

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 RASISTICKÝCH SYMBOLŮ Z OBRAZU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MATĚJ KLAPAL

BRNO 2015



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 RASISTICKÝCH SYMBOLŮ Z OBRAZU

DETECTION OF RACIST SYMBOLS IN PICTURES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MATĚJ KLAPAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. LUKÁŠ POVODA

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Matěj Klapal

ID: 158159

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Detekce rasistických symbolů z obrazu

POKYNY PRO VYPRACOVÁNÍ:

Cílem této práce je vytvořit automatický detektor závadného (rasistického) obsahu a prodeje těchto symbolů na internetu. Zpracujte rešerši závadných obrazových symbolů a vytvořte sadu těchto symbolů pro trénování. S pomocí metod detekce objektů v obraze (poskytne vedoucí práce) natrénujte vybrané detektory. Dosažené výsledky statisticky zhodnoťte. Výstupem práce bude konzolová aplikace pro použití na serveru a windows aplikace s rozhraním pro výběr souboru obrázku, případně i URL adresy obrázku.

DOPORUČENÁ LITERATURA:

- [1] Ballard, D. H.; Brown, Ch. M. Computer Vision. Englewood Cliffs, N.J.: Prentice Hall, 1982.
- [2] MARENGONI, Maurício; STRINGHINI, Denise. High level computer vision using OpenCV. In: Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2011 24th SIBGRAPI Conference on. IEEE, 2011. p. 11-24.
- [3] MATUSKA, Slavomir, et al. Classification of Wild Animals based on SVM and Local Descriptors. AASRI Procedia, 2014, 9: 25-30.
- [4] PEREZ, Claudio, et al. Gender classification from face images using mutual information and feature fusion. International Journal of Optomechatronics, 2012, 6.1: 92-119.

Termín zadání: 9.2.2015

Termín odevzdání: 2.6.2015

Vedoucí práce: Ing. Lukáš Povoda

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

ABSTRAKT

Cílem této práce je vytvořit detektor rasistických symbolů z obrazu za použití funkcí open source knihovny OpenCV. V textu je shrnut základní proces zpracování obrazových dat pomocí počítačů. Text dále obsahuje popis jednotlivých metod z této knihovny umožňujících natrénovat a poté v obraze zaznamenat a lokalizovat požadovaný objekt. Součástí tohoto textu je také porovnání úspěšností detekce za použití Haarových příznaků, LBP a histogramu orientovaných gradientů. V textu jsou také shrnuty výsledky testu detekce pro trojici podporovaných symbolů, svastiku, znaky SS a triskelion.

KLÍČOVÁ SLOVA

OpenCV, detekce symbolů, počítačové vidění, zpracování obrazu, Haarovy příznaky, Local Binary Pattern, histogram orientovaných gradientů, AdaBoost

ABSTRACT

Goal of this thesis is detector of racist symbols from the picture using functions from the open source library OpenCV. Text also summarizes description of basic processes of image processing via computers. This text contains descriptions of some methods from the library allowing us to train and afterwards detect and localize requested object. This text also compares accuracy of detection using Haar-like features, Local Binary Patterns (LBP) and histogram of oriented gradients. Text also summarizes results of a test of detection for three supported symbols, swastika, signs of SS and triskelion.

KEYWORDS

OpenCV, detection of symbols, computer vision, image processing, Haar-like features, Local Binary Pattern, histogram of oriented gradients, AdaBoost

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Detekce rasistických symbolů z obrazu“ 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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Lukáši Povodovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych rád poděkoval své rodině za podporu při tvorbě bakalářské práce, ale i během celého studia.

Brno

.....

(podpis autora)



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

PODĚKOVÁNÍ

Výzkum popsany v této 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.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 Počítačové vnímání obrazu	12
1.1 Využití počítačového vidění v různých odvětvích	12
1.2 Vlastnosti obrazu	12
1.2.1 Barevné modely	12
1.2.2 Bitová hloubka	14
1.2.3 Grafické formáty obrazů	15
2 Postup práce s obrazem	17
2.1 Snímání a digitalizace obrazu	17
2.2 Předzpracování	18
2.2.1 Jasové transformace	19
2.2.2 Geometrické transformace	19
2.2.3 Filtrace a ostření	20
2.3 Segmentace obrazu	20
2.3.1 Prahování	21
2.3.2 Detekce hran	21
2.4 Popis objektů	22
2.5 Klasifikace	23
3 Rozpoznávání objektů v obraze	24
3.1 Haarovy příznaky	24
3.2 Local binary pattern (LBP)	25
3.3 Histogram orientovaných gradientů (HOG)	27
3.4 Kaskádový klasifikátor	27
3.4.1 AdaBoost	28
4 Realizace detektoru	29
4.1 Knihovna OpenCV	29
4.2 Využití pro detekci objektů z obrazu	29
4.3 Detekovatelné symboly	30
4.3.1 Svastika	30
4.3.2 SS	31
4.3.3 Triskelion	32
4.4 Softwarová obsluha	32
4.5 Dosažené výsledky	33
4.5.1 Detektor hákových křížů	34

4.5.2	Detektor znaků SS	36
4.5.3	Detektor triskelionů	37
4.5.4	Celkové zhodnocení	39
5	Závěr	40
	Literatura	41
	Seznam symbolů, veličin a zkratk	43
	Seznam příloh	44
	A Slabý klasifikátor	45
	B Postup při detekci	46

SEZNAM OBRÁZKŮ

1.1	Skládání barev u RGB modelu [1]	13
1.2	Trojrozměrné (vlevo) a dvojrozměrné (vpravo) zobrazení modelu HSB [1]	14
2.1	Čtvercová a hexagonální vzorkovací mřížka [3]	18
3.1	Hranové Haarovy příznaky	24
3.2	Čárové Haarovy příznaky	24
3.3	Hodnota integrálního obrazu v bodě (x,y)	25
3.4	Průběh tvorby kaskádového klasifikátoru	28
4.1	Postup pro tvorbu detektoru	30
4.2	Hákový kříž [18]	31
4.3	Znak SS [18]	31
4.4	Triskelion	32
4.5	Konzolové okno	32
4.6	Příklady detekce	33
4.7	Grafické rozhraní	33
4.8	Trénovací sada – hákové kříže	34
4.9	Trénovací sada – znaky SS	36
4.10	Trénovací sada – triskelion	37
B.1	Postup při detekci	46

SEZNAM TABULEK

1.1	Přehled bitových hloubek	15
4.1	Výsledky detekce svastik – pozitivní snímky	35
4.2	Výsledky detekce svastik – negativní snímky	36
4.3	Výsledky detekce znaků SS – pozitivní snímky	37
4.4	Výsledky detekce znaků SS – negativní snímky	37
4.5	Výsledky detekce triskelionů – pozitivní snímky	38
4.6	Výsledky detekce triskelionů – negativní snímky	38

ÚVOD

V dnešní době si lze jen těžko představit život bez počítačů. V téměř každém odvětví jsou využívány technologie, které ještě před několika lety mohly být pouhou představou. Tehdy nikoho jistě nenapadlo, že budeme schopni téměř dokonale převést svět, který vidíme kolem nás, do elektronické podoby, natož aby počítač reagoval na vstupní podnět podle předem definovaných pravidel.

Tato práce je zaměřena na tvorbu automatického detektoru, který reaguje na vstupní obraz obsahující závadné, rasistické symboly. Součástí práce bylo nastudovat jaké symboly patří mezi ty zakázané a vytvořit silný klasifikátor.

V úvodu práce je popsán obraz, jeho důležité vlastnosti a parametry. Představeny jsou barevné modely, grafické formáty obrazů a uvedeny informace o bitových hloubkách. Znalost těchto parametrů je důležitá pro zpracování obrazu.

Další kapitola se zabývá přístupem počítače k obrazovým datům. Je zde vysvětleno, jakou cestu musí obraz, který člověk vnímá pomocí svého zraku, absolvovat, aby bylo dosaženo stejného zobrazení i na monitoru počítače. Předneseny jsou metody předzpracování a segmentace obrazu.

V následující kapitole jsou představeny Haarovy a Local Binary Pattern příznaky a histogram orientovaných gradientů. Tyto metody slouží k popisu obrazu a jsou používány při tvorbě klasifikátoru. Kaskádový klasifikátor je tvořen řadou slabých klasifikátorů, každý s úspěšností alespoň 50 %, pomocí algoritmu AdaBoost (Adaptive Boosting), o čemž také pojednává třetí kapitola.

Součástí práce je také konzolová a grafická aplikace, která umožňuje detekovat trojici podporovaných symbolů. Tyto symboly jsou popsány ve čtvrté kapitole, kde jsou také uvedeny výsledky a porovnání úspěšností jednotlivých klasifikátorů vytvořených pro daný symbol. Klasifikátory jsou tvořeny pomocí Haarových příznaků několika různými variantami AdaBoostu, pomocí příznaků LBP a pomocí histogramu orientovaných příznaků.

1 POČÍTAČOVÉ VNÍMÁNÍ OBRAZU

1.1 Využití počítačového vidění v různých odvětvích

Počítačové vidění je rychle se rozvíjející oblastí. Uplatnění pro automatické pracování s obrazovými materiály se dostává do více a více oborů, a tak se v dnešní době zvyšuje potřeba efektivního a rychlého zpracovávání obrazových materiálů. Rozdíl mezi lidským a počítačovým vnímáním obrazu je v mnoha oborech klíčovou záležitostí. Ať už jsou to kamery detekující pohyb v zabezpečovací technice, fotoaparáty používající detektory obličeje nebo přístroje ve zdravotnictví umožňující rozpoznávání a počítání cév z lékařských snímků v oblasti cévní analýzy.

1.2 Vlastnosti obrazu

Každý obraz je popsán řadou parametrů, které nám jej pomáhají v počítači interpretovat a usnadňují následnou práci s obrazovými daty. Určena je barva obrazu, bitová hloubka nebo grafický formát obrazu.

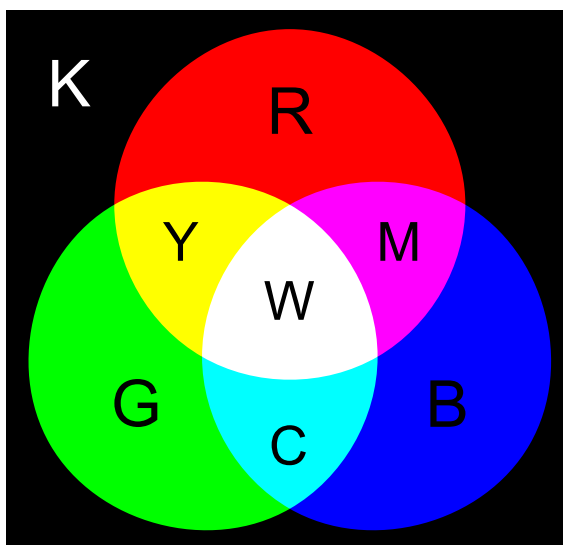
1.2.1 Barevné modely

Barva obrazu je definována pomocí barevných modelů. Barevný model je abstraktní matematický model popisující možnost reprezentování barvy obrazu maticí několika čísel. Nejpoužívanější modely jsou v současnosti RGB, CMY(K) a HSB. Každý z modelů popisuje stejnou barvu jinak. Jako příklad bude použita modrá barva.

RGB

Představa o barevném rozdělení na červenou (R), zelenou (G) a modrou (B) barevnou složku se zrodila již počátkem 19. století na základě zjištění, že lidské oko je citlivé právě na tyto barvy.

Barvy v tomto modelu jsou skládány aditivně, tedy přimícháváním jednotlivých složek o různé intenzitě. Zpravidla nabývají jednotlivé složky hodnot od 0 do 255, 0 značí nulové zastoupení dané barvy, 255 je pak maximální možné. Smíchání všech složek s maximální intenzitou má za následek barvu bílou, jak můžeme vidět na obrázku 1.1. Může být dosaženo téměř jakékoliv barvy z viditelného barevného spektra. Největší uplatnění pro model RGB je v zobrazovací technice. [1]



Obr. 1.1: Skládání barev u RGB modelu [1]

CMY(K)

Barevný model CMY je převážně využíván pro tiskařské účely. Tento model se nazývá subtraktivní, neboť ke skládání barev dochází odečítáním RGB barev od bílé. Skládá se z barev Cyan, Magenta, Yellow – tyrkysová, purpurová a žlutá. Jedná se o doplňkový model k modelu RGB. Barva Cyan vzniká směsí zelené a modré barvy, na barvě Magenta se podílí červená a modrá. Pro barvu Yellow pak kombinujeme červenou a zelenou, jak lze vidět na obrázku 1.1.

Vztah mezi modely CMY a RGB:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Je však obtížné a v tisku ekonomicky nevýhodné získat černou barvu z těchto tří složek, výslednou barvou bývá tmavě hnědozelená, a proto byla přidána do tohoto modelu černá barva, která umožňuje tisk černé a také věrnější podání tmavších barev. [1]

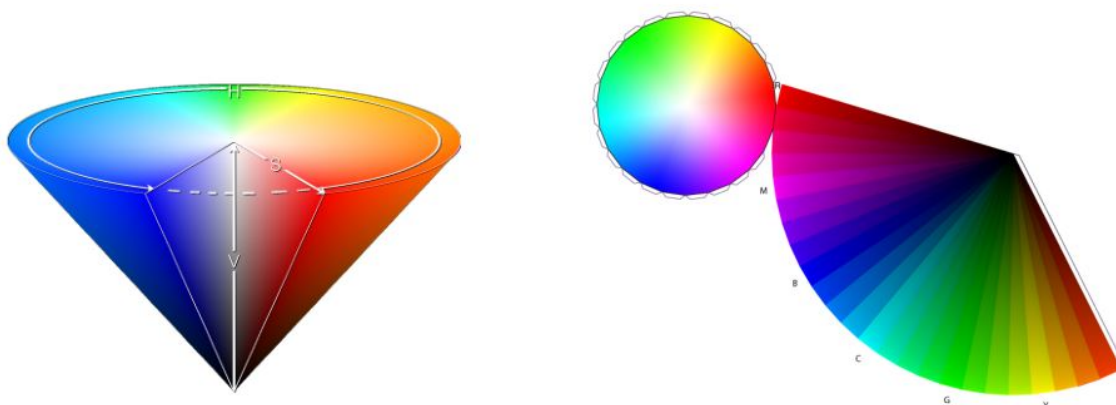
HSB

Pomocí modelů RGB či CMYK je pro člověka obtížné manipulovat s odstínem barvy, proto v roce 1978 vytvořil Alva Ray Johnson barevný model HSB, který je vyobrazený jako trojrozměrný kužel (Obrázek 1.2). Tento model nesestává z barevných složek jako v předchozích případech, je však popisován pomocí tří následujících parametrů:

- **Hue** – barevný odstín – nastavuje se pomocí 360° barevného kola, 0° pro červenou barvu, 120° pro zelenou a 240° pro modrou barvu spektra.
- **Saturation** – sytost – procentuální poměr šedi k odstínu barvy, při 100 % je barva plně sytá. Sytost jedné barvy nastavujeme na přímce od pláště ke středu kuželu rovnoběžně s plochou podstavy.
- **Brightness** – světlost (jas) – určuje, je-li barva světlá či tmavá, se zvětšujícím se jasnem dochází k zesvětlování barvy, nejsvětlejší barva je při 100 %. Jas ladíme na přímce mezi podstavou a vrcholem kuželu. [1]

Porovnání souřadnic téže tmavé modré barvy v modelech RGB, CMYK a HSB:

$$\begin{bmatrix} R = 0 \\ G = 0 \\ B = 255 \end{bmatrix} = \begin{bmatrix} C = 88 \% \\ M = 77 \% \\ Y = 0 \% \\ K = 0 \% \end{bmatrix} = \begin{bmatrix} H = 240 \% \\ S = 100 \% \\ B = 100 \% \end{bmatrix}$$



Obr. 1.2: Trojrozměrné (vlevo) a dvojrozměrné (vpravo) zobrazení modelu HSB [1]

1.2.2 Bitová hloubka

Bitová hloubka určuje, kolik jedinečných barev je dostupných v barevné paletě daného obrazu. Obraz s vyšší bitovou hloubkou může obsahovat více odstínů barev, protože existuje více bitových kombinací. Pro obraz ve stupních šedi pak bitová hloubka specifikuje, kolik jedinečných odstínů může být zobrazeno. V tabulce 1.1 je uveden přehled používaných hloubek. [2]

Tab. 1.1: Přehled bitových hloubek

Počet bitů	Počet barev	Označení
1	$2^1 = 2$	Monochrome
2	$2^2 = 4$	CGA
4	$2^4 = 16$	EGA
8	$2^8 = 256$	VGA
16	$2^{16} = 65\,536$	High Color
24	$2^{24} = 16\,777\,216$	True Color
32	$2^{32} = 4\,294\,967\,296$	Super True Color

Lidské oko je schopné rozeznat 10 milionů barev, fotografie proto nejčastěji bývají 24 bitové, kdy na jeden barevný kanál připadá právě 8 bitů, na každém ze tří kanálů lze tedy zobrazit 256 barev. Ukládání snímku ve vyšší bitové hloubce je již pro účely prohlížení nadbytečné, avšak při následném zpracovávání obrazu má své využití. [2]

1.2.3 Grafické formáty obrazů

Možnosti dalšího zpracovávání velmi závisí na kvalitě obrazu, ta se pak odvíjí od použitého formátu. Různé formáty poté mají při jejich dalším zpracování odlišné vlastnosti.

Obrazy dělíme na vektorové a bitmapové. Základní rozdíl mezi vektorovou a bitmapovou grafikou je ve způsobu ukládání. Bitmapový obraz je specifikován maticí několika bodů (pixelů) různých bitových hloubek a barev, kdežto pro vektorovou grafiku je charakteristické použití geometrických prvků podléhajících přesné definici, jako jsou body, přímky, mnohoúhelníky či křivky, kterými lze popsat jakékoliv tvary.

Vektorová grafika

Nejpoužívanější vektorové formáty:

- SVG – Scalable Vector Graphics – založeno na XML, podpora vektorových animací, využití u webových aplikací.
- PDF – Portable Document Format – vyvinut roku 1993, s kompresí, či bez ní, shodně zobrazuje na různých operačních systémech a zařízeních, používá se hlavně pro zobrazování elektronických knih či dokumentů.

Mezi hlavní výhody vektorové grafiky patří zachování kvality při změnách rozlišení obrazu, nižší velikost než bitmapy a možnost ruční editace. Vektorová grafika je však náročná na přesnost výpočtů a kvalitu algoritmů. [1]

Bitmapová grafika

Úprava bitmapových obrazů je značně výpočetně a uživatelsky jednodušší, než práce s vektorovou grafikou. Počítačové vidění, včetně algoritmů detekujících objekty v obraze, pracuje právě s tímto typem.

Bitmapové formáty dělíme na nekomprimované a komprimované. Komprimované pak mohou být neztrátové a ztrátové. [1]

Rozdělení bitmapových formátů:

- BMP
 - neztrátový,
 - hloubka: 1, 4, 8 a 24 bitů/pixel,
 - bez komprese nebo komprese typu RLE (Run Length Encoding – kódování délkou znaku) – kódovány jsou hodnoty, které se opakují,
 - výhody: jednoduchý, snadně zpracovatelný, rozšířená podpora,
 - nevýhody: slabá komprese. [1]
- GIF
 - Graphics Interchange Format,
 - hloubka: 1-8 bitů/pixel,
 - komprese LZW (Lempel, Ziv, Welch) – slovníkové kódování, neztrátové. [1]
- JPEG
 - Joint Photographic Experts Group,
 - hloubka: 24 bitů,
 - ztrátové i neztrátové verze formátu,
 - vychází z podobnosti barev dvou sousedních pixelů,
 - použití na fotografie,
 - nehodí se na tvorbu logotypů či snímků obrazovky, z důvodu artefaktů v okolí hran v obraze. [1]
- PNG
 - Portable Network Graphics,
 - hloubka až 24bitů/pixel,
 - pixely jsou zpracovány před komprimací,
 - metoda kódování L277 (Lempel, Ziv) – slovníkové kódování. [1]

2 POSTUP PRÁCE S OBRAZEM

Aby bylo možné správně detekovat v obraze určitý objekt, je nezbytné podrobit tento obraz určitému procesu. Tento postup práce s obrazem lze zjednodušeně rozdělit do následujících kroků:

1. Snímání a digitalizace obrazu – záznam a převod signálu do digitální podoby.
2. Předzpracování – odstranění vad obrazu.
3. Segmentace obrazu – rozdělení obrazu na jednotlivé objekty či oblasti.
4. Popis obrazu – kvantifikace objektů nalezených segmentací.
5. Klasifikace – rozřídění objektů podle určitých pravidel.

Uvedený postup je popsán v následujících podkapitolách. [3]

2.1 Snímání a digitalizace obrazu

Snímání obrazu, tedy zachycování požadovaného objektu za použití digitálního fotoaparátu, videokamery či mobilního telefonu, je ve své podstatě převodem optické veličiny na elektrický signál. Výsledná kvalita snímku je pak ovlivňována řadou parametrů, jako jsou např. světelné podmínky, odrazové schopnosti snímaného materiálu nebo použité snímací zařízení. Nejvyšší kvalitu snímky jsou produkovány zrcadlovkami, vývojáři mobilních telefonů mají sice snahu o co největší kvalitu fotografií, ale mezi fotografiemi z telefonů a z digitálních zrcadlovek je pořád velký kvalitativní rozdíl.

Obraz může být matematicky modelován pomocí skalární spojité funkce o dvou nebo třech proměnných. Tato funkce se nazývá *obrazová funkce*. [3] [4]

Dalším nezbytným krokem potřebným k následnému zpracování je digitalizace, tedy převod obrazové funkce na digitální obrazový signál. Digitalizace spojité obrazové funkce $f(x, y)$ probíhá pomocí **vzorkování** do dvourozměrné matice $M \times N$ bodů a **kvantováním** do K kvantizačních úrovní.

Jak vyplývá ze Shannon-Kotělnikova teoremu o vzorkování (rovnice (2.1)), vzorkovací frekvence musí být alespoň dvakrát větší, než nejvyšší kmitočet ze vzorkovaného signálu.

$$f_{vz} > 2 \cdot f_{max} \quad (2.1)$$

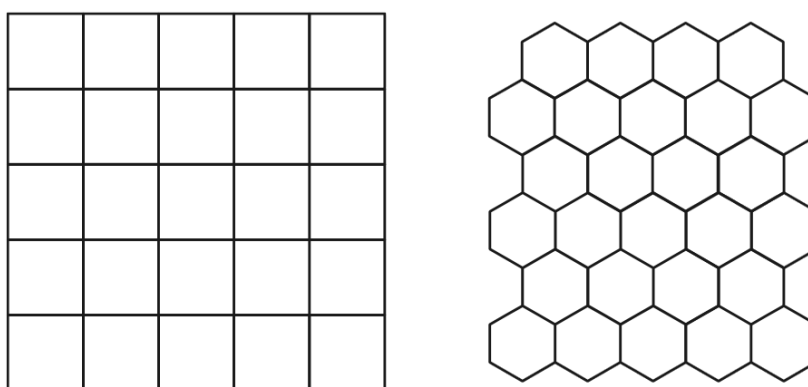
kde

f_{vz} je vzorkovací kmitočet,

f_{max} je maximální kmitočet vzorkovaného signálu.

Vztaženo na obrazový signál pak platí, že interval vzorkování musí být roven nejvýše polovině nejmenšího detailu obrazu. Pro vzorkování se zřídka používají mezní hodnoty, použit bývá většinou interval 5krát menší.

Dalším důležitým faktorem hrajícím významnou roli při procesu vzorkování je volba tzv. vzorkovací mřížky, tj. plošného uspořádání bodů při vzorkování. Mřížka je nejčastěji volena čtvercová, vyskytují se zde však problémy se spojitostí, které řeší mřížka šestiúhelníková, ta ale není tolik využívána. Na obrázku 2.1 jsou zobrazeny tyto dva typy vzorkovacích mřížek. Navzorkované body, odpovídající jednotlivým obrazovým pixelům, jsou pak pravidelně umístěny do vzorkovací mřížky po celé její ploše. [3] [4]



Obr. 2.1: Čtvercová a hexagonální vzorkovací mřížka [3]

Po vzorkování následuje proces zvaný kvantování. Proces, při kterém bude amplituda obrazového signálu převedena na digitální hodnotu. Počet kvantizačních úrovní K odpovídá počtu úrovní jasu na jednom pixelu. Tento počet by měl dosahovat takových hodnot, při kterých dojde k přesnému vyjádření malých detailů obrazu a přiblížení výsledné citlivosti lidskému zraku. [4]

2.2 Předzpracování

Snímáním či procesem digitalizace mohou v obraze vzniknout určité vady. Účelem předzpracování obrazu je eliminovat či minimalizovat výskyt těchto vad před samotným zpracováváním. [5]

Obraz může obsahovat tyto chyby:

- Šum:
 - Aditivní – vzniká při snímání či přenosu obrazu.
 - Bílý – idealizovaný, simuluje nejhorší stav šumu, ve výkonovém spektru jsou zastoupeny všechny kmitočty.
 - Kvantizační – nedostatečný počet úrovní jasu při kvantizaci.

- Multiplikatívni – televizní šum, charakteristické pro něj jsou vodorovné pruhy. [5]
- Zkreslení, může být způsobeno:
 - pohybem snímače a snímaného objektu;
 - nevhodným ostřením;
 - vadou snímače. [5]

Existuje celá řada metod, které umožňují požadovaný obraz předzpracovat a tím odstranit výše jmenované vady. Tyto metody se dělí na:

- jasové transformace,
- geometrické transformace,
- filtrace a ostření. [5]

2.2.1 Jasové transformace

Jasové korekce se využívají zejména při nerovnoměrném osvětlení obrazu nebo pro korigování kontrastu. Zpracovávaný obraz má po provedení transformace stejné parametry jako měl na začátku (bitová hloubka, rozlišení). [5] Jasové transformace se dělí v závislosti na velikosti okolí zpracovávaného bodu na:

- Globální – novou hodnotu definují hodnoty celého obrazu. Např. integrální obraz, Fourierova transformace.
- Lokální – hodnota je určena okolím zpracovávaného pixelu. Např. lokální filtrace, zvýraznění obrysů, hran.
- Bodové – hodnotu vypočítáme z hodnoty odpovídajícího pixelu. Např. jasová korekce, převodní charakteristika. [6]

2.2.2 Geometrické transformace

Pomocí geometrických transformací dosáhneme geometrických úprav obrazu, jako je změna rozlišení, rotace, posuv. Dále nám umožňují odstranit geometrická zkreslení, jako třeba zkosení vzniklé při snímání vzájemnou polohou snímače a snímaného objektu.

Geometrická transformace probíhá ve dvou krocích. Prvním krokem je plošná transformace. Jedná se o transformaci souřadnic bodů obrazové funkce

$$f(x, y) \xrightarrow{T} f'(x', y'), \quad (2.2)$$

pomocí funkce T . Může se jednat o rotaci, zkosení či změnu velikosti. Další fází geometrické transformace je aproximace jasové funkce, tedy určení jasových hodnot pro nové souřadnice obrazové funkce $f'(x', y')$ tak, aby výsledná hodnota jasu odpovídala co nejvíce skutečnosti. [6]

2.2.3 Filtrace a ostření

Filtrace slouží k potlačení vyšších frekvencí a eliminaci šumu či nežádoucích obrýsů a hran. Filtrovat lze:

- Průměrováním jednotlivých pixelů – dochází k rozmazávání hran, výpočetně nejjednodušší metoda.
- Filtrem s Gaussovým rozložením – metoda již složitější na výpočet, avšak přesnější a spolehlivější. [6] Rovnice pro dvourozměrný objekt:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2.3)$$

kde

σ je směrodatná odchylka (specifikuje velikost zpracovávaného okolí bodu x, y), x, y souřadnice obrazu,

a $G(x, y)$ je nová funkce po filtraci.

- Mediánem – číslo, které je po vzestupném uspořádání číselné posloupnosti uprostřed. Touto metodou sice nedochází k rozmazávání hran, ale jsou poškozovány tenké čáry a ořezávány rohy.

Ostření se naopak zabývá zvýrazňováním vyšších frekvencí za účelem zvýraznění hran a rohů, bohužel za cenu vyššího šumu.

Detekce hran je pro práci se symboly základním kamenem, neboť na tomto principu jsou založeny některé algoritmy pracující s obrazem. Hrana je definována jako skoková změna úrovně jasu mezi několika sousedními pixely obrazu, strmost obrazové funkce $f(x, y)$ udává velikost hrany, její směr je pak určen dvousložkovým vektorem. Detektory pracují na principu hledání maxim prvních derivací (Robertsův, Prewittové, Sobelův), nebo na poloze nulových bodů u druhých derivací (Marr-Hildreth). Detekování hran se bude více věnovat část, která se zabývá segmentací obrazu, v kapitole 2.3.2. [7]

2.3 Segmentace obrazu

Nejsložitější operací při zpracování obrazu je jeho segmentace, tedy oddělení jednotlivých významných objektů a oblastí, které budou poté dále zpracovány, od pozadí, které není tolik podstatné. Výsledkem této operace jsou oddělené všechny významné prvky, které odpovídají reálným předmětům. V takovém případě hovoříme o tzv. kompletní segmentaci. Nepodaří-li se obraz rozdělit dokonale a rozdělené části s reálnými objekty nekorrespondují, jedná se o segmentaci částečnou. Segmentování probíhá pomocí prahování nebo detekováním hran, které mohou ohraničovat dané objekty. [8]

2.3.1 Prahování

Metoda prahování je založena na předpokladu, že hodnota barvy a jasu objektů se velmi liší od těch hodnot na pixelech, které jsou specifikovány jako pozadí. Metoda s jediným prahem je patrně nejjednodušší metodou segmentace. Výstupní funkcí $h(x, y)$ pak je binární obraz reprezentovaný pouze pomocí jedniček (objekt) a nul (pozadí).

$$h(x, y) = \begin{cases} 0 & \text{pro } f(x, y) < P \\ 1 & \text{pro } f(x, y) \geq P. \end{cases} \quad (2.4)$$

Jak vyplývá z rovnice (2.4), každý pixel vstupní funkce $f(x, y)$ je porovnáván s hodnotou nastaveného prahu P , a podle jejich vzájemné velikosti je poté nastavena příslušná hodnota výsledného obrazu. Metoda s jedním prahem je použitelná spíše pro segmentování symbolů, či obrazů s velkými rozdíly mezi objektem zájmu a pozadím. Nehodí se příliš pro práci s fotografiemi, v takových případech lze použít více prahů a nastavit jim různé vlastnosti. Je také možné použít práh s dynamickou velikostí, která se liší v závislosti na okolním jasu zpracovávaného bodu dle předchozího nastavení. [8]

2.3.2 Detekce hran

Jak již bylo uvedeno v 2.2, hrana je definována jako skoková změna jasu mezi dvěma body. K určení velikosti a směru hrany slouží matematický operátor gradient. [9]

Gradient

Gradient, neboli směr růstu, můžeme vyjádřit jako výsledek prvních parciálních derivací obrazové funkce $f(x, y)$. Gradient této funkce potom definujeme jako vektor

$$\mathbf{G}[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}, \quad (2.5)$$

kde $\mathbf{G}[f(x, y)]$ označuje směr maximálního nárůstu funkce $f(x, y)$. Velikost tohoto vektoru, tzv. magnitudu, pak určíme ze vztahu

$$\mathbf{G}[f(x, y)] = \text{mag}[\mathbf{G}] = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}. \quad (2.6)$$

Magnituda nám udává velikost dané hrany obrazu. Úhel gradientu $\mathbf{G}[f(x, y)]$, označený jako φ , je poté vypočítán z

$$\varphi(x, y) = \arctan \frac{G_y}{G_x}. \quad (2.7)$$

Směr gradientu je kolmý na detekovanou hranu, není tedy obtížné získat správný směr dané hrany. Spolehlivost detekce je velmi závislá na používané konvoluční masce. Nejpoužívanější hranové detektory pracují s konvolučními maskami o rozměrech 2×2 (Robertsův) a 3×3 (např. Sobelův). Větší masky jsou jemnější, operace jsou pak méně citlivé na šum. Jedním z nejpoužívanějších 2×2 operátorů je Robertsův operátor [9]:

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}. \quad (2.8)$$

Nejvyužívanějším operátorem s maskou 3×3 je Sobelův operátor, kladoucí důraz na hodnoty poblíž středu masky:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \quad (2.9)$$

Laplaceův operátor

Na rozdíl od gradientu, Laplaceův operátor pracuje se druhými derivacemi obrazové funkce. Je definován jako:

$$\mathbf{L}[f(x, y)] \approx \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \quad (2.10)$$

tento operátor používá konvoluční masku o rozměrech 3×3 :

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (2.11)$$

Laplaceův operátor udává velikost hrany, ale neumožňuje určení jejího směru. Z důvodu nepřijatelné citlivosti na šum není tento operátor používán pro detekci hran samostatně, ale jako sekundární detektor pro upřesnění po předchozí detekci. [9]

2.4 Popis objektů

Po rozdělení obrazů do určitých segmentů, definovaných jako objekty a pozadí, je třeba tyto objekty popsat. Toho lze dosáhnout pomocí dvou způsobů. Jeden se zakládá na kvantitativním přístupu k datům, tedy popsání objektů prostřednictvím souboru číselných charakteristik. Mohou nám určovat souřadnice objektu nebo např. velikost. Druhým principem je kvalitativní nalezení určitých vztahů mezi objekty, či informace o jejich tvaru. Volba způsobu popisu je velmi důležitá vzhledem k účelu, ke kterému budou objekty sloužit.

2.5 Klasifikace

Posledním krokem procesu zpracování obrazu je klasifikace. Úkolem klasifikace je roztrždit objekty v obraze podle předem známých pravidel do určitých tříd. V závislosti na metodě popisu objektů pak volíme postup klasifikace. V případě kvantitativního přístupu k popisu se pracuje s tzv. příznaky, sadou charakteristik, určujících parametry požadovaných objektů. Pro účely detekce je nutné klasifikátor na tyto příznaky natrénovat. Více se příznakům věnuje kapitola 3.

Dalším postupem klasifikace je strukturální rozpoznávání. Pracuje s objekty, které byly popsány kvalitativním přístupem. Objekty jsou zde popsány pomocí tzv. primitiv, což jsou základní popisné elementy. Takové třídy pracují s vlastním slovníkem a gramatikou přizpůsobenou pro co nejrychlejší zpracování. [3]

3 ROZPOZNÁVÁNÍ OBJEKTŮ V OBRAZE

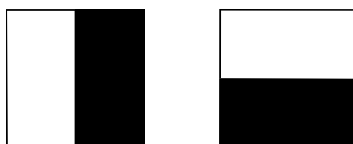
Předešlá kapitola se zabývala postupem zpracování obrazu. Tato kapitola bude věnována části zmíněné na závěr, tedy popisu obrazu.

Obraz je popisován pomocí příznaků, které umožňují natrénovat klasifikátor na požadované objekty. Popisovány zde jsou Haarovy příznaky, Local Binary Pattern (LBP) a histogram orientovaných gradientů. Tyto příznaky jsou poté použity k vytvoření klasifikátoru, který slouží pro klasifikaci a následnou detekci natrénovaných objektů v obraze.

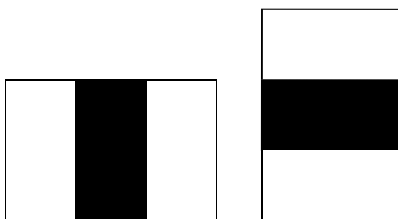
3.1 Haarovy příznaky

Základem zavedení příznaků do zpracování obrazů je fakt, že práce s příznaky výrazně urychluje celý proces oproti práci s jednotlivými pixely. Toho si byli vědomi vědci Viola a Jones [10] již v roce 2001, když použili Haarovy příznaky pro detekci objektu v obraze, konkrétně lidského obličeje.

Příznaky vypočítáme jako součet pixelů ležících v bílé ploše odečtený od součtu pixelů v ploše tmavé. V závislosti na vstupních datech pak používáme správné příznaky. Hranové příznaky (obrázek 3.1), které pracují se dvěma obdélníky, mohou dosahovat hodnot až 43 200 příznaků. Čárové příznaky pracují se třemi obdélníky, z nichž jeden je černý (obrázek 3.2). Zde můžeme mít 27 600 příznaků, pro okno s rozměry 24×24 . Existuje mnoho dalších příznaků, např. výše uvedené pootočené o 45° , středové, s tmavými okraji, či diagonální.



Obr. 3.1: Hranové Haarovy příznaky



Obr. 3.2: Čárové Haarovy příznaky

Každé z oblastí je přidělena váha, v závislosti na barvě dané oblasti. Váha obdélníkové oblasti bílé barvy je stanovena na $\omega_0 = -1$, váha černé je určena podílem

bílých a černých ploch. Hodnotou příznaku je myšlena jeho odezva na vstupní obraz. [11] Je vyjádřena vzorcem:

$$f(x) = \omega_0 r_0 + \omega_1 r_1, \quad (3.1)$$

kde

$f(x)$ – odezva Haarova příznaku na vstup,

ω_0 – váha bílé obdélníkové oblasti r_0 ,

ω_1 – váha černé obdélníkové oblasti r_1 .

Integrální obraz

Obdélníkové příznaky mohou být vypočteny velmi rychle s použitím integrálního obrazu. Integrální obraz na pozici (x, y) obsahuje součet všech pixelů nad a vlevo od požadovaného bodu (x, y) , viz obrázek 3.3. Jeho hodnota je určena ze vztahu:

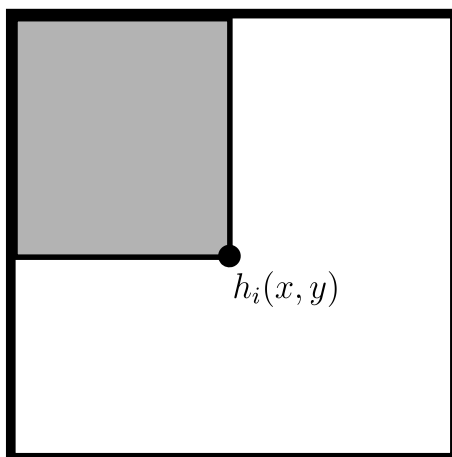
$$h_i(x, y) = \sum_{x' \leq x, y' \leq y} h(x', y'), \quad (3.2)$$

kde

$h(x, y)$ – originální obraz,

$h(x', y')$ – funkční hodnota každého bodu po bod $h_i(x, y)$,

$h_i(x, y)$ – hodnoty integrálního obrazu. [10]



Obr. 3.3: Hodnota integrálního obrazu v bodě (x, y)

3.2 Local binary pattern (LBP)

Operátor LBP je poměrně jednoduchá a účinná metoda analýzy obrazu. Byl představen, nejprve za účelem rychlejšího měření kontrastu obrazu, již v roce 1993. Hodnota

základního LBP příznaku je určena porovnáním požadovaného pixelu s jeho 8 sousedními pixely. Všechny tyto pixely jsou spolu s požadovaným umístěny do matice o rozměrech 3×3 . Střední pixel představuje práh. Ostatní pixely z této matice jsou s tímto prahem porovnány, a vzhledem k jejich vzájemné hodnotě jim je poté přidělena výsledná hodnota podle vzorce: [12]

$$LBP_{P,R} = \sum_{p=0}^{p-1} s(g_p - g_c)2^p, \quad (3.3)$$

$$s(x) = \begin{cases} 0 & \text{pro } x < 0 \\ 1 & \text{pro } f(x, y) \geq 0, \end{cases} \quad (3.4)$$

kde

- g_c je hodnota šedi středního pixelu,
- g_p je hodnota sousedních pixelů,
- x je hodnota rozdílu $g_p - g_c$,
- P je celkový počet sousedních pixelů,
- R je zpracovávaný okruh pixelu.

$$\begin{bmatrix} 2 & 3 & 2 \\ 3 & 4 & 6 \\ 7 & 5 & 8 \end{bmatrix} \xrightarrow{\text{Prahování}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & T & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.5)$$

Po provedení operace prahování (práh T) jsou přiděleny jednotlivým pixelům váhy podle mocnin čísla 2. Hodnota příznaku je odečtena z prahované matice z levého horního rohu ve směru hodinových ručiček. Výsledkem je binární číslo o počtu osmi bitů. Pro výše uvedený případ v rovnici (3.5) je to tedy

$$LBP_{8,1} = 0111\ 1000_b = 120_d.$$

Maximální počet LBP příznaků je roven jejich maximální hodnotě $2^P = 2^8 = 255$. Po obdobném určení všech příznaků z celého obrazu jsou získané hodnoty umístěny do vektoru příznaků, nebo histogramu. U příznaků LBP je snaha o to, aby byly rotačně invariantní, tedy aby neměnily svoji hodnotu při různé rotaci totožného objektu v obraze. V praxi je toho dosaženo rotací hodnot v matici nebo bitovým posunem příznaků, aby dosahoval co nejnižší binární hodnoty. Pro příklad z (3.5) to bude vypadat následovně: [12]

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & T & 1 \\ 1 & 1 & 1 \end{bmatrix} \gg \begin{bmatrix} 1 & 1 & 1 \\ 0 & T & 1 \\ 0 & 0 & 0 \end{bmatrix} \approx 0000\ 1111_b = 15_d \quad (3.6)$$

Touto operací zároveň dojde k výraznému snížení maximálního počtu LBP příznaků v jednom obraze.

Multi-scale Block LBP

Použití pouze 3×3 rozměrné matice však přináší jistá omezení. Z důvodu malé zpracovávané oblasti jsou vzájemně porovnávány pixely značně ovlivněny šumem a v případě větších objektů v obraze je nedokáže dokonale pokrýt. Z toho důvodu byla představena novější reprezentace Multi-scale Block LBP (MB-LBP), která tyto omezení eliminuje. Na rozdíl od původního LBP, jehož hodnota byla vypočtena porovnáváním hodnot jednotlivých pixelů, MB-LBP počítá svoji hodnotu porovnáváním průměrných hodnot šesti jednotlivých podoblastí. Každá podoblast obsahuje skupinu sousedících pixelů (případně pouze jediný pixel). Hodnota jednotlivých podoblastí MB-LBP může být získána velmi snadno použitím integrálního obrazu. Tím je velmi urychlen celý proces výpočtu příznaku. [13]

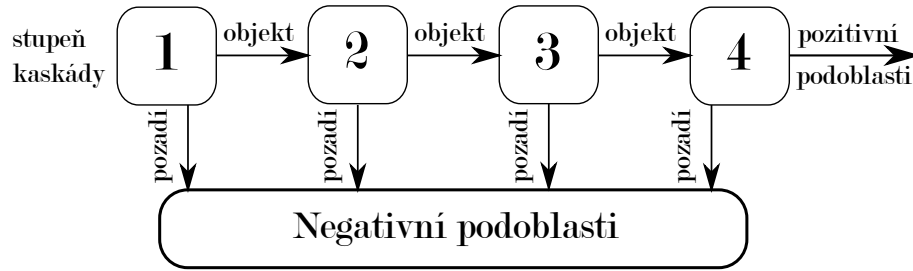
3.3 Histogram orientovaných gradientů (HOG)

Dalším způsobem, jak lze vytvořit klasifikátor, je použít histogram orientovaných gradientů (z angl. Histogram of Oriented Gradients - HOG). Základem pro tuto metodu je správná detekce hran v obraze, nutné je také určit magnitudu této hrany. Tomuto tématu byla věnována kapitola 2.3.2. Obraz je rozdělen do několika oblastí, ve kterých jsou vypočteny jednotlivé magnitudy pro danou hranu. Četnost výskytu magnitud pro danou oblast je uložena do histogramu.

3.4 Kaskádový klasifikátor

Klasifikace je, jak již bylo předesláno v kapitole 2.5, rozdělení obrazových objektů do tříd podle definovaných pravidel. Třídy jsou zde definovány dvě, pozitivní a negativní třída.

Kaskádový klasifikátor (Haar Cascade Classifier) je tvořen kaskádou, tedy celou řadou slabých klasifikátorů. Slabé klasifikátory jsou tvořeny malým množstvím příznaků (v ideálním případě je použit pouze jeden příznak), prahem, či definované pomocí histogramu. Tato řada klasifikátorů je posléze sloučena do jednoho silného klasifikátoru. Účelem kaskády je snaha o nejrychlejší nalezení požadovaného objektu. Postup trénování probíhá následovně. Nejprve jsou odstraněny všechny negativní podoblasti, takové, které neobsahují hledaný symbol. Pozitivní podoblasti pokračují k dalšímu stupni klasifikátoru. Tato metoda je nazvána Adaptive Boosting. Metoda je popsána v následující podkapitole a tento postup je zobrazen na obrázku 3.4.



Obr. 3.4: Průběh tvorby kaskádového klasifikátoru

3.4.1 AdaBoost

Metoda několika složených slabých klasifikátorů je nazvána AdaBoost, tedy Adaptive Boosting. Výstupem je lineární klasifikátor $H(x)$, může se jednat o velmi silný klasifikátor. Klasifikátor je tvořen lineární kombinací slabých klasifikátorů $h(x)$. Slabé klasifikátory vykazují velmi malé procento úspěšnosti, jsou srovnatelné s náhodným tipováním. Na vstupu se nachází trénovací množina $(x_1, y_1; x_2, y_2; \dots; x_j, y_j)$, kde x_j je n -rozměrný vektor použitých příznaků a y_j značí, jedná-li se o pozadí: $y_j = -1$, nebo je-li to objekt: $y_j = 1$. Díky použitým hodnotám $(1, -1)$ se hovoří o takzvaném diskrétním AdaBoostu (DAB). [15]

Základní podmínkou pro vytvoření jednoho slabého klasifikátoru je eliminovat alespoň 50 % podoblastí a správně je určit jako pozadí. Je-li toho dosaženo, proces pokračuje dále s vyšším stupněm klasifikátoru. Pokud se nepodaří dosáhnout požadované hranice, je vygenerován další příznak a proces se opakuje. Příklad slabého klasifikátoru na počátečním stupni tréninku je uveden v příloze A. [16][15]

Varianty AdaBoostu:

Kaskádový klasifikátor lze natrénovat použitím následujících typů boostování:

- Real AdaBoost – RAB – Na rozdíl od DAB pracuje s reálnými čísly, dosahuje vyššího výkonu a nižších chybovostí.
- Logit Boost – LB – Užitečný při zašuměných vstupních datech, dochází k nižším ztrátám.
- Gentle AdaBoost – GAB – Jemnější verze AdaBoostu, větší výkon než Discrete a Real AdaBoost, eliminuje špatně klasifikované trénovací příklady. Pro trénink je potřeba méně výpočetního výkonu. [15]

4 REALIZACE DETEKTORU

Detektor jsem vytvořil a implementoval v jazyce C++ pomocí knihovny OpenCV. Aplikace byla navržena ve vývojovém prostředí Microsoft Visual Studio 2012 ve verzi Ultimate.

4.1 Knihovna OpenCV

Knihovna OpenCV (z anglického Open source Computer Vision) je, jak již z názvu vyplývá, knihovna s veřejně dostupným kódem, pro soukromé a komerční účely k dispozici zdarma. Byla představena v roce 1999 firmou Intel, a od té doby je v neustálém vývoji. První oficiální verze byla, po sérii betaverzí, zveřejněná v roce 2006. V současné době je nejnovější finální verzí knihovny verze 2.4.11, tato verze je dostupná od 26. 2. 2015. Na tuto práci byla použita knihovna ve verzi 2.4.9, zveřejněná dne 16. 4. 2014. Knihovna nyní obsahuje již více jak 500 optimalizovaných algoritmů umožňujících zpracovávat obrazová a video data. Knihovna byla napsána v jazycích C a C++ a plně pracuje na operačních systémech Linux, Windows a Mac OS X.

Pomocí knihovny OpenCV lze jednoduše, s použitím několika příkazů, zobrazovat obrázky, načítat video, a poté upravovat jejich vlastnosti. Knihovna také může pomoci s transformací obrazu, dostupná je zde diskrétní Fourierova transformace (DFT), diskrétní kosinová transformace (DCT). Detekovat hrany umožňuje použití Sobelova operátoru, ten může být využit při segmentaci obrazu. Díky této knihovně také může být sledován pohyb ve videu, a to dokonce ve třech rozměrech. Tato knihovna má nepřehledné množství využití, pro tuto práci jsou nejvíce zajímavé algoritmy použitelné pro detekci objektu z obrazu. [17]

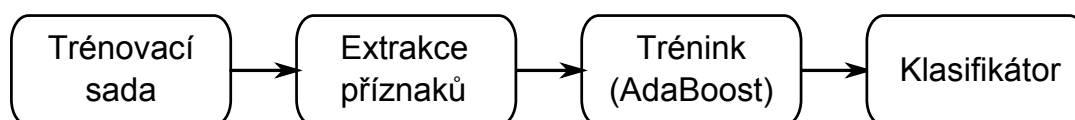
4.2 Využití pro detekci objektů z obrazu

V knihovně OpenCV jsou již předpřipraveny kaskádové klasifikátory pro detekci obličejů a částí lidského těla. Pro účely detektoru rasistických symbolů je třeba vytvořit vlastní kaskádový klasifikátor. K tomu je třeba velké množství pozitivních a negativních vzorků. Pozitivní vzorky jsou obrázky, které obsahují požadované symboly, na které bude poté detektor trénován. Je důležité, aby tyto vzorky měly stejné poměry stran. Negativními vzorky jsou myšleny ty obrázky, ve kterých není žádný z detekovaných symbolů. Tyto budou použity jako pozadí pro trénink detektoru.

Nejprve je nutné vytvořit dvojici textových souborů, jeden pro pozitivní a jeden pro negativní snímky, které budou obsahovat seznam a cestu k daným snímkům. Seznam pozitivních snímků dále musí obsahovat parametry vstupního snímku, jako

je počet symbolů v jednom snímku, počáteční souřadnice symbolu a celková výška a šířka daného snímku. V knihovně OpenCV je implementován nástroj `createsamples`, který ze snímků v textovém souboru vytvoří soubor vektorů `soubor.vec`, tento soubor je poté klíčovým pro tvorbu klasifikátoru.

Nyní následuje samotné vytvoření klasifikátoru. K tomuto účelu je použita knihovnou podporovaná funkce `traincascade`. Tato funkce vytvoří pomocí algoritmu AdaBoost z připravených vzorků v několika stupních silný klasifikátor. Je na uživateli, jakou z metod boostování si zvolí, dostupné jsou algoritmy GAB, DAB, RAB a LB. Důležitou volbou také je, bude-li klasifikátor trénován generováním Haarových či LBP příznaků, nebo pomocí histogramu orientovaných gradientů. V této práci byly použity tyto tři zmíněné druhy příznaků a jejich výsledky jsou porovnány v kapitole 4.5. Postup tvorby detektoru je zobrazen na obrázku 4.1.



Obr. 4.1: Postup pro tvorbu detektoru

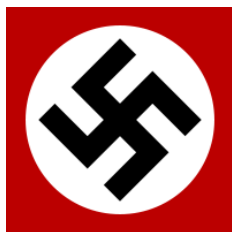
4.3 Detekovatelné symboly

V České republice není stanoven seznam zakázaných symbolů, avšak podle § 403 zákona č. 40/2009 Sb., trestní zákoník, ve znění pozdějších předpisů, je trestná podpora a propagace hnutí směřujících k potlačování lidských práv a svobod, s trestem odnětí svobody až na deset let. Subjektů vyvíjejících takové aktivity je v České republice celá řada. Velmi dobře je tato problematika popsána v publikaci [18], která slouží jako pomůcka pro policii a orgány činné v trestním řízení.

Tento detektor umožňuje detekovat svastiky, znaky SS a tzv. triskelion, logo mezinárodní organizace Blood&Honour.

4.3.1 Svastika

Hákový kříž, svastika, má svůj původ hluboko v historii. Původně byl symbolem spjatým s náboženstvím, významnou roli hrál v buddhismu (symbol Buddha), v hinduismu (symbol evoluce), výjimečně se objevil i v křesťanství. Nejstarší nálezy obsahující tento symbol se datují do 2. tisíciletí př. n. l. Svastika byla využívána po celém světě jako dekorativní prvek, objevila se dokonce i na několika stavbách na území České republiky. Svastika má podobu kříže s pravouhle zalomenými rameny. V závislosti na směru zalomení se jedná buď o levotočivou, nebo pravotočivou.



Obr. 4.2: Hákový kříž [18]

Negativně se na tomto symbolu podepsalo až 20. století, kdy od roku 1920 začala oficiálně symbol používat nacistická strana NSDAP. Nacisté vycházeli z teorií, které tvrdily, že svastika je starodávný symbol árijské rasy. V té době začala svastiku používat řada organizací sympatizujících s nacisty. Od roku 1933 se dokonce svastika objevovala na vlajce nacistického Německa (obrázek 4.2). Z těchto důvodů došlo v průběhu 30. let k výraznému omezení používání svastiky ve světě a svastika začala být chápána jako symbol náklonnosti k Hitlerovu režimu.

V současné době svastiku používají nacistické a neonacistické organizace a její použití je v řadě zemí trestné. Symboly mohou být na částech oděvu, vytetované na lidském těle, natisknuté na různé plakáty či na vlajkách. [18]

4.3.2 SS

Označení SS pochází ze slova Schutzstaffel (česky ochranný oddíl). Tento oddíl sestával za dob Hitlerovy vlády z jeho nejoddanějších následovníků a sloužil původně jako jeho osobní stráž. Ve skutečnosti však stáli za vraždami 14 milionů lidí. Symbolem organizace byly dvě za sebou jdoucí starogermánské runy Sowulo (též Sig Rune) podobající se písmenu S (obrázek 4.3). [18]



Obr. 4.3: Znak SS [18]

4.3.3 Triskelion

Název pochází z řeckého slova triskelis, trojnohý. Symbol je tvořen podobně jako svastika, s tím rozdílem, že ramena jsou pouze tři a úhel zlomu je ostrý (obrázek 4.4). Tento symbol lze také nalézt ve variantách se zakulacenými či spirálovitými tvary ramen.



Obr. 4.4: Triskelion

Symbol ve světě používá organizace AWB, afrikánské hnutí odporu z jižní Afriky, které bojovalo za udržení apartheidu a snažilo se o vznik samostatného státu. Symbol dále také používají mezinárodní neonacistické organizace Blood and Honour a Combat 18, jejichž odnože působí také na území České a Slovenské republiky pod názvem B&H Division Bohemia a Terrormachine Bohemia. [18]

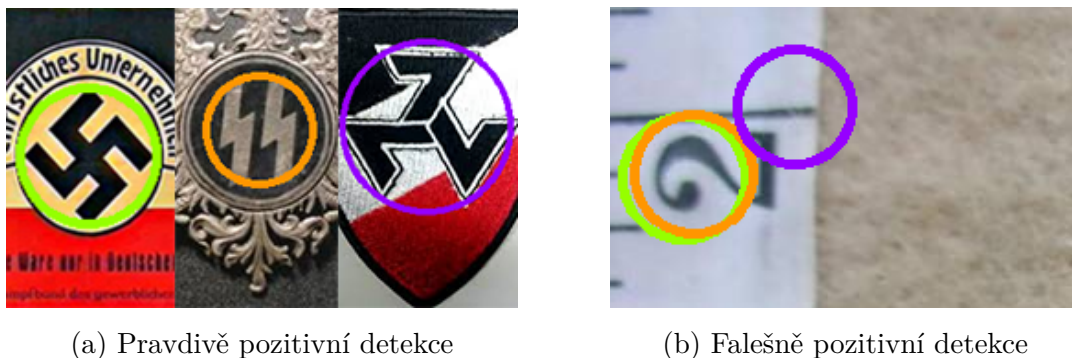
4.4 Softwarová obsluha

Jak již bylo zmíněno, program byl vytvořen v jazyce C++ ve vývojovém prostředí Visual Studio 2012 Ultimate. Program je spustitelný jako konzolová aplikace pomocí příkazového řádku, zadáním příkazu `Detektor` s jediným parametrem, kterým je název či cesta k požadovanému souboru. Na vstupu jsou podporovány grafické formáty jpg, bmp a png. Detektor po spuštění a zobrazení požadovaného obrázku umožňuje detekovat trojici podporovaných symbolů současně. Svastiky jsou označovány zeleným kolečkem kolem daného symbolu, pro znaky SS je kolečko oranžové a pro triskelion bylo zvoleno fialové kolečko, jak je zobrazeno na obrázcích 4.6a a 4.6b. Zároveň dojde k vypsání počtu detekovaných symbolů do konzolového okna, příklad na obrázku 4.5.

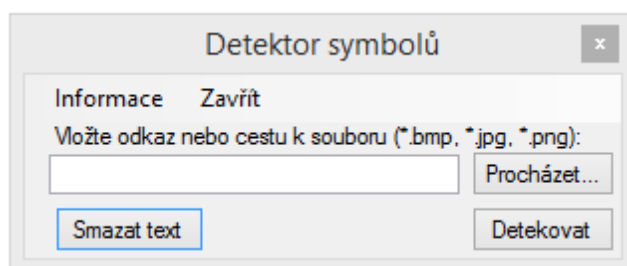
```
Celkem bylo detekováno 1 symbolu. Z toho 1 hakovych krizu, 0 log SS a 0 log blood & honour.
```

Obr. 4.5: Konzolové okno

Dále byla vytvořena aplikace s grafickým uživatelským rozhraním (obrázek 4.7). Z prostředí aplikace je možno vybírat soubory k detekci přímo ze složek, či zadávat URL odkazy na obrázky umístěné na webu. Po vybrání souboru či vložení



Obr. 4.6: Příklady detekce



Obr. 4.7: Grafické rozhraní

URL odkazu a po kliknutí na tlačítko **Detekovat** dojde ke spuštění konzolového okna a zobrazení obrázku s detekcí, resp. detekcemi.

4.5 Dosažené výsledky

V následujících podkapitolách jsou shrnuty výsledky jednotlivých vytvořených klasifikátorů pro různé symboly. Postup detekce je uveden v příloze B. Žádný ze snímků, který byl použit v trénovací sadě, nepadá do testovací sady. Úspěšnost detekce je uváděna v procentech, úspěšnost je dána poměrem detekovaných a celkovým počtem vyobrazených symbolů.

Pro testování a stanovení úspěšnosti jednotlivých klasifikátorů jsou uváděny tyto typy detekcí:

- Pravdivě pozitivní – True Positive (TP) – Dojde ke správné detekci a lokalizaci symbolu v místech jeho výskytu.
- Falešně pozitivní – False Positive (FP) – Dojde k označení místa, ve kterém se nenachází žádný požadovaný symbol.
- Pravdivě negativní – True Negative (TN) – Žádný symbol se v obraze nenachází a nic není detekováno.

- Falešně negativní – False Negative (FN) – Neúspěšná detekce, v místech výskytu daného symbolu není detekován.

Vzhledem k tomu, že se během testování vyskytla řada falešně pozitivních detekcí (obrázek 4.6b), je nutné do celkové úspěšnosti klasifikátoru také zahrnout jejich výskyt. Hodnota úspěšnosti falešně pozitivních detekcí je tedy určena vzorcem (4.1).

$$N = \left(1 - \frac{N_{FP}}{N_p}\right) * 100 [\%], \quad (4.1)$$

kde

N – úspěšnost falešně pozitivních detekcí,

N_{FP} – počet snímků s FP detekcemi,

N_p – počet všech pozitivních snímků.

4.5.1 Detektor hákových křížů

Pro tvorbu klasifikátoru detekujícího hákové kříže bylo použito 126 pozitivních (obrázek 4.8a) a 300 negativních snímků (obrázek 4.8b).



(a) Pozitivní vzorky



(b) Negativní vzorky

Obr. 4.8: Trénovací sada – hákové kříže

S touto kombinací jsem vytvořil řadu klasifikátorů a porovnával úspěšnosti jednotlivých detekcí. Byly vytvořeny klasifikátory pomocí histogramu orientovaných gradientů, pomocí příznaků LBP a pomocí nejrůznějších variant adaptivního boostingu také s Haarovými příznaky (RAB, GAB, DAB, LB). Každý klasifikátor sestával z 20 slabých klasifikátorů. Testovací sada sestávala ze 60 pozitivních snímků, na nichž se nacházelo celkem 117 symbolů. Negativních snímků, které byly rovněž součástí testovací sady, bylo také 60. Výsledky testů úspěšností jsou uvedeny v tabulce 4.1 pro test na pozitivní snímky a pro test na snímky negativní v tabulce 4.2.

Procentuálně dosáhl nejlepších výsledků klasifikátor vytvořený pomocí AdaBoostu pracující s reálnými čísly – Real AdaBoost. V porovnání s ostatními detektory dosahuje tento detektor bezkonkurenčních výsledků, co se týče falešně pozitivních detekcí. Pouze ve dvou snímcích došlo k chybnému označení místa výskytu symbolu. Co se však týče detekovaných symbolů, vykazuje nejnižší úspěšnost, podařilo se mu rozpoznat pouze 59 % z daných symbolů.

Tab. 4.1: Výsledky detekce svastik – pozitivní snímky

	RAB	LB	DAB	GAB	HOG	LBP
Celkem symbolů	117	117	117	117	117	117
Pravdivě pozitivních	69	84	82	78	71	78
Falešně negativních	48	33	35	39	46	39
Falešně pozitivních	2	16	19	9	20	141
Počet snímků s "FP"	2	10	10	8	13	47
Úspěšnost detekcí	58,97 %	71,79 %	70,09 %	66,67 %	60,68 %	66,67 %
Falešně pozitivní	96,67 %	83,33 %	83,33 %	86,67 %	88,89 %	21,67 %
Celková úspěšnost	77,82 %	77,56 %	76,71 %	76,67 %	74,79 %	44,17 %

Nejlepší poměr mezi pravdivě a falešně pozitivně detekovanými symboly má detektor vytvořený pomocí metody LogitBoost. Tento klasifikátor byl implementován do výsledného programu. Nejvyšší procento úspěšnosti detekovaných symbolů je však spojeno s dobou pro vytvoření klasifikátoru, která se pohybovala na hranici jednoho dne. Tento klasifikátor byl časově nejnáročnější. Středně úspěšné byly klasifikátory používající algoritmy Discrete a Gentle AdaBoost, zde se tvorba klasifikátoru pohybovala v horizontu tří hodin.

Druhým nejslabším z pohledu detekční úspěšnosti byl klasifikátor tvořený s využitím histogramu orientovaných gradientů, projevilo se zde také zvýšené množství falešně pozitivních detekcí. Nejhorší výsledky vykázal klasifikátor používající příznaky Local Binary Pattern. Na pouhých 47 snímcích se vyskytla falešně pozitivní detekce celkem $141 \times$. Zároveň se však klasifikátoru podařilo úspěšně lokalizovat 78 symbolů, čímž se řadil k lepšímu průměru. V porovnání s ostatními klasifikátory, u kterých se doba potřebná k vytvoření pohybovala v řádu hodin, tento byl hotov do několika minut. Zvýšení úspěšnosti detekce by bylo možné dosáhnout použitím více stupňů klasifikace, tedy použitím více slabých klasifikátorů.

Následující tabulka obsahuje výsledky testu s negativními snímky. Výsledná úspěšnost N je definována obměnou vztahu (4.1):

$$N = \left(\frac{N_{PN}}{N_n} \right) * 100 [\%], \quad (4.2)$$

kde

N – úspěšnost pravdivě negativních (PN) detekcí,

N_{PN} – počet snímků s PN detekcemi,

N_n – počet všech negativních snímků.

Výsledky z testu pozitivních snímků se potvrdily i při testu s negativními snímky. Nejmenší chybovosti dosáhl klasifikátor s Haarovými příznaky vytvořený pomocí al-

Tab. 4.2: Výsledky detekce svastik – negativní snímky

	RAB	LB	DAB	GAB	HOG	LBP
Celkem snímků	60	60	60	60	60	60
Pravdivě negativních	59	54	45	56	52	16
Falešně pozitivních	1	9	16	7	8	101
Úspěšnost	98,33 %	90,00 %	75,00 %	93,33 %	86,67 %	26,67 %

goritmu RAB. Z testovací sady došlo pouze k jedné falešně pozitivní detekci. Nejvyšší chybovost vykázal opět klasifikátor s LBP příznaky, který vyhodnotil na 44 snímcích 101 symbolů.

4.5.2 Detektor znaků SS

K vytvoření klasifikátoru detekujícího znaky SS bylo použito 116 pozitivních a 300 negativních vzorků. Příklad pozitivních vzorků je na obrázku 4.9, negativní vzorky byly použity stejné jako v minulém případě. Vytvořena byla dvojice klasifikátorů, sestávající z 20 stupňů, pracující s Haarovými příznaky. Z důvodu nejvyšších úspěšností při předchozím testování byly použity trénovací algoritmy Real AdaBoost a LogitBoost. Testovací sada čítala celkem 120 snímků, 60 negativních a stejný počet pozitivních. Na pozitivních snímcích se nacházelo 70 symbolů.



Obr. 4.9: Trénovací sada – znaky SS

V tabulce 4.3 jsou uvedeny výsledky pro test detektorů na pozitivních vstupních obrázcích. Ze sedmdesáti přítomných symbolů jich bylo správně klasifikováno pouze 38 u RAB detektoru, pro detektor LogitBoost jich pak bylo pouze 50. S vyšším počtem správně detekovaných symbolů rostla také četnost výskytu falešně pozitivních detekcí. Z důvodu menšího výskytu falešně pozitivních detekcí byl do výsledného programu implementován klasifikátor RAB.

Tabulka 4.4 obsahuje výsledky testu negativních snímků. Během testu docházelo k velkému počtu falešně pozitivních detekcí. Byla použita totožná testovací sada jako pro detektor hákových křížů. Pro LB detektor bylo správně klasifikováno pouze 48,33 % z testovaných snímků. Na 31 snímcích došlo v 75 případech k falešně pozitivní detekci. Druhý klasifikátor dosahoval lepších výsledků, 63,33 % obrázků bylo vyhodnoceno správně.

Tab. 4.3: Výsledky detekce znaků SS – pozitivní snímky

	RAB	LB
Celkem symbolů	70	70
Pravdivě pozitivních	38	50
Falešně negativních	32	20
Falešně pozitivních	28	53
Počet snímků s "FP"	17	33
Úspěšnost detekcí	54,29 %	71,43 %
Falešně pozitivní	71,67 %	45,00 %
Celková úspěšnost	62,98 %	58,21 %

Tab. 4.4: Výsledky detekce znaků SS – negativní snímky

	RAB	LB
Celkem snímků	60	60
Pravdivě negativních	38	29
Falešně pozitivních	49	75
Úspěšnost	63,33 %	48,33 %

Klasifikátor znaků SS je značně nevyhovující. Nízkou úspěšnost klasifikátorů lze přisuzovat nevhodně zvoleným trénovacím datům. Trénovací sada obsahovala jak bílé symboly na černém pozadí, tak obrácenou kombinaci. Vhodnější by patrně bylo vytvořit trénovací sady zvlášť pro jednotlivé barevné varianty, vytvořit příslušné klasifikátory, a poté zkombinovat jejich funkčnost.

4.5.3 Detektor triskelionů

Na vytvoření klasifikátoru detekujícího tyto symboly bylo použito 300 negativních a 102 pozitivních vzorků, příklad pozitivních snímků je na obrázku 4.10. Vytvořeny byly klasifikátory pomocí algoritmů Real AdaBoost a LogitBoost, složené z 20 a také 24 stupňů.



Obr. 4.10: Trénovací sada – triskelion

Testovací sada opět čítala celkem 120 snímků, na 60 pozitivních obrázcích se nacházelo celkem 67 požadovaných symbolů. Zbylých 60 snímků tvořily negativní snímky, které byly shodné pro všechny provedené testy. Výsledky jsou uvedeny v tabulkách 4.5 pro pozitivní a 4.6 pro negativní snímky.

Tab. 4.5: Výsledky detekce triskelionů – pozitivní snímky

	LB	RAB	RAB – 24	LB – 24
Celkem symbolů	67	67	67	67
Pravdivě pozitivních	49	42	31	42
Falešně negativních	18	25	36	25
Falešně pozitivních	33	1	0	0
Počet snímků s "FP"	17	1	0	0
Úspěšnost detekcí	73,13 %	62,69 %	46,27 %	62,69 %
Falešně pozitivní	71,67 %	98,33 %	100,00 %	100,00 %
Celková úspěšnost	72,40 %	80,51 %	73,13 %	81,34 %

Nejlepších výsledků v oblasti pravdivě pozitivních detekcí dosahoval 20 stupňový detektor tvořený LogitBoostem, v 73,13 % došlo ke správné klasifikaci, tento klasifikátor byl použit pro výsledný program. Na testovacích datech se však často vyskytovaly také falešně pozitivní detekce. Na sedmnácti snímcích bylo 33 chybných klasifikací. Detektor trénovaný metodou RAB dosáhl nižší úspěšnosti detekce, došlo ale pouze k jednomu výskytu falešně pozitivní detekce. Zvýšením počtu stupňů klasifikátoru se mi sice podařilo eliminovat výskyt falešně pozitivních detekcí u obou použitých metod, došlo však k výraznému poklesu úspěšných detekcí.

Následující tabulka obsahuje výsledky testu negativních snímků. Nulovou chybovost vykazovaly klasifikátory složené z 24 stupňů kaskády. Méně stupňové klasifikátory tolik úspěšné již nebyly. Klasifikátor LB označil chybné místo v obraze ve 24 případech na čtvrtině testovací sady. Algoritmus RAB je v tomto případě vhodnější, neboť došlo pouze ke třem falešně pozitivním detekcím.

Tab. 4.6: Výsledky detekce triskelionů – negativní snímky

	LB	RAB	RAB – 24	LB – 24
Celkem snímků	60	60	60	60
Pravdivě negativních	45	57	60	60
Falešně pozitivních	24	3	0	0
Úspěšnost	75,00 %	95,00 %	100,00 %	100,00 %

4.5.4 Celkové zhodnocení

U všech testovaných symbolů dosahoval nejvíce úspěšných detekcí klasifikátor s Haarovými příznaky tvořený metodou LogitBoost, bohužel za cenu vysokého výskytu falešně pozitivních detekcí. Klasifikátory trénované algoritmem Real AdaBoost obecně dosahovaly nižších pravdivě pozitivních detekcí. Vítaným jevem byl u těchto klasifikátorů výskyt nízkého počtu falešně pozitivních detekcí.

Nejúspěšnější byly klasifikátory detekující svastiky a triskeliony. Detekční úspěšnost se zde pohybovala kolem 75 %. Úspěšnost se odvíjela od trénovací sady. Pro detekci hákových křížů bylo nejněsnější nasbírat desítky trénovacích a testovacích dat. Obrázků dostupných na internetu je velké množství, v dostatečné kvalitě a různých variantách, poměrně snadno jsou dostupné různé vlajky, plakáty, tetování či kresby. Podobná situace nastala i u klasifikace triskelionů. Trénovací sada sice již ne sestávala z tolika různých variant, neonacistické organizace však tento symbol hojně využívají i v současnosti a po celém světě, takže trénovacích a testovacích vzorků bylo také dostatečné množství v dobré kvalitě.

Výrazně nižší úspěšnosti u klasifikátorů pro znaky SS jsou také dány trénovací sadou. Jak již bylo zmíněno v podkapitole 4.5.2, trénovací sada měla být rozšířena a rozdělena na symboly bílé s černým pozadím a černé s bílým pozadím. Tyto symboly také nejsou již tak hojně využívány jako předchozí zmiňované, nebylo jich proto dostatečné množství. Řada fotografií či snímků s těmito symboly byla nižší kvality, jednalo se např. o fotografie z dob 2. světové války.

Pracoval jsem také na vytvoření klasifikátoru detekujícího znaky rasistické organizace Ku Klux Klan (KKK), zde jsem se však setkal s neúspěchem již při sestavování trénovací sady. Řada fotografií, které obsahovaly tento znak, byla silně zašuměných, a těchto symbolů bylo dostupné malé množství. Symbol je navíc poměrně komplikovaný a použití Haarových příznaků nebylo nejlepším řešením. Výslednému klasifikátoru se nepodařilo správně detekovat téměř nic, naopak docházelo k velmi vysokému výskytu falešně pozitivních detekcí. Z těchto důvodů tento symbol nebyl implementován do výsledného detektoru a ani na něm nebyly provedeny testy úspěšnosti.

5 ZÁVĚR

Cílem bakalářské práce bylo vytvořit detektor závadných (rasistických) symbolů. Zhotovil jsem automatický detektor, který dokáže v obraze detekovat trojici symbolů, hákové kříže neboli svastiky, znaky SS a triskeliony, které v současnosti používá neonacistická organizace Blood & Honour.

Tento detektor pracuje s veřejně dostupnou knihovnou OpenCV a využívá jen malý zlomek z jejích užitečných funkcí. Detektor hákových křížů byl natrénován použitím Haarových příznaků, Local Binary Pattern (LBP) příznaků a pomocí histogramu orientovaných gradientů. Následně byla u jednotlivých klasifikátorů otestována úspěšnost. Dvojice algoritmů, které vytvořily nejúspěšnější klasifikátory, byla poté použita pro tvorbu klasifikátorů pro zbylé symboly.

Nejúspěšnější detektor hákových křížů dokázal správně rozpoznat 71,79 % z testovaných dat. S přihlédnutím k falešně pozitivním detekcím je výsledná hodnota úspěšnosti rovna 77,56 %.

Během testu na detekci znaků SS se vyskytovalo nejvíce chybných detekcí a nízký počet pravdivě pozitivních detekcí. Z hlediska počtu detekcí dosahoval účinnější klasifikátor úspěšnosti 71,43 %, bohužel kvůli vysokému počtu falešně pozitivních detekcí klesla výsledná úspěšnost na 58,21 %.

Detektor triskelionů v 73,13 % případů správně detekoval požadovaný symbol, celková úspěšnost se zastavila na 72,40 %. Zvýšení počtu kaskád klasifikátoru mělo za následek výrazné snížení falešně pozitivních detekcí, došlo však také ke snížení pravdivě pozitivních detekcí. Symbol se podařilo detekovat pouze v 62,69 % případů.

Pro účely testování a použití byla vytvořena konzolová a grafická aplikace. Konzolová aplikace je spustitelná přes příkazový řádek a jejím parametrem je název obrázku nebo cesta k němu. Grafické rozhraní umožňuje přímý výběr souboru ze složky a podporuje také webové odkazy.

Při zpracovávání této bakalářské práce jsem byl nepříjemně překvapen množstvím závadných symbolů, které lze nalézt na internetu. Existuje velmi mnoho internetových obchodů, ve kterých lze poměrně jednoduše nakupovat závadné zboží. Zdokonalením detekce a její implementací do webových prohlížečů by bylo možné výskyt těchto symbolů monitorovat a blokovat jejich zobrazení, případně celý web, například v rámci rodičovské ochrany.

LITERATURA

- [1] RAJMIC, Pavel. *Základy počítačové sazby a grafiky* [online]. Brno: VUT, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012 [cit. 2014-11-13]. 147 s. ISBN 978-80-214-4451-5. Dostupné z: <https://www.vutbr.cz/elearning/mod/resource/view.php?id=276386>
- [2] MCHUGH, Sean. Bit Depth. *CAMBRIDGE in COLOUR: A Learning Community for Photographers* [online]. 2014 [cit. 2014-11-13]. Dostupné z: <http://www.cambridgeincolour.com/tutorials/bit-depth.htm>
- [3] FIŘT, Jaroslav a Radek HOLOTA. *Digitalizace a zpracování obrazu* [online]. Plzeň: Nové technologie — výzkumné centrum, 2002 [cit. 2014-11-13]. 5 s. Dostupné z: <http://home.zcu.cz/~holota5/publ/DigZpr0.pdf>
- [4] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. 2. přeprac. vyd. Praha: ČVUT, 2007. 255 s. ISBN 978-80-01-03110-0.
- [5] HORÁK, Karel, Ilona KALOVÁ, Petr PETYOVSKÝ a Miloslav RICHTER. *Počítačové vidění* [online]. Brno: VUT, Fakulta elektrotechniky a komunikačních technologií, 2008 [cit. 2014-11-15]. 132 s. Dostupné z: http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove_videni_S.pdf
- [6] HORÁK, Karel. *Computer Vision: Zpracování vícerozměrných signálů* [online]. Brno: VUT, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2010 [cit. 2014-12-12]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php
- [7] HLAVÁČ, Václav. Hledání hran. *Center for Machine Perception* [online]. Praha: ČVUT, Fakulta elektrotechnická, 2014 [cit. 2014-12-12]. Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/22EdgeDetectionCz.pdf>
- [8] NIBLACK, Wayne. *An introduction to digital image processing*. [Nachdr.]. Englewood Cliffs. New Jersey: Prentice Hall International, 1988. ISBN 01-348-0600-X.
- [9] GONZALEZ, Rafael C. *Digital Image Processing: Second Edition*. 2nd Ed. Reading: Addison-Wesley Publishing Company, 1987. 503 s. ISBN 02-011-1026-1.
- [10] VIOLA, Paul a Michael JONES. *Rapid Object Detection using a Boosted Cascade of Simple Features* [online]. 2001. [cit. 2014-11-29]. 9 s. Dostupné z: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>

- [11] MAŠEK, Jan. *Detekce objektů v obraze s pomocí Haarových příznaků*: diplomová práce [online]. Brno: VUT, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 72 s. Vedoucí práce byl Ing. Radim Bureget, Ph.D. Dostupné z: http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=51400
- [12] MÄENPÄÄ, Topi. *The local binary pattern approach to texture analysis: extensions and applications* [online]. Oulu: Oulun yliopisto, 2003 [cit. 2014-12-08]. 80 s. ISBN 95-142-7076-2. Dostupné z: <http://herkules.oulu.fi/isbn9514270762/isbn9514270762.pdf>
- [13] LIAO, Shengcai, Xiangxin ZHU, Zhen LEI, Lun ZHANG a Stan Z. LI. *Learning Multi-scale Block Local Binary Patterns for Face Recognition* [online]. 2007 [cit. 2014-12-10]. 10 s. International Conference on Biometrics (ICB). Dostupné z: <http://www.cbsr.ia.ac.cn/users/scliao/papers/Liao-ICB07-MBLBP.pdf>
- [14] WILSON, Philip Ian a John FERNANDEZ, *Facial feature detection using Haar classifiers* [online]. 2006 [cit. 2014-12-10]. 7 s. Consortium for Computing Sciences in Colleges. Dostupné z: <http://nichol.as/papers/Wilson/FacialfeatureddetectionusingHaar.pdf>
- [15] KVÁČA, Jan. *Detekce aut v obrázcích s vysokým rozlišením*: diplomová práce [online]. Praha: ČVUT, Fakulta elektrotechnická, 2012 [cit. 2014-12-11]. 73 s. Vedoucí práce byl Ing. Karel Zimmermann, Ph.D. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/kvacajan_2011dipl.pdf
- [16] HRADIŠ, Michal. Boosting. *Klasifikace a rozpoznávání: Prezentace přednášek* [online]. Brno: VUT, Fakulta informačních technologií, 2013 [cit. 2014-12-12]. Dostupné z: http://www.fit.vutbr.cz/study/courses/IKR/public/stare_prednasky_2012/07_boosting/IKR-Boosting.pdf
- [17] BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly, 2008. 555 s. ISBN 978-0-596-51613-0.
- [18] MAREŠ, Miroslav. *Symboly používané extremisty na území ČR v současnosti: manuál pro Policii ČR*. Praha: Ministerstvo vnitra, 2006. 80 s.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AdaBoost	Adaptive Boosting
BMP	Windows BitMaP
C++	Programovací jazyk
CMY	Cyan, Magenta, Yellow – tyrkysová, purpurová a žlutá
CMYK	Cyan, Magenta, Yellow, Key – tyrkysová, purpurová, žlutá a černá
DAB	Discrete Adaptive Boosting
DCT	Diskrétní Cosinová Transformace
DFT	Diskrétní Fourierova Transformace
GAB	Gentle AdaBoost
GIF	Graphics Interchange Format
HOG	Histogram orientovaných gradientů
HSB	Hue, Saturation, Brightness
JPEG	Joint Photographic Experts Group
LB	LogitBoost
LBP	Local Binary Pattern
LZW	Lempel, Ziv, Welch
mag	Magnituda, velikost vektoru gradientu
MB-LBP	Multi-scale Block Local Binary Pattern
OpenCV	Open source Computer Vision
PDF	Portable Document Format
PNG	Portable Network Graphics
RAB	Real AdaBoost
RGB	Red, Green, Blue – červená, zelená, modrá
RLE	Run Length Encoding – kódování délkou znaku
SVG	Scalable Vector Graphics

SEZNAM PŘÍLOH

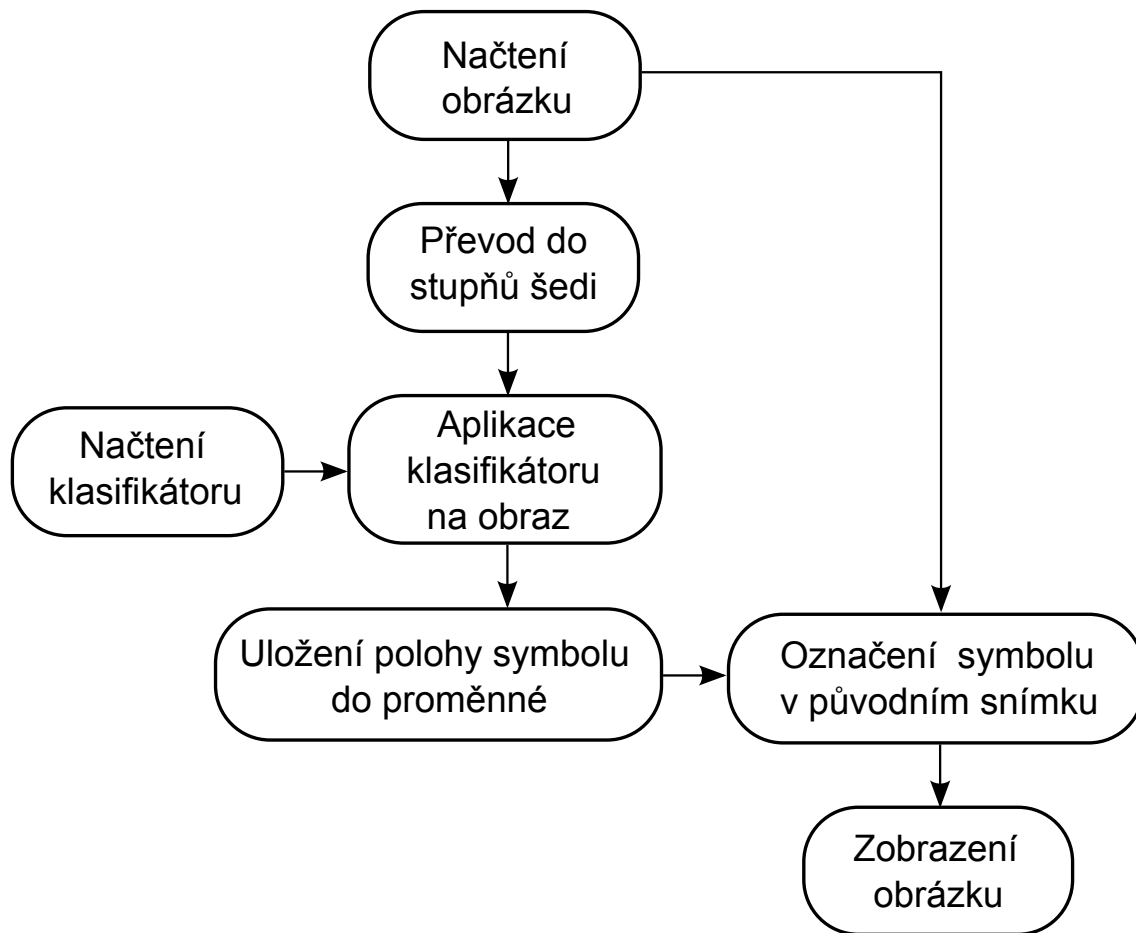
A Slabý klasifikátor	45
B Postup při detekci	46

A SLABÝ KLASIFIKÁTOR

```
<opencv_storage>
  <stage0>
    <maxWeakCount>2</maxWeakCount>
    <stageThreshold>-1.8888509273529053e-001</stageThreshold>
    <weakClassifiers>
      <_>
        <internalNodes>
          0 -1 2905831 4.2263287468813360e-005
        </internalNodes>
        <leafValues>
          -9.0243899822235107e-001 8.0952382087707520e-001
        </leafValues>
      </_>
      <_>
        <internalNodes>
          0 -1 1842669 -1.3723787560593337e-004
        </internalNodes>
        <leafValues>
          7.1355390548706055e-001 -9.2287600040435791e-001
        </leafValues>
      </_>
    </weakClassifiers>
  </stage0>
</opencv_storage>
```

B POSTUP PŘI DETEKCI

Následující vývojový diagram vyobrazuje postup při detekci symbolu, na který byl natrénován klasifikátor.



Obr. B.1: Postup při detekci