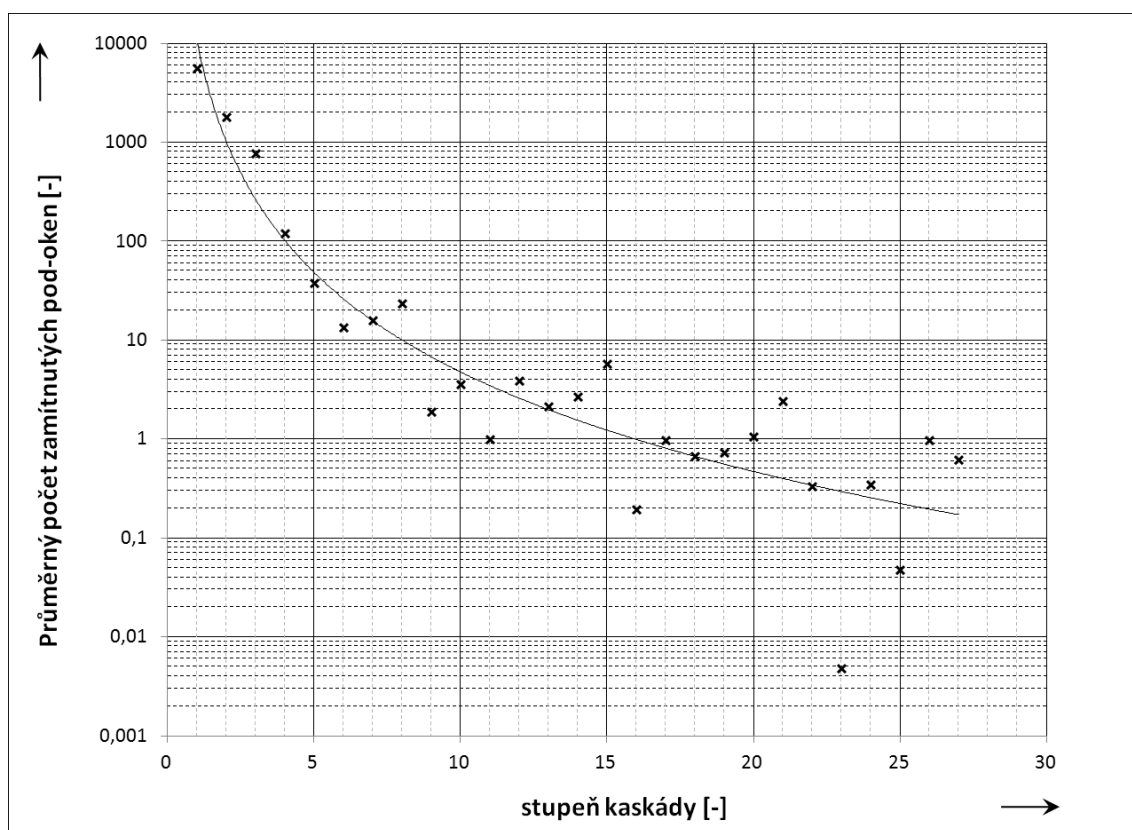
8.1.3 Vlastnosti zvolené kaskády klasifikátorů

V této kapitole bude popsána kaskáda klasifikátorů s nejlepší úspěšností detekce na testovacích datech. Jedná se o kaskádu, která byla natrénována pro fitness funkci (viz rovnice 4.3) s parametry $a = 1$, $b = 0$. Kaskáda se skládá z 27 stupňů a obsahuje celkem 338 lineárních klasifikátorů. První stupeň kaskády obsahuje 3 klasifikátory a je schopen zamítnout 66,48 % negativních pod-oken a detekovat téměř 100 % pozitivních pod-oken. Druhý stupeň obsahuje 6 klasifikátorů a zamítne 21,38 % negativních pod-oken. Třetí stupeň se skládá z 33 klasifikátorů a zamítne 9,06 % negativních pod-oken. Zbýlá negativní podokna jsou vyřazena na následujících stupních kaskády. Tyto výsledky byly získány zpracováním všech použitých testovacích i trénovacích snímků. Protože je 96,92 % pod-oken zamítnuto na prvních třech stupních kaskády, tak je detekční proces dostatečně rychlý. Na obrázku 8.4 jsou zobrazeny průměrné počty zamítnutých pod-oken v jednotlivých stupních kaskády. Obrázek 8.5 zobrazuje průměrné počty propuštěných pod-oken v jednotlivých stupních kaskády. Proces trénování této kaskády klasifikátorů trval 30 hodin.

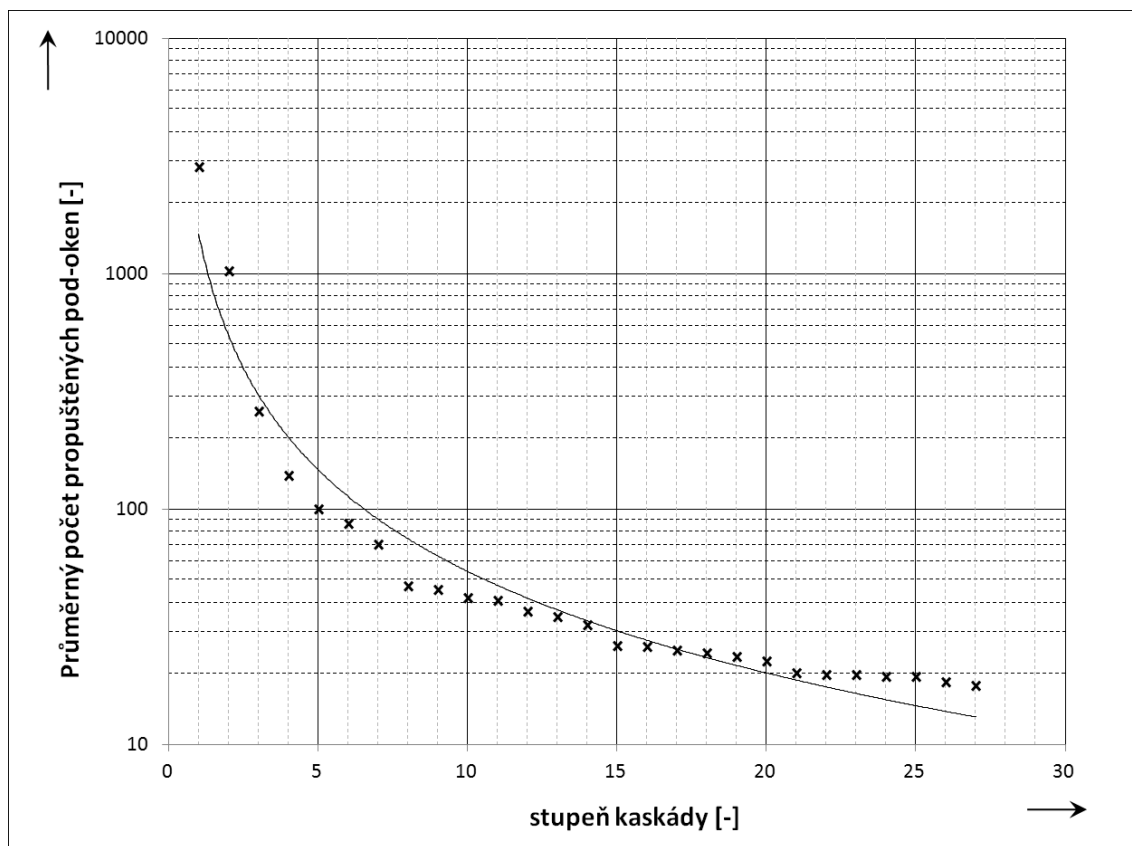
Pro standardní nastavení vstupních parametrů detektoru (viz kapitola 8.1.2) musí být ve snímku o rozlišení 388x400 pixelů kaskádou zpracováno 10 119 pod-oken. A pro snímek o rozlišení 286x322 pixelů musí být zpracováno 4 882 pod-oken. Při změně startovní velikosti detekčního okna na 48x48 pixelů (používá se pro dětské pacienty) musí být pro tento snímek zpracováno 15 755 pod-oken. Na obrázku 8.6 je v ukázkovém snímku tepny zobrazeno několik nejlepších Haarových příznaků vybraných algoritmem AdaBoost. Červeně je vyznačena oblast s váhou $w_0 = -1$ a modře oblast s váhou w_1 . Ze snímků je patrné jaké části tepny jednotlivé příznaky detekují.

Rychlost detekce

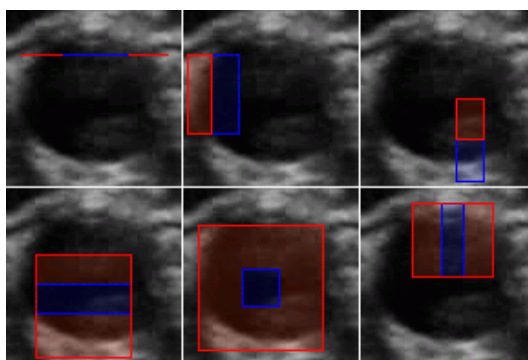
Pro snímky o rozlišení 286x322 pixelů, které byly získány z přístroje Toshiba, byla průměrná rychlost detekce 9 ms pro dospělé pacienty a 17 ms pro dětské pacienty. Pro snímky z přístroje Ultrasonix o rozlišení 388x400 pixelů byla průměrná rychlost detekce 13 ms. Parametry detekčního procesu byly nastaveny viz kapitola 8.1.2. Tato měření byla provedena na dvou-jádrovém procesoru Intel Core i5-680 s frekvencí 3,6 GHz.



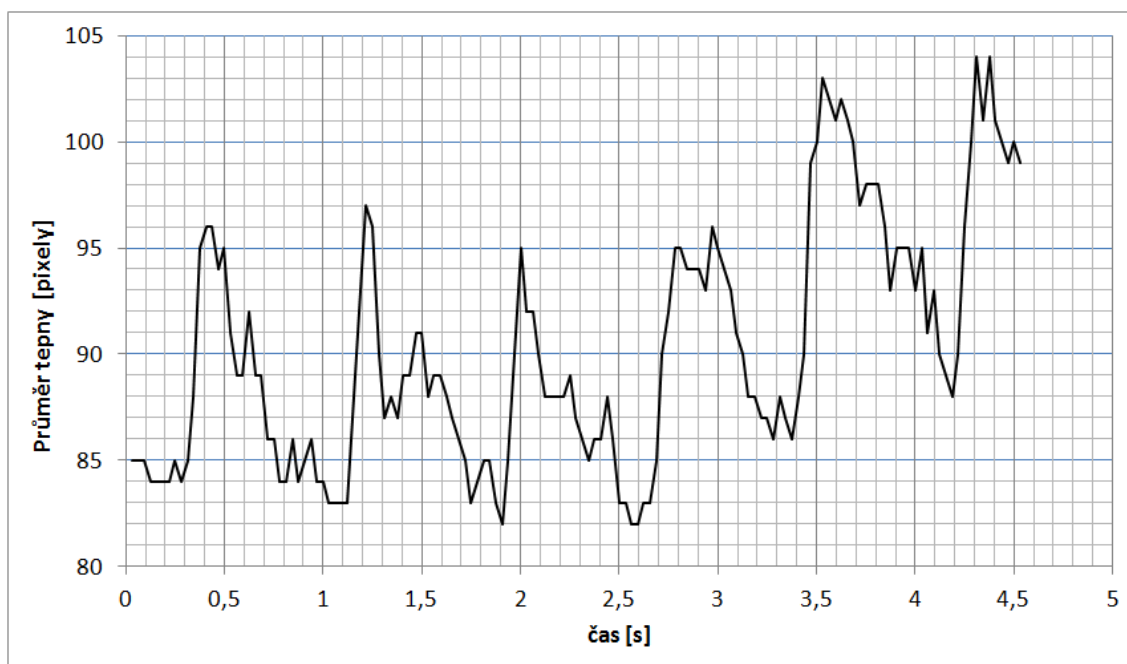
Obr. 8.4: Počty zamítnutých pod-oken v jednotlivých stupních kaskády.



Obr. 8.5: Počty propuštěných pod-oken v jednotlivých stupních kaskády.



Obr. 8.6: Ukázka několika prvních vybraných příznaků v příčném řezu tepny.



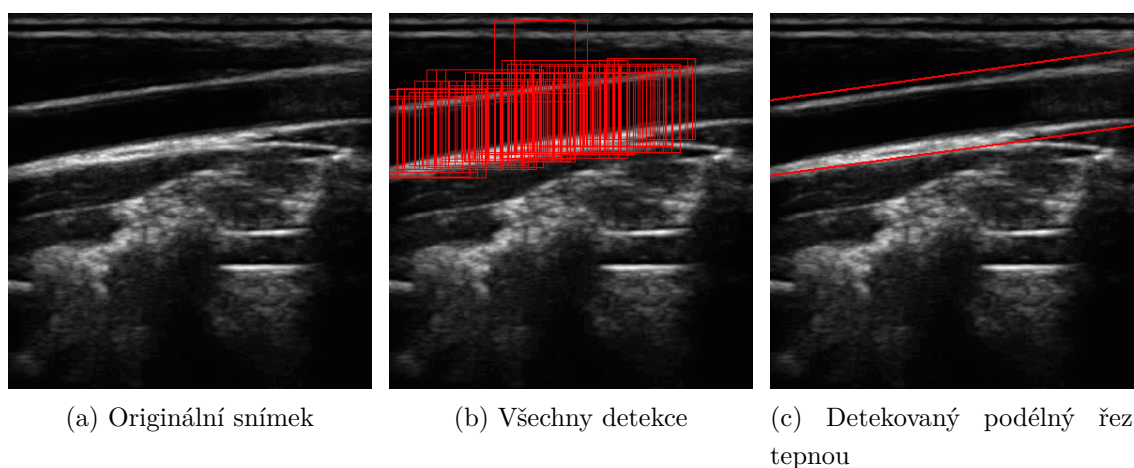
Obr. 8.7: Graf závislosti průměru tepny na čase – příčný řez.

Detekce z videosekvence

Tato kaskáda klasifikátorů byla použita pro detekci tepen z videosekvence pořízena na přístroji Ultrasonix. Videosekvence zobrazuje tepání tepny po dobu asi 4,5 s a celkový počet snímků, které byly uloženy pro detekci je 145. Postupným zpracování jednotlivých snímků byl pro každou nalezenou detekci ve snímku změřen její rozměr. Průběh tepové frekvence pacienta je zobrazen na obrázku 8.7. Z tohoto grafu jsou patrné jednotlivé tehy i měnící se průměr tepny. Výsledek je mírně zašumělý, což je dáno buď špatnou detekcí tepny, tím pádem je zobrazovaný rozměr detekované oblasti špatný (velká odchylka), nebo i při správné detekci tepny nemusí detekovaná oblast úplně odpovídat rozměrům tepny (malá odchylka). Proto vzhledem k práci [15] by mohlo zamýšlené využití navrženého detekčního algoritmu spočívat v provedení prvotní inicializace procedury měření průměru tepny v závislosti na čase, přičemž je důležité pouze označení oblasti (prstence), ve které se nachází hranice tepny a přesné určení pozice není podstatné. Po určení pozice by mohly být v tepně určeny významné body, které je pak možné v průběhu videosekvence sledovat. Tím je dosaženo přesnějších výsledků pro měření průměrů tepen.

8.2 Detekce podélných tepen

Detekce podélného řezu tepny je prováděna ze stejného důvodu jako detekce příčného řezu tepny (viz kapitola 8.1). Použité videosekvence byly zaznamenány na již zmíněném přístroji Ultrasonix. Na obrázku 8.8a je zobrazen originální snímek tepny v podélném řezu. Dále jsou zobrazeny všechny detekce (obrázek 8.8b). Nakonec jsou tyto detekce zpracovány a jsou vykresleny 2 přímky určující hranice tepny (obrázek 8.8c).



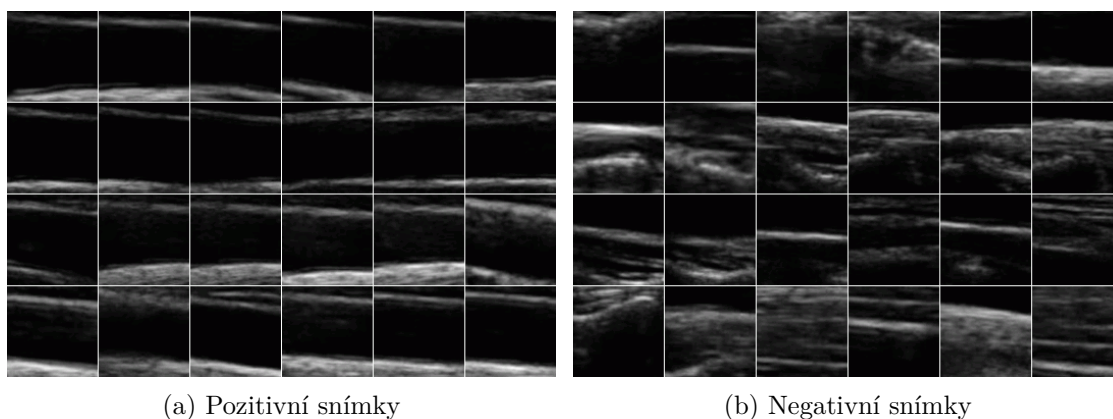
Obr. 8.8: Ukázka detekce podélného řezu tepny.

8.2.1 Použité databáze snímků

Testovací i trénovací databáze byly vytvořeny na přístroji Ultrasonix. Všechna data byla nasnímána na 15-ti dospělých pacientech, kde data 5-ti pacientů byla určena pro trénování a testování probíhalo na datech získaných od 10-ti pacientů.

Trénovací snímky

Pro vytvoření trénovacích dat bylo použito 15 videosekvencí. Z každé videosekvence byly vybrány 2 snímky, čímž vzniklo 30 velkých snímků. Následně byly k těmto snímkům vytvořeny textové soubory obsahující pozice hledaných tepen. Tyto pozice označují část podélného řezu tepny. Těchto pozic bylo v každém snímku označeno průměrně 9 a byly brány rovnoměrně z jedné strany řezem tepny k druhé. Na základě těchto souřadnic byly vygenerovány trénovací snímky. Trénovací databáze tedy čítá 260 pozitivních (obrázek 8.9a) a 180 negativních snímků (obrázek 8.9b). Trénování probíhalo na 440 snímcích.



Obr. 8.9: Ukázka trénovacích snímků podélných řezů tepen.

Testovací snímky

Testovací snímky byly vytvořeny za použití 45 videosekvencí. Z každé této videosekvence byly vybrány 3 snímky, čímž vzniklo celkem 135 testovacích snímků.

8.2.2 Výsledky testování – podélné tepny

Následující text popisuje nastavení parametrů detekčního procesu, při kterém byla otestována evolučně natrénovaná kaskáda klasifikátorů. Dále je uveden způsob testování a dosažené výsledky. V příloze jsou na obrázku B.2 zobrazeny správné detekce podélných řezů tepen. Dále jsou na obrázku B.5 zobrazeny falešně pozitivní detekce ve vybraných snímcích.

Parametry detekčního procesu

Detekční proces měl nastaveny tyto vstupní parametry: startovní velikost detekčního okna byla 72x72 pixelů, parametr zvětšení detekčního okna v každém kroku byl 1,1. Hodnota parametru posunu detekčního okna v obraze byla 0,1. Dále byla zapnuta funkce pro zpracování všech nalezených detekcí a následné vykreslení dvou přímkami ohraničujících oblast podélného řezu tepny.

Testování kaskády klasifikátorů

Pro detekci podélných řezů tepen byla natrénována s využitím genetických algoritmů jedna kaskáda klasifikátorů. Parametry rovnice 4.3 pro výpočet hodnoticí funkce byly nastaveny na $a = 10$ a $b = 1$. Při testování byla tepna určena jako správně detekovaná pokud vykreslené přímky správně určovaly její okraje. Pro tepny, které mají zakřivený tvar je obtížné určit jejich okraje pomocí přímek. V tomto případě

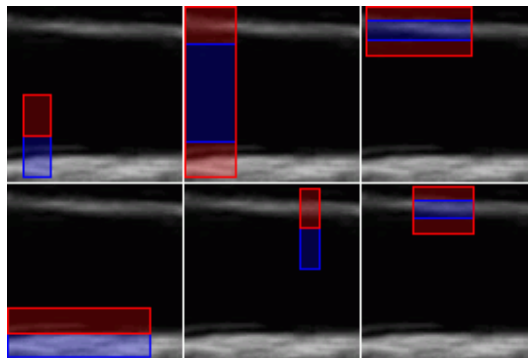
je jako správně detekovaná tepna určena ta, kde je správně označena alespoň její část. Řešením do budoucna by bylo na základě detekcí nechat místo přímky vykreslit například kubický splajn. Špatně detekovaná tepna je určena podle zobrazených přímk, které nekopírují okraje tepny. V tabulce 8.3 jsou zobrazeny výsledky přesnosti detekce podélných řezů tepen. Na testovacích snímcích byla dosažena přesnost detekce 96,26 %.

Tab. 8.3: Přesnost detekce kaskády klasifikátorů (pro detekci podélných řezů tepen).

Snímky	R_{TP} [%]
trénovací - (30)	95
testovací - (134)	96,26

Rychlost detekce

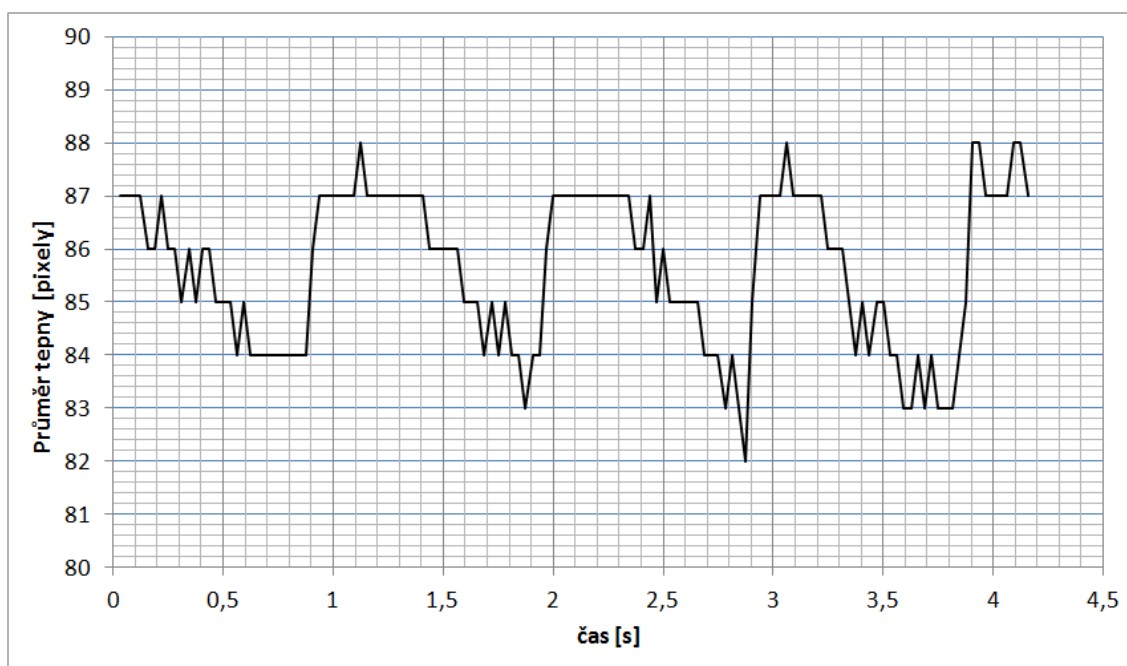
Při měření rychlosti detekce byly použity snímky o rozlišení 388x400 pixelů. Pro každý snímek muselo být kaskádou klasifikátorů zpracováno 10 119 pod-oken. Protože kaskáda dokázala na svém prvním stupni zamítnout 80 % pod-oken a na svém druhém stupni 14 % pod-oken, byl detekční proces rychlejší v porovnání s detekčním procesem příčných řezů tepen (platí pro snímky o rozlišení 388x400 pixelů). Průměrný čas detekce jednoho snímku byl 7 ms při nastavení vstupních parametrů detekčního procesu viz kapitola 8.2.2. Testování probíhalo na počítači s dvoujádrovým procesorem Intel Core i5-680 s frekvencí 3,6 GHz. Na obrázku 8.10 je v pozitivním snímku podélného řezu tepny zobrazeno několik nejlepších Haarových příznaků.



Obr. 8.10: Ukázka několika vybraných příznaků ve snímku podélného řezu tepnou.

Detekce z videosekvence

Z videosekvence, která zobrazuje tepání tepny po dobu asi 4,3 s bylo uloženo celkem 133 snímků. Dále byly v těchto snímcích detekovány tepny a měřen jejich průměr. Průběh tepové frekvence pacienta je zobrazen na obrázku 8.11. Minimální a maximální hodnoty v grafu určují průměr tepny při diastolickém a systolickém tlaku. Z grafu je patrný zašumělý průběh křivky. Při srovnání zašumění průběhů (viz obrázky 8.11 a 8.7) lze konstatovat, že tepový průběh při snímání podélného řezu tepny je méně zašumělý.

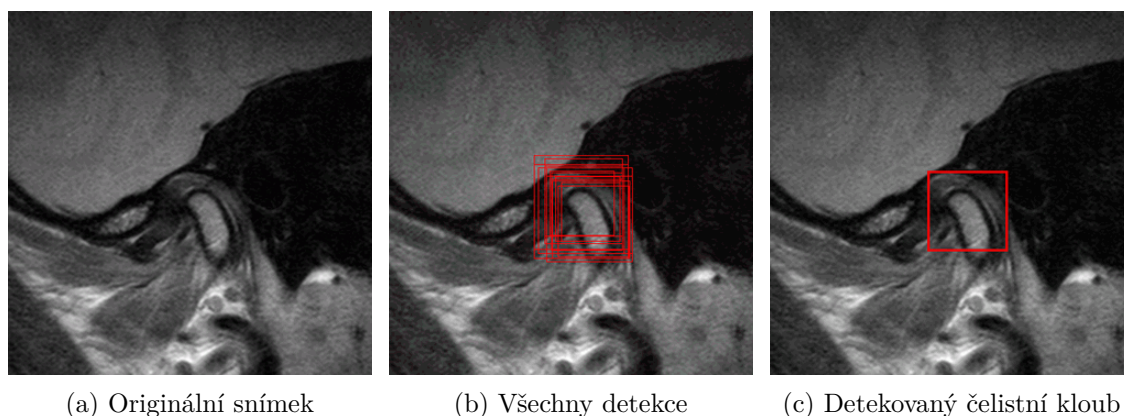


Obr. 8.11: Graf závislosti průměru tepny na čase – podélný řez.

8.3 Detekce čelistních kloubů

Dalším příkladem může být detekce čelistních kloubů ze snímků magnetické rezonance. Na základě detekce čelistního kloubu je možné změřit parametry chrupavky kloubu. Díky získaným parametrům je možné diagnostikovat nemoc jakou je porucha čelistního kloubu.

Všechny použité snímky byly získány z přístroje Philips Achieva 1,5 T. Na obrázku 8.12a je zobrazen vstupní snímek z magnetické rezonance. Dále jsou v něm zobrazeny všechny nalezené detekce (obrázek 8.12b) a nakonec jsou tyto detekce sloučeny v jednu nejvíce vypovídající (obrázek 8.12c).



Obr. 8.12: Ukázka detekce čelistního kloubu.

8.3.1 Použité databáze snímků

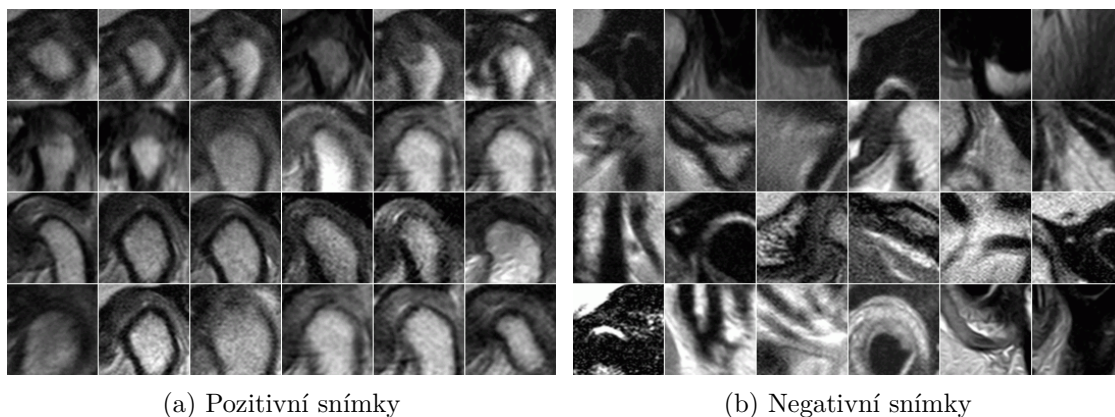
Ze snímků získaných z magnetické rezonance byly vytvořeny dvě databáze, trénovací a testovací. Všechna data byla nasnímána na osmi pacientech, kde snímky sedmi pacientů posloužily k trénování a snímky zbylého pacienta k testování. Snímky každého pacienta byly snímány od ucha směrem dále dovnitř hlavy. Rozteč mezi jednotlivými řezy (snímky) byla 2 mm. Pro trénování i testování byly vybrány pouze ty snímky, na kterých byl vidět čelistní kloub.

Trénovací snímky

Pro trénování bylo použito 70 snímků, ke kterým byly vytvořeny textové soubory obsahující pozice a velikosti hledaných čelistních kloubů. Následně byly vygenerovány trénovací pozitivní a negativní snímky o rozměrech 24x24 pixelů. Z každého velkého snímku byly vytvořeny 4 pozitivní snímky obsahující čelistní kloub. Tyto snímky tedy obsahují stejný čelistní kloub, který je ale v každém snímku zobrazen odlišně (mírně změněná velikost, posunutí). Tímto způsobem bylo z 70 velkých snímků získáno 280 pozitivních snímků (viz obrázek 8.13a) a 300 negativních snímků (viz obrázek 8.13b). Trénování tedy probíhalo na 580 snímcích.

Testovací snímky

Pro testování bylo použito 12 snímků jednoho pacienta. Tyto snímky mají rozlišení 640x640 pixelů.



Obr. 8.13: Ukázka trénovacích snímků příčných řezů tepen.

8.3.2 Výsledky testování – čelistní klouby

Z důvodů nedostatečného množství použitelných snímků pro trénování i testování, nejsou dosažené výsledky popsané v této kapitole tak vypovídající jako předchozí výsledky detekcí tepen. V příloze jsou zobrazeny správné detekce čelistních kloubů (viz obrázek B.3). Dále jsou na obrázku B.6 zobrazeny ve vybraných snímcích falešně pozitivní detekce.

Parametry detekčního procesu

Detekční proces měl nastaveny tyto vstupní parametry: startovní velikost detekčního okna byla 72x72 pixelů, parametr zvětšení detekčního okna v každém kroku byl 1,1. Hodnota parametru posunu detekčního okna v obraze byla 0,1. Dále byla přenastavena hodnota maximální velikosti detekčního okna na 144x144 pixelů. Změna této hodnoty byla provedena ve zdrojovém kódu. Tento krok byl nutný k eliminaci detekcí větších rozměrů, které znehodnocovaly celkový výsledek.

Testování kaskády klasifikátorů

Pro detekci čelistních kloubů byla natrénována s využitím genetických algoritmů jedna kaskáda klasifikátorů. Při evolučním skládání této kaskády byly pro výpočet hodnotící funkce nastaveny parametry rovnice 4.3 na $a = 1$ a $b = 0$. Tato kaskáda klasifikátorů byla testována dvěma způsoby: se započítáním všech možných detekcí a se započítáním pouze nejvýznamnější detekce. V tabulce 8.4 jsou zobrazeny výsledky přesnosti detekce pro oba dva způsoby testování. Z výsledků vyplývá, že kaskáda dosáhla 66,7 % úspěšnosti detekce na testovacích datech. Tato hodnota je nižší než hodnoty úspěšnosti kaskád klasifikátorů při detekci tepen. Důvodem této

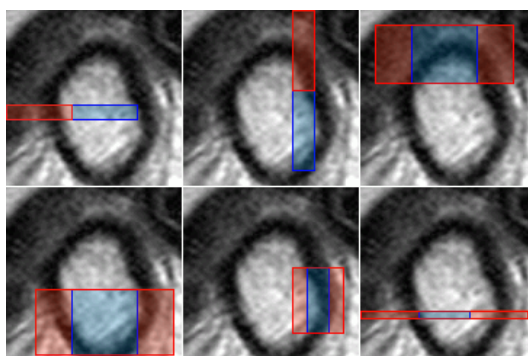
nižší úspěšnosti je omezené množství použitých dat i tvar čelistního kloubu, který se v každém řezu mění.

Tab. 8.4: Přesnost detekce kaskády klasifikátorů (pro detekci čelistních kloubů).

	1 detekce	více detekcí	
Snímky	R_{TP} [%]	R_{TP} [%]	R_{FP} [%]
trénovací - (70)	70	82,9	123
testovací - (12)	66,7	100	150

Rychlost detekce

Při standardním nastavení vstupních parametrů detekčního procesu (viz kapitola 8.3.2) musí být ve snímku o rozlišení 640x640 pixelů kaskádou zpracováno 30 271 pod-oken. Průměrná rychlost detekce činila 20 ms za použití počítače s procesorem Intel Core i5-680 s frekvencí 3,6 GHz. Na obrázku 8.14 je v pozitivním snímku čelistního kloubu zobrazeno několik nejlepších Haarových příznaků.



Obr. 8.14: Ukázka několika vybraných příznaků ve snímku čelistního kloubu.

8.4 Zhodnocení výsledků

Z dosažených výsledků je zřejmé, že nejlepší úspěšnost měl vytvořený algoritmus při trénování kaskády klasifikátorů pro detekci příčných řezů tepen, kde dosáhl úspěšnosti 97,21 % na testovacích datech z přístroje Ultrasonix a 96,99 % na testovacích datech z přístroje Toshiba. Tyto dosažené výsledky jsou poměrně vypovídající, protože bylo použito dostatečné množství různých trénovacích i testovacích snímků. Vytvořené algoritmy a natrénovaná kaskáda klasifikátorů by mohly být v budoucnu po jisté úpravě implementovány jako detekční program do některého z ultrazvukových přístrojů.

Kaskáda klasifikátorů natrénovaná pro detekci podélných řezů tepen dosáhla na testovacích datech přesnosti detekce 96,26 %. Tato hodnota přesnosti ale není zcela vypovídající kvůli malému množství trénovacích i testovacích dat. Při případném vylepšení metody pro zpracování detekcí u podélného řezu tepny, by mohly být výsledky přesnosti detekce lepší.

Nejhorší přesnost detekce (66,7 %) byla dosažena při testování kaskády klasifikátorů trénované pro detekci čelistních kloubů. V tomto případě bylo použito malé množství trénovacích i testovacích dat, ale vzhledem k proměnným tvarům čelistního kloubu v jednotlivých řezech snímků MRI, by ani použitím většího počtu trénovacích dat nebylo zřejmě dosaženo lepší přesnosti detekce. Použité metody nejsou tedy příliš vhodné pro detekci čelistních kloubů ze snímků magnetické rezonance.

Všechny natrénované detektory jsou schopné provádět detekci objektů v obraze na běžném počítači v reálném čase.

9 ZÁVĚR

Práce se zabývala detekcí objektů v obrazech. Cílem práce bylo seznámit se s metodou Viola–Jones, Haarovými příznaky a algoritmem AdaBoost. Dále měl být vytvořen v prostředí jazyka Java detektor objektů v obraze a jeho přesnost měla být ověřena na vybraných příkladech ze zdravotnictví.

V rámci práce byl vytvořen objektový detektor, který má implementován trénovací algoritmus AdaBoost a Haarovy příznaky. Pro program RapidMiner bylo vytvořeno osm operátorů, které jsou součástí Java knihovny „Image Processing Extension“ vyvíjené na Ústavu telekomunikací. Vytvořené operátory slouží: pro načítání sady obrázků, načítání kaskády klasifikátorů z *xml* souboru, vytváření trénovacích dat, natrénování modelu (kaskády klasifikátorů), testování natrénovaného modelu, zhodnocení úspěšnosti modelu, detekci objektů v obraze a úpravu natrénovaného modelu. Operátory vytvořené v této práci jsou navrženy tak, aby je bylo možné kombinovat i se standardními operátory RapidMineru. K jednotlivým operátorům byly vytvořeny návody k použití. Dále byly vytvořeny demonstrativní příklady použití operátorů spustitelné v RapidMineru. V rámci práce byly předpřipraveny trénovací databáze snímků podélných a příčných řezů tepen a snímků čelistních kloubů.

Hlavním přínosem této práce je rozšíření stávajícího Viola–Jones objektového detektoru o možnosti jeho trénování evolučními algoritmy. S využitím této metody se podařilo vytvořit detektor příčných řezů tepen s přesností 97 %, dále detektor podélných řezů tepen s přesností 96 % a detektor čelistních kloubů s přesností 67 %. Všechny natrénované detektory detekují objekty v obraze v reálném čase.

Zmíněný nástroj je v současné době součástí rozšíření RapidMineru, které je komerčně nabízeno v rámci spolupráce VUT a německé společnosti Rapid-I. Vytvořený algoritmus byl také již použit ve vědecké práci Dr. Manola Maragoudaka z „University of the Aegean“, kde byly detekovány některé body na sítnici oka. Dále byl algoritmus nasazen rakouskou firmou „Ditech“ pro vytvoření aplikace pro počítání osob v obchodě.

Na tuto práci lze do budoucna navázat například rozšířením sady Haarových příznaků, nebo přidáním jiných typů příznaků pro trénování detektoru. Dále může být detektor rozšířen o možnosti detekovat natočené objekty nebo detekovat více typů různých objektů.

LITERATURA

- [1] Ballard, D. H.; Brown, Ch. M. *Computer Vision*. Englewood Cliffs, N.J.: Prentice Hall, 1982.
- [2] Burghardt, Tilo; Calic, Janko. *Real-time Face Detection and Tracking of Animals*. [online], Neural Network Applications in Electrical Engineering, 2006
- [3] Dalal, N.; Triggs, B. *Histograms of oriented gradients for human detection*. [online], IEEE, Computer Vision and Pattern Recognition, 1990
- [4] Forsyth, D.; Ponce, J. *Computer Vision: A Modern Approach*. Prentice Hall; 2 edition (November 7, 2011)
- [5] Freund, Yoav; Schapire, Robert. *A decision-theoretic generalization of on-line learning and an application to boosting*. [online], In Computational Learning Theory: Eurocolt, 1995
- [6] Gallant, S.I. *Perceptron-based learning algorithms*. [online], IEEE, Transaction on Neural Networks, 1990
- [7] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [8] Han, Jiawei; Kamber, Micheline. *Data mining: concepts and techniques*. The Morgan Kaufmann Series in Data Management Systems, 2006
- [9] Chang, Huang; Haizhou, Al; Bo, Wu; Shihong, Lao. *Boosting Nested Cascade Detector for Multi-View Face Detection*. [online], ICPR, 2004
- [10] Jensen, O. *Implementing the Viola-Jones Face Detection Algorithm*. [online], Technical University of Denmark, 2008
- [11] Jun-Su, Jang; Jong-Hwan, Kim. *Fast and Robust Face Detection Using Evolutionary Pruning*. [online], IEEE Transactions on Evolutionary Computation, 2008
- [12] Lienhart, R.; Maydt, J. *An Extended Set of Haar-like Features for Rapid Object Detection*. [online], IEEE ICIP, 2002
- [13] Zaidi, N.A.; Suter, D. *Object Detection Using a Cascade of Classifiers*. [online], Digital Image Computing: Techniques and Applications, 2008
- [14] Přinosil, Jiří; Krolikowski, Martin. *Využití detektoru Viola-Jones pro lokalizaci obličejů a očí v barevných obrazech*. [online], Elektrorevue, 2008

- [15] Říha K, Potůček I. *The Sequential Detection of Artery Sectional Area Using Optical Flow Technique*. Proceedings of the 8th WSEAS International Conference on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING. Puerto De La Cruz, 14–16 December 2009.
- [16] Shen, Chunhua; Wang, Peng; Hengel, A. *Optimally Training a Cascade Classifier*. [online], CVPR, 2010
- [17] Shengye, Yan; Shiguang, Shan; Xilin, Chen; Wen, Gao; Jie, Chen. *Matrix-Structural Learning (MSL) of Cascaded Classifier from Enormous Training Set*. [online], CVPR, 2007
- [18] Szasz-Toth, Lorant. *Combination of Classifier Cascades and Training Sample Selection for Robust Face Detection*. [online], Universität Karlsruhe, 2009
- [19] Viola, P.; Jones, M. *Fast Multi-view Face Detection*. [online], Mitsubishi Electric Research Lab, 2003
- [20] Viola, P.; Jones, M. *Rapid Object Detection using a Boosted Cascade of Simple Features*. [online], IEEE CVPR, 2001
- [21] Viola, P.; Jones, M. *Robust Real-Time Face Detection*. [online], Int. J. Comput. Vision 57(2), (May 2004)
- [22] Weiwei, Zhang; Jian, Sun; Xiaoou, Tang. *From Tiger to Panda: Animal Head Detection*. [online], IEEE Transactions on Image Processing, 2011
- [23] Wu, Jianxin; Mullin, Matthew D; Rehg, James M. *Linear Asymmetric Classifier for cascade detectors*. [online], ICML, 2005
- [24] Ying, Ying; Han, Wang; Jian, Xu. *An automatic system for multi-view face detection and pose estimation*. [online], ICARCV, 2010

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

α	váha lineárního slabého klasifikátoru
C_F	hodnoticí (fitness) funkce
C_M	Matthewsův korelační koeficient
ϵ	chyba lineárního slabého klasifikátoru
EA	evoluční algoritmy
f	feature – příznak
HOG	Histogram of Oriented Gradients – histogramy orientovaných gradientů
HOOG	Haar of Oriented Gradients
H	množina všech lineárních slabých klasifikátorů
$h(x)$	slabý lineární klasifikátor aplikovaný na vstupní snímek x
$H(x)$	silný klasifikátor aplikovaný na vstupní snímek x
i_I	integral image – integrální obraz
i_{ISqr}	kvadrát hodnot integrálního obrazu
JGAP	Java Genetic Algorithms Package – Java knihovna genetických algoritmů
LAC	Linear Asymmetric Classifier – lineární asymetrický klasifikátor
MSL	Matrix-Structural Learning – metoda maticové struktury trénování
MRI	magnetic resonance imaging – magnetická rezonance
N_C	počet všech lineárních klasifikátorů v kaskádě
N_{CS}	maximální počet klasifikátorů v jednom stupni kaskády
N_{FN}	počet falešně negativních snímků
N_{FP}	počet falešně pozitivních snímků
N_I	celkový počet snímků
N_S	maximální počet stupňů kaskády
N_{TN}	počet správně klasifikovaných negativních snímků

N_{TP}	počet správně klasifikovaných pozitivních snímků
OpenCV	Open Source Computer Vision
p	polarita
P	prahová hodnota stupně kaskády
R_{FP}	míra falešně pozitivních detekcí
RM	RapidMiner
R_{TP}	přesnost detekce
s	kumulovaný součet hodnot pixelů v řádku
Θ	threshold – prahová hodnota slabého lineárního klasifikátoru
US	ultrasound – ultrazvukový
w	váha trénovacího snímku x
x	vstupní snímek
XML	Extensible Markup Language
y	klasifikační třída snímku x

SEZNAM PŘÍLOH

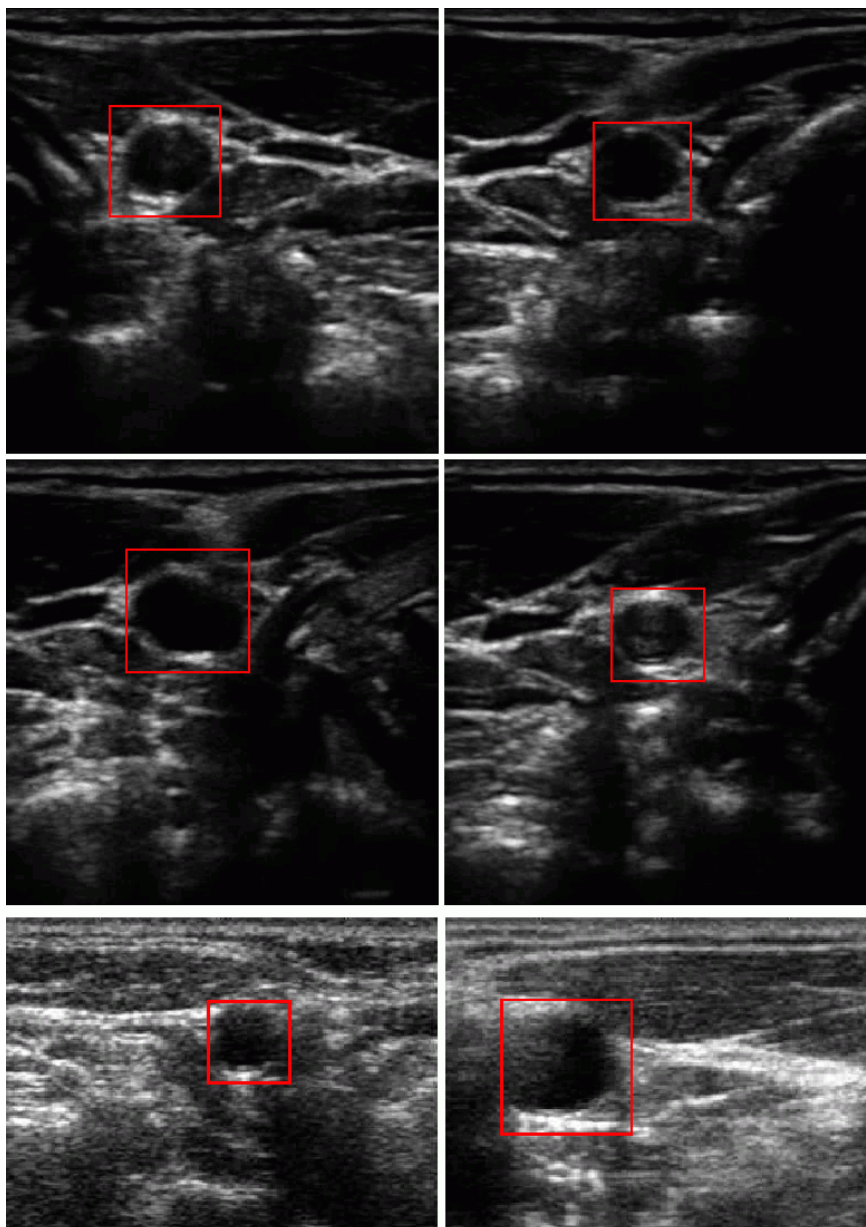
A	Obsah CD	68
B	Ukázky detekovaných objektů	69
B.1	Správně detekované objekty	69
B.2	Falešně pozitivní detekce	72

A OBSAH CD

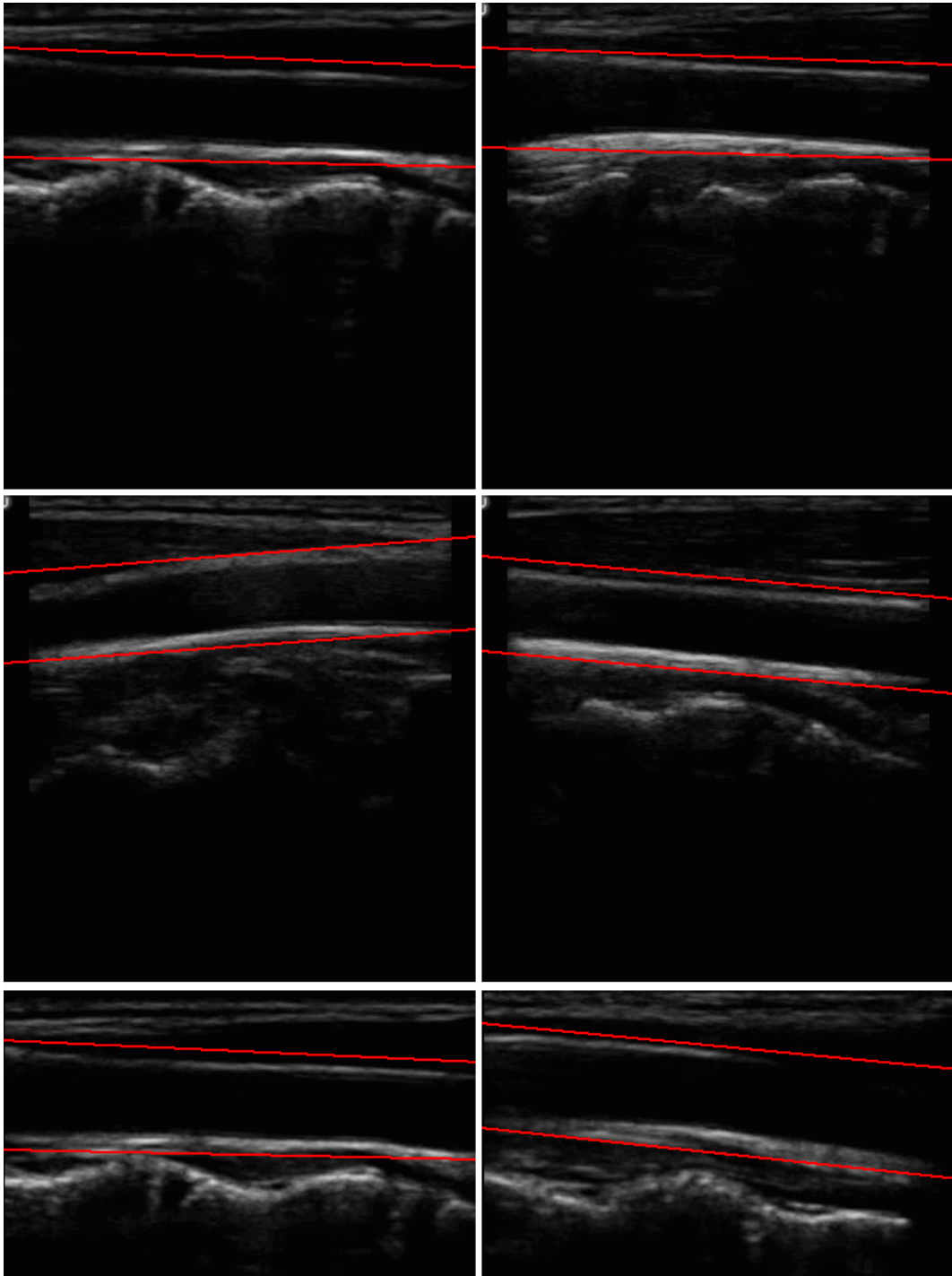
- **xmasek10_DP.pdf** – Elektronická verze této práce ve formátu PDF.
- **Data.zip** – Archiv obsahuje trénovací data, testovací data a natrénované kaskády klasifikátorů (ve formátu *xml*). Jedná se o databáze příčných a podélných tepen a čelistních kloubů.
- **ukazkove_priklady.zip** – Demonstrační příklady spustitelné v RM.
- **ImageProcessingExtension.zip** – Archiv obsahuje zdrojové kódy vytvořené v rámci této práce. Současně obsahuje i zdrojové kódy celého rozšíření.
- **rapidminer-ImageMiner-1.3.3.jar** – Knihovna obsahující vytvořené algoritmy spustitelná v RM.
- **RapidMiner_Vega.zip** – Archiv obsahuje zdrojové kódy programu RapidMiner.
- **rapidminer-5.2.003x32-install.exe** – Instalační soubor pro operační systém Windows – 32 bitová verze.
- **rapidminer-5.2.003x64-install.exe** – Instalační soubor pro operační systém Windows – 64 bitová verze.

B UKÁZKY DETEKOVANÝCH OBJEKTŮ

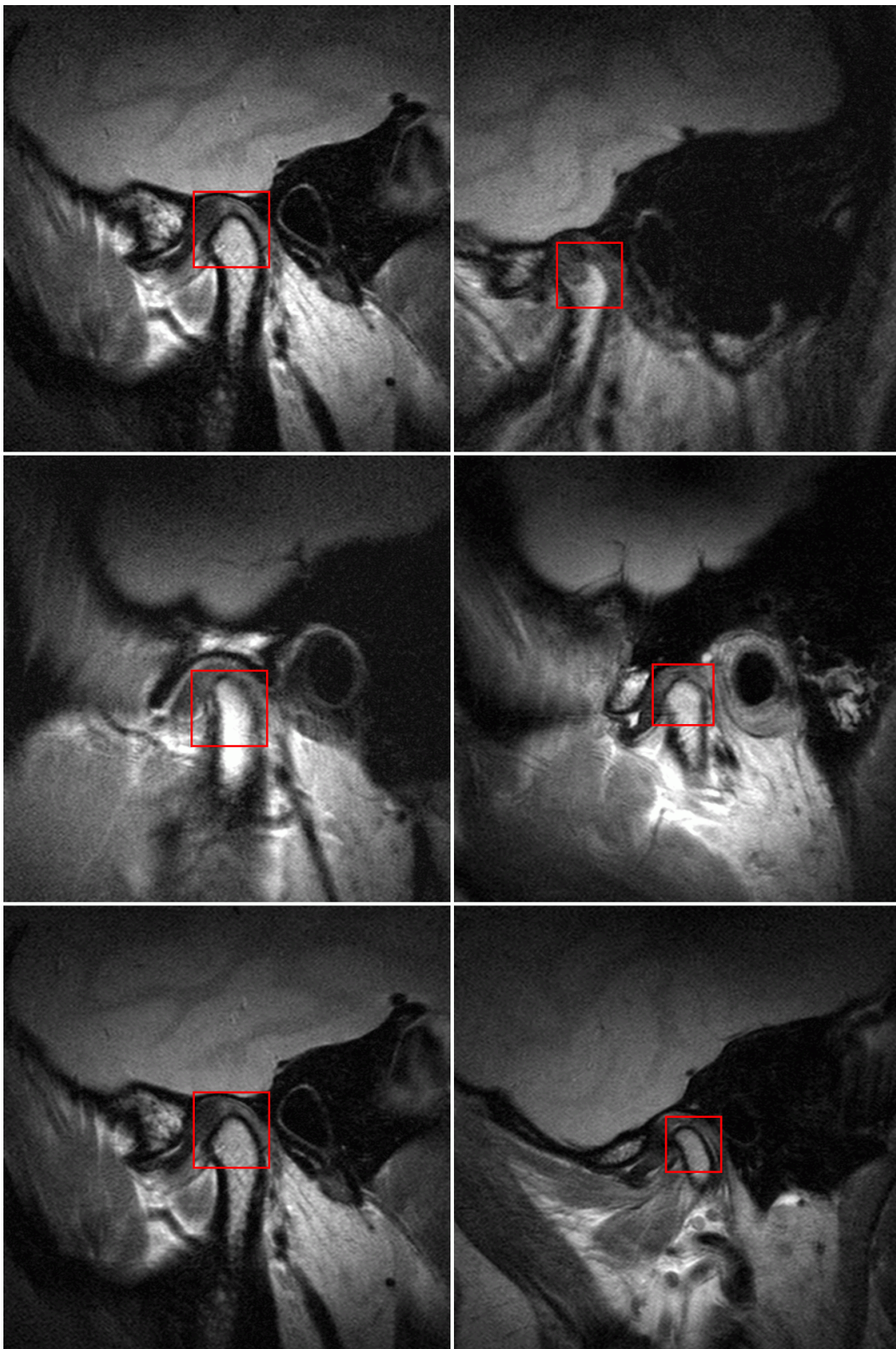
B.1 Správně detekované objekty



Obr. B.1: Ukázka správně detekovaných příčných řezů tepen.

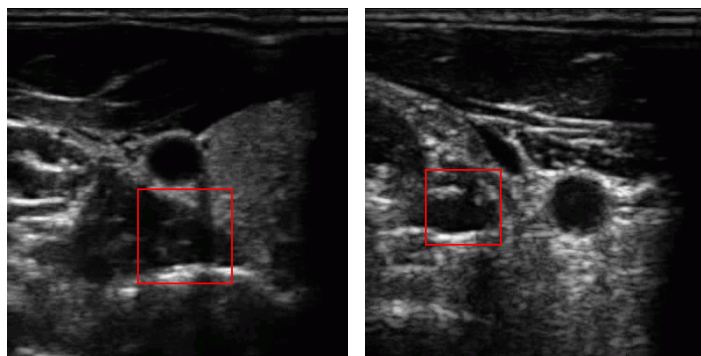


Obr. B.2: Ukázka správně detekovaných podélných řezů tepen.

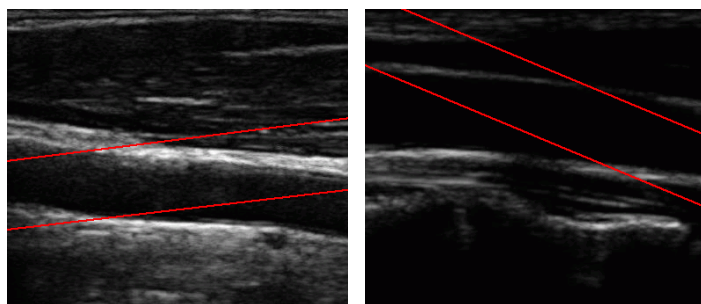


Obr. B.3: Ukázka správně detekovaných čelistních kloubů.

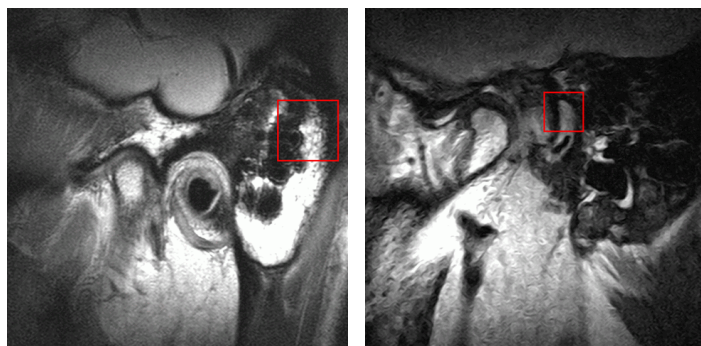
B.2 Falešně pozitivní detekce



Obr. B.4: Tepny příčné – falešně pozitivní detekce.



Obr. B.5: Tepny podélné – falešně pozitivní detekce.



Obr. B.6: Čelistní klouby – falešně pozitivní detekce.