

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NEURONOVÉ SÍŤE PRO LOKALIZACI LIDSKÉHO OBLIČEJE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAKUB LIBOSVÁR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NEURONOVÉ SÍŤE PRO LOKALIZACI LIDSKÉHO OBLIČEJE

APPLICATION OF NEURAL NETWORKS FOR HUMAN FACE LOCALIZATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAKUB LIBOSVÁR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAL ŠPANĚL

BRNO 2009

Abstrakt

Bakalářská práce pojednává o problematice detekce a lokalizace lidského obličeje v obraze. Jsou zde popsány možnosti současných metod a umělých neuronových sítí. Práce vychází z výzkumu H. Rowleyho. Prakticky je práce zaměřena na implementaci programu, který je schopen na základě trénovacích a testovacích obrázkových sad natrénovat umělou neuronovou síť pro rozpoznávání lidských obličejů.

Klíčová slova

klasifikátory, umělé neuronové sítě, detekce obličeje, cross-validation, bootstrap, ekvalizace histogramu, rozptyl

Abstract

This bachelor thesis deals with detection and localization of human upright faces in images. At first, there are considered current methods. Then the face detection concept is presented. The thesis is focused on practical implementation of artificial neural network-based face detector designed by H. Rowley. Finally, training process and results of detector are discussed in more detail.

Keywords

classifiers, artificial neural networks, face detection, cross-validation, bootstrap, histogram equalization, variance

Citace

Jakub Libosvár: Neuronové sítě pro lokalizaci lidského obličeje, bakalářská práce, Brno, FIT VUT v Brně, 2009

Neuronové sítě pro lokalizaci lidského obličeje

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla.

Veškeré literární prameny jsou uvedeny na konci dokumentu.

.....
Jakub Libosvár
17. května 2009

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce Ing. Michalu Španělovi za odbornou pomoc, cenné rady a připomínky, které velmi přispěly ke vzniku této práce.

© Jakub Libosvár, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	5
2 Klasifikátory	6
2.1 Neuronové sítě	6
2.2 AdaBoost	6
2.3 Support vector machine	7
3 Neuronové sítě	9
3.1 Biologický neuron	10
3.2 Umělý neuron	11
3.3 Skoková přenosová funkce	12
3.4 Sigmoidální přenosová funkce	12
3.5 Přenosová funkce radiální báze	13
3.6 Přenosová funkce hyperbolické tangenty	13
4 Metody pro detekci obličeje v obraze	14
4.1 Znalostní metody	14
4.2 Invariantní rysy	14
4.3 Srovnávání šablon	15
4.4 Metody založené na zjevu	16
5 Detekce obličeje pomocí neuronových sítí	17
6 Návrh detektoru	20
6.1 Výpočet rozptylu intenzit	23
6.2 Ekvalizace histogramu	23
7 Implementace	25
7.1 OpenCV	25
7.2 Annie	25
7.3 Boost	26
7.4 Třídy	26
8 Testování	28
8.1 Trénovací sady	28
8.2 Vyhodnocení	30
9 Závěr	34

A Obsah CD	37
B Manuál	39
C Konfigurační soubor	40

Seznam obrázků

2.1	Rozdělení do 2 množin lineárním klasifikátorem (Zdroj: [6])	6
2.2	Slabé klasifikátory a obecný výpočet silného klasifikátoru (Zdroj: [3])	7
2.3	Rozdělení pomocí metody SVM (Zdroj: [11])	8
3.1	Příklad neuronové sítě (Zdroj: [10])	10
3.2	Struktura biologického neuronu (Zdroj: [12])	10
3.3	Skoková přenosová funkce pro $\Theta = 0$	12
3.4	Sigmoidální přenosová funkce pro $k = 1$	12
3.5	Přenosová funkce radiální báze pro $k = 1$	13
3.6	Přenosová funkce hyperbolické tangenty	13
4.1	Ukázka detekce metodou invariantní rysy (Zdroj: [2])	15
4.2	Šablona pro detekci (Zdroj: [1])	15
4.3	Proces detekce metodou srovnávání šablon (Zdroj: [1])	15
5.1	Algoritmus detekce podle H. Rowleyho (Zdroj: [5])	18
5.2	Metoda překrývání podle H. Rowleyho (Zdroj: [5])	18
5.3	Rozhodování arbitrážní sítí (Zdroj: [5])	19
6.1	Algoritmus lokalizace obličeje	22
6.2	Rozptyly hodnot zleva: 205.005, 483.675, 637.13	23
6.3	Obličej bez ekvalizovaného histogramu	24
6.4	Obličej s ekvalizovaným histogramem	24
7.1	Diagram tříd	27
8.1	Ukázka obličejů testovací sady CBCL	28
8.2	Ukázka obrázku ze Schneidermanovy sady a vstupy pro trénování sítě	29
8.3	Průběh trénování neuronové sítě	30
8.4	Shluk detekcí okolo obličeje	31
8.5	Chybná detekce	31
8.6	Odstranění nepřekrývajících se detekcí s parametry <code>threshold(4,4)</code>	31
8.7	Příklad špatné detekce	32
8.8	Po odstranění nepřekrývajících se detekcí	32
8.9	Příklady detekcí obličejů	33

Seznam tabulek

8.1	Chybovost sítě s ekvalizací histogramu	29
8.2	Chybovost sítě bez ekvalizace histogramu	30

Kapitola 1

Úvod

Detekce obličeje je dnes hojně využívána v oblasti moderní techniky. Většina moderních fotoaparátů a mobilních telefonů je již takovým vybavením opatřena. Účelem detekce je najít takový bod ostření, aby nebyl obličej na fotografii rozmazaný. Na již detekovaný obličej lze aplikovat i detekci úsměvu. Je možné si dát fotoaparát před sebe a ten vyčkává, dokud se člověk neusměje. Při dostatečném úsměvu pak fotoaparát situaci zvěční. Dále je možné detekce uplatnit v bezpečnostních systémech. Například inteligentní budovy si takto mohou hlídat návštěvníky a mít tak přesný přehled o čase a osobě, která do budovy vstoupí nebo z ní odejde. Takové detektory musí být ovšem na profesionální úrovni a je potřeba, aby dokázaly pracovat *real-time*¹. Aby byl systém použitelný ve videosekvencích, měl by detektor dokázat lokalizovat obličeje alespoň jedenkrát za sekundu a s velkou přesností. Tato bakalářská práce se ale zaměřuje na detekci obličeje v obraze, což výrazně usnadní práci s efektivitou programu. K detekci lidského obličeje využívá umělé neuronové sítě.

Detekce obličeje znamená, že se na předloženém obrázku určitým způsobem lokalizuje místo, kde se obličej nachází. Lokalizace může být dána například bodem a rozměry obrazce, který ohraničuje obličej, nebo se obličej v obraze přímo nějak vyznačí.

Práce je rozdělena na kapitoly a sekce. Nejprve je rozebrána teorie neuronových sítí a detekce obličejů obecně. Kapitola 2 dává přehled nejdůležitějších a nejpoužívanějších klasifikátorů, mezi které patří také neuronové sítě. Těm je věnována celá kapitola 3, kde se připodobňuje umělý neuron ke svému vzoru, biologickému neuronu. Kapitoly 4 a 5 obsahují teorii k detekci obličejů v obraze.

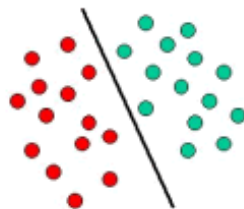
Prakticky se práce zabývá návrhem detektoru lidských obličejů v kapitole 6, jeho implementací v kapitole 7 a testováním funkčního detektoru v kapitole 8. Samotnou implementaci je možné nalézt na přiloženém CD.

¹Aby systém pracoval v reálném čase.

Kapitola 2

Klasifikátory

Klasifikátor¹ má za úkol roztrždit množinu dle určitých podobných vlastností do specifických tříd. Tato rozdělení jsou velmi důležitá v oblasti strojového učení. Klasifikátorů existuje celá řada, zde jsou uvedeny pouze některé, aby byly zřejmé principy a rozdíly mezi klasifikátory.



Obrázek 2.1: Rozdělení do 2 množin lineárním klasifikátorem (Zdroj: [6])

2.1 Neuronové sítě

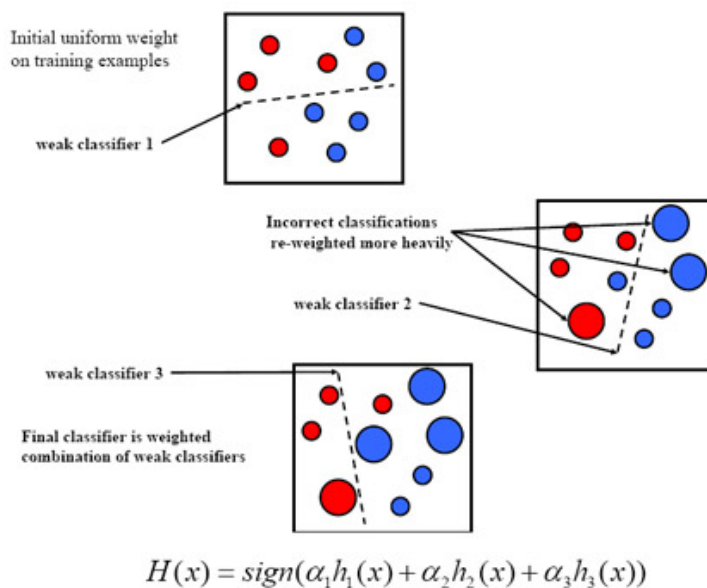
Principiálně se jedná o trénování klasifikátoru na určitých datových sadách. Postupem času se neuronová síť naučí třídit objekty podle toho, co se naučila. Tomuto tématu je věnována celá kapitola 3.

2.2 AdaBoost

Název je odvozen z anglického slova adaptive boosting neboli „přizpůsobivá podpora“. Jedná se o metodu trénování, kdy na počátku je sada slabých klasifikátorů. Procesem učení je snaha určit lineární kombinaci těchto klasifikátorů tak, aby byla chyba výsledného klasifikátoru co nejmenší. Prakticky se jedná o ladění dané váhy slabého klasifikátoru tím, že se jeho váha snižuje, resp. zvyšuje v závislosti na jeho výsledku. V cíli tak vznikne silný

¹angl. *classifier*, *class* = třída

klasifikátor. Tento způsob se hojně využívá v dolování dat. Tato sekce je převzata z [8]. Na obrázku 2.2 je ukázán výpočet výsledného klasifikátoru ze 3 slabých klasifikátorů.



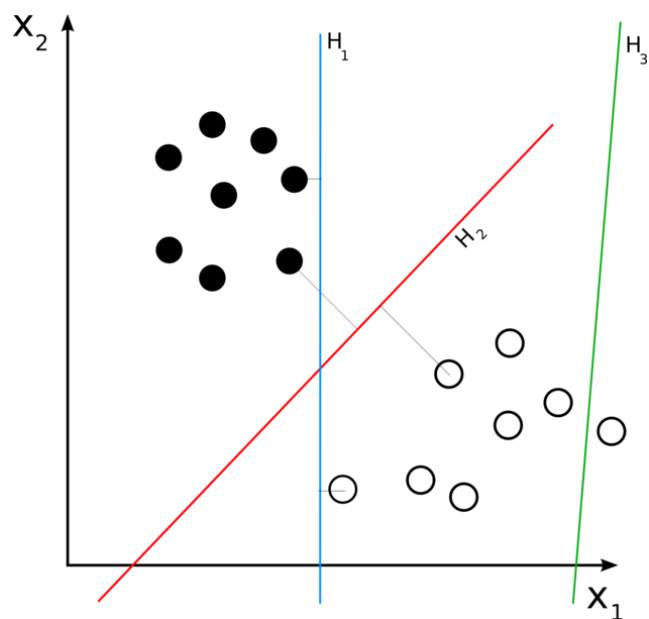
Obrázek 2.2: Slabé klasifikátory a obecný výpočet silného klasifikátoru (Zdroj: [3])

2.3 Support vector machine

Cílem této metody je nalézt nadrovinu optimálně rozdělující trénovací data v prostoru příznaků. Takové rozdělení je realizované lineární funkcí, tím se rozdělí prostor na 2 množiny. Metoda je tedy binární.

Definice 2.3.1. *Optimalita rozdělení* je definována jako maximum minima vzdálenosti mezi body rozdělovaných dat.

Obrázek 2.3 znázorňuje 2 množiny, na nichž se hledá lineární klasifikátor. Z obrázku je zřejmé, že klasifikátor $H1$ množiny dělí velmi špatně. Klasifikátor $H2$ už množiny dělí správně, ale stále nesplňují podmínku optimality rozdělení z definice 2.3.1. Klasifikátor $H3$ už podmínku splňuje.



Obrázek 2.3: Rozdělení pomocí metody SVM (Zdroj: [11])

„Důležitou součástí techniky Support vector machines je jádrová transformace prostoru příznaků dat do prostoru transformovaných příznaků typicky vyšší dimenze. Tato jádrová transformace umožňuje převést původně lineárně neseparovatelnou úlohu na úlohu lineárně separovatelnou, na kterou lze dále aplikovat optimalizační algoritmus pro nalezení rozdělující nadroviny.“

Obsah a citace převzata z [11].

Kapitola 3

Neuronové sítě

Umělé neuronové sítě, jak už název napovídá, jsou obrazem biologických struktur v živých organismech. Konkrétně se jedná o biologické neuronové sítě patřící do nervové soustavy, pomocí které se v živých organismech přenášejí informace. Tak jako systém biologických neuronů tvoří biologickou neuronovou síť, tak systém umělých neuronů vytváří umělou neuronovou síť. Podobně jako se člověk v průběhu života učí, má umělá neuronová síť také schopnost se učit.

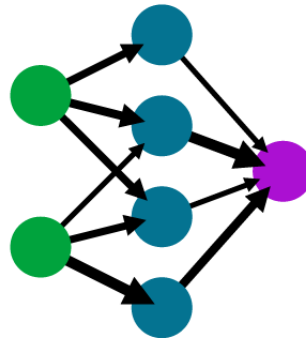
Různým uspořádáním neuronů, propojením a výběrem jejich aktivačních funkcí lze sestavit neuronovou síť pro daný problém. Umělou neuronovou síť lze rozdělit do několika vrstev:

- Vstupní vrstva (*input layer*)
- Skrytá vrstva (*hidden layer*)
- Výstupní vrstva (*output layer*)

Zatímco vstupní a výstupní vrstva je pouze jedna, skrytá vrstva se v síti může opakovat. Počet neuronů v jednotlivých vrstvách je libovolný, je možné tedy vytvořit síť s několika vstupy a několika výstupy. Vstupy i výstupy lze uvažovat jako vektory, počet jejich dimenzí může být odlišný.

Aby byly neuronové sítě použitelné, je třeba je nejprve na určitém vzorku dat natrénovat. Trénováním se rozumí proces, kdy je na vstup neuronové sítě předložen vzor dat a na výstup očekávaná hodnota. Trénovacích algoritmů existuje spousta, nejznámějším je *back-propagation algorithm*. Po jedné iteraci trénování putuje chyba trénování zpět sítě a zpětně nastavuje váhy, aby byla chyba co nejmenší. Při učení tímto způsobem se porovnají očekávaná data s výsledkem neuronové sítě a zpětně se změny hodnoty vah, aby byl výsledek co nejvíce podobný vzoru.

A simple neural network
input layer hidden layer output layer

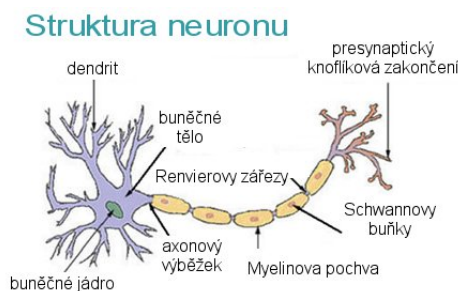


Obrázek 3.1: Příklad neuronové sítě (Zdroj: [10])

3.1 Biologický neuron

Neuron je základní stavební jednotkou nervové tkáně. Jedná se o buňku, která přenáší a také zpracovává specifické signály. Umožňuje tedy v závislosti na příchozím signálu určitým způsobem reagovat. Když se například člověk dotkne rukou horké plotny, buňky na povrchu ruky zaznamenají vysokou teplotu a pomocí nervové soustavy¹ zasílají signál do míchy. Zde je tento signál vyhodnocen jako hrozící nebezpečí pro tělo a stejným systémem je zaslán signál svalům v ruce, aby ji odtáhly pryč. Jedná se o nepodmíněný reflex.

Na obrázku 3.2 je vidět stavba biologického neuronu. Velmi důležitou jednotkou jsou dendrity, kterými neuron přijímá ze synapsí cizího neuronu signál. Jádro neuronu signál zpracuje a ten potom putuje skrze axon do synapsí, aby se mohl předat dalším neuronům.



Obrázek 3.2: Struktura biologického neuronu (Zdroj: [12])

¹Nervová soustava je složena z neuronových sítí.

3.2 Umělý neuron

Podle vzoru dendritů biologického neuronu má umělý neuron vstupy, lze je také nazývat vstupním vektorem \vec{X} . Každému vstupu je přiřazena váha, která se mění v procesu trénování sítě. Tyto váhy lze matematicky označit jako vektor vah \vec{W} . Je zřejmé, že vstupní vektor \vec{X} a vektor vah \vec{W} musí mít stejný rozměr. Jejich skalárním součinem $\vec{X} \cdot \vec{W}$ tak vznikne argument aktivační funkce $f(x)$. Tato funkce² představuje zpracování informace podobně, jako v biologickém neuronu zpracovává signál jádro. V některých případech se v umělých neuronech ještě využívá prahové aktivační hodnoty Θ , která říká, zda je neuron aktivní nebo ne. Veškeré přenosové funkce $f(x)$ musí být definovány na celém definičním oboru $D(f(x)) = \mathbb{R}$. Aktivačním funkcím jsou věnovány sekce 3.3, 3.4, 3.5 a 3.6. Výsledek Y aktivační funkce je zároveň výstupem neuronu, představuje tedy axon. Synapse axonu demonstrují propojení umělého neuronu s dalšími neurony. Výstup neuronu existuje pouze jeden, může ovšem vést do více umělých neuronů (analogie se synaptickými zakončeními axonu).

Tento proces lze jednoduše matematicky zapsat rovnicí

$$Y = f\left(\sum_{i=1}^N (x_i w_i) - \Theta\right) \quad (3.1)$$

nebo uvažujeme-li vektory

$$Y = f(\vec{X} \cdot \vec{W} - \Theta) \quad (3.2)$$

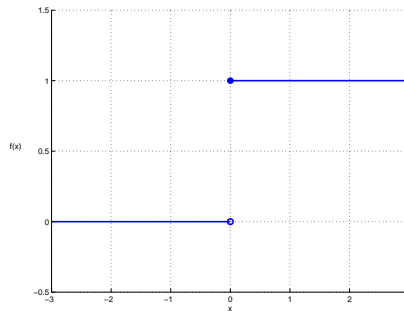
Typy přenosových funkcí jsou převzaty z [13].

²V literatuře také označovaná jako přenosová funkce, jedná se o synonymum.

3.3 Skoková přenosová funkce

U této funkce je potřeba definovat aktivační hodnotu Θ . Ta udává, jestli je výstup neuronu 1 nebo 0. Obor hodnot nabývá pouze binárních hodnot 1 a 0. Funkce nemá v bodě $f(\Theta)$ derivaci. Matematické vyjádření funkce je

$$f(x) = \begin{cases} 1 & \text{pro } x \geq \Theta \\ 0 & \text{pro } x < \Theta \end{cases} \quad (3.3)$$

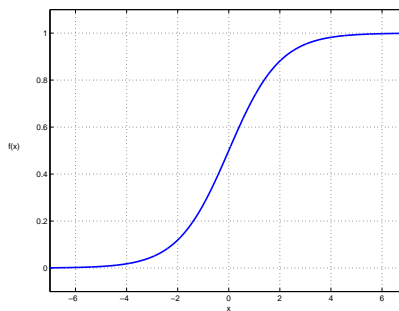


Obrázek 3.3: Skoková přenosová funkce pro $\Theta = 0$

3.4 Sigmoidální přenosová funkce

Na rozdíl od předchozí přenosové funkce má sigmoidální přenosová funkce derivaci v každém bodě. Obor hodnot spadá do intervalu $(0; 1)$ a funkce má 2 limity: $\lim_{x \rightarrow -\infty} f(x) = 0$ a $\lim_{x \rightarrow \infty} f(x) = 1$. Její předpis je

$$f(x) = \frac{1}{1 + e^{-kx}} \quad (3.4)$$



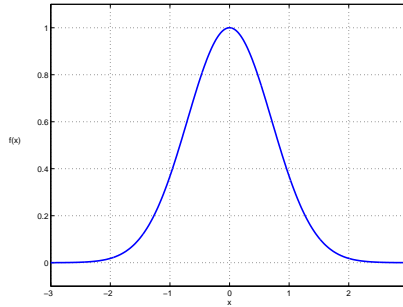
Obrázek 3.4: Sigmoidální přenosová funkce pro $k = 1$

3.5 Přenosová funkce radiální báze

Tato funkce má velmi podobný průběh jako Gaussova křivka. Jedná se o funkci sudou s oborem hodnot $H(f(x)) = (0; 1)$. Extrém v podobě globálního maxima se nachází v bodě $f(0) = 1$, funkce má 2 limity: $\lim_{x \rightarrow -\infty} f(x) = 0$ a $\lim_{x \rightarrow \infty} f(x) = 0$

Matematický zápis této funkce je:

$$f(x) = e^{-kx^2} \quad (3.5)$$

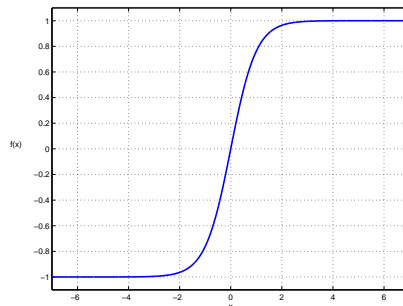


Obrázek 3.5: Přenosová funkce radiální báze pro $k = 1$

3.6 Přenosová funkce hyperbolické tangenty

Funkce je lichá, obor hodnot náleží intervalu $(-1; 1)$. Funkce má 2 limity $\lim_{x \rightarrow -\infty} f(x) = -1$, $\lim_{x \rightarrow \infty} f(x) = 1$. Důležité je okolí bodu 0. Průběh 1. derivace se okolo bodu 1,5 rapidně snižuje a okolo bodu 2 je téměř nulová. Lze tedy konstatovat, že argumenty této funkce by nejčastěji měly spadat do intervalu $\langle -2; 2 \rangle$. Předpis této funkce je:

$$f(x) = \tanh(x) \quad (3.6)$$



Obrázek 3.6: Přenosová funkce hyperbolické tangenty

Kapitola 4

Metody pro detekci obličeje v obraze

Detekce obličejů pomocí neuronových sítí není jedinou možností jak hledat obličeje v obraze. V následujících sekcích jsou stručně popsány některé metody detekující obličeje. Obsah kapitoly byl čerpán z [7].

4.1 Znalostní metody

Už z názvu vyplývá, že se jedná o metodu, kdy musíme o detekovaném objektu mít předem nějakou *znalost*. Metodu lze obecně aplikovat na detekci různých objektů, nejen obličejů. Metoda funguje tak, že popíšeme určité obličejové rysy a na základě těchto pravidel se v obraze vyhledávají objekty, které těmto pravidlům odpovídají. Výsledky této metody nejsou příliš přesné.

4.2 Invariantní rysy

Při detekci touto metodou je potřeba, aby obličeje byly dobře nasvětlené a nebyly nijak pootočené. Metoda detekuje hlavní rysy obličeje, jako jsou nos, oči, ústa, obličejová textura nebo barva obličeje. Implementace touto metodou dosahuje dobrých výsledků a je rychlá. Ukázkou této metody zachycuje obrázek 4.1.



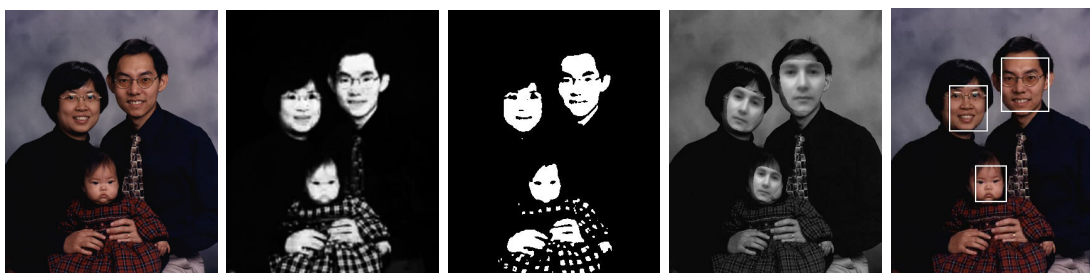
Obrázek 4.1: Ukázka detekce metodou invariantní rysy (Zdroj: [2])

4.3 Srovnávání šablon

Detekce probíhá na základě korelace s přednastavenými šablonami. Šablony obsahují buď přímo celé obličej nebo pouze jejich části. Vytvořit takové šablony je časově náročné a velice pracné, protože se většinou vytvářejí ručně. To je velkou nevýhodou této metody. Proces detekce znázorňuje obrázek 4.3, šablonu obrázek 4.2.



Obrázek 4.2: Šablona pro detekci (Zdroj: [1])



Obrázek 4.3: Proces detekce metodou srovnávání šablon (Zdroj: [1])

4.4 Metody založené na zjevu

Tyto metody vyžadují určitá vzorová data a program se na jejich základě učí taková data klasifikovat. Sběr těchto vzorových dat je jedinou nevýhodou metody. Na základě vzorového modelu se potom detekuje obličej v jednotlivých částech obrazu. Mezi takové metody patří i detekce pomocí neuronových sítí. Je zde dosahováno dobrých výsledků.

Kapitola 5

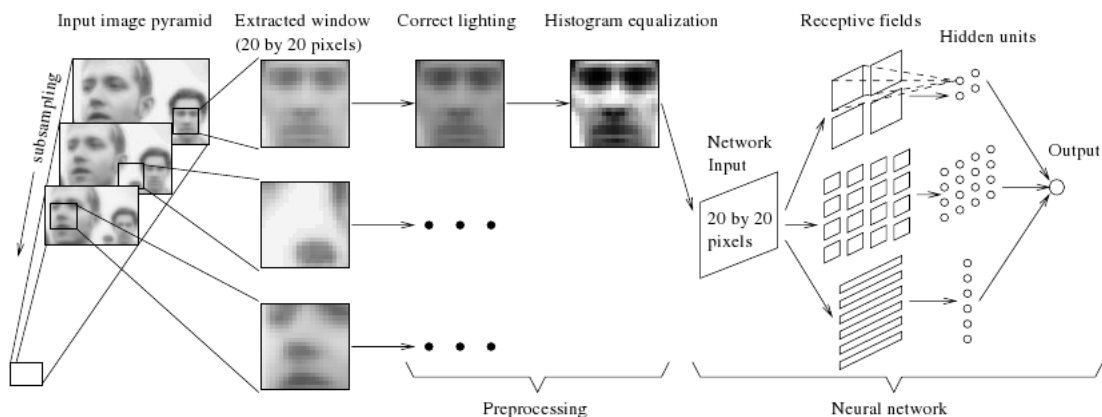
Detekce obličeje pomocí neuronových sítí

Tato kapitola seznamuje s architekturou umělé neuronové sítě publikovanou v roce 1996 H. Rowleym, v roce 1998 potom rozšířena v [5]. Cílem je funkční detektor obličeje, který na základě vah mezi jednotlivými neurony rozhodne, zda se jedná o obličej či nikoliv. Detekce je úspěšná pouze na obličeje nijak nenatočené a vyobrazené pouze zepředu. Snahou je detekovat obličeje v co nejmenším rozlišení. Velké obličeje lze zmenšit na požadovanou velikost, ovšem dosáhnout zvětšení s nízkým zkreslením menších obličejů lze stěží nebo vůbec. Uvažováním obrazů v šedém odstínu se zásadně zmenší množina možných obrazců.

H. Rowley se ve svém návrhu zaměřil na malé obličeje o rozměru 20x20 pixelů. Vstupem neuronové sítě budou tedy intenzity barev jednotlivých pixelů. Z barevného obrázku s barevnou hloubkou 24 bitů lze vypočítat intenzitu barvy pomocí vzorce

$$G = 0,299R + 0,587G + 0,114B, \quad (5.1)$$

kde R , G , B jsou jednotlivé barevné složky. V pravé části obrázku 5.1 je vidět architektura navržené sítě. Ve vstupní vrstvě neurony reprezentují jednotlivé pixely obrázku, celkem tedy je 400 neuronů ve vstupní vrstvě. V návrhu se vyskytuje pouze jediná skrytá vrstva, kterou lze rozdělit na 3 části. Každá část zpracovává určitý segment vstupního obrazu a svůj výsledek zasílá do vrstvy výstupní. Jednotlivé části skryté vrstvy se zaměřují na specifické elementy v obličeji, jako jsou oči, ústa, nos. Ve skryté vrstvě se nachází celkem 27 neuronů. Vrstva výstupní sestává pouze z jednoho neuronu, který na svém výstupu očekává výsledky jednotlivých částí skryté vrstvy. Tento neuron produkuje výstup, podle kterého lze řídit rozhodování o existenci obličeje.



Obrázek 5.1: Algoritmus detekce podle H. Rowleyho (Zdroj: [5])

Po projití celého obrazu a získání pozic detekovaných obličejů následuje fáze, která se snaží odstranit co nejvíce chybných detekcí. H. Rowley experimentoval s různými metodami a jejich kombinacemi.

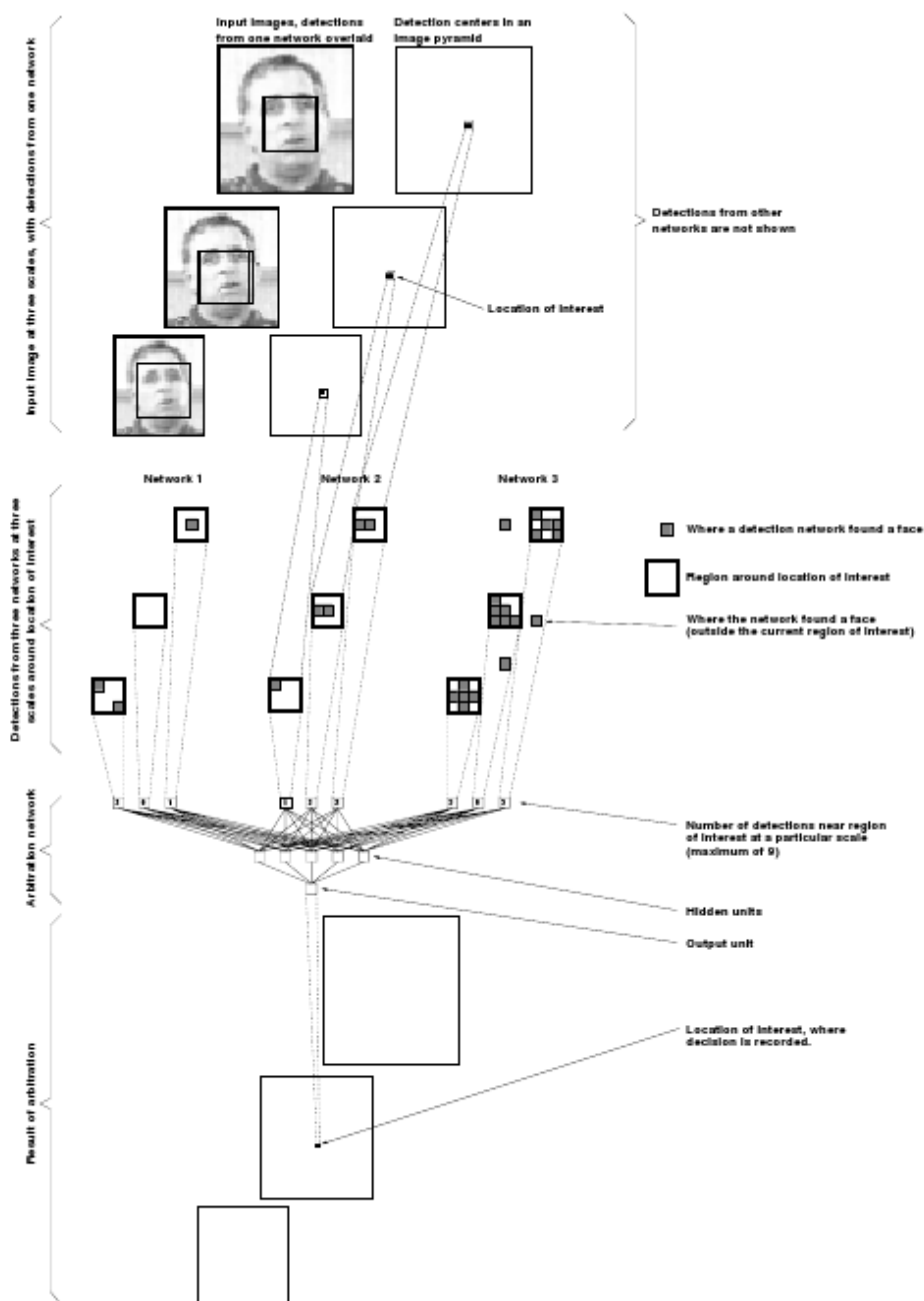
Jako první zmiňuje tzv. *prahování*. Je to funkce, které se zadají 2 parametry určující způsob odstranění detekce. První parametr udává vzdálenost překrývajících se detekcí a druhý počet překrývaných obličejů. V případě jeho nedosažení je detekce odstraněna.

Druhý způsob je *překrývání*. Překrytím se nemyslí jednotlivé detekce, ale výsledky více sítí. K této metodě je tedy potřeba natrénovat alespoň 2 sítě, přičemž každé se na počátku trénování inicializují jiné váhy. Tím se dosáve dvou různých klasifikátorů. Na každou oblast v obraze se pak aplikují obě sítě a detekce je kladná pouze v případě kladných detekcí obou sítí. Tento způsob detekce ukazuje obrázek 5.2.



Obrázek 5.2: Metoda překrývání podle H. Rowleyho (Zdroj: [5])

Třetí způsob je podobný předchozí metodě. Změnou je rozhodování o zachování detekce, která je praktikována arbitrážní neuronovou sítí. Každá síť vloží na vstup této sítě svoje výsledky a na základě průchodu arbitrážní sítě se detekce zavrhnou nebo vykreslí. Nastínění této metody zachycuje obrázek 5.3.



Obrázek 5.3: Rozhodování arbitrážní sítě (Zdroj: [5])

Kapitola 6

Návrh detektoru

Jak již bylo zmíněno, detektor využívá ke klasifikaci neuronové sítě. Návrh vychází z architektury sítě H. Rowleyho popsané v kapitole 5. Modifikace jsou pouze malé, nejvíce v části předzpracování obrazu. Normalizované okénko¹ je posunováno po jednotlivých pixelech, čímž se docílí přesnějšího detekování za cenu delšího času. Přesný algoritmus detekce lidského obličeje v obraze znázorňuje obrázek 6.1.

Jelikož je ve vstupní vrstvě k dispozici pouze hodnota intenzity barvy daného pixelu, přítomnost aktivační funkce by byla bezúčelná. Proto se aktivační funkce implementuje pouze do skryté a výstupní vrstvy. Jako aktivační funkce byla zvolena funkce hyperbolické tangenty popsána v sekci 3.6. To má za následek hodnoty od -1 do 1 na výstupech neuronů. Prahovou hodnotu, která rozhoduje, zda obraz je či není obličej, stanovil H. Rowley na 0 . Hodnoty větší říkají, že se jedná o obličej, hodnoty menší vypovídají o opaku. V této práci je po fázi testování sítě práh zvětšen na hodnotu 0.9 , čímž se znatelně sníží počet špatných detekcí.

Aby byla neuronová síť použitelná ke klasifikaci, musí se v první řadě dostatečně natrénovat. Za trénovací algoritmus byl po vzoru H. Rowleyho zvolen *backpropagation algorithm*². Znamená to, že se neuronové síti na vstup vloží vzorek dat a na výstup požadovaný výsledek sítě. Po průchodu sítí se chyba výpočtu sítě šíří sítí zpět a váhy se upravují tak, aby bylo dosaženo správného výsledku. V takovém procesu je potřeba znát derivaci aktivační funkce.

$$\frac{d}{dx} \tanh(x) = 1 - \tanh^2(x) \quad (6.1)$$

Po každém trénování je třeba zjistit chybovost sítě. Ta se získá z testovacího procesu. Na určitém testovacím vzorku se síti předkládají data a statisticky se vyhodnocuje jejich výsledek. Výsledkem je průměr chybovosti na obličejích a „neobličejích“. Ideální by bylo, kdyby chybovost sítě klesala. V praxi se ale chyba také zvyšuje, proto se musí detekovat op-

¹Normalizované okénko je vyříznutý obrázek z celkového obrazu o rozměru 20×20 , který je vstupem neuronové sítě.

²Algoritmus zpětného nastavení vah.

timální natrénování. Je nutné si pamatovat chybovost předchozích testů. Pokud dojde k situaci, že je v paměti dosažena hodnota menší než chybovost aktuálního trénování a zároveň počet hodnot v paměti odpovídá hodnotě nastavené v konfiguračním souboru, trénování sítě je úspěšně ukončeno. Znamená to, že bylo nalezeno globální minimum v průběhu trénování.

V průběhu se mění parametry trénování. Na začátku je chybovost okolo 50 %, váhy se proto nastavují s větším rozdílem, než při dotrénování sítě. Počáteční parametr úprav vah je nastavitelný v konfiguračním souboru. Při trénování je parametr fixně zmenšován. Když dosáhne chybovost hodnoty menší než 24 %, je citlivost³ nastavena na 0.0001 a při chybovosti menší 18 % na 0.00006, což zaručí za cenu delšího trénování přesnější klasifikaci.

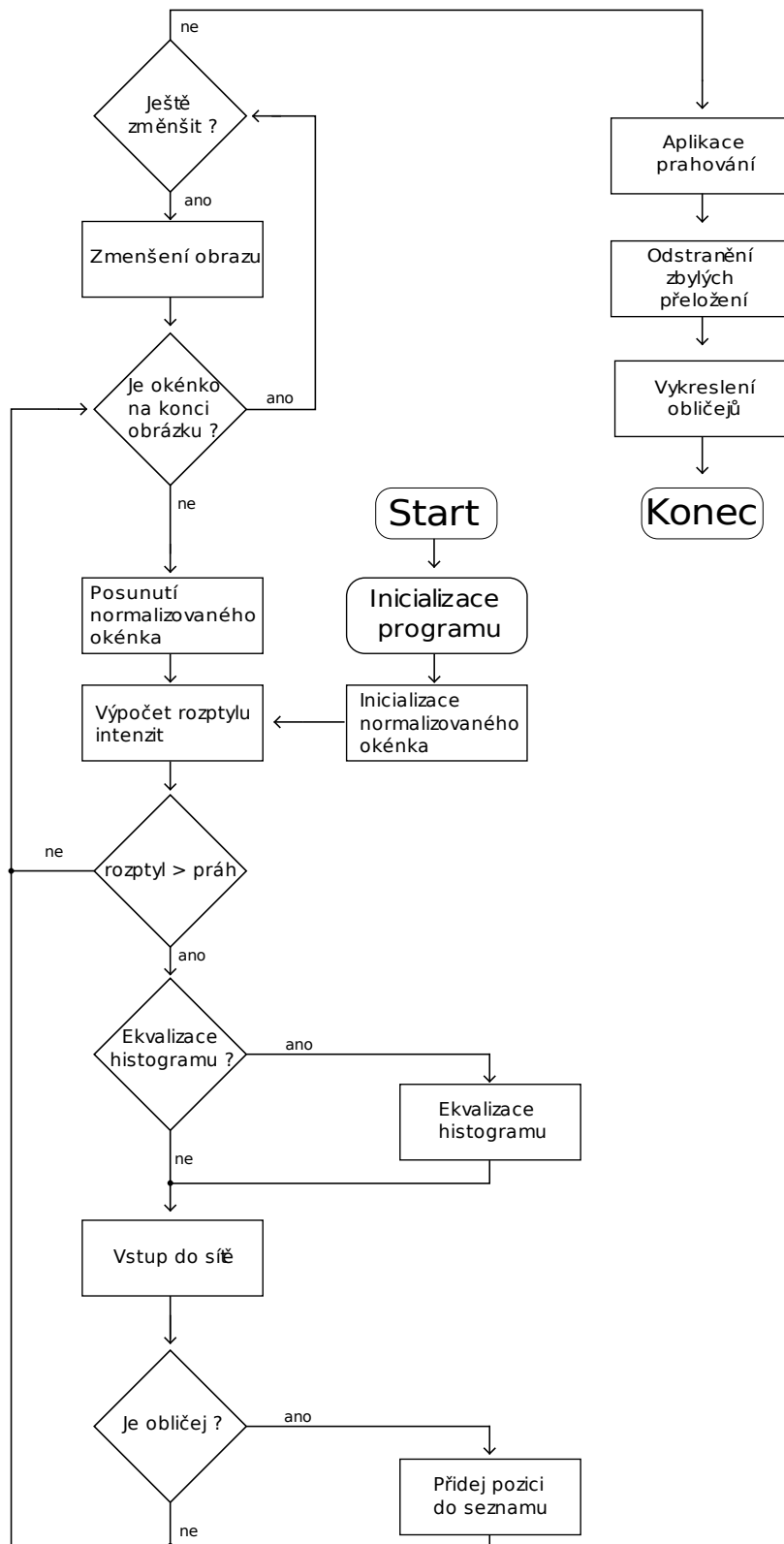
Abyste měli trénování co nejlepší výsledek, obličejové sady do trénovací sady prokládány obrázky bez obličejů. Takové obrázky je možné brát ze sady CBCL nebo si je přímo generovat. O trénovacích sadách je možné se více dočíst v sekci 8.1. Výstup sítě se trénuje na hodnotu 1 pro obličej a -1 pro zbytek. Pro dosažení lepších výsledků trénování se na trénovací sadu „neobličejů“ aplikuje tzv. *bootstrap* – pokud je při testování „neobličej“ označen jako obličej, je tento obrázek přidán do trénovací sady. Výsledky s využitím *bootstrap* jsou značně lepší.

Na větší obrázky, ve kterých se lokalizuje obličej, se uplatňuje tzv. *image pyramid*. V principu jde o zmenšování obrazu o určité procento. Na každý takto zmenšený obraz se aplikuje detektor tím způsobem, že se „vyřízne“ malý obrázek, který se vloží na vstup neuronové sítě. Postupně se tak do neuronové sítě vloží každý potenciální obličej. Obraz se zmenšuje 1.2krát.

Pro představu při faktoru zmenšení 1.2 a vstupním obrazu o rozlišení 1600x1200 pixelů je potřeba aplikovat jeden natrénovaný klasifikátor 6 263 896krát. Takové časté používání neuronové sítě není příliš efektivní. K detekci obličejů je možné použít i dvě neuronové sítě a detekci prohlásit za kladnou pouze v případě kladné shody obou sítí. Každý detekovaný obličej se ukládá do seznamu lokací, aby se v závěrečné fázi mohly odstranit chybné detekce. Ty se odstraňují *prahováním*. To znamená, že detekce musí dosáhnout stanovených prahových hodnot. Odstranění probíhá jako funkce $\text{threshold}(x, y)$, kde x je parametr vypovídající o minimální vzdálenosti od okraje první detekce k okraji druhé detekce a y udává minimální počet překrývajících se detekcí. Při nesplnění těchto podmínek je detekce ze seznamu odstraněna. Nakonec se odstraní všechny detekce, které se kryjí s jinou.

Podstatná je fáze předzpracování obrazu, kdy se aplikují na vstupní obraz výpočet rozptylu hodnot a ekvalizace histogramu. Tímto lze získat z neuronové sítě lepší výsledky.

³Citlivostí je myšlena hodnota, o kterou se změní váha mezi neurony při jedné iteraci trénování.



Obrázek 6.1: Algoritmus lokalizace obličeje

Fází předzpracování vstupního obrazu do klasifikátoru se opět značně zmenší množina možných vstupů. Obličej budou mít s jistotou tmavá místa v oblasti očí, nosních dírek a v místě mezi lícemi a nosem. Rozptyl hodnot tak bude díky osvětlení nabývat vyšších hodnot. Proto lze eliminovat vstupní obrazy, pokud se zahrnou jednobarevné obrazy nebo obrazy s malým rozptylem hodnot. Tím se také sníží počet aplikování detektoru na normalizované okénko a sníží celkový čas běhu programu. Více o rozptylem hodnot a jeho výpočtu se lze dočíst v podsekcí 6.1. Za účelem přesnější detekce se aplikuje na vstupní obraz z normalizovaného okénka ekvalizace histogramu popsaná v sekci 6.2.

6.1 Výpočet rozptylu intenzit

Jeden pixel šedého odstínu nabývá celočíselných hodnot v intervalu $\langle 0; 255 \rangle$, obrázek je tedy diskrétní systém.

Definice 6.1.1. *Rozptyl* je definován jako střední hodnota kvadrátů odchylek od střední hodnoty. Odchylku od střední hodnoty, která má rozměr stejný jako náhodná veličina, zachycuje směrodatná odchylka⁴.

Definice je citována z [14].

Výsledek výpočtu je využíván k určení výskytu obličeje. Pokud je rozptyl obrázku menší než prahová hodnota, obrázek vůbec do sítě nevstupuje. Příklady jsou na obrázku 6.2. Vzhledem k tomu, že pravděpodobnosti výskytu hodnot jsou stejné, lze výpočet rozptylu provést vztahem

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - E(x))^2 \quad (6.2)$$

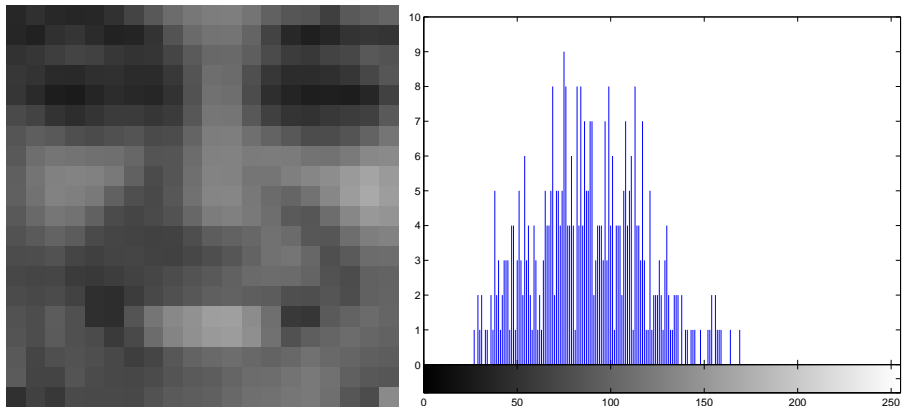


Obrázek 6.2: Rozptyly hodnot zleva: 205.005, 483.675, 637.13

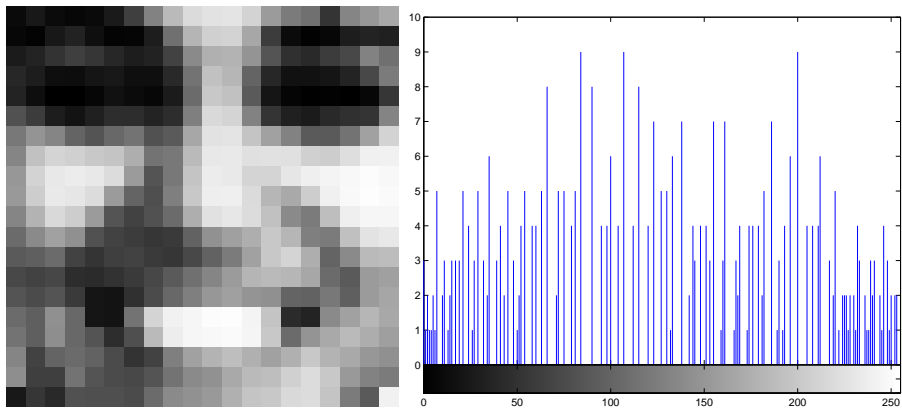
6.2 Ekvalizace histogramu

Tato fáze předzpracování upravuje vstup neuronové sítě tak, aby si byly obličej už při trénování co nejvíce podobné. Jedná se o úpravu kontrastu pomocí histogramu obrázku. Cílem je rozložit zastoupení hodnot šedi do celého intervalu $\langle 0; 255 \rangle$. Rozdíly před a po ekvalizaci histogramu lze pozorovat na obrázcích 6.4 a 6.3. Obsah této sekce čerpá z [9].

⁴Pro upřesnění se směrodatná odchylka značí σ , rozptyl je pak jejím kvadrátem σ^2 .



Obrázek 6.3: Obličej bez ekvalizovaného histogramu



Obrázek 6.4: Obličej s ekvalizovaným histogramem

Kapitola 7

Implementace

K implementaci byl zvolen jazyk C++. Důvodem jsou dostupné knihovny potřebné k řešení daného problému, výhody objektově orientovaného programování a modulárnost jazyka. Program byl vyvíjen pro platformu linux, překládán překladačem gcc 4.3.2. Možné výskyty chyb v programu jsou řešeny odchyťáváním výjimek. Zdrojový kód je rozdělen na 3 moduly:

- `localize` – obsahuje hlavní program a využívá zbylých dvou modulů
- `network` – specializuje se na veškerou práci s neuronovými sítěmi
- `image` – implementuje všechnu práci s obrazovými daty

Program neobsahuje žádné grafické rozhraní, jedná se o konzolovou aplikaci.

7.1 OpenCV

Tuto knihovnu vyvíjenou společností Intel lze využít pod BSD licencí. Existuje také lepší, placená OpenCV knihovna, pro účely této práce ale postačila knihovna k volnému stažení. Jde o knihovnu pro různé práce s obrazy nebo videi. V této práci je knihovna využívána pro načítání obrazu, ekvalizaci histogramu, zmenšování vstupního obrazu, vyhrazení okolí zájmu¹, ukládání obrazu a vykreslování čtverců okolo nalezených obličejů.

Při kompilaci programu musí být již na daném počítači OpenCV knihovna zkompilována. Je to jediná závislost programu.

7.2 Annie

Annie je knihovna pro tvorbu neuronových sítí, pokud programátora nezajímá vnitřní implementace neuronové sítě. Jsou zde třídy pro jednotlivé neurony, vrstvy, vstupní a výstupní vektory a spousta dalších užitečných věcí pro tvorbu sítí. Jde o velmi povedenou knihovnu,

¹Okolím zájmu je myšleno normalizované okénko.

ale ještě není kompletní. V dokumentaci se občas vyskytnou metody, které v knihovně zatím nejsou implementovány.

Pro tvorbu neuronových sítí existuje také *FANN*² knihovna, která umí pouze propojit veškeré neurony se všemi neurony vrstvy následující, což u architektury sítě H. Rowleyho nelze využít. Proto byla zvolena právě knihovna Annie, ve které lze přesně říci, do kterého konkrétního neuronu vede výstup stávajícího neuronu.

Veškeré zdrojové a hlavičkové soubory jsou součástí balíčku programu, překlad by měl proběhnout i bez zkompilevané knihovny na dané architektuře počítače.

7.3 Boost

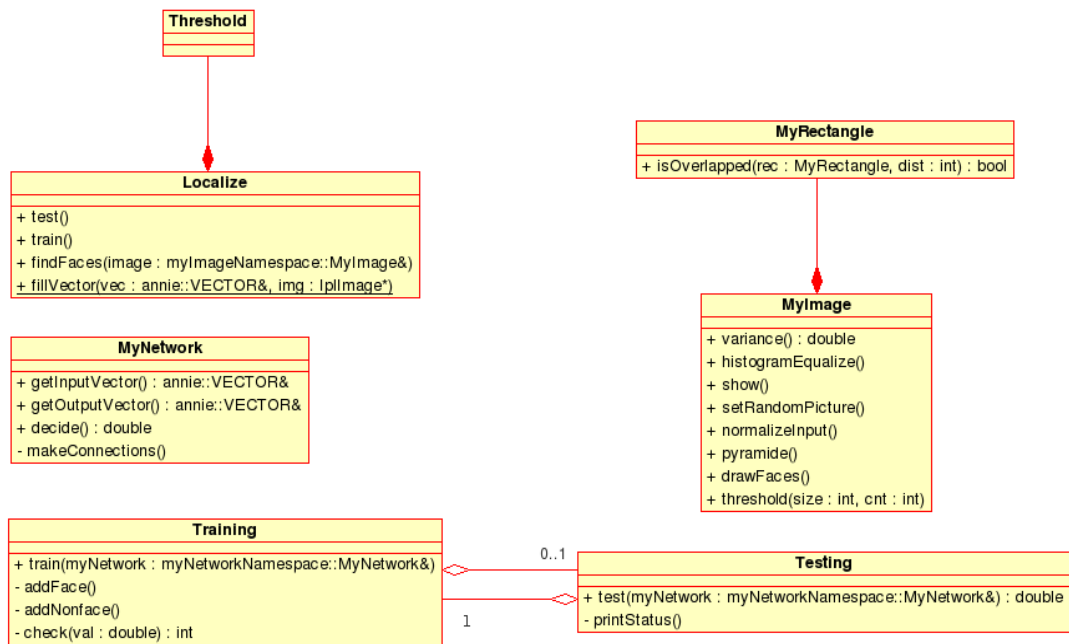
Boost knihovna je velmi rozsáhlá a má velice bohaté využití. V programu byla užita část pro práci se soubory v daném souborovém systému. Knihovna obsahuje širokou škálu funkcí, metod a tříd, ať už z oblasti matematiky, programování paralelních procesů, zpracování obrazů až po operace pro vstupně/výstupní zařízení. Ke správnému běhu programu není nutné mít tuto knihovnu zkompilevanou na počítači.

7.4 Třídy

Při programování bylo využito objektové orientace. Implementováno bylo 8 tříd, kde každá třída se specializuje na určitou svoji část. Zjednodušený diagram tříd je možné pozorovat na obrázku 7.1.

Mezi nejdůležitější třídy patří *Training* a *Testing*, jejichž metody zaručují trénování sítě s co nejmenší možnou chybovostí. Důležitá je také třída *Localize*, která rozhoduje o průběhu programu. Třída *MyImage* pracuje s obrázkem a třída *MyNetwork* s neuronovými sítěmi.

²*Fast Artificial Neural Network*



Obrázek 7.1: Diagram tříd

Kapitola 8

Testování

8.1 Trénovací sady

Dříve než dojde k procesu trénování, je třeba posbírat a vybrat nejvhodnější trénovací a testovací data. Každá sada se dělí na obrázky obsahující obličej a na obrázky bez obličejů. Takové rozdělení na trénovací a testovací sadu se nazývá *cross-validation*.

Velice vhodným kandidátem pro trénování neuronové sítě byla CBCL¹ sada z MIT². Obrázky již byly rozděleny do sad a ořezány na obličej. Jediný problém spočíval v jejich velikosti. Obrázky měly rozměr 19x19 pixelů. Po konverzi na správný rozměr se sada stala ideální. Příklad obličeje z této trénovací sady je ukázán u obrázku s ekvalizací histogramu 6.3. Sada obsahuje

- trénovací sadu o 2 429 obličejích a 4 548 „neobličejích“
- testovací sadu obsahující 472 obličej a 23 573 „neobličej“

Testovací obličej nejspíše příliš kvalitní. V sadě se nachází spousta rozostřených obličejů, které svým obsahem obličeje ani nepřipomínají. Příklady testovací sady jsou na obrázcích 8.1. Tato sada je k dispozici ke stažení na [4].



Obrázek 8.1: Ukázka obličejů testovací sady CBCL

Vzhledem k faktu, že se v testovací sadě nachází poměrně málo obličejů, byly některé obličej přesunuty z trénovací do testovací sady za účelem přesnějšího sledování chybovosti sítě při trénování.

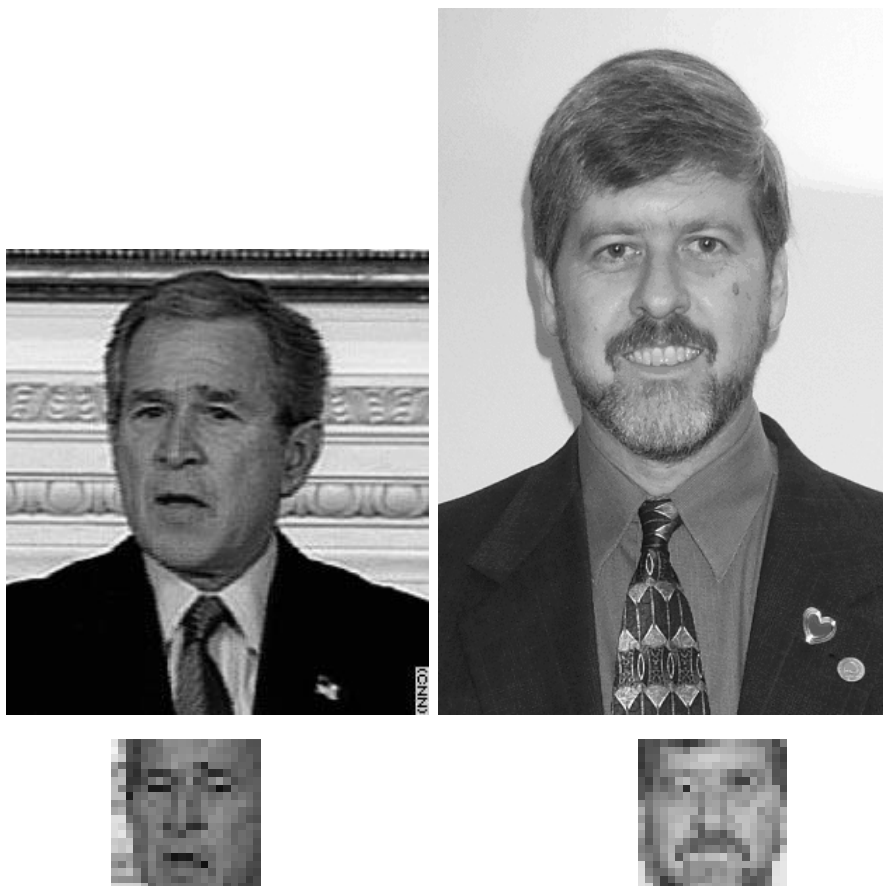
Druhá sada byla použita H. Schnedermanem a T. Kanadem při jejich výzkumu na Carnegie Mellon University. Sada se skládá z fotografií, které zachycují celé postavy s obličejem.

¹Center of biological and computational learning

²Massachusetts Institute of Technology

Je zde k dispozici soubor popisující lokace jednotlivých obličejů. Některé trénovací obrázky nejsou popsány. Nevýhodou je způsob, jakým jsou obličeje ohraničeny. Zatímco u obličejů sady CBCL je ohraničena pouze tvář, tato sada zachycuje celou hlavu člověka. Proto jsou v rozích různé intenzity, což klasifikátor lehce znepřesňuje. Dále se zde vyskytují obličeje různě natočené, některé nejsou brány z přímého pohledu. Nicméně program je přizpůsoben také k trénování pomocí této sady. Pro testování byly použity sady od M. Hradiše.

Úspěšnost detektoru s ekvalizací histogramu na jednotlivých trénovacích a testovacích sadách je možné vidět v tabulce 8.1 a bez ekvalizace histogramu v tabulce 8.2.



Obrázek 8.2: Ukázka obrázku ze Schneidermanovy sady a vstupy pro trénování sítě

Sada	Obličeje	Neobličeje
CBCL	81.5678 %	0.861155 %
Schneiderman	64.0000 %	0.390277 %
Upravená CBCL	11.7589 %	0.861155 %

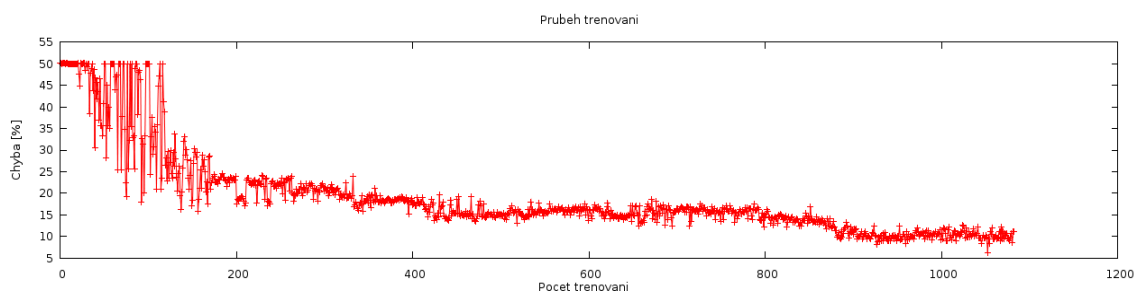
Tabulka 8.1: Chybovost sítě s ekvalizací histogramu

Sada	Obličeje	Neobličeje
CBCL	92.5847 %	0.449667 %
Schneiderman	43.6585 %	0.666016 %
Upravená CBCL	25.7905 %	0.436941 %

Tabulka 8.2: Chybovost sítě bez ekvalizace histogramu

8.2 Vyhodnocení

Chybovost sítě se při trénování zaznamenává do souboru `training-d_m_y_h_m_s.log`. Z takového logu lze vytvořit graf, který přesně znázorňuje trénování sítě. Ukázka průběhu trénování sítě je na obrázku 8.3. Testováním bylo prokázáno, že při trénování „neobličejů“ byl výsledek příznivější, pokud se trénovací vzorky generovaly z obrázků neobsahujících obličeje.

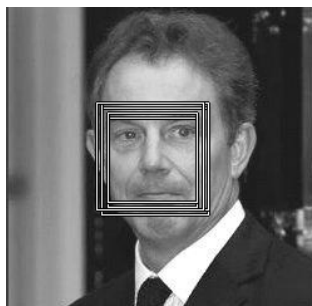


Obrázek 8.3: Průběh trénování neuronové sítě

Při lokalizaci hraje velkou roli konečná fáze, kde se vykreslují pouze překrývající se detekce. V průchodu obrazem se na jednotlivé výřezy aplikují dvě neuronové sítě. První je natrénována s aplikací ekvalizace histogramu a druhá bez fáze předzpracování. Obličej se zaznamenává pouze v případě shody obou sítí, že se na daném místě obličej nachází. Kvůli citlivosti sítě se okolo obličejů vytvoří shluk detekcí. Pokud dojde k chybné detekci³, projeví se na místě pouze jednou nebo dvakrát. To velmi usnadňuje hledání chybných detekcí, protože detekce nesplňující parametry překrývání jsou zahazeny. Příklad shluku detekcí ukazuje obrázek 8.4.

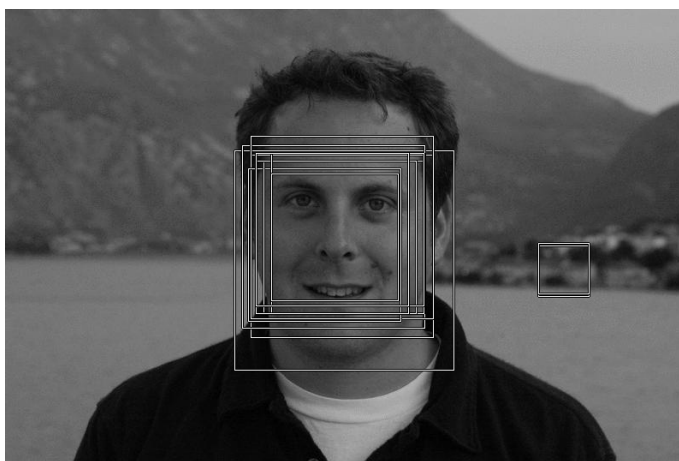
Není tam ale vidět chybná detekce, která je zřetelná na obrázku 8.5. Chybná detekce tvoří také malý shluk, ovšem pouze o četnosti 2. Proto musí být parametr pro odstranění chybných detekcí dostatečně velký.

³Chybné detekce jsou označovány jako *false positive*.



Obrázek 8.4: Shluk detekcí okolo obličeje

Po odstranění detekcí, které nesplňují podmínky funkce `threshold`, obrázek nabyde finálního stavu. Ten zobrazuje obrázek 8.6.



Obrázek 8.5: Chybná detekce



Obrázek 8.6: Odstranění nepřekrývajících se detekcí s parametry `threshold(4,4)`

Může se také stát, že v některých částech obrazu vytvoří neuronová síť velký shluk detekcí. Zde už nepomůže ani odstranění nepřekrývajících se detekcí a i ve výsledku se na špatném místě detekuje obličej. Příklad takové detekce je ukázán na obrázcích 8.7 a 8.8.



Obrázek 8.7: Příklad špatné detekce

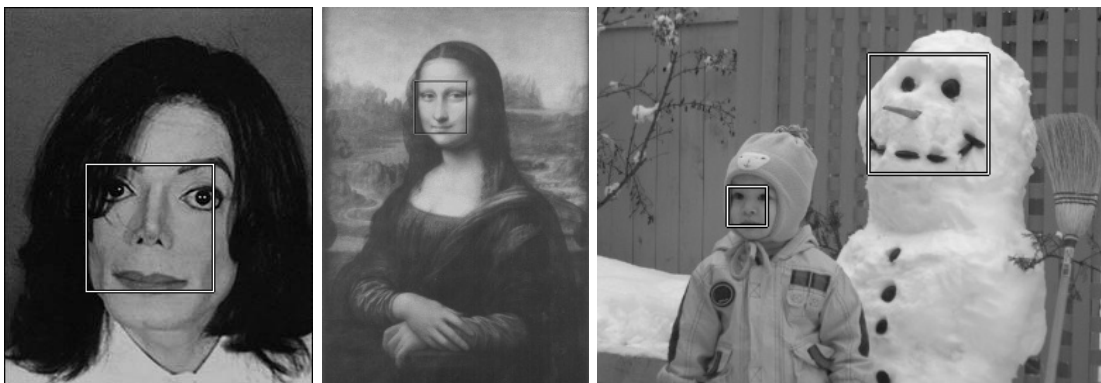


Obrázek 8.8: Po odstranění nepřekrývajících se detekcí

Detekce trvala průměrně okolo jedné minuty, v nejhorších případech až šest minut. Dobu detekce ovlivňuje obsah a hlavně rozměr obrazu. Na obrázcích 8.9 jsou uvedeny různé příklady detekcí obličejů, povedené i nepovedené.



 保良局李城璧中學 F.1D
Po Leung Kuk Lee Shing Pik College 2002-03



Obrázek 8.9: Příklady detekcí obličejů

Kapitola 9

Závěr

Cílem bakalářské práce bylo seznámit se s principy neuronových sítí a sestavit funkční lokalizátor obličejů v obraze za pomoci těchto sítí. Velikou inspiraci poskytla práce H. Rowleyho, podle jehož návrhu byl detektor sestaven.

Nejlepší výsledky detekce dosahovala síť trénovaná na upravené sadě CBCL a s fází předzpracování obrazu. Důvodem zlepšení v testovací fázi je přítomnost dobře osvětlených ostřejších obličejů. Trénování sítě trvalo v závislosti na inicializovaných hodnotách vah a generovaných „neobličejích“ různou dobu. Nejkratší trénování zabralo asi dvě hodiny, ale detektor nebyl použitelný. Důvodem bylo špatné nastavení parametrů trénování. Nejdelší trénování trvalo okolo šestnácti hodin. Nelze ale říct, že čím delší trénování, tím kvalitnější detektor. Při trénování bylo využito shellových skriptů, sítě byly trénovány sekvenčně za sebou s různými parametry a trénovacími sadami.

Cíle bakalářské práce se podařilo dosáhnout, lokalizace obličejů nejsou pochopitelně stoprocentní. Lokalizace funguje na obličejích nenatočených a fotografovaných zepředu, určitá malá tolerance natočení tam ale je. Dalším rozšířením této práce by mohla být detekce invariantní vůči rotacím. Dále by se mohlo zefektivnit hledání obličejů. Toho by bylo možné docílit, pokud by se při nedetekovaném obličejích normalizované okénko posunulo o více pixelů. Při kladné detekci by se případně mohlo o pixel vrátit, aby bylo okolo obličejů co nejvíce detekcí za účelem pozdější aplikace prahování.

Literatura

- [1] CHANG, H.; ROBLES, U.: Face Detection. [online] [cit. 5.5.2009].
URL <<http://www-cs-students.stanford.edu/~robles/ee368/results.html>>
- [2] DRAKOS, N.; MORE, R.: Optimal Edge-Based Shape Detection. [online] [cit. 5.5.2009].
URL <http://www.umiacs.umd.edu/~hankyu/shape_html/>
- [3] KIM, K.: Computer Vision Final Project. [online] [cit. 5.5.2009].
URL <http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject_KH.htm>
- [4] MIT: CBCL face database. [online] [cit. 11.11.2008].
URL <<http://cbcl.mit.edu/software-datasets/FaceData2.html>>
- [5] ROWLEY, H. A.; BALUJA, S.; KANADE, T.: Neural Network-Based Face Detection. *IEEE Transactions On Pattern Analysis and Machine intelligence*, ročník 20, 1998: s. 23–38.
- [6] StatSoft: Support Vector Machines. [online] [cit. 5.5.2009].
URL <<http://www.statsoft.com/textbook/stsvm.html>>
- [7] VLACH, J.; PŘINOSIL, J.: Lokalizace obličej v obraze s komplexním pozadím. *Elektrorevue*, ročník 12, 2007: s. 1–2.
- [8] Wikipedia: AdaBoost - Wikipedia, the free encyclopedia. [online] [cit. 24.4.2009].
URL <<http://en.wikipedia.org/wiki/AdaBoost>>
- [9] Wikipedia: Histogram equalization - Wikipedia, the free encyclopedia. [online] [cit. 16.11.2008].
URL <http://en.wikipedia.org/wiki/Histogram_equalization>
- [10] Wikipedia: Neural network - Wikipedia, the free encyclopedia. [online] [cit. 29.10.2009].
URL <http://en.wikipedia.org/wiki/Neural_network>

- [11] Wikipedia: Support vector machine - Wikipedia, the free encyclopedia. [online] [cit. 24.4.2009].
URL <http://en.wikipedia.org/wiki/Support_vector_machine>
- [12] Wikipedie: Neuron - Wikipedie, otevřená encyklopedie. [online] [cit. 29.10.2008].
URL <<http://cs.wikipedia.org/wiki/Neuron>>
- [13] Wikipedie: Neuronová síť - Wikipedie, otevřená encyklopedie.
URL <http://cs.wikipedia.org/wiki/Neuronov%C3%A1_s%C3%AD%C5%A5#P.C5.99enosov.C3.A9_funkce>
- [14] Wikipedie: Rozptyl (statistika) - Wikipedie, otevřená encyklopedie. [online] [cit. 14.3.2009].
URL <[http://cs.wikipedia.org/wiki/Rozptyl_\(statistika\)](http://cs.wikipedia.org/wiki/Rozptyl_(statistika))>

Dodatek A

Obsah CD

- `poster.png` – plakát prezentující tuto práci
- `program/`
 - `config.cfg` – konfigurační soubor pro trénování, testování a užívání neuronové sítě
 - `doc/` – HTML dokumentace ke zdrojovému kódu generovaná nástrojem Doxygen
 - `Doxyfile/` – konfigurační soubor pro generování dokumentace Doxygenem
 - `images/` – adresář obsahující trénovací a testovací sady
 - `include/` – adresář se soubory potřebnými k překladu
 - `libs/` – adresář se zkompilevanými knihovnami důležitými pro linkování
 - `Makefile` – `make program` přeloží, `make pack` vytvoří archiv `xlibos01-bp.tar.gz` a `make clean` odmaže objektové soubory k linkování a binární spouštěcí soubor
 - `networks/` – adresář s natrénovanými neuronovými sítěmi použitelnými k lokalizaci obličejů
 - `README` – textový soubor s popisem spouštění programu a popisem konfiguračního souboru
 - `src/` – adresář se zdrojovými soubory programu
- `report/` – zdrojové soubory tohoto dokumentu
 - `bp.tex` – preambule
 - `cls/` – adresář s definicí dokumentu
 - `czechiso.bst` – český standard pro citace ČSN ISO 690
 - `desky.tex` – desky
 - `detection/` – adresář s obrázky detekce
 - `fig/` – adresář s obrázky

- Makefile
- obsah.tex – obsah dokumentu
- prilohy.tex – přílohy dokumentu

Dodatek B

Manuál

Program se přeloží příkazem `make`. `make clean` odstraní binární a objektové soubory a `make pack` archivuje soubory do `bp-xlibos01.tar.bz2`.

- `./localize -test networkName` – otestuje síť jménem `networkName` na testovací sadě specifikované v konfiguračním souboru
- `./localize -train [trainedNetwork]` – trénuje síť podle parametrů nastavených v konfiguračním souboru, pokud je přidán název sítě `trainedNetwork`, pokračuje se v trénování této sítě
- `./localize picture [outputPicture]` – hledá obličej v obraze `picture` a výsledek zobrazí, pokud je přidán název `outputPicture`, je výsledek uložen do tohoto souboru
- `./localize --help` – vypíše tuto nápovědu

Dodatek C

Konfigurační soubor

`config.cfg`

- `//` – na začátku řádku značí komentář
- `threshold="x,y"` – `x` je počet pixelů, které se mají při odstranění nepřekrývajících detekcí od kraje prohledávat a `y` minimální počet překrytí
- `noHistNet="path/to/network"` – cesta k neuronové síti, která byla trénována bez fáze předzpracování obrazu
- `net="path/to/network"` – cesta k neuronové síti, která byla trénována s fází předzpracování obrazu¹
- `hist="x"` – `x` je hodnota 1 v případě, že při trénování/testování je vyžadována fáze předzpracování obrazu, hodnota 0 pokud vyžadována není
- `learningRate="x"` – `x` je citlivost nastavování vah při trénování
- `momentum="x"` – `x` je citlivost pro doznívání trénování v další iteraci
- `batch="x"` – `x` je počet tváří a „netváří“ v jedné trénovací iteraci, celkem bude v trénovací iteraci tedy $2x$ obrázků
- `globalsize="x"` – `x` je počet větších minim v trénování pro zastavení trénování
- `trainfaces="/path/to/dir"` – cesta k adresáři s obličejovou sadou pro trénování
- `trainnonfaces="/path/to/dir"` – cesta k adresáři s „neobličejovou“ sadou pro trénování
- `testfaces="/path/to/dir"` – cesta k adresáři s obličejovou sadou pro testování
- `testnonfaces="/path/to/dir"` – cesta k adresáři s „neobličejovou“ sadou pro testování

¹Musí být zadána cesta alespoň k jedné síti – `net` nebo `noHistNet`.