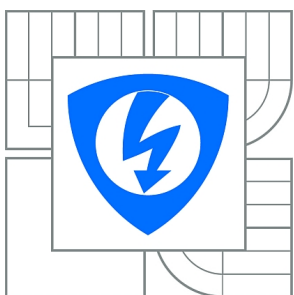


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE SKUPIN OSOB V DATABÁZI OBRAZŮ

DETECTION OF GROUPS OF PEOPLE IN IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

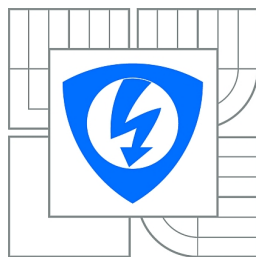
ONDŘEJ MIKULČÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR ČÍKA, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Ondřej Mikulčík

ID: 120930

Ročník: 3

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Detekce skupin osob v databázi obrazů

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte současný stav a techniky pro detekci objektů v obraze. Zaměřte se zejména na detekci skupiny podobných objektů. Navrhněte metodu, která bude vhodná k detekci skupin osob na předloženém snímku. Tento návrh implementujte v programovém prostředí RapidMiner, ověřte jeho výsledky a porovnejte je s výsledky současných metod.

DOPORUČENÁ LITERATURA:

[1] BURGER, Wilhelm; BURGE, Mark J. Principles of Digital Image Processing: Fundamental Techniques. Londýn : Springer, 2009. 272 s. ISBN 978-1848001909.

[2] GONZALEZ, Rafael C.; WOODS, Richard E. Digital Image Processing. 3. Londýn : Pearson Pentice Hall, 2008. 954 s. ISBN 978-0-13-505267-9.

Termín zadání: 6.2.2012

Termín odevzdání: 31.5.2012

Vedoucí práce: Ing. Petr Číka, Ph.D.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá dvěma metodami pro detekci objektů v obrazech. První metodou je Viola-Jones, druhou je metoda histogramů orientovaných gradientů. Začátek práce se zabývá teoretickým popisem metod. V dalších částech je prezentována tvorba trénovacích databází, implementace metod samotných a implementace nových metod pro detekci skupin osob v programu RapidMiner. V závěru jsou zhodnoceny výsledky z testování a využití metod pro detekci skupin osob v databázi obrazů.

KLÍČOVÁ SLOVA

Detekce obličejů, postav a jejich skupin, AdaBoost, Viola-Jones, Histogramy orientovaných gradientů, SVM

ABSTRACT

This work describes two methods for detecting objects in images. The first method is the Viola-Jones, the second is the method of histograms oriented gradients. Start of work deals with the theoretical description of the methods. In the other parts of this work is presented creation of the training databases, implementation methods and implementation of new methods for group detection in the RapidMiner. In conclusion, the results and the use of methods for detection of groups of people in the database of images are evaluated.

KEYWORDS

Detection of faces, figures and groups, AdaBoost, Viola-Jones, Histograms of oriented gradients, SVM

MIKULČÍK, Ondřej *Detekce skupin osob v databázi obrazů*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2011. 41 s. Vedoucí práce byl Ing. Petr Číka, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Detekce skupin osob v databázi obrazů“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Chtěl bych poděkovat Bc. Janu Maškovi a Bc. Martinu Králíkovi za cenné rady ohledně detekčních algoritmů, a dále mému vedoucímu Ing. Petru Číkovi, Ph.D. za cenné připomínky a odborný dohled při vypracování této bakalářské práce.

Brno

.....

(podpis autora)

Výzkum popsáný v této bakalářské 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	11
1 Metody detekce hran v obraze	12
1.1 Metoda Viola-Jones	12
1.1.1 AdaBoost	12
1.1.2 Haarovy příznaky	14
1.1.3 Integrální obraz	15
1.1.4 Vlastní detekce pomocí kaskády	16
1.2 Metoda HOG	17
1.2.1 Předzpracování obrazu	17
1.2.2 Histogramy orientovaných gradientů a jejich výpočet	17
1.2.3 Orientace gradientů	18
1.2.4 Normalizace a rozdělení bloků, tvorba vektoru	18
1.2.5 SVM	19
2 Tvorba databáze	20
2.1 Tvorba databáze pro Viola-Jones	20
2.1.1 Tvorba databáze pro pozitivní příznaky	20
2.1.2 Tvorba databáze pro negativní příznaky	20
2.2 Tvorba databáze pro HOG	21
2.2.1 Tvorba databáze pro pozitivní příznaky	21
2.2.2 Tvorba databáze pro negativní příznaky	21
3 Implementace v RapidMineru	22
3.1 Prostředí RapidMineru	22
3.2 Implementace Viola-Jones	22
3.3 Implementace HOG	23
4 Výsledky testování	25
4.1 Testování Viola-Jones	25
4.2 Testování HOG	26
5 Nové metody pro detekci skupin osob v obraze	28
5.1 Princip metod	28
5.1.1 Nalezení všech bodů zájmu	29
5.1.2 Výběr bodů zájmu patřících do skupiny	29
5.1.3 Výstup	30

5.2	Implementace operátoru do RapidMineru	30
5.3	Implementace operátoru v RapidMineru	31
5.4	Vstupní parametry operátoru	32
5.4.1	Display	32
5.4.2	Object color a Group color	33
5.4.3	Min. people	33
5.4.4	Distance	33
5.5	Výstup operátoru	34
6	Testování nových metod	35
6.1	Testování Group Visualizeru	35
6.2	Testování s algoritmem Viola-Jones	36
6.3	Testování s algoritmem HOG	38
7	Závěr	39
	Literatura	40
	Seznam symbolů, veličin a zkratk	41

SEZNAM OBRÁZKŮ

1.1	a) 1. slabý klasifikátor, b) 2. slabý klasifikátor, c) silný klasifikátor . . .	14
1.2	Haarovy příznaky: a) hranové, b) čárové, c) diagonální	14
1.3	Haarovy příznaky na obličejích	15
1.4	Vlevo vstupní obraz, vpravo integrální obraz	15
1.5	Výpočet hodnot jasů vyznačeného obdélníku	16
1.6	Blokové schéma zapojení klasifikátorů do kaskády	16
1.7	Algoritmus HOG	17
1.8	Detekce hran: a) originální obrázek, b) Sobelův operátor, c) Robert- sův operátor, d) Laplaceův operátor, e) rozdíl Gaussových rozostření	18
1.9	Rozdělení negativních a pozitivních příznaků pomocí nadroviny . . .	19
2.1	Příklad pozitivní databáze pro Viola-Jones	20
2.2	Příklad negativní databáze pro Viola-Jones	21
2.3	Příklad pozitivní databáze pro HOG	21
3.1	Trénovací algoritmus Viola-Jones	22
3.2	Detekční algoritmus Viola-Jones	23
3.3	Trénovací algoritmus modulární HOG	23
3.4	Trénovací algoritmus diffusion HOG	24
3.5	Detekční algoritmus HOG	24
4.1	Příklad pozitivních detekcí Viola-Jones	25
4.2	Výsledky třífázového testování Viola-Jones	26
4.3	Příklad pozitivních detekcí HOG	27
4.4	Výsledky dvoufázového testování HOG	27
5.1	Blokové schéma operátoru	28
5.2	Výběr bodů zájmu	29
5.3	Výpočet vyznačení skupiny	31
5.4	Group Visualizer v RapidMineru	32
5.5	Implementace Group Visualizer s detektorem Viola-Jones	32
5.6	Vstupní parametry	33
5.7	Výstup operátoru Group Visualizer do konzole	34
5.8	Ideální grafický výstup Group Visualizeru	34
6.1	Test Group Visualizeru	35
6.2	Test Group Visualizeru	36
6.3	Test Group Visualizeru pomocí Viola-Jones	36
6.4	Test Group Visualizeru pomocí Viola-Jones	37
6.5	Test Group Visualizeru pomocí Viola-Jones	37
6.6	Test Group Visualizeru pomocí HOG	38
6.7	Test Group Visualizeru pomocí HOG	38

SEZNAM TABULEK

4.1	Výsledky testování Viola-Jones	25
4.2	Výsledky testování HOG	27

ÚVOD

S rozvojem výpočetní techniky rostou i možnosti programovacích jazyků. Trendem dnešního světa je předat co nejvíce zodpovědnosti z člověka na počítač, protože je rychlejší, výkonnější a ekonomičtější. V oboru nazývaném počítačové vidění je snahou programátorů, aby počítače „porozuměly“ grafické informaci. To ovšem na rozdíl od textové informace není vůbec jednoduché. Zatímco v textu se počítač orientuje velmi snadno, „rozumět“ obrázku se musí nejprve naučit.

Velmi důležitou součástí je detekce osob. V době Internetu a digitálních fotoaparátů se dá využít pro usnadňování práce s fotoalbami, ve videu může na průmyslových kamerách odhalovat nepovolané osoby.

Tato technologie je založena na strojově se učících algoritmech, které rozpoznávají objekty podle natrénované množiny pozitivních a negativních obrázků. Každý učící se algoritmus se natrénovaný objekt snaží detekovat pomocí různých příznaků. Tato práce se bude zabývat dvěma detekčními algoritmy, na kterých budou postaveny nové metody pro detekci skupin osob. Cílem bude rozhodnout, který detekční algoritmus je k detekci skupin osob vhodnější.

Tato bakalářská práce je součástí projektu výzkumné skupiny Signal Processing Laboratory (SPLab), kterou vede Ing. Radim Burget, Ph.D. Snahou této skupiny je začlenit obrazové techniky detekce do programu RapidMiner, který je kompletně napsán v jazyce Java a je volně šířitelný. Díky tomu jsou detekční algoritmy dostupné všem lidem a ne jen firmám vyvíjejícím jejich komerční verze. Využití těchto technologií je velmi široké a mohlo by být obrovským přínosem.

1 METODY DETEKCE HRAN V OBRAZE

1.1 Metoda Viola-Jones

Poprvé tento objektový detektor představili P. Viola a M. Jones už v roce 2001. Pracuje se šedo-tónovými obrazy a skládá se ze tří základních částí, které budou popsány v následující podkapitole. Pro přehlednost jsou to: integrální obraz, Haarovy příznaky a učící se algoritmus AdaBoost [3]. Metoda Viola-Jones je ve světě proslulá a osvědčila se zejména v oblasti detekce obličejů, a to hlavně kvůli své rychlosti a spolehlivosti.

1.1.1 AdaBoost

Aby se vůbec mohlo cokoliv v grafické informaci detekovat, musí se nejprve algoritmu říci, co má hledat. K tomuto účelu slouží učící se algoritmus Adaptive Boosting (AdaBoost). Ten si umí rozřadit trénovací data do dvou tříd (pozitivní, negativní) a při samotné detekci rozlišit, která informace patří do které třídy. Klasifikační algoritmus AdaBoost vychází z metody strojového učení. Základem je čerpání tzv. slabých klasifikátorů z trénovací množiny, jejichž přesnost se pohybuje kolem 50%. Jakmile jsou vybírány další slabé klasifikátory s obdobnou přesností, vzniká z nich tzv. silný klasifikátor, jehož klasifikační přesnost je proměnlivě vysoká v závislosti na datech v trénovací množině. Snahou učícího se algoritmu AdaBoost je tedy vytvořit ze slabých klasifikátorů H_t co nejpřesnější silný klasifikátor $H(x)$, který vznikne jejich lineární kombinací [3].

Trénovací množina S , která je vstupem algoritmu, se skládá z dvojice $(x_1, y_1), \dots, (x_m, y_m)$, kde x_m odpovídá získané hodnotě příznaku a y_m odpovídá třídě příznaku. Třídy jsou v našem případě dvě, což znamená, že y_i může nabývat hodnot $\{-1, 1\}$. Hodnotu 1 získává klasifikátor, který označuje hledaný objekt, zatímco -1 všechny ostatní. Dále AdaBoost používá cyklus, který z trénovací množiny vybírá slabé klasifikátory váhami D_t . Tyto váhy jsou na začátku nastaveny na stejné hodnoty. Učící se algoritmus obsahuje čtyři body a úkolem každého cyklu je určit právě jeden odpovídající slabý klasifikátor [3]. Algoritmus pro vstupní množinu $S = (x_1, y_1), \dots, (x_m, y_m)$ nejprve inicializuje váhy pomocí rovnice

$$D_1(i) = \frac{1}{m}, \quad (1.1)$$

poté může začít algoritmus samotný pro $t = 1 \dots T$ (T značí počet iterací cyklu):

1. Výběr nejlepšího slabého klasifikátoru (s nejmenší chybou ε_t), při daných vahách D_t .
2. Ověření chyby ε_t vybraného klasifikátoru (musí být menší než 0.5).
3. Výpočet váhy α_t slabého klasifikátoru v lineární kombinaci $H(x)$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right). \quad (1.2)$$

4. Aktualizace vah D_t

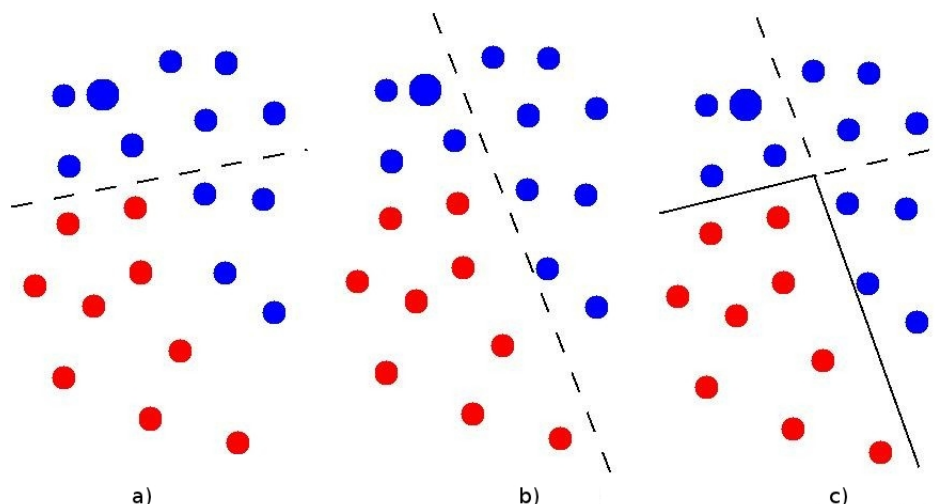
$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}, \quad (1.3)$$

kde Z_t je normalizační faktor [5].

Slabý klasifikátor tedy není vybírán náhodně, ale vždy je vybrán ten, který má v trénovací množině nejmenší chybu ε_t . K zajištění konvergence je potřeba, aby tato chyba byla menší než 0,5. Výpočet koeficientu slabého klasifikátoru v lineární kombinaci α_t je navržen tak, že v každém kroku učení je snaha minimalizovat odhad chyby výsledného klasifikátoru. Na konci cyklu je provedena aktualizace vah, která způsobí, že se váha špatně klasifikovaných měření zvětší a váha správně klasifikovaných zmenší. V příštím kroku tedy bude hledán takový slabý klasifikátor, který lépe klasifikuje doposud špatně provedená měření. To znamená exponenciální redukci chyby v závislosti na rostoucí velikosti trénovací množiny. Výsledkem je silný klasifikátor

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right). \quad (1.4)$$

Toto je velká výhoda oproti jiným strojově se učícím algoritmům. Avšak může dojít k jevu zvanému přetrénování, což znamená ztrátu rozpoznávání natrénovaných klasifikátorů až přílišnou mírou natrénovaných dat. Naštěstí se tak stává velice zřídka.[3] Na obrázku 1.1 je příklad tvorby silného klasifikátoru.



Obr. 1.1: a) 1. slabý klasifikátor, b) 2. slabý klasifikátor, c) silný klasifikátor

1.1.2 Haarovy příznaky

V předchozí kapitole bylo popsáno, jak algoritmus AdaBoost postupuje při výběru klasifikátorů. Jako takový klasifikátor používá Viola-Jones odezvu na tzv. Haarovy příznaky. Tyto příznaky pracují na rozdílech jasu a využívají se hlavně kvůli velmi rychlé výpočetní rychlosti. Haarovy příznaky jsou tvořeny obdélníkovými oblastmi různých typů. [3]. Příklady jsou vidět níže na obr. 1.2:



Obr. 1.2: Haarovy příznaky: a) hranové, b) čárové, c) diagonální

Velikost těchto příznaků je různá, od nejmenší možné se zvětšuje až na velikost vstupního obrazu. Než AdaBoost začne vybírat klasifikátor s nejmenší chybou, vygeneruje množinu všech příznaků v obraze. V okně 24 x 24 pixelů se jich může vyskytovat až 160 000, což není zase takový problém hlavně díky již zmiňované výpočetní rychlosti. Číselné vyjádření Haarova příznaku je určeno rozdílem sumy pixelů obrázku odpovídajících tmavé části od sumy pixelů obrázku odpovídajících světlé části [3]. Příklad je uveden na obrázku 1.3.



Obr. 1.3: Haarovy příznaky na obličejí

1.1.3 Integrální obraz

Již ve velmi malých obrázcích se vyskytují Haarovy příznaky v řádech desítek tisíc. Aby byl výpočet hodnot jasů co nejjednodušší, provádí se v tzv. integrálním obraze. Ten vytvoříme podle rovnic

$$s(x, y) = s(x, y - 1) + I(x, y), \quad (1.5)$$

$$I_{int}(x, y) = I_{int}(x - 1, y) + s(x, y). \quad (1.6)$$

Slovně to znamená, že každá jasová hodnota pixelu je součtem všech jasových hodnot pixelů předcházejících [3], viz obrázek 1.4.

0	1	1	1	0	1	1	1
1	2	2	3	1	4	7	3
1	2	1	1	1	7	11	16
1	3	1	0	1	3	16	21

Obr. 1.4: Vlevo vstupní obraz, vpravo integrální obraz

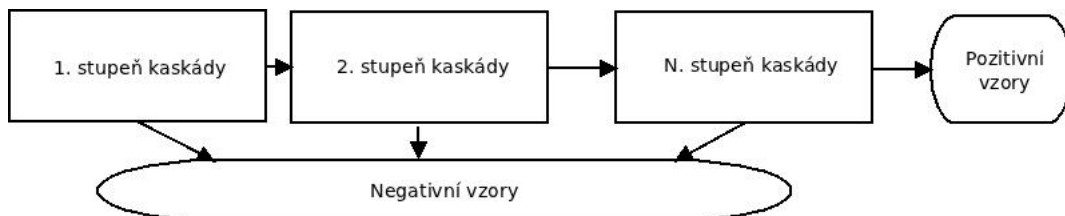
Nejvyšší hodnota jasů v integrálním obraze vždy ustane v pravém dolním rohu. K výpočtu sumy hodnot jasů libovolného obdélníku v obraze postačí jeho rohové hodnoty jasů, na které budou použity tři operace sčítání nebo odčítání [3]. K výpočtu obdélníku na obr. 1.5 jsou potřeba následující operace: $16 - 3 - 1 + 1 = 13$.

0	1	1	1
1	4	7	3
1	7	11	16
1	3	16	21

Obr. 1.5: Výpočet hodnot jasů vyznačeného obdélníku

1.1.4 Vlastní detekce pomocí kaskády

Klasifikátor který posuzuje, jestli na daném vstupu je, nebo není hledaný objekt, nebere detekovaný obraz jako celek. Snaží se proskenovat celý obrázek pohybujícími se subokny, které mění polohu i velikost, aby určil, zda-li se na něm daný objekt vyskytuje. Respektive porovnává příznaky v subokně s pozitivními a negativními příznaky v trénovací množině. Toto je výpočetně velice náročné, (pro obrázek 320 x 240 px je nutno zpracovat přibližně 500 000 suboken), a proto se tyto data zpracovávají tzv. kaskádovým zapojením více klasifikátorů [3]. Příklad zapojení je na obr. 1.6.

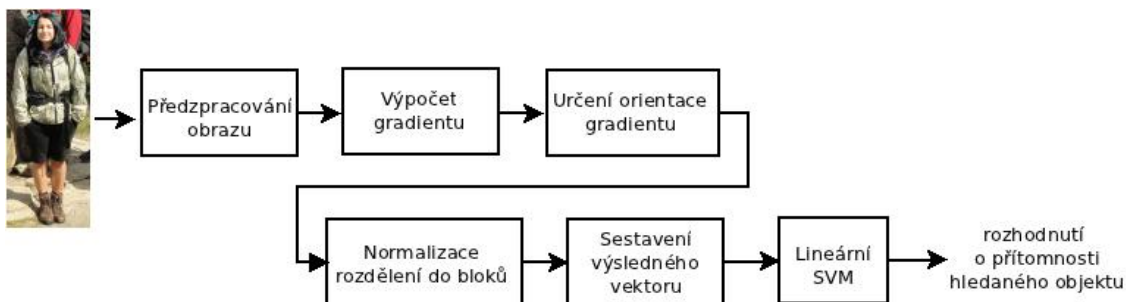


Obr. 1.6: Blokové schéma zapojení klasifikátorů do kaskády

Aby algoritmus Viola-Jones při detekci dosahoval vysoké úspěšnosti, je nutné mít kvalitní silný klasifikátor zahrnující stovky klasifikátorů slabých. Kaskádové zapojení je řešeno tak, že nejjednodušší klasifikátory jsou na prvních stupních kaskády. To zajišťuje vyřazení většiny nehledaných objektů už na začátku a šetří tak čas silnějším klasifikátorům na konci, které jsou náročnější na výpočet. Viola-Jones tedy vychází z předpokladu, že negativních identifikací se v obrázku vždy vyskytuje více. [3]

1.2 Metoda HOG

Tato metoda je založena na histogramech orientovaných gradientů. Základem je idea, že hledaný objekt může být nalezen pomocí intenzity gradientů a hran objektu, přestože není známa jejich přesná poloha. Metoda se opět skládá z více částí. Průběh algoritmu je zobrazen na obrázku 1.7, který byl inspirován v [1].



Obr. 1.7: Algoritmus HOG

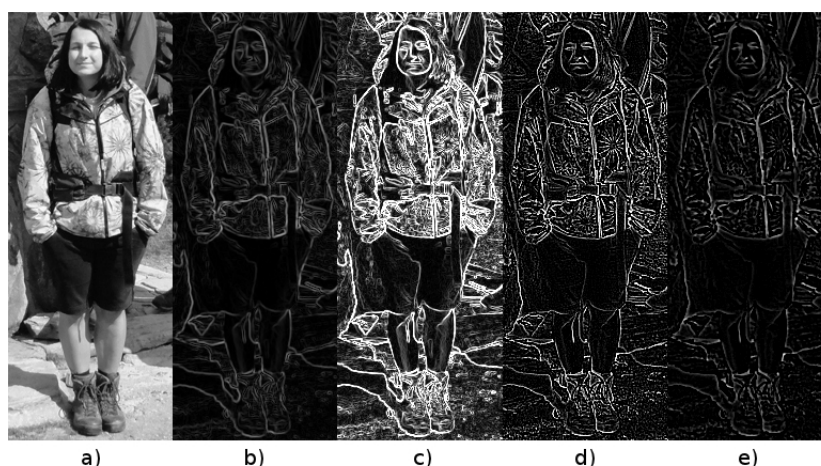
1.2.1 Předzpracování obrazu

V obrázku se mohou vyskytovat nežádoucí vlivy, např. jasové znečištění. Aby byl výsledek co nejpřesnější, je vhodné obrázky v trénovací sadě nějakým způsobem srovnat, konkrétně využít filtr. Může být použito vyjádření v jednokanálovém zobrazení, tedy ve stupních šedi. Obvykle je ale použito zobrazení v tříkanálovém zobrazení barevného modelu RGB [1].

1.2.2 Histogramy orientovaných gradientů a jejich výpočet

Jak už bylo zmíněno, metoda využívá pro detekci hledaných objektů (v tomto případě postav) hodnoty orientovaných gradientů. V praxi je obrázek rozdělen na síť stejně velkých bloků (8 x 8 pixelů). Pro každý blok je poté spočítán jednorozměrný histogram orientovaných gradientů, tedy směr růstu. Tyto gradienty určují postavu v obraze. K určení její polohy je nutné využít libovolnou metodu pro detekci hran. Aby byla metoda úspěšná, musí být provedena jak ve vertikálním, tak i v horizontálním směru. Vektory významných hran (ty, které určují postavu) mají rozsah 0°- 360°. Jelikož je tento rozsah velký, musí být rozdělen na devět tzv. binů, kdy každý bin bude obsahovat 40°. Lokální histogram vytvořený z těchto binů obsahuje informace o směru jednotlivých gradientů a složením všech histogramů odpovídajících buněk se vytvoří výsledný vektor. Ten je nakonec použit k samotné detekci [1].

Pro výpočet gradientů tedy musí být nejprve nalezeny významné hrany. Ty jsou zjištěny pomocí konvoluce obrázku derivační maskou $[-1,0,1]$, samostatně v horizontálním i vertikálním směru. K detekci hran mohou být použity i jiné masky, např. Sobelův, Robertsův nebo Laplaceův operátor. Výpočet gradientů pro tříkanálové zobrazení (barevný RGB model), znamená výpočet pro každý kanál zvlášť. Další metodou detekce hran je rozdílů Gaussových rozostření, které používá algoritmus Scale Invariant Feature Transformation (SIFT). [1] Na obrázku 1.8 jsou vyobrazeny všechny zmiňované případy.



Obr. 1.8: Detekce hran: a) originální obrázek, b) Sobelův operátor, c) Robertsův operátor, d) Laplaceův operátor, e) rozdíl Gaussových rozostření

1.2.3 Orientace gradientů

V předchozí podkapitole bylo zmíněno o rozdělení buněk do binů. Každý pixel v buňce nese informaci o směru gradientu a právě z nich je sestaven histogram. Nepoužijeme rozsah 0° - 360° , ale 0° - 180° . Je to kvůli zvýšení výpočetní rychlosti (každý bin odpovídá 20°). Každá buňka je tedy reprezentována vektorem (složený z devíti binových hodnot) a z těchto dílčích vektorů je sestaven celý obraz [1].

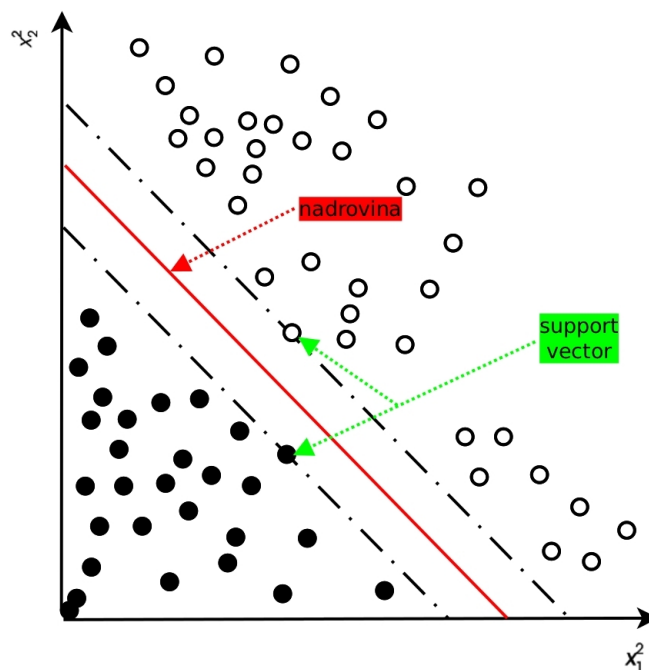
1.2.4 Normalizace a rozdělení bloků, tvorba vektoru

Hodnoty gradientů se v celém obraze liší vlivem kontrastu a osvětlení. Je proto vhodné seskupit buňky do bloků a ty normalizovat. Obraz je možné rozdělit i do kruhových bloků, ale tento algoritmus používá čtvercové rozdělení. Jak již bylo poznamenáno, každá čtvercová buňka obsahuje určitý počet pixelů (6-8 x 6-8 pixelů) a

je rozdělena na 9 binů po 20°. Blok se pak skládá z několika buněk, nejlepší je však o velikosti 3x3 [1]. Různé nastavení těchto parametrů bude testováno.

1.2.5 SVM

Stejně jako u algoritmu Viola-Jones, i tady se musí nějakým způsobem rozdělit trénovací data do dvou tříd, tedy na pozitivní a negativní příznaky. V předchozím případě vše zajišťoval AdaBoost, zde je využito služeb strojově se učícího algoritmu Support Vector Machine (SVM). Tento nástroj pracuje s tzv. podpůrnými vektory. Jeho výhodou je, že dokáže oddělit nelineárně ohraničená data (např. kružnicí) lineární funkcí. Aby to dokázal, musí si do dvourozměrného vstupního prostoru přidat další dimenzi, neboli nadrovinu. Tím se stane, že se jedna skupina dat (např. ty v kružnici) posune podle jedné z os. Tím pádem už je možné oddělit tato data lineární rovinou. Klasifikace dat je tím lepší, čím je prostor mezi oddělenými daty prázdnější, tedy bez bodů. Krajiní body ležící na hranici rozdělení se nazývají podpůrné vektory. Pro názornost je vše vyobrazeno na obr. 1.9 [6].



Obr. 1.9: Rozdělení negativních a pozitivních příznaků pomocí nadroviny

2 TVORBA DATABÁZE

Jelikož je pro každou metodu potřeba vytvořit jinou trénovací databázi, je tato kapitola rozdělena na dvě podkapitoly. Všechny obrázky které byly použity byly buď staženy z Internetu, nebo jsou to fotografie vlastní. Všechny obrázky z Internetu byly zveřejněny pod licencí Creative Commons. To znamená, že je povoleno je využívat nebo měnit bez porušení autorských práv. Veškerá práce s nimi byla provedena ve volně šiřitelném programu pro úpravu obrazů GNU Image Manipulation Program (GIMP).

2.1 Tvorba databáze pro Viola-Jones

Detekční okno této metody je 24x24 pixelů. Jednotlivé obrázky v trénovací množině tedy budou čtvercové, aby při zpracování AdaBoostem nedocházelo k jejich deformaci. Dále musí být všechny v jednobarevném šedotónovém zobrazení.

2.1.1 Tvorba databáze pro pozitivní příznaky

Do čtvercového okna by se obdélníková postava vměstnávala jen s příliš obtížně. Aby detektor nedetekoval i nežádoucí předměty, zaměřilo se trénování jen na hlavu. Obrázky byly ručně ořezány na velikost 90 x 90 pixelů a teprve poté hromadně zmenšeny na 24 x 24 pixelů. To převážně kvůli lepší manipulaci s nimi. Pozitivní trénovací databáze obsahuje obličejové zahrnující celou hlavu ze všech možných úhlů, kromě pohledu zezadu. Příklad databáze je vidět na obr. 2.1 a čítá 500 vzorků.



Obr. 2.1: Příklad pozitivní databáze pro Viola-Jones

2.1.2 Tvorba databáze pro negativní příznaky

Stejně jako pozitivní databáze, i negativní musí obsahovat vzorky čtvercového tvaru. Opět mají velikost 24 x 24 pixelů a byly náhodně vystřiženy z databáze obrázků neobsahující člověka. Příklad je vidět na obr. 2.2.



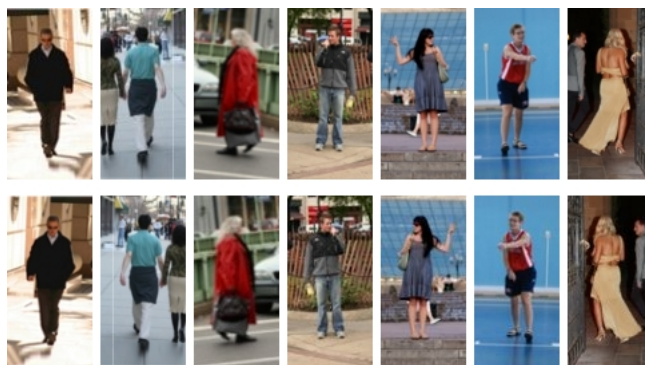
Obr. 2.2: Příklad negativní databáze pro Viola-Jones

2.2 Tvorba databáze pro HOG

Na rozdíl od Viola-Jones, tato metoda má rozměry detekčního okna 128 x 64. Pozitivní i negativní databáze tedy byla tvořena v obdélníkovém tvaru. Obrázky byly ponechány v tříkanálovém zobrazení RGB a nebyly nijak dále upravovány.

2.2.1 Tvorba databáze pro pozitivní příznaky

Každý obraz v této databázi obsahuje celou postavu člověka. Jelikož HOG funguje na detekci hran, nebude tolik vadit okolní šum. Každá postava byla vycentrována na střed. Takto bylo ručně zpracováno 250 obrázků. Následně byly hromadně převráceny podle souměrné svislé osy. Databáze tedy čítá 500 vzorků. Příklady jsou vidět na obr. 2.3.



Obr. 2.3: Příklad pozitivní databáze pro HOG

2.2.2 Tvorba databáze pro negativní příznaky

Stejně jako u Viola-Jones, i tady byly obrázky náhodně vystřiženy z databáze snímků neobsahujících postavu. Jediný rozdíl je ve velikosti, která je stejná jako u pozitivní databáze metody HOG.

3 IMPLEMENTACE V RAPIDMINERU

V této kapitole bude popsáno a ukázáno prostředí RapidMiner a implementace jednotlivých algoritmů.

3.1 Prostředí RapidMineru

Program RapidMiner je volně šiřitelný software pro doslova dolování dat a textu. Výzkumná skupina SPLab se pokouší jeho pole působnosti rozšířit o obrazové funkce jako je například právě detekce objektů. Každý projekt v RapidMineru je tvořen tzv. „bloky“. Každý z těchto „bloků“ symbolizuje jeden operátor. Všechny jsou napsány v jazyce Java za pomoci vývojového prostředí Eclipse. Tyto „bloky“ jsou poté spojeny a tvoří cestu od počátečních operátorů až ke koncovým.

3.2 Implementace Viola-Jones

Viola-Jones byl v RapidMineru implementován za pomoci dvou projektů. První projekt obsahuje trénovací algoritmus, druhý potom detekční. Oba algoritmy by mohly být implementovány i v jednom projektu, ale to není praktické, protože trénovací algoritmus je časově příliš náročný.

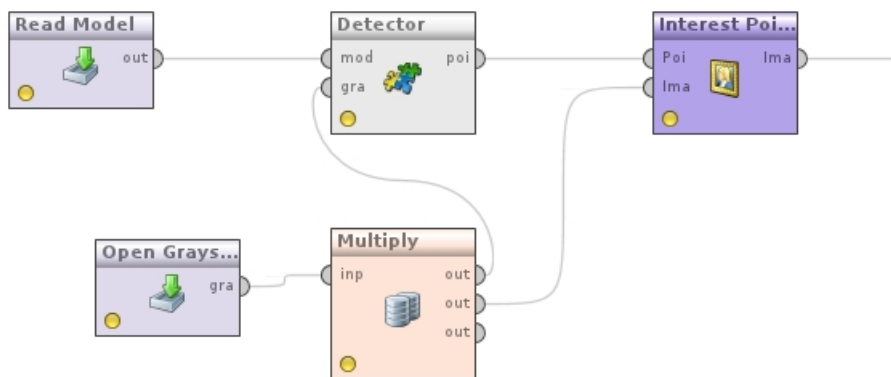
Trénovací algoritmus na obr. 3.1 se tedy skládá ze tří operátorů. První se nazývá *Read Image Set* a slouží k načtení negativní a pozitivní databáze obrázků. Operátor *Viola-Jones* už je samotný AdaBoost. V kontextovém menu tohoto operátoru se nastavuje cesta k souboru s uloženým nastavením kaskády. Výstupem AdaBoostu je kaskádový model, podle kterého bude následně probíhat detekce. Poslední operátor už jen zapíše výsledný model na pevný disk.



Obr. 3.1: Trénovací algoritmus Viola-Jones

Detekční algoritmus na obr. 3.2 obsahuje operátorů více. *Read Model* načte předchozí uložený model. *Open Grayscale Picture* načítá obrázek, na kterém se bude detekovat. Odtud načtený obraz putuje do *Multiply*, který jen větví cestu do dvou

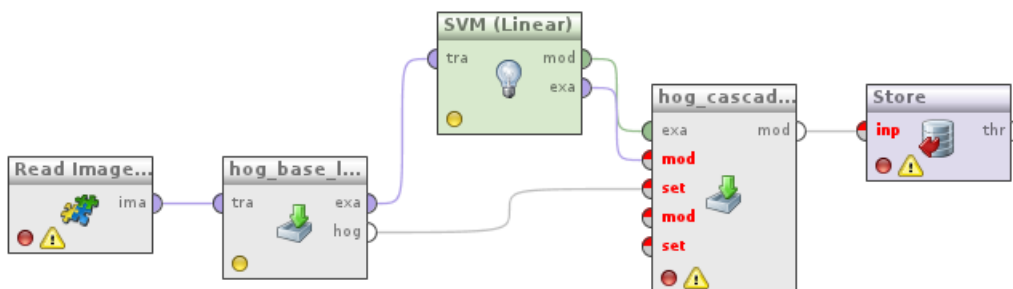
dalších operátorů. Ten první je *Viola-Jones Detector*, jeho úkolem je samotná detekce objektů. U tohoto operátoru se nastavuje např. velikost detekčního okna, nebo použití funkce minimal neighbours (nastavení minimálního počtu detekce objektu, aby byl brán jako detekovaný). Druhý nese název *Interest Point Visualizer* a vyznačuje detekovaná místa.



Obr. 3.2: Detekční algoritmus Viola-Jones

3.3 Implementace HOG

Implementací metody HOG v RapidMineru existuje více. První se nazývá modulární HOG, druhá diffusion HOG. Modulární HOG umožňuje natrénovat kaskádu více klasifikátorů. Příklad trénování modelu s jedním stupněm kaskády je na obr. 3.3.



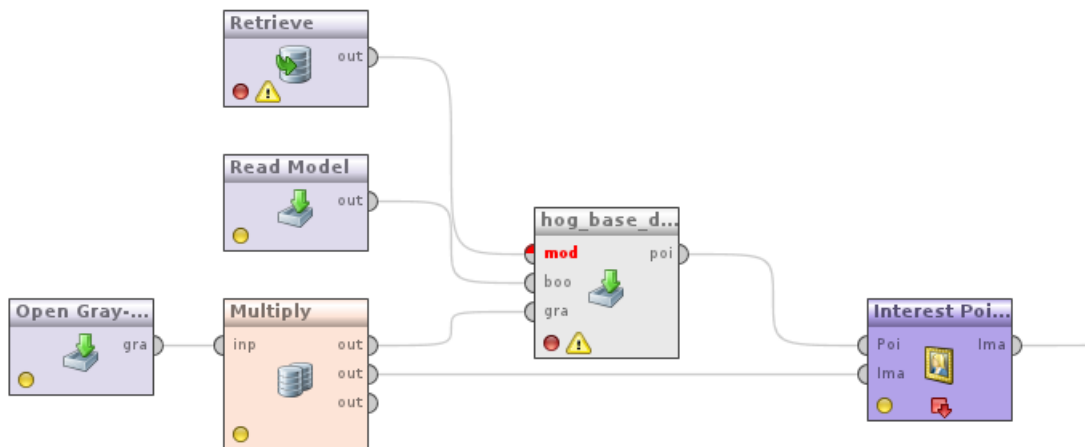
Obr. 3.3: Trénovací algoritmus modulární HOG

Operátor *Hog Base Learner* pro každý obraz v trénovací množině provede výpočet deskriptoru, zde se nastavují velikosti bloků a počet binů. Operátor *SVM* je vektorový klasifikátor a na jeho výstupu je klasifikační model. *Hog cascade* řetězí jednotlivé modely a vytváří z nich jeden klasifikátor. Diffusion HOG narozdíl od modulárního využívá ke klasifikaci algoritmus Adaboost. Využívá se výhradně ke zkrácení detekční doby. Jeho implementace je vidět na obrázku 3.4.



Obr. 3.4: Trénovací algoritmus diffusion HOG

Detekční algoritmus k oběma předchozím metodám tvoří operátory, které jsou vyobrazeny na obrázku 3.5.



Obr. 3.5: Detekční algoritmus HOG

Operátory *Retrieve* a *Read Model* načítají natrénované modely. Operátor *Hog Base Detector* slouží k detekci pomocí těchto modelů. Model připojený na vstup *mod* je trénovaný modulární metodou. Ten druhý je trénován pomocí diffusion HOG. Detektor pracuje buď s oběma modely, nebo pouze s modulárním.

4 VÝSLEDKY TESTOVÁNÍ

Testování probíhalo na dvou testovacích databázích, pro obě metody zvlášť. V obou následujících kapitolách budou prezentovány výsledky.

4.1 Testování Viola-Jones

Testovací množina se skládala z 200 vzorků, jejichž rozlišení bylo 112 x 92 pixelů. Polovina obsahovala obličej, druhá polovina nikoliv. K sestavení pozitivních testovacích a doplnění trénovacích obrázků byla využita databáze vytvořená tvůrci článku [2]. Testování probíhalo ve třech fázích, kdy v každé fázi byl natrénován model s jiným počtem trénovacích obrázků:

1. fáze: 500 pozitivních, 1500 negativních.
2. fáze: 3000 pozitivních, 6000 negativních
3. fáze: 6000 pozitivních, 12000 negativních

Tab. 4.1: Výsledky testování Viola-Jones

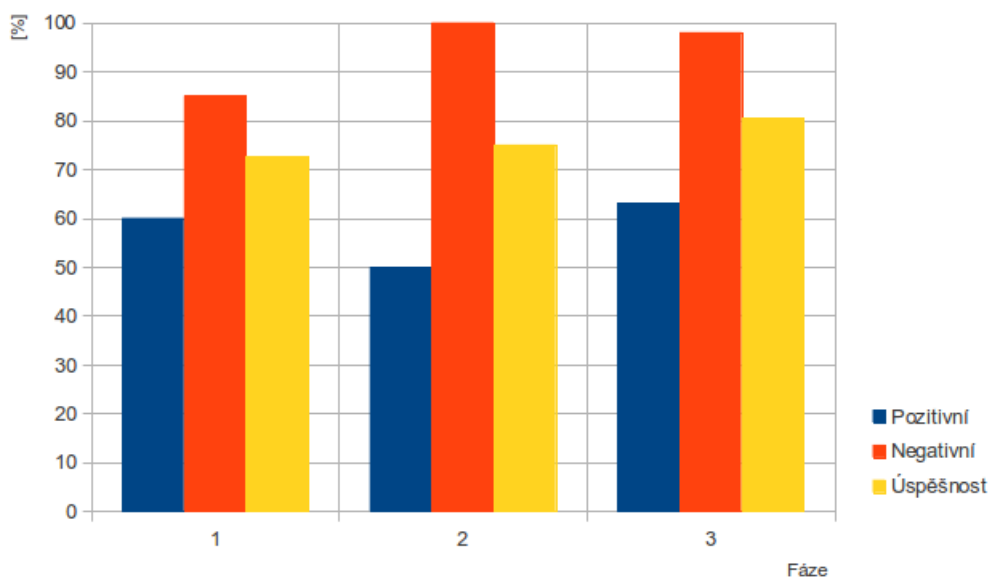
Fáze	Pozitiv	Detekované	Negativ	Detekované	Úspěšnost
1.	100	60	100	85	72.5 %
2.	100	50	100	100	75 %
3.	100	63	100	98	80.5 %

Výsledky testování jsou vidět v tabulce 4.1 výše. Ta obsahuje počet správných pozitivních a negativních detekcí v každé fázi, ze kterých je potom vyjádřena procentuální úspěšnost ze všech 200 vzorků. Dále z tabulky vyplývá, že se zvětšením počtu negativních vzorků trénovací databáze může dojít ke snížení úspěšnosti detekce vzorku pozitivního. Příklad správné pozitivní detekce včetně dvojité je vidět na obr. 4.1.



Obr. 4.1: Příklad pozitivních detekcí Viola-Jones

Doba trénování u tohoto algoritmu je závislá na velikosti trénovacích databází. Model první trénovací fáze trval 6 hodin. Fáze dvě zabrala 5 dní a 12 hodin. Poslední fáze trénovací dobu o mnoho neprodloužila, trvala 6 dní a 17 hodin. Samotná detekce potom trvala necelou vteřinu. Testování probíhalo na dvoujádrovém počítači a RapidMineru bylo přiděleno 2500 MB paměti, přičemž Adaboost využíval pouze 1600 MB. Pro všechny fáze byly trénovány kaskády klasifikátorů nastavené podle článku [4]. Výsledky testování zobrazeny níže v grafu 4.2.



Obr. 4.2: Výsledky třífázového testování Viola-Jones

4.2 Testování HOG

Testovací množina obsahovala opět 200 vzorků. K doplnění trénovacích databází byla tentokrát použita databáze INRIA, od tvůrců článku [1]. Testované snímky měly rozlišení 70 x 134 pixelů. Postava na nich však nepřekonala velikost detekčního okna. Testování probíhalo tentokrát dvoufázově, kdy v první fázi byl natrénován model s jedním stupněm kaskády, ve fázi druhé potom model s devíti stupni kaskády.

1. fáze: 500 pozitivních, 1500 negativních.
2. fáze: 2400 pozitivních, 3000 negativních.

Z výsledků v tabulce 4.2 vyplývá, že nejdříve se úspěšnost detekce drží kolem 30%, ale poté vzroste k 70 %. Za tento skok může trénování pomocí devítistupňové

kaskády. Stabilnější výsledky poskytuje detekce pozitivních vzorků, kde je úspěšnost 60 - 80%. Příklady pozitivních detekcí jsou k dispozici na obrázku 4.3.

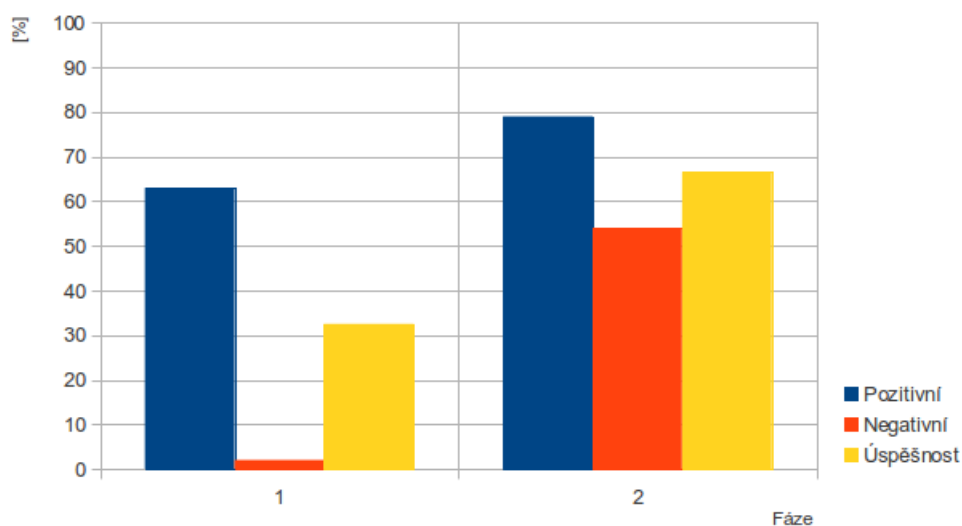
Tab. 4.2: Výsledky testování HOG

Fáze	Pozitiv	Detekované	Negativ	Detekované	Úspěšnost
1.	100	63	100	2	32.5 %
2.	100	79	100	54	66.5 %



Obr. 4.3: Příklad pozitivních detekcí HOG

Při stejné konfiguraci systému trvalo trénování modelů mnohem méně než v předchozím případě. První fáze trvala v řádech několika minut a druhá nepřestoupila hranici 2 hodin. Tato metoda je velmi náročná na operační paměť, trénování fáze 2 využilo bezmála 3 GB RAM. Výsledky testování jsou vidět níže v grafu 4.4.



Obr. 4.4: Výsledky dvoufázového testování HOG

5 NOVÉ METODY PRO DETEKCI SKUPIN OSOB V OBRAZE

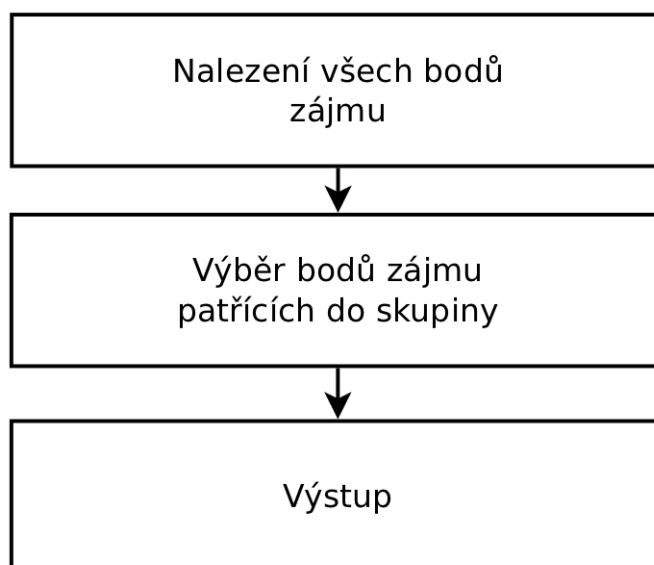
V rámci řešení bakalářské práce bylo nutné navrhnout a implementovat nové metody pro detekci skupin osob v databázi obrazů. Cílem těchto metod nebude jen detekovat osoby jako takové, ale i rozhodnout, zda-li patří do jedné skupiny nebo ne. Nové metody budou postaveny na známých detekčních algoritmech Viola-Jones a HOG.

5.1 Princip metod

Každá detekovaná osoba na obraze je reprezentována tzv. bodem zájmu. Každý bod zájmu je definován čtyřmi parametry:

1. Souřadnicí na ose x .
2. Souřadnicí na ose y .
3. Výškou detekované osoby.
4. Šířkou detekované osoby.

Souřadnice na osách x a y určují právě střed detekované osoby. Nová metoda bude porovnávat vzdálenosti těchto středů a vybírat ty, které patří do skupiny. Návrh průběhu metody je vidět na blokovém schématu 5.1.



Obr. 5.1: Blokové schéma operátoru

5.1.1 Nalezení všech bodů zájmu

Detekční algoritmus Viola-Jones nebo HOG na obraze vyhledá všechny body zájmu. Ze všech nalezených nové metody vybírají jen ty, které patří do skupiny.

5.1.2 Výběr bodů zájmu patřících do skupiny

Výběr detekovaných objektů patřících do skupiny závisí na vektorové vzdálenosti mezi jednotlivými body zájmu. Maximální vektorovou vzdálenost určuje uživatel, viz. podkapitola 5.3 Vstupní parametry. Pro další postup je tedy nezbytné vypočítat vektorovou vzdálenost mezi všemi body zájmu. Pro dva tyto body např. A a B je nutné postupovat nejdříve podle vzorce pro výpočet vektoru \vec{AB} ,

$$\vec{AB} = [B_x - A_x, B_y - A_y], \quad (5.1)$$

kde $A_{x,y}$ a $B_{x,y}$ jsou souřadnice bodů A a B na osách x a y . Následně je vypočítána délka tohoto vektoru podle Pythagorovy věty

$$|\vec{AB}| = \sqrt{AB_x^2 + AB_y^2}, \quad (5.2)$$

kde AB_x a AB_y jsou souřadnice vektoru \vec{AB} na osách x a y .



Obr. 5.2: Výběr bodů zájmu

Na obrázku 5.2 je vyobrazen příklad výběru bodů zájmu do skupiny. První nalezený bod je A, druhý je bod B. Jelikož je bod B dostatečně blízko, bod A je vybrán jako součást skupiny. Poté je hledán blízký bod pro bod B. Jelikož bod C není dostatečně blízko, pokračuje výpočet vzdálenosti k bodu D. Zde už vektorová délka odpovídá a bod B je také vybrán. Dále je hledán blízký bod k bodu C atd.

5.1.3 Výstup

Výstupem metod jsou právě ty body zájmu, ke kterým byl nalezen blízký bod. Ostatní body zájmu jsou zahozeny.

5.2 Implementace operátoru do RapidMineru

Implementace operátorů do RapidMineru probíhá za pomoci jazyku Java. Všechny operátory pro práci s obrazy se vyskytují v projektu Image processing extension. Obě implementace nových metod pro detekci skupin osob využívají nový operátor, který byl pojmenován *Group Visualizer* a nachází se v souboru `GroupVisualizer.java`, ten je součástí přílohy. *Group Visualizer* lze z hlediska implementace rozdělit na několik částí:

1. Uložení všech bodů zájmu.
2. Vyhledání bodů zájmu patřících do skupiny.
3. Výpočet velikosti okna vyznačujících skupinu.

V prvním kroku operátor pomocí cyklu `for` zapíše všechny detekované body zájmu do kontejneru třídy `Vector`. Body zájmu jsou ukládány pomocí třídy `Interest Area` díky níž jsou do jednoho pole kontejneru uloženy všechny čtyři parametry jednoho bodu zájmu.

V kroku druhém byl použit cyklus `for` vnořený do druhého. Tento algoritmus zajišťuje porovnávání všech bodů a výpočet vektorových vzdáleností za pomoci metod z třídy `Math`. Jakmile je pro první bod nalezen vhodný blízký bod, je zapsán do nového kontejneru `Vector`. Vnořený cyklus se ukončí příkazem `break` a vnější cyklus se posune o jedno pole doprava k dalšímu bodu. Vnořený cyklus se opakuje. Tímto způsobem jsou porovnány všechny nalezené body zájmu. Příklad uložení bodů do kontejneru `Vector`:

```
Vector [A; B; C; D; E; F]
```

Všechny body, které jsou uloženy v druhém kontejneru jsou vykresleny na detekovaný obraz pomocí metody `drawRect` z třídy `Graphics`. Tato metoda potřebuje

čtyři vstupní parametry: souřadnice x a y , výšku a šířku obdélníka. Pro jednotlivé body určil tyto parametry detektor, k vykreslení skupiny je nutné je vypočítat.

Ve třetím kroku tedy dochází k výpočtu souřadnic pro ohraničení celé skupiny. Do čtyř proměnných pojmenovaných X_{min} , Y_{min} , X_{max} a Y_{max} jsou uloženy maximální a minimální hodnoty souřadnic všech bodů vyskytujících se v dané skupině. Souřadnicový systém na detekovaných obrazech má hodnotu nula v levém horním rohu. Detekovaná skupina je pro názornost ohraničena na obrázku 5.3.



Obr. 5.3: Výpočet vyznačení skupiny

Do dalších dvou proměnných pojmenovaných $maxWidth$ a $maxHeight$ jsou uloženy největší výšky a šířky detekovaných bodů zájmu. Tyto hodnoty jsou přičítány k celkové výšce a šířce okna vyznačující skupinu a to tak, aby ohraničila všechny detekované objekty.

5.3 Implementace operátoru v RapidMineru

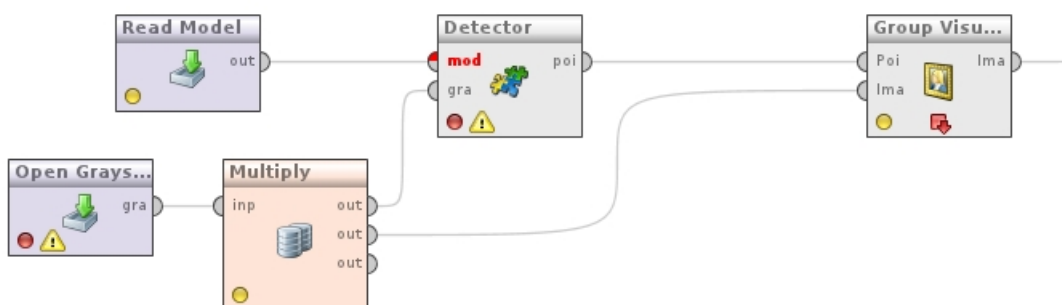
Operátor *Group Visualizer* implementován v RapidMineru představuje jeden blok. Má dva vstupy a jeden výstup. Na první vstup nazvaný *Poi* musí být přiveden výstup z detektoru, který mu předá všechny detekované body zájmu. Na druhý vstup nazvaný *Ima* musí být přiveden detekovaný obraz. Stejný obraz je pak zobrazen na

výstupu *Ima* i s vyobrazenými detekovanými objekty. Blok je zobrazen na obrázku 5.4.



Obr. 5.4: Group Visualizer v RapidMineru

Implementaci *Group Visualizeru* do detekčního algoritmu Viola-Jones ukazuje obrázek 5.5. Z něj je jasné, že nahrazuje původní operátor *Interest Point Visualizer*. V detekčním algoritmu pro metodu HOG je implementace obdobná.



Obr. 5.5: Implementace Group Visualizer s detektorem Viola-Jones

5.4 Vstupní parametry operátoru

Group Visualizer pracuje s pěti vstupními parametry. První tři jen upravují grafický výstup operátoru, poslední dva zásadně ovlivňují jeho průběh. Vstupní parametry ukazuje obrázek 5.6.

5.4.1 Display

U parametru Display jsou na výběr tři možnosti. Při výběru volby „Group“ operátor zobrazí jen detekovanou skupinu a v pravém dolním rohu obrázku vypíše počet detekovaných objektů, které do ní patří. Volba „Objects“ zase zobrazí jen ony detekované objekty. Doporučená volba „All“ zobrazí obě předchozí možnosti.

5.4.2 Object color a Group color

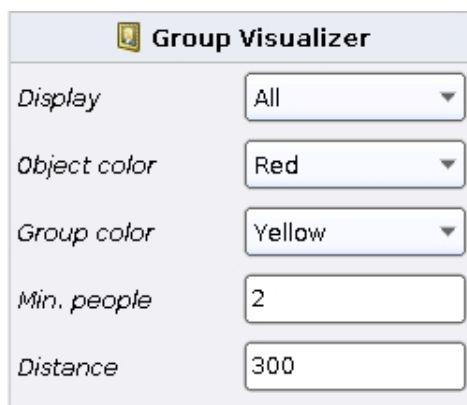
Podle názvu je jasné, že tyto parametry nastavují jen barvu ohraničení detekovaných skupin a objektů. Na výběr jsou barvy: černá, modrá, zelená, oranžová, červená, bílá a žlutá.

5.4.3 Min. people

Podle tohoto parametru operátor pozná, zda-li bude vykreslovat grafický výstup. Pokud detekční algoritmus vyhledá méně detekovaných objektů patřících do jedné skupiny, než je určeno v tomto parametru, na výstupu se nezobrazí žádný detekovaný objekt. Výchozí nastavení parametru je roven 2.

5.4.4 Distance

Tento parametr určuje minimální vektorovou vzdálenost mezi středy dvou detekovaných objektů patřících do skupiny. Jinými slovy, pokud operátor vypočítá větší vektorovou vzdálenost mezi dvěma objekty než je zadáno v tomto parametru, nebude je považovat jako skupinu. Vzdálenost je počítána v pixelech a rozsah parametru je od nuly do maximální hodnoty typu Integer. Výchozí nastavení je 300 pixelů.



The image shows a control panel titled "Group Visualizer" with a small icon of a group of people. It contains five rows of controls:

Parameter	Value
Display	All
Object color	Red
Group color	Yellow
Min. people	2
Distance	300

Obr. 5.6: Vstupní parametry

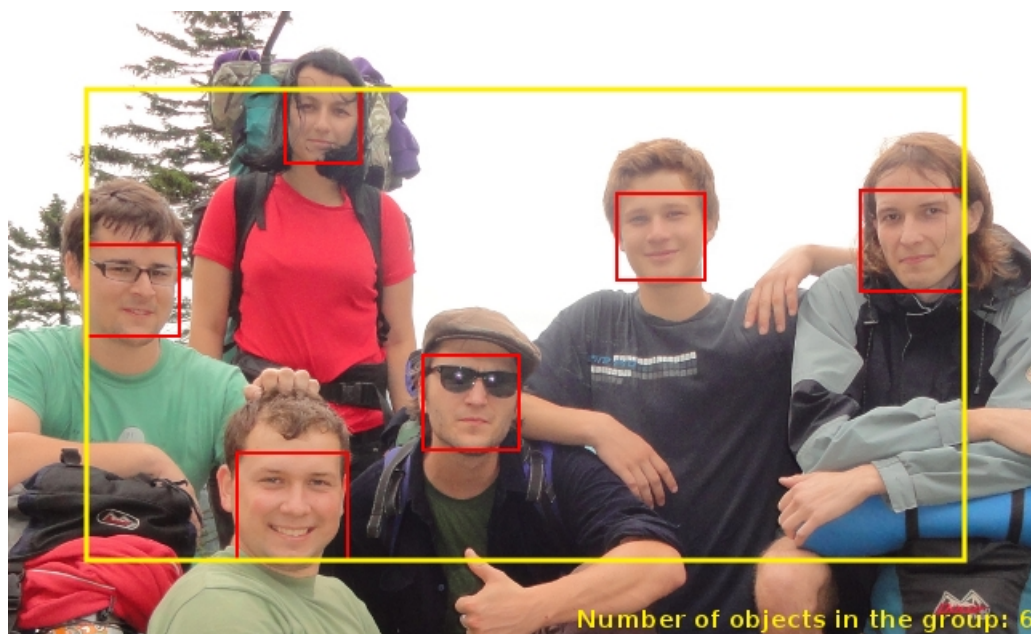
5.5 Výstup operátoru

Výstupy z operátoru jsou dva, jeden do konzolového okna ve vývojovém prostředí Eclipse, druhý vizuální v RapidMineru. Ten první vypisuje souřadnice všech nalezených bodů, vektorové vzdálenosti mezi nimi a počet bodů patřících do stejné skupiny. Jeho část může vypadat např. jako na obrázku 5.7.

```
Ax: 80  
Ay: 294  
Bx: 74  
By: 66  
Vector length: 228.07893370497854  
Number of objects in the group: 5
```

Obr. 5.7: Výstup operátoru Group Visualizer do konzole

Druhý výstup vizuálně zobrazuje nalezené body na detekovaném obraze v RapidMineru. Navíc vyznačuje, které spolu patří do skupiny a v pravém dolním rohu jejich počet vypisuje. Příklad znázorňuje obrázek 5.8.



Obr. 5.8: Ideální grafický výstup Group Visualizeru

6 TESTOVÁNÍ NOVÝCH METOD

Výsledky testování operátoru *Group Visualizer* jsou přímo závislé na natrénovaných modelech detekčních algoritmů Viola-Jones a HOG. Proto byl nejdříve otestován *Group Visualizer* samotný. Přestože oba detekční algoritmy vykazovaly úspěšnost detekce přes 66 %, výsledky detekce skupin na větších obrazech není ideální. Děje se tak především kvůli variabilitě pozadí na snímcích, proměnlivé velikosti osob a dalším vlivům neblaze ovlivňujícím výkonnost detektoru.

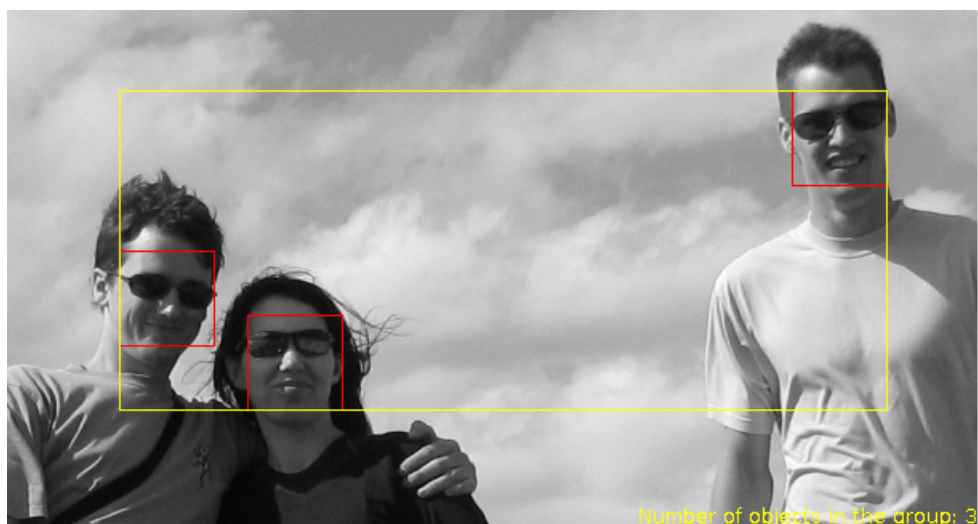
6.1 Testování Group Visualizeru

Group Visualizer byl testován za pomoci operátoru *Point of interest generator*, který umožňuje generovat body zájmu. Vygenerovat body lze náhodně, ale lze nastavit i jejich pevnou polohu. Při náhodné generaci byl nastavován počet bodů zájmu v rozmezí 2 až 100. Ve 100 % případů operátor spočítal přesný počet zobrazených bodů patřících do skupiny. Poté byly na obr. 6.1 a 6.2 nastaveny tři body zájmu přesně na obličeje tří znázorněných osob. Výsledek se vstupním parametrem *Distance* nastaveným na 300 pixelů je vidět na obr. 6.1.



Obr. 6.1: Test Group Visualizeru

Operátor opět správně vyhodnotil, které body zájmu patří do skupiny a započítal do ní pouze dva obličeje nalevo obr. 6.1. Ve výstupu z konzole operátoru vyšlo najevo, že třetí obličej je od nejbližšího bodu skupiny vzdálen 430 pixelů. Grafický výstup *Group Visualizeru* s parametrem *Distance* nastaveným na 450 pixelů je na obr. 6.2.



Obr. 6.2: Test Group Visualizeru

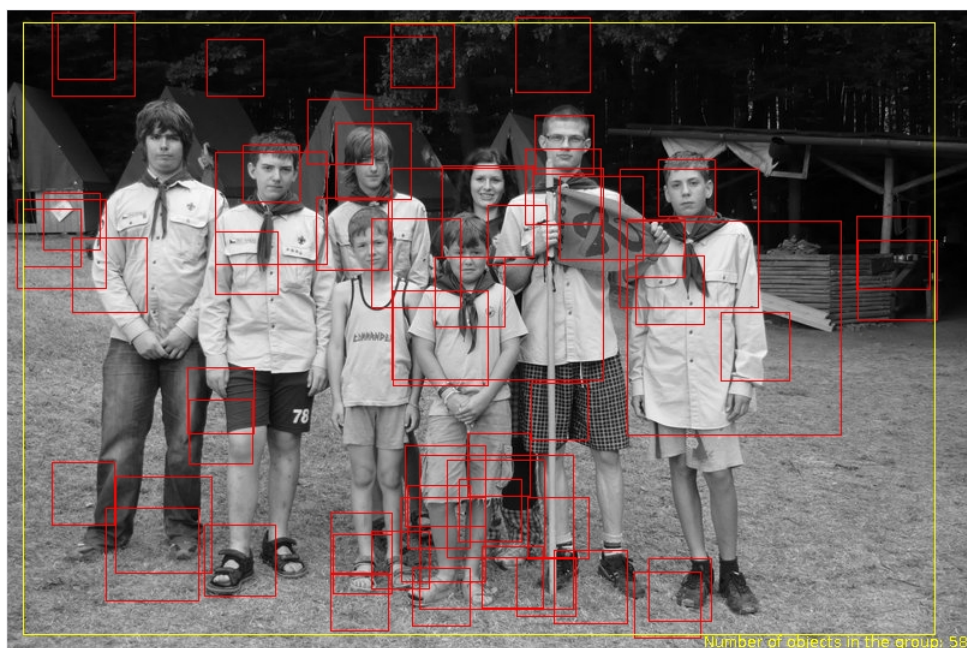
6.2 Testování s algoritmem Viola-Jones

Z deseti testovacích snímků o proměnlivé velikosti metoda většinou ohraničila skupinu osob, ovšem zdaleka to nebylo díky dokonalé funkčnosti. K testování byl použit natrénovaný model ze třetí fáze testování Violy-Jones (viz. kapitola testování Viola-Jones).

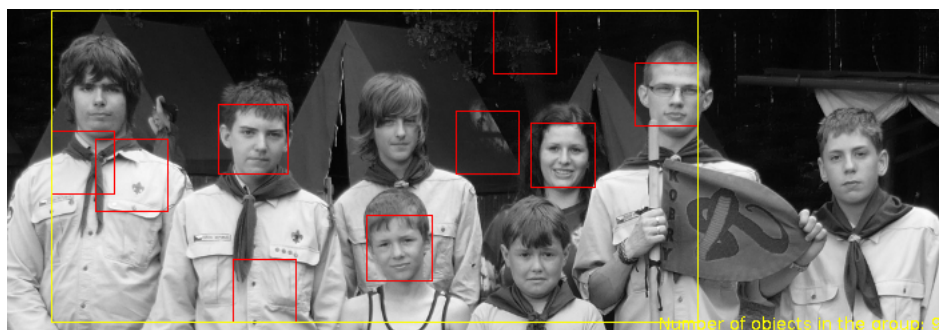


Obr. 6.3: Test Group Visualizeru pomocí Violy-Jones

Obr. 6.3. ilustruje jen ohraničení skupiny. Zdánlivě proběhla detekce v pořádku, avšak tato metoda by měla ohraničovat jen detekované obličeje. Dále *Group Visualizer* na výstupu vypisuje, že počet nalezených osob ve skupině je 58. To je zcela jistě špatně. Testovaný model nebyl dostatečně natrénován na přílišnou variabilitu prostředí. Ze všech deseti testovacích snímků metoda neurčila ani jedinou správný počet detekovaných osob, avšak se snižujícím se výskytem okolí kolem obličejů na obrázku se číslo přibližovalo pravdě. Všechny detekované objekty jsou vidět na obr. 6.4, detekce na stejném snímku ořezaném o okolí na obr. 6.5.



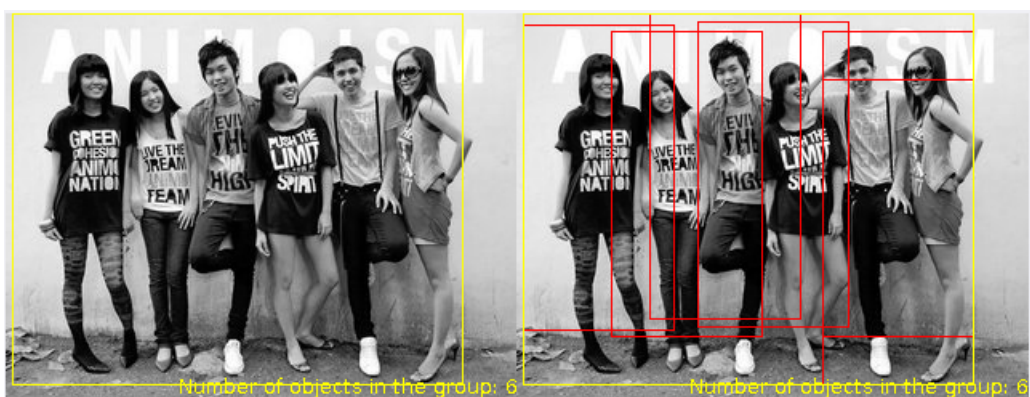
Obr. 6.4: Test Group Visualizeru pomocí Violy-Jones



Obr. 6.5: Test Group Visualizeru pomocí Violy-Jones

6.3 Testování s algoritmem HOG

Tato metoda při testování vykazovala velmi podobné výsledky jako předchozí. Testován byl natrénovaný model kaskády z druhé fáze testování algoritmu HOG. K tomu byl natrénován jeden model diffusion HOG, který ale výsledky výrazně neovlivňoval, jen zkracoval detekční dobu obrazu. Z deseti testovacích snímků proměnlivých velikostí metoda určila správný počet detekovaných objektů jen dvakrát a to na obrazech s nejmenším okolním šumem. Jakmile detekovaný obraz nabýval na velikosti, zvětšovala se doba detekce. Počet falešných detekcí závisel opět na typu okolí. Názorné ukázky detekcí jsou vidět na obrázcích 6.6 a 6.7.



Obr. 6.6: Test Group Visualizeru pomocí HOG



Obr. 6.7: Test Group Visualizeru pomocí HOG

7 ZÁVĚR

Cílem této práce bylo nastudovat moderní metody detekce skupin objektů a posoudit, která by byla nejvhodnější k detekci skupin osob. Moderní metody přímo k detekci skupin objektů ale neexistují. Bylo tedy nutné vycházet z metod detekujících jednotlivé objekty. Práce se zabývala dvěma známými detekčními algoritmy Viola-Jones a HOG. K oběma metodám byly vytvořeny trénovací databáze, díky kterým byly natrénovány klasifikační modely.

Algoritmus Viola-Jones byl natrénován na detekci obličejů a na testovací sadě s malými obrázky typu obličej / pozadí dosahoval úspěšnosti až 80 %. Algoritmus HOG pracoval s klasifikací celé postavy a nejlepší natrénovaný model na obdobné testovací sadě osoba / pozadí dosahoval úspěšnosti až 66 %.

Ke zmiňovaným detektorům byl implementován nový operátor a vznikly dvě nové metody, které jsou schopné pracovat s detekcí skupin. Výsledky z testování jsou bohužel více než zklamáním. U obou metod docházelo u snímků s větším okolním šumem k mnohonásobným falešným detekcím. Vyladění detektoru se bohužel nezlepšilo ani po několika testováních z různými parametry detektoru.

Podprůměrné výsledky testování jsou pravděpodobně špatně natrénovanými modely, ať už špatnou volbou trénovacích databází, nebo nevhodně nastavenou kaskádou klasifikátorů. V potaz by se měla brát i možnost nevhodné implementace detekčních algoritmů, které jsou k dispozici v projektu Image processing extension v RapidMineru. U Violy-Jones ztěžovala natrénování více variant klasifikačních modelů velmi dlouhá trénovací doba, která dosahovala bezmála 6 a půl dne, což je téměř týden. Podle článku [4] je však možné s touto metodou dosahovat lepších výsledků. U algoritmu HOG zase zabraňovala natrénování modelů s vyšším počtem trénovací databáze přílišná spotřeba paměti RAM. Trénovací proces si vyžádal přinejmenším 3.5 GB operační paměti. Článek [1] ovšem dokazuje, že dalším rozvojem této metody je možné detekovat mnohem účinněji. Do budoucna je tedy nutné zlepšit úspěšnost detekčních algoritmů, nově implementovaný operátor *Group Visualizer* pracoval bezchybně.

LITERATURA

- [1] Dalal, N.; Triggs, B.: Histograms of Oriented Gradients for Human Detection. In *International Conference on Computer Vision & Pattern Recognition*, ročník 2, editace C. Schmid; S. Soatto; C. Tomasi, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, s. 886–893.
URL <<http://lear.inrialpes.fr/pubs/2005/DT05>>
- [2] Georghiades, A.; Belhumeur, P.; Kriegman, D.: From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, ročník 23, č. 6, 2001: s. 643–660.
- [3] Přinosil, J.; Krolikowski, M.: Využití detektoru Viola-Jones pro lokalizaci obličejů a očí v barevných obrazech. *Elektrorevue*, 2008.
- [4] Viola, P.; Jones, M. J.: Robust Real-Time Face Detection. *International Journal of Computer Vision*, ročník 57, č. 2, Květen 2004: s. 137–154, ISSN 0920-5691, doi:10.1023/B:VISI.0000013087.49260.fb.
URL <<http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb>>
- [5] Šochman, J.: AdaBoost. Cvičení z RPZ, 2005.
URL <<http://cmp.felk.cvut.cz/cmp/courses/recognition/Labs/adaboost/adaboost.pdf>>
- [6] Žižka, J.: SVM. Studijní materiály předmětu FI, 2005.
URL <http://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AdaBoost Adaptive Boosting

GIMP GNU Image Manipulation Program

HOG Histogram of Oriented Gradient

SIFT Scale Invariant Feature Transformation

SPLab Signal Processing Laboratory

SVM Support Vector Machine