

**Detekce obličejů pomocí
příznakového rozpoznání**

**Face Detection using
Feature-Based Detectors**

Zadání bakalářské práce

Student: **Marek Merta**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Detekce obličejů pomocí příznakového rozpoznání
Face Detection using Feature-Based Detectors**

Zásady pro vypracování:

S rozvojem kamerových systémů se v posledních letech zvýšilo i jejich využití v oblasti detekce a rozpoznání lidských tváří. Detekce lidských tváří pomocí kamerových systémů je využívána například v moderních fotoaparátech, sociálních sítích nebo v bezpečnostních systémech. Cílem této práce bude otestování renomovaných metod v aplikaci detekce lidských tváří v obrazech.

1. Seznamte se se základními pojmy v oblasti detekce obličejů.
2. V popisu se zaměřte zejména na příznakové rozpoznávání pomocí renomovaných příznaků (např. HOG, Haar, LBP).
3. Seznamte se s knihovnou OpenCV.
4. S pomocí knihovny OpenCV tyto detektory implementujte.
5. Experimentálně ověřte funkčnost, přesnost a rychlost těchto detektorů na renomovaných datasetech lidských tváří.
6. Svě závěry řádně zdokumentujte v textu práce.

Seznam doporučené odborné literatury:

- [1] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. vol. 1, pp. I-511-I-518 vol.1 (2001)
- [2] Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: ICB. pp. 828-837 (2007)
- [3] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886-893 vol. 1 (june 2005)

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radovan Fusek**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2014

Marek Miska
.....

Abstrakt

Cílem této práce je zpracovat oblast detekce obličejů, která je dnes nedílnou součástí sociálních sítí nebo bezpečnostních systémů a podobně, zejména na základě příznakového rozpoznání pomocí renomovaných příznaků HOG, Haar a LBP. Dalším bodem je prozkoumat a popsat, na jakých principech tyto příznaky fungují. Následně také implementovat detekci obličejů s použitím kaskádového klasifikátoru natrénovaného na výše zmíněné příznaky. K implementaci využijeme knihovnu OpenCV 2.4.8. Pro experimentální testování jednotlivých implementací je připravena vlastní testovací sada obrazů, na kterých bude provedeno měření rychlosti, přesnosti a funkčnosti těchto detektorů. Nakonec bude provedeno porovnání naměřených dat.

Klíčová slova: Detekce obličejů, HOG, Haar, LBP, Kaskádový klasifikátor, OpenCV

Abstract

The purpose of this paper is to process the field of face detection. In today's world it forms an indispensable part in - social networks or security systems. It mainly utilizes thematic recognition based on renowned features of HOG, Haar and LBP. The aim of the following section is to investigate and describe the rules of how these features function. In addition, the paper will carry out an implementation of face detection with the use of cascade classifier that is tailored to the features mentioned above. For the implementation the library OpenCV 2.4.8 will be used. In terms of experimental testing of individual implementations, a separate testing set of images is prepared. It will be used to perform a test of speed, accuracy and functionality of these detectors. The paper will conclude with performing a comparison analysis of available data.

Keywords: Face detection, HOG, Haar, LBP, Cascade classifier, OpenCV

Seznam použitých zkratk a symbolů

AdaBoost	– Adaptive boosting
GLOH	– Gradient Location and Orientation Histogram
HOG	– Histogram of Oriented Gradients
LBP	– Local Binary Patterns
MB-LBP	– Multi-scale Block Local Binary Patterns
SIFT	– Scale-Invariant Feature Transform
SURF	– Speeded-Up Robust Features

Obsah

1	Úvod	5
2	Detekce lidských obličejů	6
2.1	Předzpracování obrazu	6
3	Příznakové rozpoznání	8
3.1	Histogram orientovaných gradientů	9
3.1.1	Výpočet histogramu orientovaných gradientů	10
3.1.2	Rozdělení modulů gradientu do binů	11
3.1.3	Shlukování pixelů do bloků a normalizace	11
3.1.4	Shrnutí	12
3.2	Lokální binární vzor	12
3.2.1	Předzpracování	13
3.2.2	Výpočet lokálních binárních vzorů	13
3.2.3	Extrakce příznaků a klasifikace	14
3.2.3.1	Ahonenova metoda	14
3.2.3.2	Spatial pyramid match	15
3.2.3.3	Naive Bayes Nearest Neighbor	15
3.2.4	Shrnutí	16
3.3	Haarovy příznaky	16
3.3.1	Extrakce příznaků	16
3.3.2	Integrovaný obraz	17
3.3.3	Klasifikace nalezených příznaků	17
3.4	SIFT, SURF a GLOH	18
3.4.1	SIFT	18
3.4.2	SURF	18
3.4.3	GLOH	19
4	Implementace pomocí knihovny OpenCV	20
4.1	AdaBoost	20
4.2	Kaskáda klasifikátorů	21
4.3	Vlastní provedení implementace	22
4.3.1	Experimentální testování	24

4.3.2	Vliv velikosti trénovací množiny	25
4.3.3	Vliv parametrů na testování	26
4.3.4	Vliv předzpracování	28
4.3.5	Vliv počtu stupňů kaskády	30
4.4	Zhodnocení implementace a testování	31
5	Závěr	33
6	Reference	34
	Přílohy	34
A	Zdrojový kód detektoru lidských tváří	35

Seznam obrázků

1	Proces detekce objektu. Zdroj: autor	6
2	Příklad obrazu převedeného na kontury. Zdroj:[1]	8
3	Proces konvoluce. Zdroj: http://cs.wikipedia.org/wiki/Konvoluce	10
4	Znázornění orientovaných gradientů. Zdroj: http://pavel.campr.cz/public/scikits-image-hog-fix/hog_lena_old.png	11
5	(a) 3x3 LBP (b) 9x9 MB-LBP. Obrázek převzat z webové prezentace [4].	13
6	Tvorba LBP histogramu. Obrázek převzat z webové prezentace [5].	14
7	Příklady Haarových příznaků. Obrázek převzat z webové prezentace [6].	17
8	První dva příznaky vybrané AdaBoostem. Obrázek převzat z webové prezentace [6].	20
9	Kaskáda klasifikátorů. Obrázek převzat z webové prezentace [6].	21
10	Výsledky měření se stanovenými základními parametry. Zdroj: autor	25
11	Vliv velikosti trénovací množiny. Zdroj: autor.	26
12	Vliv změny parametrů. Zdroj: autor.	27
13	Výsledky měření pro parametry -scaleFactor 1.1 -minNeighbors 9. Zdroj: autor	27
14	Výsledky měření pro parametry -scaleFactor 1.2 -minNeighbors 3. Zdroj: autor	28
15	Výsledky měření pro parametry -scaleFactor 1.05 -minNeighbors 3. Zdroj: autor	28
16	Vliv předzpracování. Zdroj: autor.	29
17	Výsledky měření po aplikaci Gaussova rozostření. Zdroj: autor	29
18	Výsledky měření po aplikaci jednoduchého rozostření. Zdroj: autor	30
19	Výsledky měření po aplikaci vyvážení histogramu. Zdroj: autor	30
20	Vliv počtu stupňů kaskády. Zdroj: autor.	31
21	Výsledky měření za použití kaskády HOG příznaků o 14 stupních. Zdroj: autor	31

Seznam výpisů zdrojového kódu

1	Ukázka zdrojového kódu v jazyce Java. Zdroj: autor	22
---	--	----

1 Úvod

Souběžně s rozvojem zařízení zaznamenávajících obraz dochází k rozvoji softwaru, který s těmito obrazy pracuje a který práci s nimi zjednodušuje a umožňuje provádět úkony v mnohem širší míře než jaké by byla schopna lidská složka. Tady se objevuje nutnost umět za použití těchto softwarů daný obraz vyhodnotit, rozpoznat a lokalizovat jednotlivé objekty v něm se nacházející. Detekce lidských tváří nachází v dnešní době široké uplatnění, ať už se jedná o dnes velmi oblíbené a společností vyhledávané sociální sítě, či moderní fotoaparáty, ve kterých se detekce tváře projeví na zlepšeném zaostřování. Dalším nezanedbatelným uplatněním je užití těchto aplikací v bezpečnostních kamerových systémech či letištních kontrolách, kde je taktéž užití počítačového zpracování obličejů spolehlivější než lidský faktor a stalo se běžnou součástí bezpečnostních kontrol.

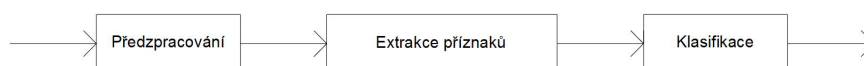
Samozřejmě je třeba brát v úvahu fakt, že lidská tvář je objekt proměnný a vzhledem k velké rozmanitosti tváří dochází k nepřesnostem v detekci, které se neustálým zlepšováním algoritmu a zařízení, které obraz zaznamenávají potlačují. Proto existují různé způsoby, jak obraz zpracovat a extrahovat z něj hledané objekty s co největší přesností a v co nejkratším čase. Jednotlivé algoritmy se svou výkonností liší a proto je třeba zvolit podle aktuálních požadavků buď metodu přesnější nebo metodu méně přesnou avšak co se zpracování týče rychlejší.

Tato práce se bude zabývat detekcí lidských obličejů na základě příznakového rozpoznání se zaměřením na konkrétní příznaky HOG, Haar a LBP. Pro tyto potřeby bude vytvořena testovací sada obrazů obličejů z čelního pohledu v běžném životě.

2 Detekce lidských obličejů

Dnešní doba nabízí velké možnosti zachycení a ukládání obrazu. Díky tomu může docházet k jeho zpracování, které dnes zahrnuje spoustu technik i manipulací a tím pádem nabízí nepřehledné množství využití v praxi. Jednou z oblastí zpracování obrazu je detekce objektů v obraze. Je mnoho způsobů, jak extrahovat objekty z obrazu, ale cílem této práce a také jejím hlavním přínosem je zpracovat detekci objektů, konkrétně lidských tváří, s použitím příznakového rozpoznání. V dalších částech se práce bude zabývat už výhradně detekcí lidského obličeje.

Detekce lidského obličeje je speciálním případem detekce objektů. Vzhledem k rozmanitosti barvy očí, pleti, také poměru velikostí obličejových částí a různým tvarům obličeje, se metoda detekce příznaků a následná klasifikace, zda se v dané oblasti může vyskytovat lidská tvář, jeví velmi vhodná pro použití. Z toho také vyplývá potřeba velkého množství pozitivních obrazů obličeje - přesně ohraničené oblasti obrazu, kde se vyskytuje obličej a negativních obrazů - obrazy, kde se obličej určitě nevyskytuje.



Obrázek 1: Proces detekce objektu. Zdroj: autor

Před začátkem práce s obrazem může být vhodné použít předzpracování obrazu, kterým se dá zlepšit výsledek zpracování.

Samotná detekce obličeje začíná hledáním příznaků charakteristických pro lidskou tvář. K tomuto účelu slouží příznakové deskriptory. V této práci budou hlavním zaměřením především deskriptory HOG, Haar a LBP. Tyto příznaky se následně zpracují kaskádovým klasifikátorem, který rozhodne o možném výskytu obličeje. V dnešní době existuje mnoho implementací algoritmů, řešících tuto problematiku a dosahujících vysokého procenta úspěšnosti.

2.1 Předzpracování obrazu

Výsledky zpracování obrazu ovlivňují do značné míry vstupní data. Ty vždy obsahují ať už drobnou nebo výraznější deformaci. Detekci mohou negativně ovlivnit například nerovnoměrné rozložení jasu nebo velký podíl šumu a spousta dalších rušení. Předzpra-

cování obrazu silně závisí na metodě extrakce příznaků. Mezi techniky předzpracování obrazu patří:

- Filtrace a odstranění šumu
- Adaptivní filtrace
- Zvýraznění kontrastu
- Potlačení vlivu osvětlení a jiných nehomogenit
- Vyhlazení a ostření

Knihovna OpenCV nabízí možnosti eliminace některých nežádoucích efektů. Pro odstranění šumu jsou v dokumentaci dobře popsány například Gaussian filtr nebo Normalized box filtr. Pokud jsou v obraze stupně šedi příliš nízké nebo naopak příliš vysoké, nabízí se využít roztažení histogramu obrazu tak, aby byly využity intenzity pixelů po celé stupnici intenzit.

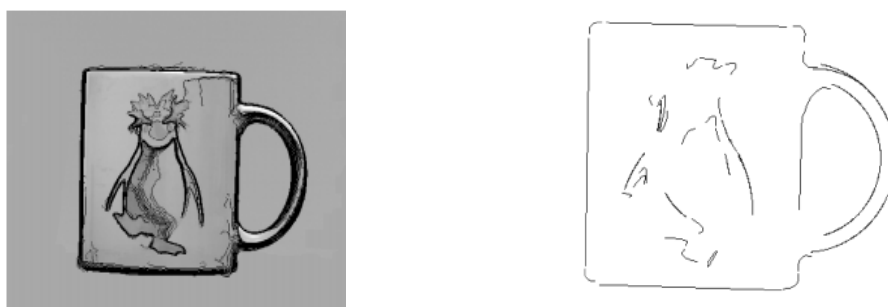
3 Příznakové rozpoznání

V následující fázi dochází k manipulacím s obrazem vedoucím k extrakci příznaků typických pro lidské obličej. V tuto chvíli by bylo na místě nejprve vysvětlit, co spojení "příznaky objektu" vůbec znamená. Jedná se o měřitelné fyzikální veličiny, které se liší v závislosti na jednotlivých objektech. Při výběru příznaků je nutné postupovat tak, aby jejich konečný výběr byl co nejvíce informativní. Při větším množství vybraných příznaků získáme vyšší procento úspěšnosti klasifikátoru, avšak na úkor vyšší časové náročnosti. Zvyšování počtu příznaků zlepšuje výsledky detekce pouze do určitého množství, další zvyšování už nevede k žádnému zlepšení.

Dříve než dojde k detekci samotného příznaku, dochází k rozebrání obrazu na jednotlivé kontury. Kontury, někdy také nazývány lokální hrany, jsou spojovány do větších celků-globálních hran. Existuje různé kategorizování těchto celků, které si každý upraví nebo přizpůsobí svým potřebám. Při členění do skupin se nejčastěji vyskytují pojmenování:

- hrana
- roh
- oblouk a další...

Zde může dojít k zavádějícímu použití slova "hrana". Pro vysvětlení, jedná se o spojování malých hran, ve skutečnosti spíše orientovaných bodů, do větších celků, blízcích se podobou skutečným hranám.



Obrázek 2: Příklad obrazu převedeného na kontury. Zdroj:[1]

I přes spojování lokálních hran do globálních, bývá nalezeno množství hran v řádech tisíců. Všechny nasbírané hrany musejí být nakombinovány do menšího počtu příznaků.

Vzniklé příznaky mohou být stále kombinovány do větších, ze kterých se vyberou pouze ty nejdůležitější.

Pro účely nalezení příznaků vzniklo velké množství algoritmů, které se stále vyvíjejí a zlepšují. Tato práce je zaměřena na tyto příznakové detektory:

- HOG
- Haar
- LBP
- SIFT, SURF a GLOH

3.1 Histogram orientovaných gradientů

Příznakový detektor HOG, celým názvem Histogram of Oriented Gradients, je používán pro zpracování obrazu za účelem detekce objektů. Tato technika je založena na výskytu orientovaného gradientu v lokálních částech obrazu. Gradient je směr největšího růstu obrazové funkce (od černé po bílou). Hrany jsou kolmice na směr gradientu [2] a dochází v nich ke změně směru a rychlosti růstu obrazové funkce. Tato metoda byla poprvé představena na konferenci "International Conference on Computer Vision & Pattern Recognition" v roce 2005. V článku [3] byla tato metoda publikována jejími průkopníky Navneet Dalal a Bill Triggs. Ve své publikaci popisují postup při detekci lidských postav již zmiňovanou metodou. Při návrhu této metody se osvědčilo dodržovat následující kritéria:

- RGB spektrum bez gama korekce
- konvoluční maska $[-1,0,1]$
- rozdělení gradientů do 9 binů
- uskupení 4 bloků s rozměry 8×8 pixelů
- detekční okénko 64×128 pixelů
- lineární SVM klasifikátor

Ve shrnutí - nejdůležitější je použít barevné spektrum bez korekce, množství orientačních binů a přiměřeně velké, přísně normalizované překrývající se bloky deskriptoru.

3.1.1 Výpočet histogramu orientovaných gradientů

Celý proces hledání orientovaných gradientů začíná rozdělením obrazu do buněk o velikosti 2x2 bloku. Tyto buňky jsou svou plochou z 50 procent překryté, jak ve směru horizontálním, tak i ve směru vertikálním. Každá z těchto buněk o velikosti 2x2 obsahuje 4 bloky, každý o velikosti 8x8 pixelů.

Výkon detektoru je silně závislý na způsobu výpočtu gradientů, nicméně se dá konstatovat, že nejlepších výsledků dosahujeme těmi nejjednoduššími postupy. Ve své práci testovali gradienty získané za použití Gaussova vyhlazování následovaného aplikací jedné z mnoha diskretních derivací. Derivace obrazových dat jako taková se nejčastěji provádí za použití konvoluce.

Poznámka 3.1 S konvolucí je možné se často setkat při zpracování 2D diskretního obrazu v počítačové grafice. Základem konvoluce je použití vhodné konvoluční masky. Je zvoleno místo na obraze, kam je umístěna konvoluční maska. Všechny pixely původního obrazu jsou vynásobeny hodnotou v odpovídající buňce masky, čímž bude získán nový pixel, který odpovídá průměru hodnot původních pixelů.



Obrázek 3: Proces konvoluce. Zdroj: <http://cs.wikipedia.org/wiki/Konvoluce>

Navneet Dala a Bill Triggs testovali použití různých druhů konvolučních masek, přičemž nejlepších výsledků dosahovala konvoluční maska ve tvaru $[-1,0,1]$. Použití větších masek se ukázalo jako nevhodné, jelikož snižovaly výkon detektoru.

3.1.2 Rozdělení modulů gradientu do binů

V dalším kroku následuje zpracování informací, které poskytují vypočtené gradienty. Orientace gradientu se pohybuje v rozmezí $0^\circ - 360^\circ$, pro zjednodušení náročnosti výpočtu se ukázalo výhodnější použít rozsah $0^\circ - 180^\circ$, který nezpůsobil žádné výraznější zhoršení. Tento rozsah se dále rozdělí do devíti částí, každá po 20° . Tyto oblasti se označují jako biny.

Podle autorů metody se nad každou buňkou obrazu vytvoří histogram orientovaných hran, kde sloupce histogramu odpovídají jednotlivým binům. Každý pixel má vliv na podobu výsledného histogramu, a to tak, že se zařadí do příslušného binu podle hodnoty gradientu, který ho pomyslně překrývá. Navneet Dalal a Bill Triggs také ověřili, že zvětšování počtů binů pozitivně ovlivňuje výkonnost detekce, to platí pro rozsah $0^\circ - 180^\circ$.



Obrázek 4: Znáznornění orientovaných gradientů. Zdroj: http://pavel.campr.cz/public/scikits-image-hog-fix/hog_lena_old.png

3.1.3 Shlukování pixelů do bloků a normalizace

Výsledný výkon detektoru závisí na histogramech jednotlivých buněk. Ty mohou vypovídat o existenci hrany velmi odlišně v závislosti na jasových rozdílech a do jisté míry i intenzitě šumu v jednotlivých buňkách. Rozdíly v kontrastu popředí a pozadí obrazu jsou normalizovány, čímž se napomáhá k lepším výsledkům. Autoři uvádějí testování velkého množství normalizačních schémat. Většina z nich je založena na shlukování buněk do větších bloků a normalizace kontrastu pro každý blok samostatně. Konečný deskriptor se pak skládá z množiny informací, které poskytuje normalizovaná buňka odvozená z jednotlivých bloků. Se shlukováním buněk souvisí i výše zmíněné překrývání buněk. Tímto dochází k normalizaci některých bloků vícekrát, pokaždé ale ve vztahu k jiné buňce.

Navzdory vzniklé redundanci jsou však výsledky uvedené v publikaci představitelů této metody zlepšené.

3.1.4 Shrnutí

Pro implementaci byl zvolen výkonný klasifikátor SVM a trénovací sady obrazů o rozměrech 64x128. Použití algoritmu SVM však není nutností, lze využít také algoritmus AdaBoost. Vytvořený klasifikační model byl podroben testování. V závěrečných testech se metoda HOG ukázala jako schopná konkurovat ostatním metodám a dokonce být i lepší. Představitelé této metody podrobili všechny části procesu detekce lidí touto metodou v reálném čase mnoha testům a experimentům. Výsledkem je velmi schopný postup detekce lidí případně jiných objektů v obraze. Hlavním přínosem této práce není samotné použití histogramu orientovaných gradientů, ale nalezení několika optimalizací při implementaci.

3.2 Lokální binární vzor

Další možností extrakce příznaků z obrazu je metoda LBP (Local Binary Pattern). Stejně jako metoda HOG, tak i tato metoda umožňuje detekci lidských obličejů v ne vždy zrovna ideálních podmínkách osvětlení nebo úhlu obličeje k zařízení zaznamenávajícím obraz. Práce s lokálními binárními vzory je poměrně jednoduchá, není pro výpočetně příliš náročná a ve výsledku nabízí silný nástroj k detekci lidských tváří. Testováním a optimalizací metody LBP přímo za účelem detekce lidského obličeje se zabývali vědci Daniel Maturana, Domingo Mery a Alvaro Soto. Výsledky jejich bádání zveřejnili ve svém článku Face Recognition with Local Binary Patterns, Spatial Histograms and Naive Bayes Nearest Neighbor classification. [4]. V jejich práci rozdělují proces detekce lidských tváří do čtyř hlavních částí:

- Předzpracování: Tato část zahrnuje pouze aplikaci Tanova a Triggsova algoritmu pro normalizaci osvětlení.
- Výpočet LBP: Ve druhé fázi procesu dojde k výpočtu hodnoty LBP pro každý pixel zvlášť s následným vytvořením popisu textury obrazu.
- Extrakce lokálních příznaků: Lokální příznaky jsou nalezeny v menších regionech, do kterých je obraz rozdělen, na základě histogramů vytvořených z LBP hodnot pixelů.

- Klasifikace: Porovnání příznaků získaných z testovacího obrazu s tréningskými daty klasifikátoru.

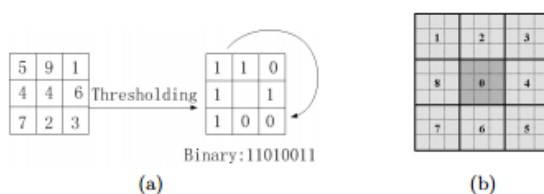
První dva kroky jsou společné pro většinu metod detekce objektů, proto se autoři výše zmíněného článku detailněji zabývali pouze třetí a čtvrtou částí procesu detekce.

3.2.1 Předzpracování

Předzpracování, které Maturana a kolektiv zvolili obsahuje pouze použití jednoho algoritmu, konkrétně Tanova a Triggsova, který však není předmětem této práce. Stačí pouze zmínit, že se jedná o normalizaci osvětlení obrazu. Ačkoliv se toto předzpracování může jevit velmi jednoduše, přesto se ukázalo jako velmi efektivní v kombinaci s určováním lokálních binárních vzorů.

3.2.2 Výpočet lokálních binárních vzorů

Výpočet hodnot LBP probíhá pouze pro vnitřní pixely obrazu, jelikož okrajové pixely nemají dostatek sousedních pixelů pro řádný výpočet. Základní princip výpočtu pro konkrétní pixel obrazu spočívá v prahování hodnot osmi nejbližších pixelů. To znamená, že výpočet probíhá nad devíti pixely, které jsou uspořádány do čtverce 3×3 a výsledná hodnota je přiřazena centrálnímu pixelu ve formě binárního čísla. Zde dochází k experimentování s velikostí vzdálenosti od centrálního pixelu, která je označována parametrem r a počtem bodů, nad kterým je prahování prováděno - parametr s . Označení pro základní výpočet by tedy bylo: $LBP_{8;1}$.



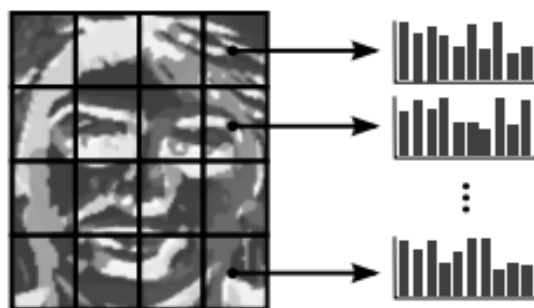
Obrázek 5: (a) 3×3 LBP (b) 9×9 MB-LBP. Obrázek převzat z webové prezentace [4].

Dobré výsledky této metody vedly k jejímu transformování a vylepšování. Jednou z mnoha úprav je MB-LBP (Multi-scale Block Local Binary Patterns), kterou prezentovali Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang a Stan Z. Li v článku Learning Multi-scale Block Local Binary Patterns for Face Recognition [5]. Na rozdíl od LBP dochází

u MB-LBP k výpočtu na základě průměrných hodnot bloků pixelů namísto jednotlivých pixelů. Tento přístup nabízí hned několik výhod, mezi které patří: větší robustnost, dochází k detekci příznaků jako větších celků, efektivnější výpočet za předpokladu použití integrálního obrazu.

3.2.3 Extrakce příznaků a klasifikace

Za předpokladu, že je obraz rozdělen do menších bloků, se pro každý blok vytvoří histogram z odpovídajících LBP hodnot. Výsledkem této metody je množina histogramů popisujících příznaky obrazu pro detekci.



Obrázek 6: Tvorba LBP histogramu. Obrázek převzat z webové prezentace [5].

Poté přichází na řadu samotná klasifikace lidského obličeje. Maturana, Mery a Soto otestovali nejen původní Ahonenovu metodu, ale také dva další postupy - Spatial Pyramid Match a Naive Bayes Nearest Neighbor.

3.2.3.1 Ahonenova metoda Ahonenova metoda je původní postup, pro který je typické rozdělení detekovaného obrazu mřížkou na čtvercové nepřekrývající se oblasti. Každé oblasti odpovídá zvlášť jeden LBP histogram. Ahonen nazval množinu histogramů jako "spatially enhanced histogram". Všechny obrazy z trénovací sady jsou taktéž převedeny na výše zmíněnou množinu. Nakonec je mezi testovací množinou histogramů a trénovacími množinami histogramů provedena klasifikace podle nejbližších sousedů, která stanoví výsledek. U této klasifikace je trénovací množina předzpracována tak, aby příznaky měli hodnotu 0 a rozsah v prostoru 1, poté je klasifikovaný příznak zařazen do té skupiny, kam patří většina jeho prostorových sousedů. Konkrétně v tomto případě

autoři použili x^2 pro porovnání vzdáleností histogramů.

$$x^2(x, y) = \sum_{i=1}^D \frac{(x_i - y_i)^2}{x_i + y_i} \quad (1)$$

Poznámka 3.2 Klasifikace podle nejbližších sousedů patří mezi neparametrické metody klasifikace, které jsou založeny na slabších předpokladech. Podstatou je hledání přímo a posteriori pravděpodobnosti, což znamená pravděpodobnosti založené na smyslové zkušenosti, čímž dojde k přibližnému řešení problému.

Parametr D znamená míru prostorového rozsahu prostorově vylepšených histogramů. V testech se tento postup ukázal jako výkonnější než klasifikace na základě průniků histogramů.

3.2.3.2 Spatial pyramid match Jedním z parametrů Ahonenova postupu je velikost oblastí, do kterých je obraz rozdělen. Stejně jako u jiných metod tak v tomto případě přichází na řadu otázka, do kolika a na jak velké oblasti obraz rozdělit. Některé testy poukazují na to, že použití větších oblastí nabízí pokrytí širšího spektra chyb v detekci. To vedlo Maturanu a kolektiv k vytvoření vícerozměrného histogramu, konkrétně prostorové pyramidy histogramů. Proces výstavby pyramidy z histogramů v prostoru je podobný Ahonenově skládání histogramů v tom, že se používají různé velikosti regionů a také podobné zřetězení hodnot do výsledného vektoru. Avšak díky své struktuře je pyramidou poskytována větší preciznost při zpracování. Tato struktura se může skládat z různého počtu stupňů. Začíná se stupněm nula, kdy dojde k vytvoření jednoho histogramu ze všech hodnot LBP. Pro druhý stupeň se obraz rozdělí na čtyři stejné oblasti a do pyramidy se přidá histogram každé z nich. Tento proces se opakuje tolikrát, kolik je stupňů pyramidy, s tím rozdílem, že se vždy dělí všechny oblasti obrazu z předchozího stupně. Výsledná pyramida je zapsána ve formě jednoho vektoru. Samotná klasifikace je pak také prováděna metodou klasifikace podle nejbližších sousedů avšak v modifikované podobě.

3.2.3.3 Naive Bayes Nearest Neighbor Obě předchozí metody porovnávají lokální příznaky nalezené na stejných pozicích v obraze. Tato metoda nabízí větší flexibilitu při porovnání. Rozdělení obrazu i použití LBP histogramu zůstává stejné, avšak jednotlivé histogramy se nespojují do jednoho výsledného. Popis obličeje je pak reprezentován množinou vektorů namísto jednoho vektoru.

3.2.4 Shrnutí

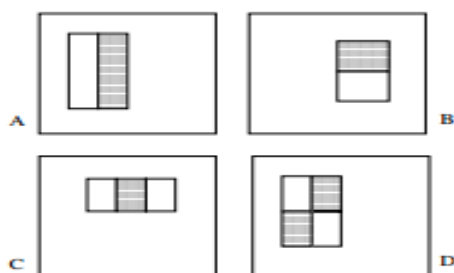
Hlavním závěrem jejich práce se zdá být zjištění, že použití algoritmu klasifikace Naive Bayes Nearest Neighbor zlepšuje výsledky detekce, avšak s tím, že zachovává původní algoritmus založený na LBP při použití rozsáhlých databází lidských obličejů. Metoda NBNN překonává původní algoritmus LBP také v případě, že testovací sada obrazů obsahuje pouze ideální snímky lidských tváří z čelního pohledu. Všechna tato vylepšení jsou výsledkem použitých principů, které jsou popsány v kapitole 3.2.3.3.

3.3 Haarovy příznaky

V této kapitole bude prozkoumána další metoda sloužící k extrakci příznaků z obrazu. Nejenom v případě této metody, ale i obou předchozích je výhodou v práci s příznaky, které popisují jednotlivé pixely, namísto pixelů samotných. Tento přístup ke zpracování obrazu zvyšuje rychlost detekce a umožňuje mnohem komplexnější přístup. Tímto tématem se zabývali autoři článku Rapid Object Detection using a Boosted Cascade of Simple Features [6]- Paul Viola a Michael Jones. V jejich experimentu pracovali s jednoduchými příznaky založenými na Haarových příznacích, avšak oproti těmto se vyznačovaly ještě větší komplexností. Systém, který Viola a Jones vyvinuli je schopen zpracovat patnáct snímků o rozměrech 384x288 pixelů za sekundu za použití 700MHz Intel Pentium 3. Je třeba se zmínit, že se jednalo pouze o obrazy ve stupních šedi. Výkon takového detektoru umožňuje nasazení do běžného života a jeho praktické využití. Navzdory vysokému výkonu je tento systém poměrně jednoduchý.

3.3.1 Extrakce příznaků

Základní princip Haarových příznaků spočívá v rozdílu součtu pixelů ve dvou různých rektangulárních oblastech. Tyto oblasti jsou stejného tvaru, velikosti a jsou buď horizontálně nebo vertikálně sousedící. Dále existují příznaky, u kterých se porovnává mezi třemi nebo čtyřmi oblastmi avšak za stejných podmínek. Z toho vyplývá, že pokud je použitý obraz o velikosti 24x24 pixelů, tak se v něm dá nalézt přes 180 000 příznaků.



Obrázek 7: Příklady Haarových příznaků. Obrázek převzat z webové prezentace [6].

Příznaky takto založené jsou citlivé na výskyt hran a dalších jednoduchých struktur v obraze.

3.3.2 Integrální obraz

Pravoúhlé oblasti představující Haarův příznak lze vypočítat poměrně jednoduchým způsobem za použití integrálního obrazu. Integrální obraz reprezentuje původní obraz a to tak, že každá oblast je označena jako součet hodnot pixelů v oblastech od aktuální pozice nahoru a doleva. Toho lze dosáhnout jedním průchodem původního obrazu. Výpočet jednotlivých oblastí je dále již jednoduchý.

3.3.3 Klasifikace nalezených příznaků

Ke klasifikaci Haarových příznaků použili Viola a Jones AdaBoost, který vybere menší množinu silných příznaků a sestaví klasifikátor z trénovacích dat. Zajímavostí je, že se v každém obraze musí zpracovat větší množství příznaků než je samotných pixelů. Vychází se ze zjištění, že výkonný klasifikátor pracuje s pouze několika silnými příznaky. Otázkou zůstává, jak tyto příznaky vybrat.

Vzhledem k tomuto cíli se využívá slabého učícího algoritmu, který je navržen tak, aby vybral jeden pravoúhlý příznak nejlépe oddělující pozitivní vzorky od negativních. Ve výsledku je pro tento příznak nejmenší procento chybných klasifikací. Proces AdaBoostu vybírá v prvním kole příznaky s chybovostí kolem 0,1, v dalších kolech se objevují příznaky s chybovostí 0,4 - 0,5. Ve výsledku sestavili klasifikátor z 200 příznaků s mírou detekce 95% a mírou nesprávně pozitivních 1 z 14084. To vše proběhlo pro obraz o roz-

měrech 384x288 za 0,7 sekundy. Ačkoliv je tento čas lepší než u jiných metod, stále není dostatečný pro běžné využití.

3.4 SIFT, SURF a GLOH

Metoda příznakového rozpoznání nabízí obecný postup pro detekci jakýchkoliv objektů, to se o všech metodách detekce říci nedá. Nezáleží také na velikosti daného objektu nebo jakékoliv drobné změně oproti trénovacímu obrazu. Díky těmto vlastnostem je tato metoda velice oblíbená, a díky tomu dochází k mnohým vylepšením jednotlivých postupů a vzniku postupů nových. SIFT, SURF a GLOH jsou keypoint detektory, to znamená, že jsou založeny na hledání zájmových bodů v obraze. Tyto zájmové jsou extrahovány za účelem poskytnutí příznaků objektu.

3.4.1 SIFT

Představitelem algoritmu Scale-Invariant Feature Transform je David Lowe. Svůj přínos v této oblasti zveřejnil v článku Distinctive Image Features from Scale-Invariant Keypoints [7]. Tímto přístupem se z obrazu extrahuje velké množství příznaků, například obraz o rozměrech 500x500 pixelů může čítat kolem 2000 příznaků. Toto množství je minimalizováno kaskádovým filtrem příznaků.

Průběh algoritmu SIFT se dá rozdělit do čtyř částí. Prvním krokem je prohledání celého obrazu ve všech měřítkách. To se efektivně provede za použití Gaussových funkcí. Ve druhé fázi se lokalizují klíčové body. Nalezeným bodům se ve třetím kroku přiřadí orientace. Na závěr se vytvoří deskriptor z gradientů, které se určují pro klíčové body v aktuálním měřítku.

Tato metoda se ukázala užitečná převážně díky nezávislosti na velikosti a měřítku klíčových bodů. Vzhledem k extrakci většího množství klíčových bodů je tato metoda úspěšná jak v detekci velkých, tak i malých objektů. Výpočty za účelem získání klíčových bodů a jejich popisu jsou efektivní a dají se provést v téměř reálném čase.

3.4.2 SURF

Dalším algoritmem pro získání příznaků je metoda SURF (Speeded-Up Robust Features). Tato metoda je novější obdobou algoritmu SIFT, takže pro ni také platí, že je nezávislá na velikosti příznaku i na jeho rotaci. Poprvé byla prezentována v roce 2006 na kon-

ferenci v Rakousku. Dobrou dokumentací k této metodě je článek Speeded-Up Robust Features(SURF) [8], jehož autory jsou Herbert Bay a kolektiv.

Algoritmus SURF probíhá ve dvou fázích. V prvním kroku se hledají klíčové body, k tomu se využívá integrální obraz. K označení důležitých bodů v integrálním obraze se používá detektor založený na výpočtu Hessovy matice. Klíčové body musí být nalezeny v různých měřítkách. Druhým krokem je vytvoření deskriptoru z popisu okolí klíčových bodů. Toto je realizováno pomocí gradientů podobně jako u algoritmu SIFT.

3.4.3 GLOH

Nakonec bude představena metoda GLOH (Gradient Location and Orientation Histogram). Zpracované postupy této metody publikovali Krystian Mikolajczyk a Cordelia Schmid v článku A Performance Evaluation of Local Descriptors. Autoři tohoto článku však nejsou původními představiteli této metody. Stejně jako algoritmus SURE, tak i algoritmus GLOH je obdobou metody SIFT. Rozdíl se vyskytuje hlavně v rozložení binů, do kterých se rozdělují gradienty.

4 Implementace pomocí knihovny OpenCV

V této kapitole bude ověřena funkčnost, přesnost a rychlost výše popsaných příznakových detektorů k detekci lidských obličejů v praxi. K implementaci bude využit interface knihovny OpenCV pro Javu. OpenCV je knihovna zaměřená na zpracování obrazu v reálném čase. Vydaná je pod licencí BSD, čímž nabízí volné využití jak pro akademické účely, tak i pro komerční použití. Navíc je tato knihovna multiplatformní.

Testovány budou příznaky HoG, Haar a LBP. Pro zrychlení běhu detekce bude sestavena kaskáda z klasifikátorů, které budou získány pomocí učícího algoritmu AdaBoost. K trénování klasifikátorů se použije vlastní sada trénovacích obrazů.

4.1 AdaBoost

Základní princip AdaBoostu spočívá v redukcí obrovského množství extrahovaných příznaků obrazu do menší skupiny, avšak velmi výkonné. Ve výsledku je vytvořeno různé množství nelineárních silných příznaků. Každý z těchto silných příznaků je lineární množina skládající se z několika slabých příznaků. Proces učení se skládá z velkého množství kol, kde v každém kole je vybrán nejsilnější příznak, který nejlépe vyhovuje pravidlům, a to takovým, že má největší procento úspěšného nalezení v trénovacím obraze a nejmenší procento chybného nalezení. Použití tohoto algoritmu zmínili v článku [6] i Viola a Jones.

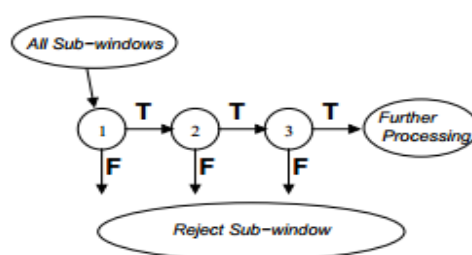
Při použití Haarových příznaků vybral AdaBoost jako první příznak obličeje oblast očí a tváří, vycházející z předpokladu, že oblast očí je vždy tmavší. Příznak vybraný na druhé pozici vychází z předpokladu, že oblast mezi očima je světlejší než samotné oblasti očí.



Obrázek 8: První dva příznaky vybrané AdaBoostem. Obrázek převzat z webové prezentace [6].

4.2 Kaskáda klasifikátorů

Pro zvětšení výkonu detekce je použito kaskádové uspořádání klasifikátorů podle Violy a Jonese [6], díky čemuž se získá výrazné snížení časové náročnosti. Podstatou této metody je, že silné menší a přesto více efektivnější klasifikátory mohou být sestaveny tak, aby vyřadili negativní okénka co nejdříve a množství chybně označených okének se bude blížit nule. Vyřazením velkého množství negativních okének zůstane menší množství s možností výskytu hledaného objektu pro další zpracování. Tímto se zkrátí čas celého procesu.



Obrázek 9: Kaskáda klasifikátorů. Obrázek převzat z webové prezentace [6].

Kaskáda má sama o sobě strukturu rozhodovacího stromu a je rozdělená do několika stupňů. První stupeň obsahuje jeden nebo pár silných klasifikátorů, druhý stupeň může být složen už z poněkud většího množství klasifikátorů a tak dále. Pokud okénko neprojde kterýmkoliv bodem, je okamžitě vyřazeno. Takto proplovají kaskádou všechna okénka dokud nejsou vyřazena nebo přijata jako hledaná.

Stupně kaskády jsou sestaveny z trénovaných klasifikátorů za použití AdaBoostu a snížení intenzity chybně označených negativních okének. Za povšimnutí stojí fakt, že u základního použití AdaBoostu bez kaskády bylo cílem získat příznaky s co nejmenším procentem chybně označených pozitivních příznaků.

Proces trénování kaskády klasifikátorů s sebou nese různé otázky. Ve většině případů budou klasifikátory s více příznaky dosahovat vyšší frekvence detekcí a nižší frekvence chybných označení. Zároveň však větší množství příznaků v klasifikátoru způsobí růst výpočetního času. Ideálním řešením je nalézt vhodný poměr mezi počtem stupňů kaskády, počtem příznaků v každém stupni a vhodnou hranici označení každého stupně. V praxi se hledá toto optimum obtížně. Proto se používá postup, kdy se v každém stupni kaskády snažíme o snížení chybných pozitivních označení a také o snížení detekční frekvence.

Trénování každého stupně probíhá dokud není dosažený bod, který je určen minimálním snížením chybných pozitivních označení a maximálním snížením detekcí.

Kaskáda příznaků pro detekci obličeje navržená Violou a Jonesem čítala 38 stupňů obsahujících přes šest tisíc příznaků. V porovnání s jinými systémy je až několikanásobně rychlejší. Jejich trénovací sadu obrazů tvořilo 4916 ručně označených obličejů s rozlišením 24x24 pixelů. Negativních obrazů obličeje použili 9544, které manuálně prozkoumali, aby se v nich nevyskytovaly obličeje. Rozložení příznaků do prvních pěti vrstev vypadalo následovně: 1, 10, 25, 25, 50. Pro prohledání obrazu v různých měřítkách se mění rozlišení detektoru namísto rozlišení obrazu.

4.3 Vlastní provedení implementace

Pro ověření funkčnosti výše zmíněných příznaků je v praxi použit programovací jazyk Java a knihovna pro zpracování obrazu - OpenCV. Výsledkem bude aplikace pro detekci lidských tváří pomocí příznaků HOG, Haar a LBP za použití kaskády klasifikátorů a učicího algoritmu AdaBoost.

Pro detekci objektů za použití kaskády klasifikátorů nabízí knihovna OpenCV třídu *CascadeClassifier*. K implementaci je nutné znát alespoň tyto dvě metody:

- public boolean load(java.lang.String filename)
- public void detectMultiScale(Mat image, MatOfRect objects, double scaleFactor, int minNeighbors, int flags, Size minSize, Size maxSize)

Metoda *load* slouží k načtení XML souboru popisující kaskádový klasifikátor, přebírá jeden parametr, který popisuje cestu k souboru XML. Druhá zmíněná metoda slouží k detekci objektů v obraze, přičemž dochází ke změně velikosti okénka prohledávání. Metoda *detectMultiscale* přebírá sedm parametrů, kde první dva určují zdrojový obraz a úložiště pro nalezené objekty. Parametr *scaleFactor* určuje jakým poměrem se bude měnit velikost prohledávacího okénka. Dalšími parametry jsou nutný počet sousedících pozitivních okének pro označení a parametr *flags*, který nyní nemá velké využití, proto bude nastaven na nulu. Poslední 2 parametry určují velikost prohledávacího okénka. Použití této třídy je popsáno v následujícím výpise úryvku zdrojového kódu.

```
CascadeClassifier faceDetector = new CascadeClassifier();
// volba kaskady
// 1 ... vlastní hog
```

```
// 2 ... vlastni haar
// 3 ... vlastni lbp
// 4 ... opencv lbp
// 5 ... opencv haar
int FeatureType = 2;
String cascade = "";

switch(FeatureType){
    case 1: cascade = "my_hog.xml";
        break;
    case 2: cascade = "my_hog14.xml";
        break;
    case 3: cascade = "my_lbp.xml";
        break;
    case 4: cascade = "opencv_lbp.xml";
        break;
    case 5: cascade = "opencv_haar.xml";
        break;
}

String xmlCascade;
xmlCascade = getClass().getResource(cascade).getPath();
xmlCascade = xmlCascade.substring(1);

//nacteni kaskady
faceDetector.load(xmlCascade);

if (faceDetector.empty()){
    System.out.println("xml_soubor_nenahran");
    System.exit(-1);
}
int index = 0;

for(File f : imagesArray){

    //nacteni obrazu
    Mat img = Highgui.imread(f.toURI().getPath().substring(1));

    //detekce
    MatOfRect faceDetections = new MatOfRect();
```

```
faceDetector.detectMultiScale(img, faceDetections, 1.05, 9, 0, new Size(24, 24), new  
Size(450, 450));
```

Výpis 1: Ukázka zdrojového kódu v jazyce Java. Zdroj: autor

Knihovna OpenCV nabízí vlastní Xml soubory popisující kaskádu klasifikátorů za použití příznaků Haar a LBP. Pro příznaky HOG bohužel tento soubor neposkytuje, proto je nutné si ho vytvořit. K těmto účelům je nabízena konzolová aplikace *opencv_traincascade*. Této aplikaci je předán vektor s pozitivními obrazy, seznam negativních obrazů, typ příznaku, šířka a výška pozitivních obrazů a počet stupňů výsledné kaskády. Pro případ parametrů HOG, je zvolena výška a šířka vstupních dat stejně jako pro případ parametrů LBP a to i přes skutečnost, že Dalal a Triggs používali u svého postupu [3] tyto parametry výrazně vyšší. Důvodem je možnost lepšího srovnání výsledků za předpokladu použití stejných hodnot parametrů a také fakt, že Dalal a Triggs detekovali lidské postavy namísto obličejů. Ostatní parametry jsou ponechány v původním nastavení. Sada obrazů pro trénování obsahuje 1946 negativních a 1744 pozitivních, vlastnoručně označených vzorků. Trénovací množina byla vytvořena ze skupin obrazů dostupných volně na internetu. Doba nutná pro sestavení kaskád s uvedenými parametry se pohybovala v rámci pár minut, avšak se zvětšováním stupně kaskády rostla exponenciálně. Pro účely testování jsou vytvořeny tři vlastní soubory s názvy a některými důležitými parametry:

- `my_hog.xml -featureType HOG -minPos 1600 -minNeg 1946 -w 24 -h 24 -numStages 12`
- `my_lbp.xml -featureType LBP -minPos 1600 -minNeg 1946 -w 24 -h 24 -numStages 12`
- `my_hog14.xml -featureType HOG -minPos 1600 -minNeg 1946 -w 24 -h 24 -numStages 14`

4.3.1 Experimentální testování

Za účelem testování byla vytvořena vlastní sada dvaceti testovacích obrazů, které zachycují lidské obličej v různých úhlech, za různého osvětlení a s různorodým výrazem ve tváři. Obsaženy jsou také osoby různé barvy pleti, osoby s brýlemi nebo s přítomností vousů ve tváři.

Tabulka na obrázku 10 znázorňuje testování výše zmíněné sady obrazů. Ve výsledném čase je obsažena i doba nutná pro označení místa detekce a uložení této kopie do adresáře. V měření byly použity tyto argumenty:

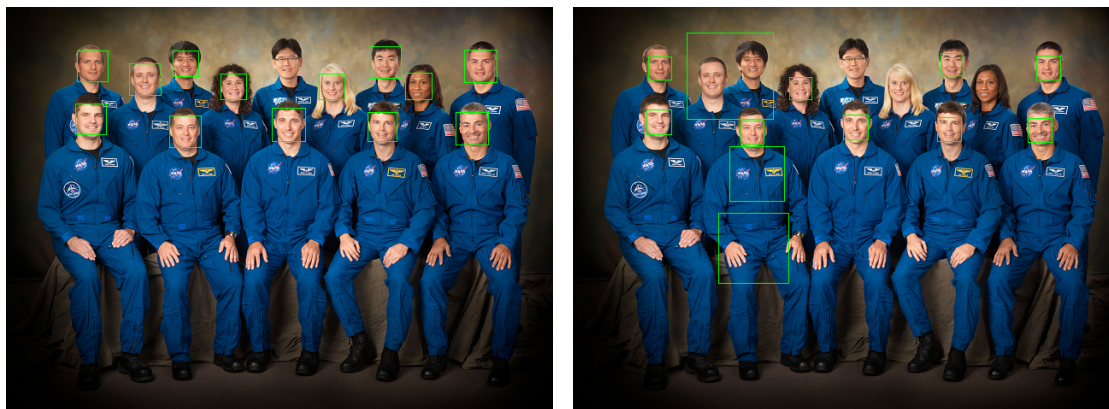
- `detectMultiscale(img, faceDetections, 1.05, 9, 0, new Size(24, 24), new Size(450, 450));`

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	OpenCV_lbp	13	1	2	2	3	1	3	1	5	5	7	7	6	2	7	2	2	1	5	11
	OpenCV_haar	14	1	2	2	3	1	3	1	6	4	7	7	6	2	7	2	2	1	7	12
	my_hog	10	1	0	2	3	1	3	1	5	2	2	3	5	2	4	1	2	1	0	0
	my_lbp	9	1	1	2	3	1	3	1	4	3	5	5	6	2	5	1	2	1	0	1
Počet falešně pozitivních detekcí obličejů	OpenCV_lbp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	OpenCV_haar	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	my_hog	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0
	my_lbp	3	0	3	1	2	0	0	1	0	0	0	1	0	13	0	0	6	2	0	1
Rychlost detekce v sekundách	OpenCV_lbp	3,04																			
	OpenCV_haar	7,65																			
	my_hog	4,22																			
	my_lbp	2,11																			

Obrázek 10: Výsledky měření se stanovenými základními parametry. Zdroj: autor

4.3.2 Vliv velikosti trénovací množiny

Jednou z vlastností, která může ovlivnit výsledky měření je velikost testovací množiny. Sada obrazů, která byla vytvořena pro tvorbu vlastní kaskády obsahuje 1946 pozitivních a 1744 negativních obrazů, přitom kaskády obsažené v knihovně OpenCV byly vytvořeny z až několikanásobně větší množiny. Klasifikátory vybrané z většího množství dat jsou výkonnější, avšak vzhledem k množství mají vyšší časové nároky. Porovnání mezi výkonností a časy je možné pozorovat v tabulce na obrázku 10.



(a) OpenCV_lbp.xml.

(b) My_lbp.xml.

Obrázek 11: Vliv velikosti trénovací množiny. Zdroj: autor.

4.3.3 Vliv parametrů na testování

Dalším kritériem, které se ukázalo jako ovlivňující výsledky měření je kompromis mezi parametry metody *detectMultiscale* a to poměrem jakým se zvětšuje detekční okénko při každém průchodu a počet označených obrazů nutný pro klasifikaci místa jako hledaného objektu. Byly voleny tyto parametry:

- `detectMultiscale(img, faceDetections, 1.1, 9, 0, new Size(24, 24), new Size(450, 450));`
- `detectMultiscale(img, faceDetections, 1.2, 3, 0, new Size(24, 24), new Size(450, 450));`
- `detectMultiscale(img, faceDetections, 1.05, 3, 0, new Size(24, 24), new Size(450, 450));`

Pokud se detekční okénko zvětšuje příliš pomalu, tak se zlepšuje přesnost detekce, ale roste výpočetní čas a naopak. Může také docházet méně často k chybným označením, to však až v kombinaci s parametrem `minNeighbors`, který svým způsobem může filtrovat negativní nálezy. Naměřené hodnoty jsou zdokumentované v obrázcích 11, 12 a 13. Jednotlivé obrázky odpovídají posloupnosti seznamu použitých parametrů.



(a) ScaleFactor 1,2; minNeighbors 3.

(b) ScaleFactor 1,05; minNeighbors 3.

Obrázek 12: Vliv změny parametrů. Zdroj: autor.

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	OpenCV_lbp	10	0	1	2	3	1	3	1	4	3	5	4	6	2	5	1	2	1	4	4
	OpenCV_haar	14	1	2	2	3	1	3	1	5	3	5	7	6	2	7	2	2	1	5	9
	my_hog	2	0	0	1	1	1	2	0	3	0	0	0	2	1	0	0	0	1	0	0
	my_lbp	9	1	0	2	2	1	3	1	2	3	5	3	4	2	2	0	2	1	0	0
Počet falešně pozitivních detekcí obličejů	OpenCV_lbp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OpenCV_haar	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	my_hog	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	my_lbp	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0
Rychlost detekce v sekundách	OpenCV_lbp	1,85																			
	OpenCV_haar	4,72																			
	my_hog	2,71																			
	my_lbp	1,54																			

Obrázek 13: Výsledky měření pro parametry -scaleFactor 1.1 -minNeighbors 9. Zdroj: autor

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	OpenCV_lbp	11	0	2	2	3	1	3	1	4	4	5	6	6	2	6	2	1	1	5	8
	OpenCV_haar	14	1	2	2	3	1	3	2	6	4	5	7	6	2	7	2	2	1	6	12
	my_hog	7	1	0	1	1	1	3	0	3	0	1	3	2	1	0	1	2	1	0	0
	my_lbp	10	1	0	2	2	1	3	1	5	4	5	4	5	2	4	1	2	1	0	0
Počet falešně pozitivních detekcí obličejů	OpenCV_lbp	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
	OpenCV_haar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	my_hog	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0
	my_lbp	4	1	3	0	0	0	0	1	2	0	0	2	0	10	0	1	4	1	1	1
Rychlost detekce v sekundách	OpenCV_lbp	1,54																			
	OpenCV_haar	3,12																			
	my_hog	1,9																			
	my_lbp	1,26																			

Obrázek 14: Výsledky měření pro parametry -scaleFactor 1.2 -minNeighbors 3. Zdroj: autor

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	OpenCV_lbp	14	1	2	2	3	1	3	1	6	5	7	7	6	2	7	2	2	1	11	13
	OpenCV_haar	14	1	2	2	3	1	3	2	6	5	7	7	6	2	7	2	2	1	10	13
	my_hog	14	1	2	2	3	1	3	2	6	2	4	6	6	2	7	2	2	1	1	1
	my_lbp	12	1	1	2	3	1	3	1	4	4	7	6	6	2	6	1	2	1	1	5
Počet falešně pozitivních detekcí obličejů	OpenCV_lbp	4	0	0	0	0	0	1	1	0	2	0	2	1	0	1	2	1	0	0	1
	OpenCV_haar	1	1	0	0	0	0	3	2	0	0	1	2	2	0	0	1	1	1	1	0
	my_hog	6	1	2	4	0	0	0	5	0	6	7	2	2	5	2	1	11	2	2	5
	my_lbp	12	0	5	5	4	0	2	8	2	5	5	6	6	19	0	9	14	3	2	5
Rychlost detekce v sekundách	OpenCV_lbp	3,17																			
	OpenCV_haar	7,79																			
	my_hog	4,3																			
	my_lbp	2,3																			

Obrázek 15: Výsledky měření pro parametry -scaleFactor 1.05 -minNeighbors 3. Zdroj: autor

4.3.4 Vliv předzpracování

Předzpracováním je možné odstranit některé nechtěné vlivy působící na obraz v době pořizování záznamu. Pro redukci šumu je v knihovně nabízeno Gaussovo rozostření nebo jednoduché rozostření. Pokud je obraz v některých místech příliš tmavý nebo příliš světlý může být použito vyvážení histogramu. Popsaná předzpracování byla aplikována a

použita v kombinaci s testovanými detektory. Výsledky měření jsou obsaženy v tabulkách na obrázcích 14, 15, 16.



(a) My_hog.xml.

(b) My_hog.xml s použitím Gaussova rozostření.

Obrázek 16: Vliv předzpracování. Zdroj: autor.

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	OpenCV_lbp	12	1	2	2	3	1	3	1	5	5	7	7	6	2	7	2	2	1	5	10
	OpenCV_haar	14	1	2	2	3	1	3	1	6	4	7	7	6	2	7	2	2	1	6	12
	my_hog	10	1	1	2	3	1	3	1	5	1	4	3	5	2	4	1	2	1	0	0
	my_lbp	13	1	1	2	3	1	3	1	5	3	5	5	6	2	6	1	2	1	0	2
Počet falešně pozitivních detekcí obličejů	OpenCV_lbp	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
	OpenCV_haar	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0
	my_hog	0	0	0	0	0	0	0	1	0	1	2	0	0	0	0	0	1	1	1	0
	my_lbp	7	1	2	0	2	0	1	4	0	0	1	3	0	15	0	1	6	2	0	1
Rychlost detekce v sekundách	OpenCV_lbp	3,19																			
	OpenCV_haar	8,05																			
	my_hog	4,11																			
	my_lbp	2,23																			

Obrázek 17: Výsledky měření po aplikaci Gaussova rozostření. Zdroj: autor

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	OpenCV_lbp	13	1	2	2	3	1	3	1	6	5	7	7	6	2	7	2	2	1	5	8
	OpenCV_haar	14	1	2	2	3	1	3	1	6	4	7	7	6	2	7	2	2	1	6	11
	my_hog	7	1	1	2	3	1	3	1	5	1	3	2	5	2	4	0	2	1	0	0
	my_lbp	12	1	1	2	3	1	3	1	4	4	5	5	6	2	6	1	2	1	0	3
Počet falešně pozitivních detekcí obličejů	OpenCV_lbp	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	
	OpenCV_haar	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	1	0	0
	my_hog	2	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0
	my_lbp	10	0	2	0	2	0	2	3	1	0	0	3	0	17	0	2	8	2	2	1
Rychlost detekce v sekundách	OpenCV_lbp	3,38																			
	OpenCV_haar	8,25																			
	my_hog	4,13																			
	my_lbp	2,38																			

Obrázek 18: Výsledky měření po aplikaci jednoduchého rozostření. Zdroj: autor

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	OpenCV_lbp	13	1	2	2	3	1	3	1	5	5	6	7	6	2	7	2	2	1	6	10
	OpenCV_haar	14	1	2	2	3	1	3	2	6	4	7	7	6	2	7	2	2	1	6	11
	my_hog	3	1	0	1	2	1	3	1	5	1	2	3	5	2	3	1	2	1	0	0
	my_lbp	10	1	1	2	3	1	3	1	3	3	5	5	5	2	5	0	2	1	0	0
Počet falešně pozitivních detekcí obličejů	OpenCV_lbp	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
	OpenCV_haar	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0
	my_hog	3	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	0	0
	my_lbp	1	0	3	0	3	0	0	1	0	0	1	2	0	15	0	0	7	1	0	1
Rychlost detekce v sekundách	OpenCV_lbp	3,13																			
	OpenCV_haar	8,09																			
	my_hog	4,42																			
	my_lbp	2,23																			

Obrázek 19: Výsledky měření po aplikaci vyvážení histogramu. Zdroj: autor

4.3.5 Vliv počtu stupňů kaskády

S rostoucím počtem stupňů kaskády je rostoucí také preciznost rozpoznávání lidských obličejů dané kaskády. Bohužel s tímto přichází i nepatrně rostoucí časová náročnost. Toto potvrzuje porovnání mezi tabulkou na obrázku 10, kde je použita kaskáda o dvanácti stupních a tabulkou na obrázku 17, kde je použita kaskáda o čtrnácti stupních, za předpokladu zachování parametrů, jak trénovacích, tak i detekčních.



(a) My_hog.xml.

(b) My_hog14.xml.

Obrázek 20: Vliv počtu stupňů kaskády. Zdroj: autor.

Označení testovaného obrazu	Typ příznaků	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet obličejů v obraze		14	1	2	2	3	1	3	2	6	5	10	7	6	2	7	2	2	1	12	13
Počet správně detekovaných obličejů	my_hog	10	1	0	2	2	1	3	1	3	2	2	5	6	2	7	2	2	1	0	1
Počet falešně pozitivních detekcí obličejů	my_hog	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
Rychlost detekce v sekundách	my_hog	4,67																			

Obrázek 21: Výsledky měření za použití kaskády HOG příznaků o 14 stupních. Zdroj: autor

4.4 Zhodnocení implementace a testování

Knihovna OpenCV se ukázala jako silný nástroj pro zpracování obrazu a v tomto případě pro implementaci detekce lidských obličejů. Možnost mnoha způsobů práce s obrazem doplňuje dobrou dokumentací ve své webové prezentaci. Výhodou je také podpora více platforem. Samotná implementace detektoru lidských tváří za použití příznakového rozpoznání a knihovny OpenCV nebyla příliš složitá.

V praktické části došlo také k otestování jednotlivých metod za použití různých filtrů, parametrů a trénovacích dat. Jednoznačně nejvíce projevila svůj vliv na výsledky měření

velikost trénovací sady obrazů. Pro detekci objektů jako jsou obličeje je nutností velká a rozmanitá trénovací sada, aby při měření byly eliminovány chyby způsobené velkou rozmanitostí tvarů obličeje, barvy pleti a dalších vlastností typických pro lidskou tvář. S tímto souvisí i počet stupňů kaskády příznaků. Větší počet stupňů kaskády měl také poměrně výrazný vliv na přesnost detekce i za cenu drobného zpomalení.

Vliv předzpracování a parametrů už neměl tak zcela jednoznačné výsledky jako vlivy v případě předchozím. Parametry detekce se dají upravit tak, aby dávaly lepší výsledky detekce, avšak za předpokladu, že jsou známy nějaké konkrétní vlastnosti testovaného obrazu, jako například očekávaná velikost lidských tváří v obraze a nebo jiné vlastnosti. Samozřejmě se lze ale přiblížit k hodnotám parametru, které budou mít dostatečně uspokojující výsledky pro libovolný obecný obraz. Předzpracování hraje také velmi důležitou roli v samotném procesu detekce. Ukázalo se ale, že vliv předzpracování se liší v závislosti na každém konkrétním obraze.

V porovnání rychlosti měření dopadlo LBP z knihovny OpenCV lépe než Haar, avšak na úkor preciznosti. Obdobně dopadlo i porovnání vlastního zpracování kaskády LBP, ale ve vztahu k HOG.

5 Závěr

V této práci byly popisovány a experimentálně ověřeny metody pro získání příznaků z obrazu za účelem detekce lidských obličejů. S použitím knihovny OpenCV byla provedena implementace těchto postupů a následně byly provedeny testy a porovnání rychlosti, přesnosti a funkčnosti.

Pro účely testování bylo použito pět kaskád příznaků a to dvě kaskády klasifikátorů příznaků LBP a Haar z knihovny OpenCV a 3 vlastní kaskády klasifikátorů příznaků LBP, HOG s dvanáctistupňovou kaskádou a HOG se čtrnáctistupňovou kaskádou.

Z výsledků měření, které byly provedeny za účelem srovnání jednotlivých metod pro potřeby této práce, byly vyvozeny následující závěry. Velký vliv na kvalitu detekce lidských obličejů má velikost a rozmanitost trénovací množiny obrazů. V průběhu testování byly v této části odhaleny nedostatky v podobě nedostatečné trénovací množiny obrazů, jak v počtu, tak v různorodosti dat. Zde je možné dosáhnout dalším vývojem podstatnějšího zlepšení výkonu vlastních kaskád klasifikátorů. Další možností, jak dosáhnout lepších výsledků, se zdá být změna parametrů detektoru, která by měla probíhat v závislosti na potřebách konkrétně prováděné detekce s přihlédnutím k fyzickému výkonu systému. Vliv na samotnou detekci může také mít použití předzpracování obrazu. V této práci byly použity filtry Gaussovo rozostření, jednoduché rozostření a ekvalizace histogramu. Není možné stanovit nejlepší volbu předzpracování, protože vliv předzpracování je pro každý obraz individuální v závislosti na jeho kvalitě.

Hlavním přínosem této práce je základní přehled detektorů objektů na základě příznakových rozpoznání HOG, Haar a LBP, dále jejich porovnání a přednosti v závislosti na vlastnostech testovací sady a v závislosti na vlivech velikosti trénovací sady, změny parametrů detektoru, předzpracování obrazu a vlastnosti kaskády. Byla vytvořena konzolová aplikace, která je schopna úspěšně detekovat lidské obličeje pro účely testování, nikoliv však pro běžnou praxi.

Marek Merta

6 Reference

- [1] HOWARTH, J.W.; BAKKER, H.H.C.; FLEMMER, R.C. *Feature-based Object Recognition*. Massey University, Nový Zéland.
- [2] ŠŤASTNÝ, Jiří. *Netradiční metody a algoritmy pro rozpoznávání objektů technologické scény*. Edice Habilitační a inaugurační spisy, sv. 198. Brno: Vutium, 2006. ISBN 80-214-3117-2.
- [3] DALAL, Navneet; TRIGGS, Bill. *Histograms of Oriented Gradients for Human Detection*. International Conference on Computer Vision & Pattern Recognition (CVPR '05) 1 (2005) 886–893.
- [4] MATURANA, Daniel; MERY, Domingo; SOTO, Álvaro. *Face Recognition with Local Binary Patterns, Spatial Pyramid Histograms and Naive Bayes Nearest Neighbor classification*. Departamento de Ciencias de la Computación Pontificia Universidad Católica Santiago, Chile.
- [5] LIAO, Shengcai; ZHU, Xiangxin; LEI, Zhen; ZHANG, Lun; LI, Stan Z. *Learning Multi-scale Block Local Binary Patterns for Face Recognition*. Čínská akademie věd, Peking, 2007.
- [6] VIOLA, Paul; JONES, Michael. *Rapid Object Detection using a Boosted Cascade of Simple Features*. Cambridge, 2001. ISBN 0-7695-1272-0/01.
- [7] LOWE, DVID G.. *Distinctive Image Features from Scale-Invariant Keypoints*. University of British Columbia, 2004.
- [8] BAY, Herbert; ESS, Andreas; TUYTELAARS, Tinne; GOOL, Luc Van. *Speeded-Up Robust Features (SURF)*. Zurich a Leuven, 2008.

A Zdrojový kód detektoru lidských tváří

Příloha na DVD.

Obsažen je Java projekt vytvořený ve vývojovém prostředí Eclipse. Součástí je i dokumentace obsažená ve složce "FaceDetection/doc".

Ke spuštění je nutné mít nainstalovanou knihovnu OpenCV verze 2.4.8 a nastavit správnou cestu ke knihovně.