

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

DETEKCE OBJEKTŮ V OBRAZE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

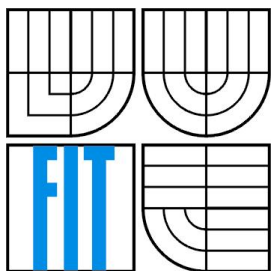
AUTHOR

TOMÁŠ VAŮKO

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

DETEKCE OBJEKTŮ V OBRAZE

OBJECT DETECTION IN IMAGES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ VALKO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MIROSLAV ŠVUB

BRNO 2009

Abstrakt

Detekce objektů v obraze je velmi populární téma už několik let. Značí o tom mnoho prací v této oblasti počítačové vědy. V této práci se věnujeme klasifikaci objektů, konkrétně lidských obličejů, jako jedním z nejzajímavějších objektů na zpracování. Ke klasifikaci používáme neuronové sítě, naučené na databáze obličejů. Zkoumá se vliv velikosti databáze a předzpracování digitálního obrazu na učení neuronové sítě. Práce implementuje jednoduchý detektor a lokalizátor obličejů. Obsahuje shrnutí úspěšných i méně úspěšných výsledků a naznačuje další možný vývoj systému v budoucnu.

Abstract

Object detection in images is quite popular topic for years. What stands for it are a lot of works from this area of computer science. This thesis is about object classification, specifically human faces, which are one of the most interesting objects for processing. For classification we use neural networks, learned on face database. We study what influence has size of face database and preprocessing of digital image on neural network learning. This project implements simple face detector and localizer. It summarizes more and less successful results and indicates possible ways of system development in the future.

Klíčová slova

Umělá neuronová síť, počítačové vidění, obličejová databáze, FANN, CImg, rozpoznávání objektů, rozpoznávání vzorů, lokalizace obličeje

Keywords

Artificial neural network, machine vision, face database, FANN, CImg, object recognition, pattern recognition, face localization

Citace

Vařko Tomáš: Detekce objektů v obraze, bakalářská práce, Brno, FIT VUT v Brně, 2009.

Detekce objektů v obraze

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Miroslava Švuba. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Vaľko
15. května 2009

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Miroslavovi Švubovi za celoroční přístup, ochotu a pomoc. Dále bych rád poděkoval autorům knihoven Stefanu Nissenovi (FANN) a Davidovi Tschumperlé (CImg). Na závěr bych poděkoval své přítelkyni, své rodině a svým přátelům, kteří my byli oporou.

© Tomáš Vaľko, 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.....	2
2	Spracovanie obrazu a detekcia objektov	3
2.1	Základné pojmy pri práci s digitálnym obrazom	3
2.2	Obrazové klasifikátory.....	6
3	Neurónové siete	11
3.1	Model umelého neurónu	11
3.2	Typy neurónových sietí	13
3.3	Učenie neurónovej siete.....	14
4	Metódy detekcie tvári.....	16
4.1	Metódy založené na vzhľade	18
5	Návrh aplikácie	19
5.1	Motivácia	19
5.2	Trénovanie neurónovej siete	19
5.3	Testovanie neurónovej siete	20
5.4	Lokalizácia tvári pomocou neurónovej siete	21
6	Implementácia.....	23
6.1	Fast Artifical Neural Network Library	23
6.2	Cool Image Library.....	24
6.3	Modul Train.....	24
6.4	Modul Test.....	25
6.5	Modul Main	25
7	Experimentálne výsledky	26
7.1	Neurónová sieť ako detektor.....	26
7.2	Neurónová sieť ako lokalizátor.....	30
8	Záver	34

1 Úvod

Od počiatku zostavenia prvého počítača v päťdesiatych rokoch minulého storočia sa tomuto vynálezu pripisovali veľké úspechy a veľké očakávania. Z informatiky ako vedy o počítačoch sa stala jedna z najdynamickejšie sa rozvíjajúcou vednou disciplínou. Rozčlenila sa do mnohých odvetví, ovplyvnila iné vedné disciplíny a našla si svoje miesto aj v bežnom živote. Prvé počítače slúžili človeku ako nástroj automatizácie, ktorý odbremenil človeka od rutinných činností. Postupom času narastal výpočtový výkon počítačov a čoraz častejšie sa začali objavovať snahy priblížiť počítač človeku ešte viac, bližšie živej prírode. A tak ako aj iné vedné disciplíny hľadajú inšpiráciu v prírode, tak aj informatika z nej čerpá. Počítač už nie je len „hlúpy“ stroj, dokazujú to oblasti výskumu ako umelá inteligencia, genetické algoritmy či počítačové videnie. Napriek tomu všetkému počítače v inteligencii za človekom stále zaostávajú. Mnohé úlohy, ktoré človek zvládne bez problémov musia byť pre počítač zjednodušené. Napríklad v oblasti spracovania obrazu, kedy človek nemá problém v obraze rozoznávať jednotlivé objekty počítač až tak úspešný nie je. Hoci si počítače pripísali mnohú úspechy, stále pretrvávajú očakávania aké boli na začiatku ich vzniku. Z toho vyplýva, že vývoj v tejto oblasti vedy sa nikdy nezastaví.

Táto práca sa zaoberá jednou z oblastí počítačovej grafiky, spracovaním obrazu. Predstavuje úvod do problematiky detekcie objektov v obraze a detailnejšie popisuje detekciu tváří pomocou neurónových sietí. Súhrne popisuje celý postup vývoja od výberu objektu detekcie, predspracovaním obrazu cez tréning neurónovej siete až po testovanie a vyhodnotenie úspešnosti.

Po úvodnej časti nasleduje druhá časť, v ktorej sa zoznámime so základnými teoretickými a praktickými informáciami o digitálnom obraze, jeho reprezentáciou a spracovaním. Ďalej sa oboznámime s problematikou detekcie objektov v obraze a rôznymi obrazovými klasifikátormi. Tretia časť obsahuje nevyhnutný teoretický základ potrebný k pochopeniu neurónových sietí. Nasledujúca dve časti popisujú celkový návrh aplikácie a jej implementáciu. Záverečné časti publikácie testujú aplikáciu, zhrňujú dosiahnuté výsledky a diskutujú o ďalších možnostiach vývoja aplikácie.

2 Spracovanie obrazu a detekcia objektov

Skôr ako začneme so spracovaním obrazu a detekciou objektov definujeme si pojmy a postupy, bez ktorých by sme sa nezaobišli.

2.1 Základné pojmy pri práci s digitálnym obrazom

Oboznámime sa s celkom základnými a pri práci s digitálnym obrazom nevyhnutnými informáciami. Ekvalizácia histogramu a zmena rozmerov obrázkov patria medzi najčastejšie operácie s digitálnym obrazom a preto sa bez nich nezaobídeme ani mi. Ale najprv pekne od začiatku.

2.1.1 Digitálny obraz

Digitálny obraz je dvojrozmerný obraz reprezentovaný 0 a 1 (binárne). Podľa toho, či je rozlíšenie obrazu fixné môžeme rozlíšiť dva typy vektorový alebo rastrový. Pojem digitálny obraz sa v praxi používa najčastejšie na označenie rastrového typu obrazu, s ktorým budeme pracovať aj mi.

Rastrový obraz obsahuje konečnú množinu hodnôt, obrazových častí nazývaných pixely. Tie sú usporiadané do riadkov a stĺpcov a ich adresovanie je zhodné s adresovaním prvkov v matici.

$$I(x, y) = B$$

I z anglického image (obraz) označuje celú maticu digitálneho obrazu. Každý bod tohto obrazu má hodnotu B . Hodnota B reprezentuje farbu daného bodu (pixlu). Môže byť zložená z troch hodnôt (RGB), vtedy je obrázok farebný alebo jej hodnota je jediné číslo z daného rozsahu (napr. od 0 do 16), vtedy ide o obrázok v stupňoch šedi. Čím je rozsah hodnôt, ktoré môže bod nadobudnúť, väčší tým je väčšia farebná hĺbka obrazu.

V tejto práci budeme používať obrazy, kde B bude reprezentované jediným číslom z intervalu $\langle 0, 255 \rangle$. Čiže pôjde o obrázky s 256 stupňami šedej farby.

Je mnoho súborových formátov, v ktorých je rastrový obraz uložený v počítači napr. RAW, JPEG alebo GIF.

2.1.2 Ekvalizácia histogramu

Ekvalizácia histogramu je metóda spracovania obrazu upravením kontrastu, využíva sa k tomu histogram obrazu. Prehľadne popísaná je v [1], odkiaľ sme aj mi čerpali cenné informácie. Histogram znázorňuje početnosť jednotlivých intenzít v obraze. Ekvalizácia histogramu sa používa najmä pri

šedotónových obrazoch. Je možné túto metódu použiť aj pre farebné obrazy, vtedy sa ekvalizuje každý farebný kanál zvlášť. Používa sa k zlepšeniu kontrastu obrazu, napríklad vtedy, keď je svetlý objekt na svetlom pozadí alebo naopak tmavý objekt na tmavom pozadí. Zlepšuje kvalitu podexponovaných a preexponovaných obrazov. Distribuuje najviac frekventované hodnoty v histograme a tým dokáže zlepšiť lokálny kontrast bez zmeny globálneho kontrastu. Pre šedotónový obrázok je pravdepodobnosť výskytu pixlu s hodnotou intenzity i definovaná ako:

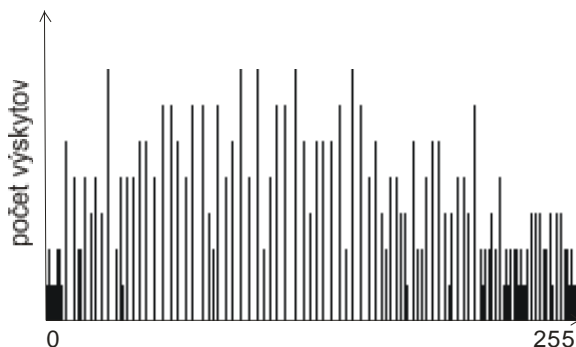
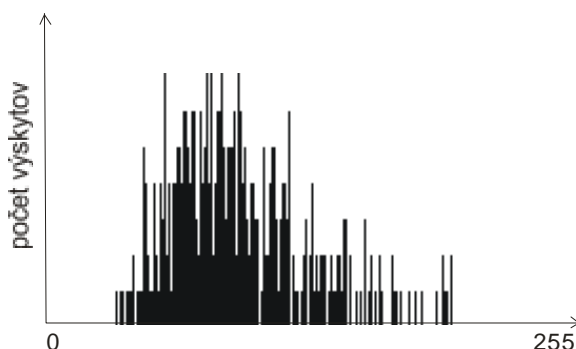
$$p_x(i) = p(x=i) = \frac{n_i}{n}, 0 \leq i < L \quad (2.1)$$

Kde n_i je počet výskytov intenzity i , L je celkový počet stupňov šedi a n je celkový počet pixlov obrazu. V podstate je to histogram obrazu normalizovaný na hodnoty z intervalu $<0,1>$. Kumulatívna distribučná funkcia je potom definovaná nasledovne:

$$cdf_x(i) = \sum_{j=0}^i p_x(j) \quad (2.2)$$

Výsledná ekvalizácia histogramu je definovaná vzťahom:

$$y_i' = y_i(\max\{x\} - \min\{x\}) + \min\{x\} \quad (2.3)$$



Obrázok 2.1: Ukážka neekvalizovaného a ekvalizovaného obrázku, vrátane histogramov.

2.1.3 Zmena rozmeru obrazu (presamplovanie)

Pri zmene rozmeru obrazu dochádza ku zmene celkového počtu pixlov v obraze, tým nastáva znateľné zhoršenie kvality obrazu. Preto sa pri zmene rozmerov obrazu používa technika, ktorá túto deformáciu z veľkej časti potláča, interpolácia. Viac o tejto problematike v [2].

Interpolácia pracuje tak, že používa známe hodnoty pixlov na určenie hodnôt pixlov neznámych. Farba bodu obrazu, ktorý po zmene rozmerov v obraze zostal sa určí pomocou váženého priemeru okolitých vynechaných bodov. Interpoláčne algoritmy rozdeľujeme do dvoch základných skupín, adaptívne algoritmy, ktoré sa menia podľa toho čo interpolujú (ostré hrany, hladká textúra) a neadaptívne algoritmy.

Neadaptívne algoritmy sú také, ktoré nerozlišujú rôzne typy pixlov v obraze ale berú všetky za rovnocenné. Medzi takéto algoritmy patria interpolácie napr. nearest neighbor, bilineárna, bikubická a iné. Čím viac pôvodných pixlov použijú na určenie farby nového pixlu, tým je interpolácia presnejšia ale zároveň tým narastie aj čas potrebný na spracovanie celého obrazu. Jednou zo základných interpolácií je nearest neighbor, ktorá spotrebuje najkratší výpočtový čas zo všetkých interpolácií, lebo na určenie novej hodnoty pixlu použije iba jednu hodnotu pixlu pôvodného. Bilineárna interpolácia používa 4 okolité body a bikubická až 16 bodov.

$$W_x = \begin{cases} 1 & x = 0 \\ (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 < |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 < |x| < 2 \\ 0 & \text{inak} \end{cases}$$

$$p(t) = \frac{1}{2} \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} a_{-1} \\ a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

Obrázok 2.2: Ukážka výpočtu bikubickej interpolácie pomocou algoritmu konvolúcie
a maticový zápis (a typicky -0,5 alebo -0,75),
prevzaté z http://en.wikipedia.org/wiki/Bicubic_interpolation

2.1.4 Normalizácia

Pri tej všetkej rozmanitosti, ktorá sa vyskytuje pri práci s tvármi, o ktorej sme hovorili v predchádzajúcej časti je vhodná ešte jedna úprava vstupných dát, normalizácia. Samotný pojem normalizácia označuje zjednotenie do rovnakej formy. Zahrňuje to viacero krokov, ktoré upravujú vstupnú množinu dát tak aby bola uniformná.

V prvom kroku ide o výber farebnej hĺbky spracovaného obrazu. Pri metódach klasifikácie obrazu sa farebné obrazy skoro vôbec nepoužívajú a prevládajú obrazy v stupňoch šedi. Farebné

obrazy sú pre tento účel zbytočne objemné a nemajú žiaden prínos. Práve naopak, došlo by k výraznému nárastu časovej a priestorovej zložitosti. Obrazy v stupňoch šedi môžu mať rôznu farebnú hĺbku napríklad 64, 128 či 256 odtieňov. Pre ďalší vývoj je nutné vybrať si jednu farebnú hĺbku a tú používať pri práci so systémom. Mi budeme používať obraz v stupňoch šedi s 256 odtieňmi.

Na záver prevedieme hodnoty vstupného obrazu, ktoré sú v našom prípade z intervalu 0 až 255 vrátane buď na hodnoty v rozsahu $\langle 0,1 \rangle$ popisuje rovnica 2.4 alebo na hodnoty z rozsahu $\langle -1,1 \rangle$, čo popisuje rovnica 2.5. Tieto rozsahy sú bežne používané detektormi obrazov, čiže aj neurónovými sieťami.

$$x_n = \frac{x}{L-1} \quad (2.4)$$

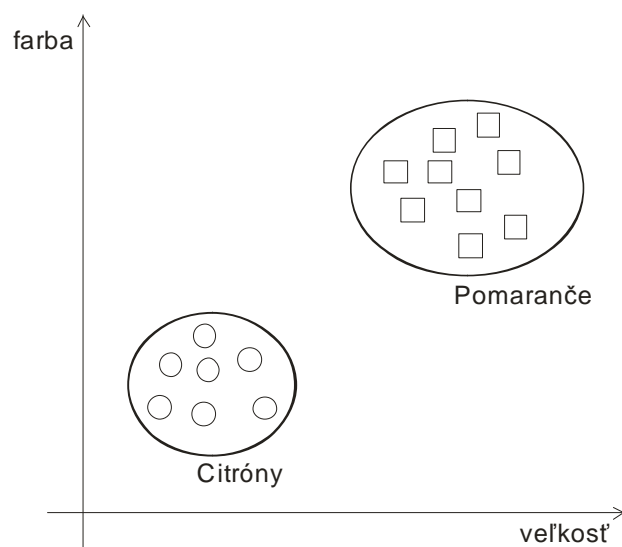
$$x_n = 2 \cdot \left(\frac{x}{L-1} - 0,5 \right) \quad (2.5)$$

Kde x je hodnota bodu pôvodného, x_n hodnota bodu normalizovaného a L je počet odtieňov šedej farby.

2.2 Obrazové klasifikátory

Žiadna úloha z oblasti počítačového videnia sa nezaobíde bez pomoci rozpoznávania. Rozpoznávanie vzorov sa používa na klasifikáciu objektov a častí obrazu, ide o spracovanie obrazu smerom zdola nahor. K tejto klasifikácii sú potrebné znalosti, bez ktorých by rozpoznávanie vzorov v obraze nebolo možné. Nato aby sme odlíšili jeden objekt od druhého potrebujeme poznať práve tieto znalosti, všeobecné vlastnosti a typické črty objektu, podľa ktorých sa rozhodujeme. Poznatky k problematike obrazových klasifikátorov boli čerpané z [3]. Uvedieme si jednoduchý príklad.

Majme nádobu plnú citrónov a pomarančov. Našou úlohou bude oddeliť citróny od pomarančov. Ak by sme nemali žiadne znalosti o týchto objektoch tak by nám neostávalo nič iné len každý jeden ochutnať a podľa toho sa rozhodnúť. Mi však poznáme typické črty citróna (menšia veľkosť, žltá farba) a pomaranča (väčšie rozmery, oranžová farba). Tie nám stačia nato, aby sme rozpoznali jednotlivé objekty a oddelili ich od seba do dvoch tried.



Obrázok 2.3: Rozpoznanie citrónov a pomarančov, črty sú veľkosť a farba

Existuje viacero možností reprezentácie znalostí. Mi si predstavíme techniky, ktoré sú používané najčastejšie.

2.2.1 Reprezentácia znalostí

Znalosti a ich reprezentácia je problematika, ktorou sa zaoberá umelá inteligencia. Oblasť počítačové videnie využíva poznatky z tejto oblasti. Medzi hlavné techniky reprezentácie znalostí používané v umelej inteligencii sú formálne gramatiky a jazyky, predikátová logika, vytváracie pravidlá, sémantické siete a rámce. Dobrá voľba spôsobu reprezentácie znalostí je jeden z najdôležitejších krokov v riešení daného problému. Techniky reprezentácie znalostí prehľadne prezentuje práca [3].

Opisy, črty

Opisy a črty sa nepovažujú za korektnú techniku reprezentácií znalostí, napriek tomu sa môžu použiť ako súčasť komplexnejšej reprezentačnej štruktúry. Opisy zvyčajne reprezentujú nejaké skalárne vlastnosti objektov, tie sa nazývajú črty. Jediný opis objektu je nedostačujúci, preto sa opisy kombinujú a vznikajú vektory črt.

Formálne gramatiky a jazyky

Ak chceme popísať štruktúru objektu, opis črtami je nevhodný. Popis štruktúry objektu sa skladá z primitív a vzťahov medzi nimi. Primitíva sú reprezentované informáciou o ich type. Jednými z najjednoduchších foriem reprezentácie štruktúry sú reťaze, stromy a grafy.

Jeden objekt môže byť popísaný jednou z spomenutých štruktúr, no na popis celej triedy objektov jediná štruktúra nestačí. Trieda štruktúr popisujúcich objekt je gramatika a jazyk. Gramatiky a jazyky poskytujú pravidlá, ktoré definujú ako zostrojiť reťaze, stromy alebo grafy z množiny symbolov (primitív).

Predikátová logika

Predikátová logika zavádza matematický formalizmus na odvodenie nových znalostí zo starých znalostí použitím matematickej dedukcie. Predikátová logika kombinuje použitie logických kvantifikátorov (\exists, \forall) a logických operátorov ($\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$). Základnými stavebnými blokmi predikátovej logiky sú pravidlá modus ponens a rezolúcia.

Používa sa na popis objektov, ktoré sú definované logickými premennými. Nevýhodou tejto techniky reprezentácie znalostí je, že nepodporuje použitie s neistými či neúplnými informáciami. Vždy potrebuje kompletný popis, pravda (*true*) alebo nepravda (*false*).

Vytváranie pravidiel

Vytváranie pravidiel tvoria širokú škálu reprezentácie znalostí, sú založené na dvojiciach podmienka, akcia (*ak* podmienka X *potom* akcia Y). Dôležitou vlastnosťou je procedurálny charakter reprezentácie znalostí, z ktorej vyplýva, že nie všetky informácie o objekte musia byť uvedené.

2.2.2 Klasifikácia

Objekt z pohľadu počítačového videnia je oblasť segmentovaného obrazu. Skupina objektov môže byť rozdelená do disjunktných množín, ktoré majú nejaké spoločné črty, nazývame ich triedy.

Rozpoznávanie objektov je založené na pridelovaní tried objektom, tento proces sa nazýva klasifikácia (zariadenie sa nazýva klasifikátor). Počet tried je zvyčajne vopred známy, dá sa odvodiť zo špecifikácie problému. Klasifikátor, podobne ako človek, nerozhoduje o triede podľa samotného objektu, ale podľa jeho vlastností. Takéto vnímanie objektu je označené ako vzor (pattern). Vzor označovaný ako vektor vzorov alebo vektor črt ($x = x_1, x_2, \dots, x_n$) popisuje objekt. Potom klasifikátor nerozpoznáva objekty ale ich vzory. Klasifikátor má n vstupov a 1 výstup, každý vstup sa použije na zadanie informácii o jednej črte z vektora črt. Rozpoznávanie objektov a rozpoznávanie vzorov sa považuje za zhodné.

Základné kroky rozpoznávania vzorov sú: voľba objektu (I), zostrojenie jeho formálneho popisu, vzniká vzor, ktorý spracuje klasifikátor (K) a zadelí objekt do triedy (T_n). Zjednodušene sa postup klasifikácie dá zapísať:

$$I \rightarrow K \rightarrow T_n$$

Klasifikátor založený na diskriminačnej funkcii je deterministický stroj, jeden vzor x bude vždy klasifikovaný do rovnakej triedy T . Objavuje sa problém, že vzor x môže reprezentovať objekty z rozličných tried. Potom klasifikátor je úspešný pre nejaké objekty a pre iné nie je. Preto nastavenie optimálneho klasifikátora by malo byť pravdepodobnostné.

Schopnosť nastaviť parametre klasifikácie z množiny príkladov je veľmi dôležitá činnosť, ide o učenie klasifikátora. Množina, na ktorej prebieha učenie klasifikátora sa nazýva **trénovacia množina**. Skladá sa zo skupiny objektov (reprezentovaných vektormi črt) a informáciami o

ich korektnej klasifikácií. Kvalita nastavenia klasifikátora závisí na kvalite a veľkosti trénovacej množiny, ktorá je vždy konečná. Preto na návrh a nastavenie klasifikátora nie je možné použiť všetky objekty, ktoré budú neskôr klasifikované. To znamená, že klasifikátor bude spracovávať aj objekty na ktorých nebol naučený a mal by byť schopný ich rozpoznať. Metódy slúžiace na návrh klasifikátora musia byť indukčné, v tom význame, že informácie získané z prvkov obsiahnutých v trénovacej množine musia byť generalizované aby pokryli celú množinu vektorov čít.

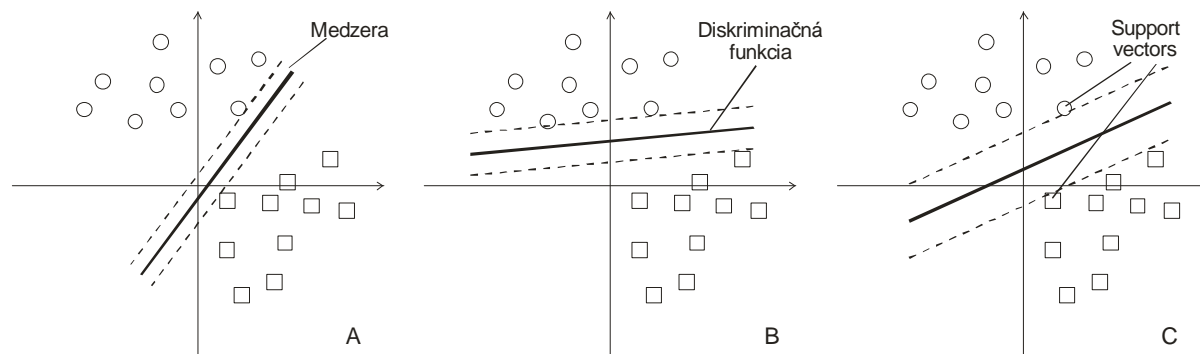
Čím väčšiu trénovaciu množinu použijeme na nastavenie klasifikátora, tým je väčšia šanca, že bude pracovať správne. Ak poznáme štatistické vlastnosti vzorov, môžeme odhadnúť potrebnú (postačujúcu) veľkosť trénovacej množiny, problémom je, že tieto štatistické vlastnosti v praxi zvyčajne nie sú známe.

Niektoré klasifikačné metódy nepotrebujú trénovaciu množinu na učenie, ide o učenie bez učiteľa (unsupervised learning). Metóda zhuková analýza (Cluster analysis) delí množinu spracovávaných vzorov na podmnožiny (zhluky) na základe vzájomnej podobnosti prvkov podmnožiny. Viac v nasledujúcej kapitole.

2.2.3 Typy klasifikátorov

Support Vector Machines

Support vector klasifikácia je relatívne nový prístup v rozpoznávaní vzorov. Táto technika dosahuje optimálnu klasifikáciu dvoch oddeliteľných tried tým, že maximalizuje šírku medzery (*margin*) medzi týmito dvoma triedami. Šírka medzery je definovaná ako vzdialenosť medzi diskriminačnou nadrovinou v n -dimenzionálnom priestore vektorov čít a najbližšími trénovacími vzormi, tie sa nazývajú *support vectors*. Support vectors špecifikujú diskriminačnú funkciu. Hlavnou výhodou tejto techniky je schopnosť nájsť optimálnu diskriminačnú nadrovinu medzi dvoma triedami v situácii, kedy existuje viacero nadrovin oddeľujúcich tieto triedy. Pôvodná klasifikácia pomocou support vectors bola určená na separáciu lineárne oddeliteľných tried. Toto obmedzenie bolo neskôr prekonané.



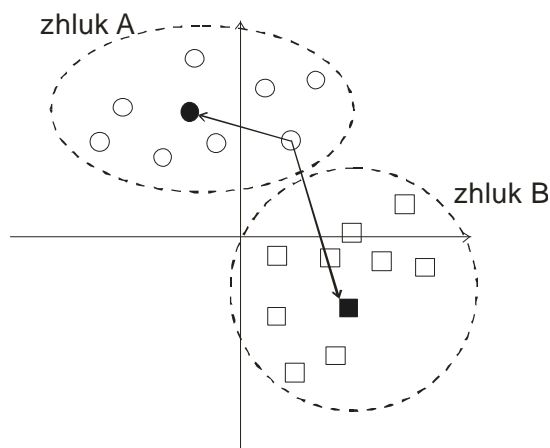
Obrázok 2.4: (A, B) Príklady nie optimálnej diskriminačnej funkcie. (C) Optimálna lineárna diskriminačná funkcia, maximalizuje veľkosť medzery. Prevzaté z [3]

Zhluková analýza

Ako sme už spomínali, existujú metódy klasifikácie, ktoré nepotrebujú trénovaciu množinu na učenie. Zhluková analýza patrí medzi tieto metódy a je možné ju využiť na klasifikáciu práve vtedy, ak z nejakých dôvodov nemáme k dispozícii trénovaciu množinu.

Metódy zhlukovej analýzy rozdeľujeme do dvoch základných skupín, hierarchické a non-hierarchické. Hierarchické metódy zostavujú strom zhlukov tak, že množinu vzorov delia na dve najviac odlišné podmnožiny. Podobne delia vzniknuté podmnožiny. Non-hierarchické metódy sekvenčne pridelujú každý vzor to jedného zhlukov. Non-hierarchická zhluková analýza môže byť s parametrami alebo bez. Bezparametrový prístup je obľúbený, jednoduchý a prakticky použiteľný. Známy príklad metódy non-hierarchickej bezparametrovej zhlukovej analýzy je *MacQueen k-means*.

Predpokladáme, že poznáme počet zhlukov K . Ak nie, tak ho môžeme určiť ako počet tried, ktoré sú najpravdepodobnejšie pre určenie výsledku alebo použijeme inú (komplexnejšiu) metódu, ktorá nepotrebuje túto informáciu. V prvom kroku sú zostrojené štartovacie body zhlukov. Tieto body sú buď zvolené náhodne alebo ako prvé K vzory zo zhlukovej množiny vzorov. Ak máme k dispozícii príklady zhlukov je výhodné ich použiť ako štartovacie body zhlukov. Metóda sa skladá z dvoch základných fáz. V prvej fáze sa vzory priradia k existujúcim zhlukom podľa najkratšej vzdialenosti. Po tomto kroku je znovu určený stred zhluku ako ťažisko všetkých vzorov obsiahnutých v zhluku. V druhej fáze dochádza znovu ku priradovaniu vzorov do novoupravených zhlukov (zmenili sa ich stredy a veľkosti). V druhej fáze nedochádza k opätovnému prepočítavaniu ťažiska zhlukov. Z toho vyplýva, že štartovacie body zvolené na začiatku metódy nemusia ostať v rovnakom zhluku aj na jej konci.



Obrázok 2.5: Čierne body – štartovacie, priradzovanie do zhlukov na základe vzdialenosti, $K = 2$

Neurónové siete

Väčšina prístupov založených na neurónových sieťach pozostáva z elementárnych častí, neurónov. Neuróny majú viacero vstupov a generujú jediný výstup. Každý vstup má pridelenú váhu a výstup sa určí ako funkcia váženej sumy vstupov. Rozpoznávanie vzorov je jedna z mnohých aplikácií neurónových sietí. Neurónové siete ako klasifikátor budeme používať aj v tejto práci, viac kapitola 3.

3 Neurónové siete

Pojem neurónová sieť môžeme chápať z dvoch pohľadov. Z biologického hľadiska ide o sieť nervových buniek (neurónov) v centrálnom alebo periférnom nervovom systéme. Z technického hľadiska ide o výpočtový model zostavený na základe biologického nervového systému. Kvôli odlíšeniu týchto dvoch interpretácií pojmu neurónová sieť sa pre výpočtový model zaviedol pojem Umelá neurónová sieť (ANN – Artificial neural network).

ANN patrí do oblasti umelej inteligencie, skladá sa z navzájom prepojených neurónov, ktoré spolu komunikujú pomocou signálov.

Výhody:

- ANN dokáže spracovať úlohy, ktoré lineárny systém nezvládne
- Ak prvok ANN zlyhá, výpočet môže pokračovať bez problémov kvôli paralelizmu
- ANN sa učí a nepotrebuje byť preprogramovaná
- ANN je možné použiť v ľubovoľnej aplikácii

Nevýhody:

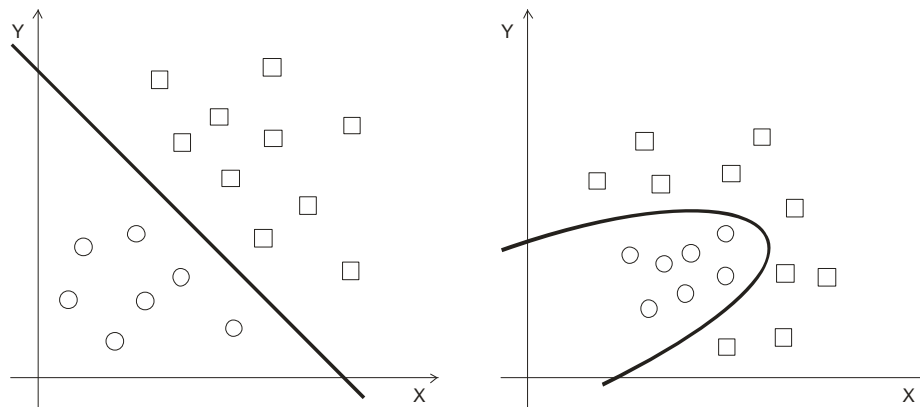
- ANN potrebuje tréning aby mohla pracovať
- Potrebuje veľký výpočtový výkon pre veľké ANN

3.1 Model umelého neurónu

Umelý neurón je základný výpočtový prvok umelých neurónových sietí. Má viacero vstupov (vstupný vektor) ale iba jeden výstup. Funkcia neurónu je transformácia vstupného vektora na výstupnú hodnotu. Vstup získava od iných prvkov siete, prípadne z externého zdroja. Každý vstup má pridelenú synaptickú¹ váhu w , ktorá sa môže meniť počas tréningu siete. Existuje viacero modelov umelého neurónu, my si predstavíme najjednoduchší model, *Perceptrón*.

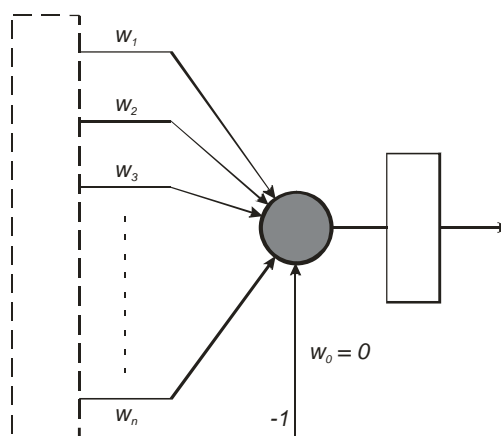
Perceptrón je určený na dichotomickú klasifikáciu, čo je rozdelenie do dvoch tried, pri ktorých sa predpokladá že triedy sú lineárne oddeliteľné v priestore.

¹ synapsia – funkčné spojenie medzi nervovými bunkami



Obrázok 3.1: Dichotomická klasifikácia, vľavo 2 triedy lineárne oddeliteľné v priestore, vpravo lineárne neoddeliteľné

Perceptrón bol po prvý krát predstavený F. Rosenblattom v roku 1958. Obsahuje tri vrstvy: senzorovú, asociatívnu a výstupný neurón.



Obrázok 3.2: Topológia perceptrónu, prevzaté z [6]

Asociatívne neuróny sú určené na stanovenie príznakov vhodných na klasifikáciu a v literatúre sú niekedy označované ako ϕ funkcie. Spojenie medzi senzorovou a asociatívnou vrstvou má hodnoty váh pevne dané, preto sa procesu učenia nezúčastňuje. Na procese učenia sa zúčastňuje spojenie medzi asociatívnou vrstvou a výstupným neurónom, kde sú hodnoty synaptických váh premenlivé. Vstup do výstupného neurónu je daný rovnicou.

$$in(t) = \sum_{j=1}^n w_j(t)x_j(t) - \phi \quad (3.1)$$

Kde n je počet neurónov v asociatívnej vrstve, $w_j(t)$ sú synaptické váhy, x_j je prvok vstupnej množiny X a je ϕ prah.

Výstup prechádza prahovaním a je daný vzťahom.

$$out(t) = \begin{cases} 1 \rightarrow in(t) \geq 0 \\ 0 \rightarrow in(t) < 0 \end{cases} \quad (3.2)$$

Základnú funkciu umelého neurónu popisuje vzťah:

$$y = g(u(x)) \quad (3.3)$$

Kde y je výstup neurónu, x je vstupný vektor, u je bazová funkcia neurónu a g je výstupná funkcia neurónu.

Neuróny rozdelujeme do viacerých kategórií podľa použitej bazovej funkcie. Hlavné typy sú neuróny s lineárnou bazovou funkciou – LBF (definovaná rovnicou 3.4) a neuróny s radiálnou bazovou funkciou – RBF (definovaná rovnicou 3.5) podľa [7].

$$u = \sum_{i=1}^N w_i x_i \quad (3.4)$$

$$u = \|\vec{x} - \vec{w}\| = \sqrt{\sum_{i=1}^N (x_i - w_i)^2} \quad (3.5)$$

Ako sme už spomínali, *perceptrón* delí priestor, tvorený všetkými možnými hodnotami vstupného vektora do dvoch tried. V prípade LBF je deliacim objektom nadrovina, v prípade RBF je deliacim objektom hypersféra.

Výstup umelého neurónu je definovaný výstupnou (aktivačnou) funkciou g . Výstupná funkcia môže byť binárna, vtedy sú jej výstupy prvky z množiny $\{0,1\}$ alebo bipolárna, kedy výstupy funkcie sú prvky z množiny $\{-1,1\}$. V prípade spojitých funkcií sú výstupné hodnoty z intervalu $\langle 0,1 \rangle$ respektíve $\langle -1,1 \rangle$ podľa oboru hodnôt výstupnej funkcie.

Pre neuróny typu LBF sa najčastejšie používa ako výstupná funkcia sigmoida, ktorá má obor hodnôt z intervalu $\langle 0,1 \rangle$ a je definovaná vzťahom:

$$y = \frac{1}{1 + e^{-\lambda u}} \quad (3.6)$$

Prípadne symetrická sigmoida, ktorá má obor hodnôt z intervalu $\langle -1,1 \rangle$:

$$y = \frac{2}{1 + e^{-2\lambda u}} - 1 \quad (3.7)$$

Parameter λ určuje strmosť funkcie. Tento parameter v podstatnej miere ovplyvňuje chovanie neurónovej siete. Neuróny s veľkou strmosťou výstupnej funkcie behom učenia rýchlo konvergujú a ich výstup sa takmer stáva binárny, respektíve bipolárny. V praxi sa takéto neuróny používajú ako klasifikátory, kedy je požadovaná jasná odozva.

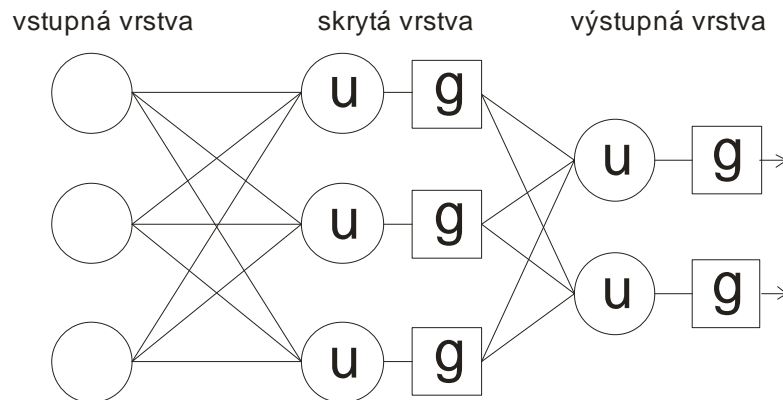
Pre neuróny typu RBF sa najčastejšie používa ako výstupná funkcia gaussova, ktorá má obor hodnôt z intervalu $\langle 0,1 \rangle$ a je definovaná vzťahom:

$$y = e^{-\left(\frac{u}{\sigma}\right)^2} \quad (3.8)$$

3.2 Typy neurónových sietí

Samotný neurón ako výpočtová jednotka neprináša žiadne zlepšenie výpočtu, až spojením jednotlivých neurónov do štruktúry nazývanej neurónová sieť dosiahneme požadované zlepšenie. Tak

ako jeden neurón dokázal rozdeliť priestor do dvoch tried tak neurónová sieť dokáže rozdeliť priestor do viacerých tried. Každá sieť sa skladá z viacerých vrstiev, ktoré môžeme rozdeliť na tri podkategórie. Vstupná vrstva, ktorá sa niekedy kvôli svojej jednoduchosti v odbornej literatúre neoznačuje ani ako vrstva. Niekoľko skrytých vrstiev, ktoré nie sú pre užívateľa viditeľné a výstupná vrstva, ktorá slúži na získanie výsledku.



Obrázok 3.3: Neurónová sieť

Neurónové siete môžeme klasifikovať podľa rôznych hľadísk, napr. podľa stupňa prepojenia, smeru prepojenia atď. Z hľadiska prepojenia neurónov v sieti rozlišujeme doprednú neurónovú sieť (feed-forward neural network), v ktorej je spojenie realizované tak, že výstupy neurónov v jednej vrstve sú privedené na vstupy neurónov nasledujúcej vrstvy alebo rekurentnú neurónovú sieť (recurrent neural network), ktorá získava spätnú väzbu.

Pre každý typ neurónovej siete existuje špecifická oblasť použitia a odpovedajúci algoritmus učenia. Na rozpoznávacie úlohy sa často používa sieť typu viacvrstvový perceptrón (MLP–Multi Layered Perceptron). Táto sieť je umelou analógiou štruktúry, v ktorej sú usporiadané nervové bunky spracovávajúce obraz z ľudského oka. Patrí do kategórie doprednej neurónovej siete.

Parametrom spojenia dvoch neurónov je synaptická váha. Privedením vstupov na vstupnú vrstvu neurónovej siete dochádza ku vygenerovaniu vzruchov (signálov). Tieto sú prostredníctvom váh medzi neurónmi buď zosilnené, alebo tlmené. Jednotlivé vzruchy prechádzajú celou sieťou, sú transformované neurónmi v jednotlivých vrstvách až končia vo vrstve výstupnej. Odozva nenaučenej neurónovej siete je náhodná.

3.3 Učenie neurónovej siete

Učenie (trénovanie) neurónovej siete je proces, pri ktorom si neurónová sieť mení hodnoty váh na základe daného tréningového pravidla. Existujú tri hlavné prístupy k učeniu neurónových sietí. Kontrolované učenie (supervised learning), pri ktorom je výstup siete porovnávaný s očakávaným výsledkom. Nekontrolované učenie (unsupervised learning), kedy sieti nie je dodaný očakávaný

výsledok, ale sieť musí sama, na základe štatistických charakteristických vlastností, objaviť výstupné kategórie. A nakoniec posilnené učenie (reinforcement learning), ktoré stojí medzi hore uvedenými dvoma prístupmi.

Častou metódou učenia umelých neurónových sietí je spätné šírenie chyby (backpropagation), ktorá patrí do kategórie kontrolovaného učenia. Uvedieme si popis algoritmu z hľadiska jeho hlavného princípu. Viac o algoritme backpropagation nájdeme v práci [8]:

1. Vyber vzor z trénovacej množiny a daj ho na vstup siete
2. Vypočítaj výstup siete a porovnaj ho s očakávaným výstupom
3. Na základe porovnania uprav hodnoty váh každého neurónu
4. Vráť sa ku kroku 1, kým nevyčerpáš všetky vzory trénovacej množiny
5. Opakuj kým je celková chyba menšia ako požadovaná chyba

Matematickým vyjadrením tohto postupu je minimalizácia hodnoty energetickej funkcie, ktorá je definovaná vzťahom:

$$E = \frac{1}{2} \sum_{i=1}^N (y_i - d_i)^2$$

Kde y je vstup a d je výstup siete.

4 Metódy detekcie tvári

Ako bolo uvedené v predchádzajúcej časti, digitálny obraz sa rôznymi metódami predspracováva. Je to kvôli tomu, že dáta z viacerých obrazoch toho istého objektu sú variabilné. Na rozdiel od klasických vstupov, kde získané hodnoty sú jednoznačné, napríklad vstup z klávesnice. Preto pred samotnou klasifikáciou obrazu je potrebné obraz upraviť aby sa jednotlivé obrazy od seba príliš nelíšili. Predspracovanie sa skladá z:

- **ekvalizácie histogramu**
- **zmeny rozmeru (presamplovanie)**
- **normalizácie**
- **filtrácie obrazu:** k detekcii je použitá iba určitá časť frekvenčného spektra obrazu

Ako som spomenul v úvode, v tejto práci sa zameriam na detekciu ľudských tvári, ktoré do tejto problematiky vnášajú ešte viac rozmanitosti. Čiže okrem rozdielov spôsobených rôznymi svetelnými podmienkami pri vytvorení obrazu, musíme zobrať do úvahy ešte veľkosť, pozíciu, orientáciu (vzpriamená, natočená) a postoj (spredu, z profilu) tváre. Ďalej rozmanité výrazy tváre (úsmev, žmurknutie, grimasa, atd.), všeobecné odlišnosti ako fúzy, briadka, okuliare.

Tieto jednotlivé možnosti výzoru tváre je možné prehľadne zatriediť do kategórií, toto zatriedenie je prevzaté z [4].

- **postoj:** Obraz tváre môže byť voči kamere natočený vo všetkých troch osiach (spredu, 45 stupňov, z profilu, hore nohami) a niektoré typické črty tváre ako je oko alebo nos môžu byť čiastočne alebo úplne neviditeľné, čo robí najväčšie problémy niektorým metódam detekcie tvári v obraze.
- **variabilné rysy tváre:** Ide o črty tváre ako je briadka, fúzy a okuliare, ktoré sa na tvári vyskytujú alebo nie. Navyše je pri týchto črtách veľká možnosť variability ako je ich tvar, veľkosť či farba.
- **výrazy tváre:** Vzhľad tváre je priamo ovplyvnený každou ľudskou emóciou, ľudská mimika je veľmi bohatá na výrazy.
- **zakrytie:** Tváre môžu byť čiastočne zakryté inými objektmi. V prípade skupinových fotiek môže dôjsť k zakrytiu tváre inou tvárou.
- **orientácia obrazu:** Tvár môže byť snímaná z rôznych uhlov (zhora, zdola, zboku, atd.). Tváre na obraze vyzerajú podstatne rovnako ako v prvej kategórii (postoj).
- **okolnosti obrazu:** Pri vytvorení fotky, faktory ako svetelnosť (spektrum, umiestnenie zdroja a intenzita) a vlastnosti kamery (senzor, odozva, šošovky) ovplyvňujú vzhľad tváre.

Je mnoho faktorov, ktoré ovplyvňujú spracovávaný obraz. Preto je pred návrhom systému dôležité preskúmať všetky možnosti, ktoré môžu nastať. Vybrať si cieľ, čo chceme detekovať

(vzpriamené tváre, aj natočené tváre, jednotlivé časti tváre) a podľa toho zvoliť vhodnú metódu pre detekciu. V tejto práci sa zameriame na detekciu úplných vzpriamených tvári pomocou neurónových sietí.

Dôležité je odlíšiť dva pojmy, ktoré sa budú v tejto ale aj v iných prácach zameraných na spracovanie obrazu vyskytovať. Sú to detekcia a lokalizácia. Cieľom detekcie je určenie či sa hľadaný objekt, v našom prípade ľudská tvár, nachádza alebo nenachádza v spracovávanom obraze. Lokalizácia prichádza na rad až po detekcii, kedy hodnotnejšie prezentuje výsledok detekcie. Určí presnú lokalizáciu nájdenej tváre, prípadne tvári v obraze.

Existuje mnoho prác, ktoré sa venujú detekcií ľudských tvári. Každá je niečím výnimočná no zároveň preberá niečo z iných prác. Postupom času sa vyčlenili spoločné vlastnosti prác, podobné rysy, či prístupy ku detekcii a tak môžeme jednotlivé metódy detekcie tvári rozdeliť do štyroch základných skupín podľa [4].

- **Metódy založené na znalostiach (Knowledge-based methods):** Tieto metódy kódujú ľudské znalosti o tom, aké elementy typicky obsahuje tvár. Často vystihujú vzťah medzi črtami tváre. Tieto metódy sú vhodné hlavne pre lokalizáciu tvári.
- **Príznakové invariantné metódy (Feature invariant approaches):** Tieto algoritmy sú namierené na nájdenie štruktúrnych črt a podľa nich určia polohu tváre. Črty existujú v obraze pri rôznych postojoch, uhloch pohľadu a svetelných podmienkach. Tieto metódy sú vhodné hlavne pre lokalizáciu tvári.
- **Šablónové porovnávanie (Template matching methods):** Niekoľko štandardných vzorov tváre je použitých na popis tváre ako celku alebo na popis jednotlivých črt tváre. Detekcia je vypočítaná ako korelácia medzi vstupným obrazom a štandardnými vzormi. Tieto metódy sa používajú na detekciu a lokalizáciu tvári
- **Metódy založené na vzhľade (Appearance-based methods):** Na rozdiel od šablónového porovnávania sú vzory naučené zo súboru tréningových obrázkov, ktorý by mal obsahovať reprezentatívnu vzorku tvári. Tieto natrénované modely sa potom použijú na detekciu. Tieto metódy sa uplatňujú hlavne na detekciu.

Veľký záujem o problematiku, mnoho práci, mnoho riešiteľov, viaceré metódy riešenia, to všetko podnecovalo k porovnávaniu výsledkov jednotlivých výskumov a súpereniu. Najmä v kvôli jednotnosti v oblasti detekcie tvári, sa zaviedli pojmy, ktoré sa používajú v hodnotení úspešnosti jednotlivých metód. Prvý takýto pojem je *miera detekcie (detection rate)* definovaná ako pomer medzi počtom správne detekovaných tvári ku počtu celkových tvári v obraze. Vo všeobecnosti pri detekcii nastávajú dve typy chýb. *Počet chybné detekovaných tvári ako netváre (false negatives)*, kedy nie sú detekované všetky tváre na obrázku. Druhá chyba je *počet chybné detekovaných netvári ako tváre (false positives)*, keď nastane to, že dáta, ktoré tvármi nie sú algoritmus označí ako tváre.

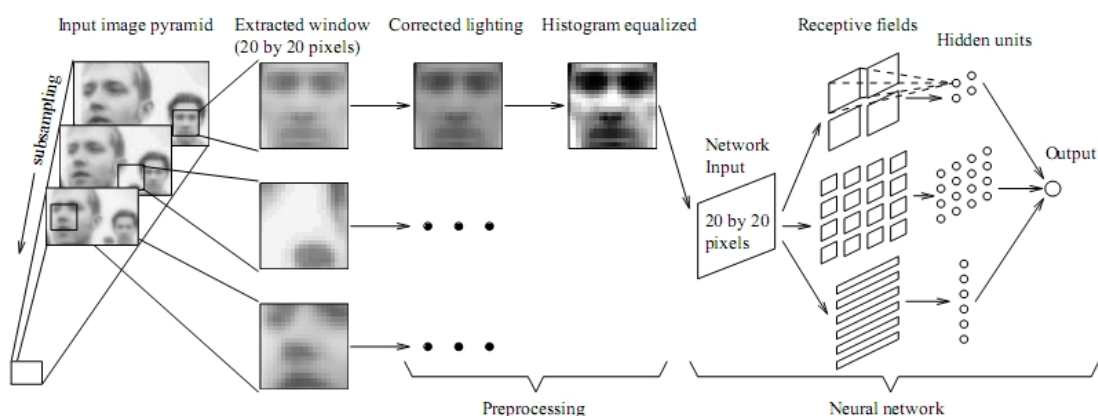
4.1 Metódy založené na vzhľade

Metódy založené na vzhľade sú, čo sa týka zástupcov, jednou z najpočetnejších skupín metód detekcií tvári. Do tejto kategórie patria metódy Eigenfaces, Neurónové siete, Support Vector Machines a iné. Tieto metódy sa spoliehajú na postupy zo štatistickej analýzy a strojového učenia aby našli relevantné charakteristiky obrázkov s tvárou (face) a bez nej (nonface).

Metódy založené na vzhľade môžu byť považované za pravdepodobnostné systémy. So vstupnými dátami, čo sú celé obrazy alebo ich časti, pracujú ako s náhodnou veličinou. Na klasifikovanie, či ide o tvár alebo netvár môže byť použitý Bayesovského klasifikátor alebo najväčšia pravdepodobnosť. Ďalší prístup, ktorý sa používa v metódach založených na vzhľade je nájdenie diskriminačnej funkcie. Tá dokáže oddeliť triedu tvárových obrazov od netvárových.

V tejto práci sme si zvolili metódu práve z tejto kategórie, na detekciu tvári používame neurónové siete. Neurónové siete si najprv predstavíme ako klasifikátor tvári, povieme si o princípe ich použitia a v 3. časti sa zoznámime s teóriou ohľadom neurónových sietí.

Neurónové siete boli úspešne použité pri problémoch s rozpoznávaním vzorov. Ich výhodou je realizovateľnosť tréningu systému na zachytenie komplexných vlastností tvárových vzorov. Avšak majú aj jeden závažný nedostatok, architektúra neurónovej siete musí byť značne vyladená (nastavenie mnohých parametrov ako počet vrstiev, počet prepojení a iné, viac v 3. časti) aby podávala výnimočné výsledky. Po natrénovaní neurónovej siete prebieha detekcia tak, že na vstupy neurónovej siete sú privedené dáta zo spracovaného obrazu a na výstupe neurónovej siete je hodnota, ktorá určuje či je daný vstup tvár alebo nie. Inšpirovala ma úspešná práca autorov Rowley, Baluja, Kanade [5], z ktorej je prevzatý **Obrázok 4.1**. Popisuje základný algoritmus použitý na detekciu tvári.



Obrázok 4.1: Detekcia tvári pomocou neurónovej siete

5 Návrh aplikácie

5.1 Motivácia

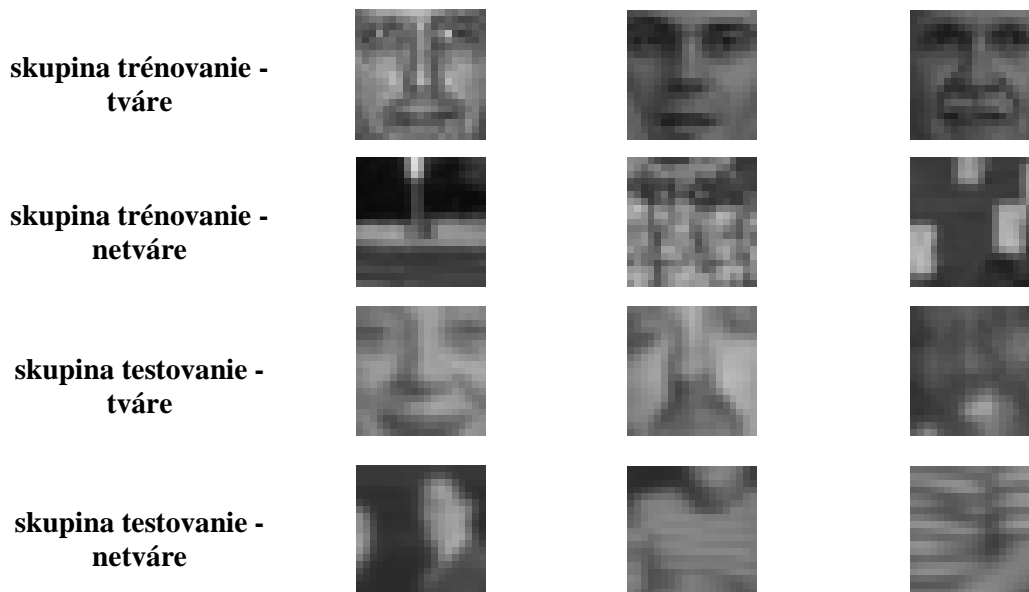
Téma tejto práce je Detekcia objektov v obraze. Táto téma je široká, pokrýva problematiku detekcie objektov v statickom ale aj dynamickom obraze s možnosťou detekcie ľubovoľného objektu (prípadne objektov). Preto bolo potrebné sa rozhodnúť a vybrať si jednu oblasť výskumu. Po konzultácii s vedúcim práce sme sa rozhodli pre detekciu objektov v statickom obraze. Nasledoval výber objektu detekcie. Vzhľadom na popularnosť detekcie tvári v dnešnej dobe, dostupnosti databáz obrázkov obsahujúcich tváre a množstva prác s významnými výsledkami na porovnanie úspešnosti systému bolo rozhodnutie jednoznačné. Téma tejto práce bola špecifikovaná ako Detekcia tvári v statickom obraze.

5.2 Trénovanie neurónovej siete

Návrh systému sa skladá z viacerých krokov. Prvým z nich je návrh vytvorenia a trénovania neurónovej siete, nevyhnutnej k následnému vývoju aplikácie. Ako som spomínal v kapitole 4.1, architektúra neurónovej siete má mnoho nastaviteľných parametrov. Tento systém bude používať plne prepojenú doprednú neurónovú sieť typu Back Propagation (algoritmus učenia siete so spätným šírením chyby, popísaný v kapitole 3.3). Sieť sa skladá z troch vrstiev (vstupná, skrytá, výstupná), ktorých počet je pevne daný. Jediný parameter, ktorý môže užívateľ meniť je počet neurónov v skrytej vrstve. Ďalším nemenným nastavením z pohľadu užívateľa je voľba bázevej (aktivačnej) a výstupnej funkcie neurónov. Pre detekciu tvári podľa [9] dáva najlepšie výsledky nasledujúce nastavenie. Ako aktivačná funkcia neurónov je zvolená sigmoida a ako výstupná funkcia je zvolená gaussova funkcia.

Následne je potrebné zvoliť databázu tvárových a netvárových obrázkov. Na základe odporúčania vedúceho bakalárskej práce sme zvolili databázu CBCL (Center for Biological and computational learning)². Obsahuje dve skupiny obrázkov na trénovanie a testovanie, obrázky sú uložené vo formáte PGM (Portable Gray Map) o rozmeroch 19×19 pixlov. Skupina určená na trénovanie neurónovej siete obsahuje 2 429 tvári (face) a 4548 netvári (nonface). Skupina určená na testovanie vytvorenej neurónovej siete sa skladá zo 472 tvári a 23 573 netvári.

² Databáza CBCL je dostupná na adrese <http://cbcl.mit.edu/software-datasets/FaceData2.html>



Obrázok 5.1: Ukážky dát z databázy CBCL

Celkovo databáza obsahuje cez 31 tisíc obrazových dát. Táto databáza nám značne uľahčila prácu, keby sme si zvolili iný objekt detekcie tak len zostavenie podobnej databázy by zabralo veľké množstvo času a úsilia.

Pred natrénovaním neurónovej siete je potrebné vstupné dáta spracovať. Dosiahneme tým zmenšenie variability obrázkov v databáze. Pre každý obrázok z databázy sa vypočíta histogram, ktorý sa následne ekvalizuje. Nové hodnoty obrázkov sa prevedú (normalizujú) do intervalu $\langle 0,1 \rangle$. Až takto upravené obrázky budú posielané na vstup neurónovej siete.



Obrázok 5.2: Ukážka ekvalizovaných obrázkov z Obrázok 5.1

Výsledkom tejto fázy bude úspešne natrénovaná neurónová sieť.

5.3 Testovanie neurónovej siete

Po úspešnom natrénovaní neurónovej siete je potrebné overiť ako sa táto sieť správa v úlohe detektora ľudských tvári. Zdôrazním, že v tejto fáze ide o detektor a nie o lokalizátor. Na tento účel nám posluží taktiež databáza CBCL popísaná v predchádzajúcej časti, konkrétne skupina obrázkov určených na testovanie.

Neurónová sieť použitá na klasifikáciu bude mať jeden neurón vo výstupnej vrstve, to znamená, že na výstupe dostaneme jedno číslo, ktoré bude určovať či vstup privedený na neurónovú sieť je alebo nie je tvár. Zvolíme si jednotnú reprezentáciu, číslo 1 na výstupe bude znamenať, že ide o tvár. Naopak číslo 0 symbolizuje, že vstupom nie je tvár. Túto reprezentáciu volíme vzhľadom na zvolenú aktivačnú a výstupnú funkciu. Obor hodnôt oboch týchto funkcií je z intervalu $\langle 0,1 \rangle$,

popísané v kapitole 3.1. Z toho vyplýva, že výstupné hodnoty neurónovej siete sú reálne čísla. To je dôvodom zavedenia prahu (threshold), ktorý má jednoduchú rozhodovaciu funkciu. Všetky výstupy neurónovej siete, ktoré majú väčšiu hodnotu ako zvolený prah, sú označené ako tváre. Ostatné výstupy sú označené ako netváre.

V tejto fáze overíme funkčnosť natrénovanej neurónovej siete ako detektora tvári. Na vstup siete dáme postupne všetky obrázky z testovacej skupiny a na výstupe získame číselnú reprezentáciu klasifikácie. Z výstupu spočítame chybu siete:

$$chyba = \sum_{i=1}^N (ocakavana_hodnota - vystup_siete)^2$$

Kde N je počet obrázkov v testovacej skupine, *ocakavana_hodnota* je 1,0 pre tváre a 0,0 pre netváre.

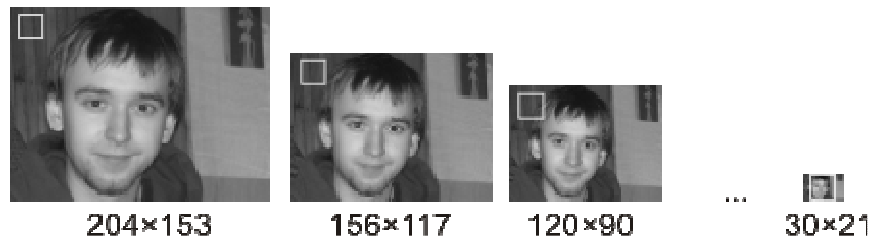
Ak sieť v tomto teste zlyhá, nebude schopná správne detekovať tváre (chyba-*false negatives*), prípadne bude detekovať netváre ako tváre (chyba-*false positives*) je potrebné vrátiť sa ku predchádzajúcemu kroku a natrénovať novú sieť. Ak sieť v teste uspeje je vyhovujúca na použitie v najnáročnejšej časti tejto práce, lokalizátore tvári.

5.4 Lokalizácia tvári pomocou neurónovej siete

Posledným krokom pri návrhu systému je použitie natrénovanej neurónovej siete na lokalizáciu tvári v obraze. Sieť v tejto práci je učená na obrázkoch o rozmeroch 19×19 pixlov, preto aj pri spracovaní detekovaného obrazu je potrebné vybrať z obrazu časti (ďalej nazývané okná) s rozmermi 19×19 pixlov. Nato aby sme mohli detekovať a lokalizovať tváre rôznych rozmerov v obraze (nie len tie, ktoré sa rozmermi zmestia do okna), je potrebné spracovaný obraz zmenšovať. Zmenšovanie prebieha delením koeficientom 1,3 pokiaľ sú rozmery (šírka a zároveň výška) spracovaného obrazu väčšie ako okno. Algoritmus detekovania celej plochy obrázka v rôznych veľkostiach popisuje nasledujúci pseudokód.

1. pokračuj pokiaľ sú rozmery obrázka väčšie alebo rovné ako rozmer okna (19×19)
2. načítaj plochu obrázku o rozmeroch okna
3. ekvalizuj a klasifikuj neurónovou sieťou
4. ak je to tvár tak zapamätaj polohu
5. pokračuj na krok 2. kým neprejdeš s oknom celý obrázok
6. zmenší obrázok delením 1,3 a choď na krok 1.

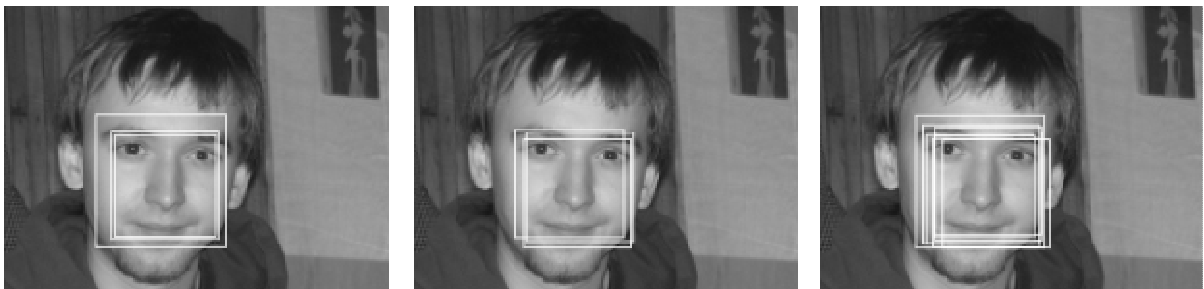
Tento proces znázorňuje obrázok **Obrázok 5.3**, na obrázku si môžeme všimnúť ako biely štvorec označujúci okno (19×19 pixlov) postupne zaberá väčšiu časť až v poslednom kroku zaberá takmer celý obrázok. K ďalšiemu zmenšeniu nedôjde lebo obrázok by mal rozmery 23×16 pixlov, čo nespĺňa podmienku o rozmeroch obrazu, tie musia byť obe väčšie ako rozmery okna. Pri zmenšovaní rozmerov spracovaného obrazu sú hodnoty intenzít nových bodov vypočítané pomocou bikubickej interpolácie.



Obrázok 5.3: Zmenšovanie rozmeru obrázka

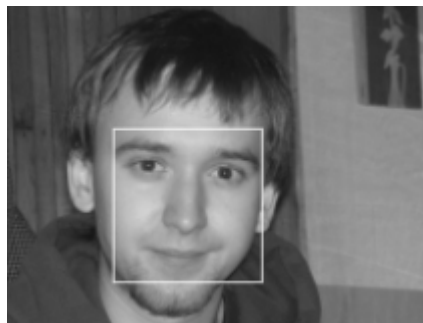
Pre každé okno obrázku sa prevedie výpočet histogramu, ekvalizácia a normalizácia. Presne tak ako to bolo aj pri tréovaní a testovaní neurónovej siete. Následne sa okno dá na vstup neurónovej siete, ktorá určí či ide o tvár alebo nie. Všetky pozitívne detekcie si systém pamätá spolu s ich pozíciou v obraze, tým sa jeho úloha mení na lokalizátor.

Vzhľadom na veľký rozdiel medzi rozmermi spracovaného obrazu a rozmermi okna bude neurónová sieť klasifikovať značné množstvo vstupných dát, ku ktorým sa pridajú ďalšie vstupné dáta vzniknuté zmenšovaním obrazu. Objavuje sa problém násobných detekcií. Ide o situáciu kedy je jedná tvár na obraze detekovaná viac násobne. Môže ísť o zanorené násobné detekcie, posunuté prípadne kombinované násobné detekcie.



Obrázok 5.4: Ukážky násobných detekcií, vľavo zanorené, v strede posunuté a vpravo kombinované

Problém násobných detekcií respektíve lokalizácií je možné vyriešiť viacerými spôsobmi napríklad využitím základných matematických operácií ako sú prienik a zjednotenie. V tomto systéme budeme zlučovať okna násobnej detekcie na princípe zjednotenia. Z zjednoteniu dvoch okien dôjde vtedy, ak plocha vymedzená ich prienikom tvorí minimálne 80% celkovej plochy aspoň jedného z okien. Pri zjednotení, na rozdiel od metódy prieniku, dochádza k zväčšeniu rozmerov okna.



Obrázok 5.5: Ukážka odstránenia násobných detekcií z Obrázok 5.4 metódou zjednotenia

6 Implementácia

Na implementáciu navrhnutého systému bol zvolený jazyk C++. Systém bol vyvíjaný a odladený na operačnom systéme Microsoft Windows XP s 32 bitovou architektúrou. Pôvodne mal byť systém implementovaný aj pre operačný systém Linux, no po dohode s vedúcim práce bolo od tohto zámeru upustené.

Systém používa dve knižnice. Fast Artificial Neural Network Library (FANN)³, na prácu s neurónovými sieťami a Cool Image Library (CImg)⁴, na prácu s obrazovými dátami. Ako vyplýva z návrhu aplikácie, systém tvoria 3 moduly: trénovací (train), testovací (test) a lokalizačný (main). Navyše je implementovaný hlavičkový súbor `functions.h`, ktorý obsahuje funkcie zdieľané jednotlivými modulmi (napr. `equalization()`). Aplikácia nemá grafické užívateľské rozhranie, je ovládaná z konzoly. Nasledujúcich podkapitolách sa budeme venovať použitým knižniciam a implementovaným modulom.

6.1 Fast Artificial Neural Network Library

Ide o Open Source knižnicu na prácu s neurónovými sieťami. Jej autorom je Steffan Niessen a je implementovaná v jazyku C. Šírená je pod licenciou GNU Lesser General Public License (LGPL)⁵. V dobe vývoja systému bola dostupná verzia 2.1.0beta. Táto knižnica bola vybraná na základe odporúčania vedúceho práce.

Práca s knižnicou FANN je jednoduchá. Po skompilovaní a prilinkovaní knižnice stačí vložiť hlavičkový súbor `fann.h` do zdrojového textu, v ktorom chceme používať naprogramované funkcie tejto knižnice. V našom systéme používame FANN vo všetkých troch moduloch, lebo neurónová sieť je kostrou celého projektu.

Proces vytvorenia siete (modul train) obsahuje volanie funkcií `fann_create_standard`, `fann_train_on_data` a `fann_save`, ktoré ako je zrejmé z ich názvov, vytvoria sieť, natrénujú ju na dátach a uložia.

Moduly test a main používajú rovnaké funkcie knižnice FANN `fann_create_from_file` a `fann_run`. Tie načítajú uloženú neurónovú sieť a použijú ju na klasifikáciu.

³ FANN je dostupná na adrese <http://leenissen.dk/fann/>

⁴ CImg je dostupná na adrese <http://cimg.sourceforge.net/>

⁵ Dostupná na adrese <http://www.gnu.org/copyleft/lesser.html>

6.2 Cool Image Library

Knižnica CImg, ako je označovaná častejšie, je Open Source knižnica určená na prácu s obrázkami. Jej autorom je David Tschumperlé a implementovaná je v jazyku C++. Distribuovaná je pod licenciou CeCILL-C⁶, ktorá je podobná licencií GNU LGPL. V dobe vývoja systému bola dostupná verzia 1.3.0. Táto knižnica bola taktiež vybraná na základe odporúčania vedúceho práce.

Veľkou výhodou tejto knižnice je to, že sa nemusí kompilovať a linkovať k projektu, čo sú pri niektorých knižniciach priam nadľudské výkony. Jediné, čo je nutné pred používaním funkcií z knižnice CImg spraviť je vložiť hlavičkový súbor *CImg.h* do zdrojového textu. Táto knižnica je v našom systéme používaná v module main.

V module main najprv načítame požadovaný obrázok konštruktorom *CImg<datovy_typ> image(nazov_obrazku)* a následne k nemu môžeme pristupovať ako k dvojrozmernému poľu, *image(x,y)*. Do spracovaného obrázku kreslíme nájdené lokalizácie tvári pomocou metódy *draw_line()*, zobrazujeme výsledok *main_disp(image,"Title")* prípadne ukladáme pomocou metódy *save()*.

6.3 Modul Train

Modul train implementuje vytvorenie, trénovanie a uloženie neurónovej siete. Reaguje na tri parametre, ktorých poradie je pevne dané.

1. Názov neurónovej siete
2. Počet neurónov v skrytej vrstve
3. Prepínač identifikujúci zapnutie či vypnutie ekvalizácie

Príklady korektného spustenia:

- `train`
- `train network.net 30 on`
- `train network2.net 60 off`

Modul je možné spustiť aj s neúplným zadáním parametrov, ktoré je možné vynechávať postupne sprava. Nezadané parametre sa implicitne nastavujú takto:

1. `network.net`
2. `30`
3. `on`

Pri spustení modulu s prepínačom `-h`, sa vypíše návod na použitie.

⁶ Dostupná na adrese http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.txt

6.4 Modul Test

Modul test je určený na testovanie neurónovej siete. Používa sa s tromi parametrami, ktorých poradie je nemenné.

1. Názov neurónovej siete
2. Prah
3. Zapnutie či vypnutie ekvalizácie

Príklady korektného spustenia:

- `test`
- `test network.net 0.6 on`
- `test network2.net 0.9 off`

Ak je modul spustený bez parametrov, použije sa implicitné nastavenie:

1. `network.net`
2. `0.5`
3. `on`

Modul reaguje na prepínač `-h` vypísaním návodu na použitie.

6.5 Modul Main

Posledný model main plní úlohu lokalizátora tvári v obraze. Reaguje na štyri parametre, ktorých poradie je pevne dané:

1. Názov neurónovej siete
2. Názov spracovaného obrázka
3. Prah
4. Zapnutie či vypnutie ekvalizácie

Príklady korektného spustenia:

- `main network2.net tom.pgm 0.5`
- `main network.net mona.pgm 0.9 on`
- `main network.net soccer.pgm 0.85 off`

Tento modul na rozdiel od predchádzajúcich, nie je možné spustiť bez parametrov. Jediný nepovinný parameter je nastavenie ekvalizácie. Implicitne je ekvalizácia zapnutá, jedine hodnota štvrtého parametru `off` ju vypne. Oproti predchádzajúcim modulom sa líši aj v tom, že má grafický výstup (dve okná Main a Merge). Okno Main reaguje na klávesy `Q`, `Esc` (zatvorenie aplikácie) a `S` uloženie výsledku lokalizácie do dvoch obrázkov. Spoločné s predchádzajúcimi modulmi má totožné reagovanie na prepínač `-h`, ktorý vypíše na štandardný výstup návod na použitie.

7 Experimentálne výsledky

Táto kapitola je aspoň pre mňa, najzaujímavejšou kapitolou celej práce. Testovali sme učenie neurónovej siete, testovali sme detekčné schopnosti siete pri rôznych podmienkach, testovali sme lokalizačné schopnosti siete. To všetko nájdete v tejto časti prehľadne reprezentované grafmi a ukážkami úspešných aj neúspešných lokalizácií tvári.

7.1 Neurónová sieť ako detektor

Učenie neurónovej siete je proces, pri ktorom sú menené synaptické váhy spojení jednotlivých neurónov v sieti. Deje sa na základe tréningu siete na databáze objektov. Proces učenia siete ovplyvňujú rôzne faktory, tie boli skúmané v tejto kapitole.

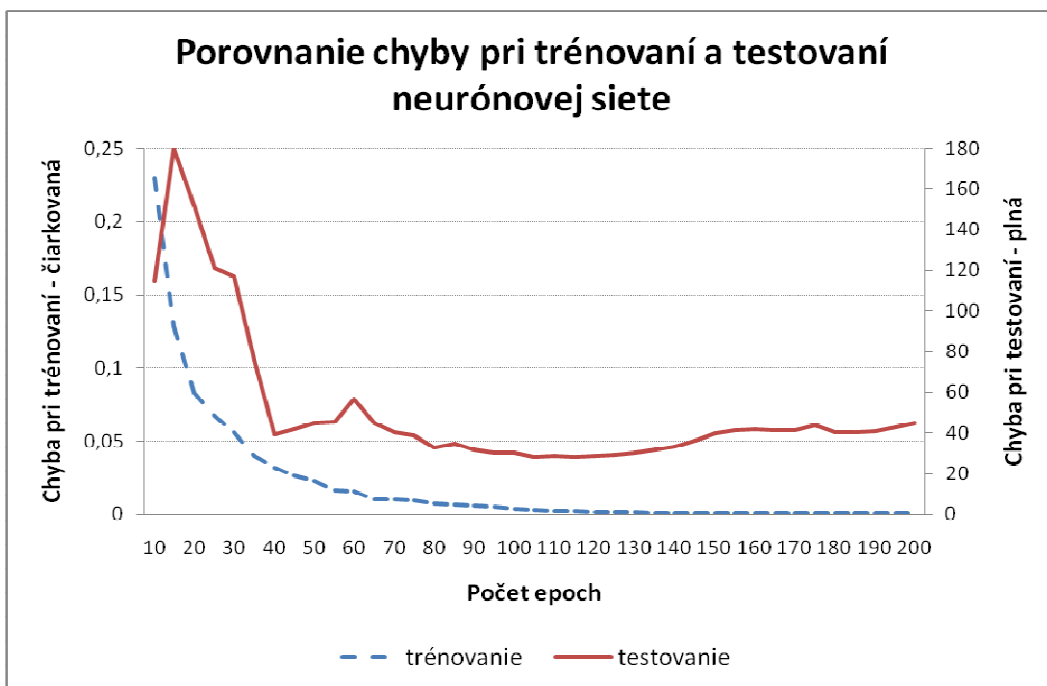
7.1.1 Pribeh chyby pri učení

Počas tréningu siete sa chyba siete na tréningovej množine stále znižuje. Dochádza ku zakolísaniam, kedy po jednej epoche chyba vzrastie ale následne klesne, čiže z celkového pohľadu chyba klesá. To neplatí pre chybu na testovacej množine, ktorá po určitom počte epoch začne stúpať. Deje sa to kvôli tomu, že sieť stráca generalizačnú schopnosť. To znamená, že dokáže poznávať len tváre z tréningovej množiny. Nasledujúci experiment z časti potvrdzuje naše predpoklady.

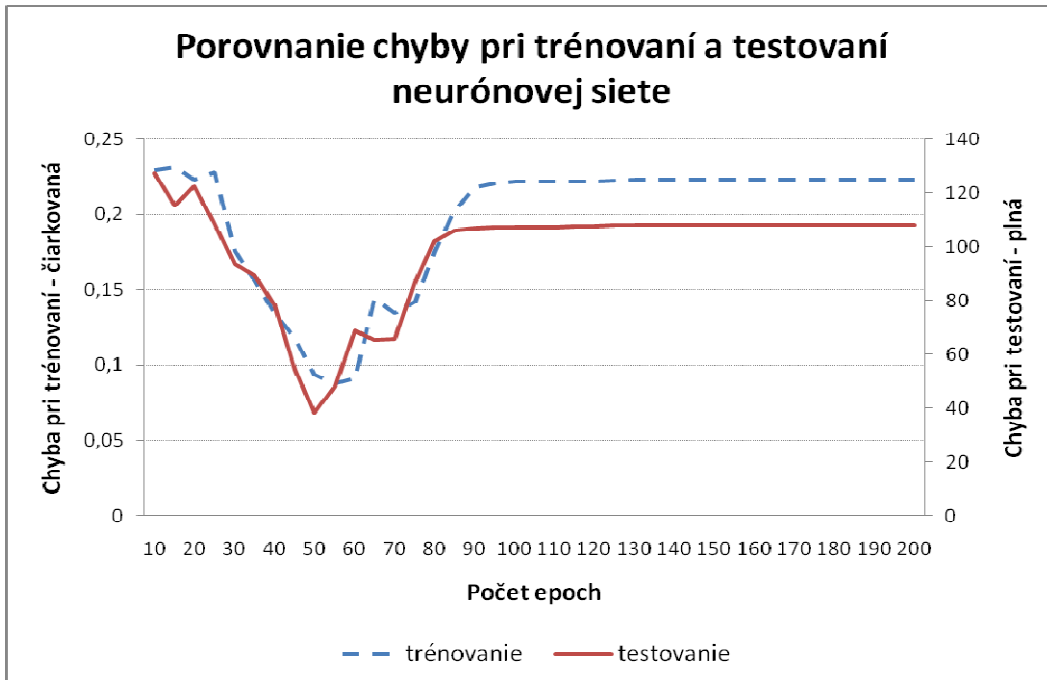
Popis experimentu:

- zvolíme si, ktoré nastavenia siete (procesu tréningu) chceme skúmať
 1. veľkosť tréningovej množiny
 2. zapnutá a vypnutá ekvalizácia
- opakujeme meranie pre každé možné nastavenie
- výsledky zobrazíme v podobe grafu

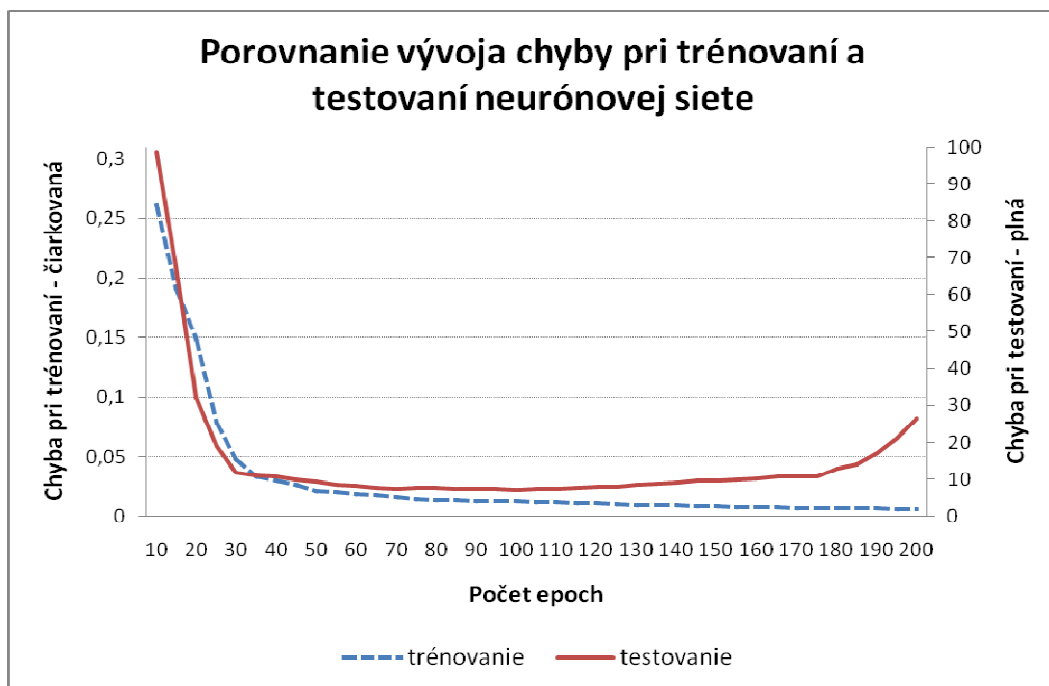
Zvolili sme dve veľkosti tréningovej množiny. Prvá obsahovala 500 tvári a 1000 netvári (*malá*), druhá sa skladala z 2174 tvári a 4348 netvári (*veľká*). Celkový počet kombinácií menených nastavení je štyri, *malá* so zapnutou ekvalizáciou, *malá* s vypnutou ekvalizáciou, *veľká* so zapnutou ekvalizáciou a *veľká* s vypnutou ekvalizáciou. Neurónová sieť obsahovala 30 neurónov v skrytej vrstve.



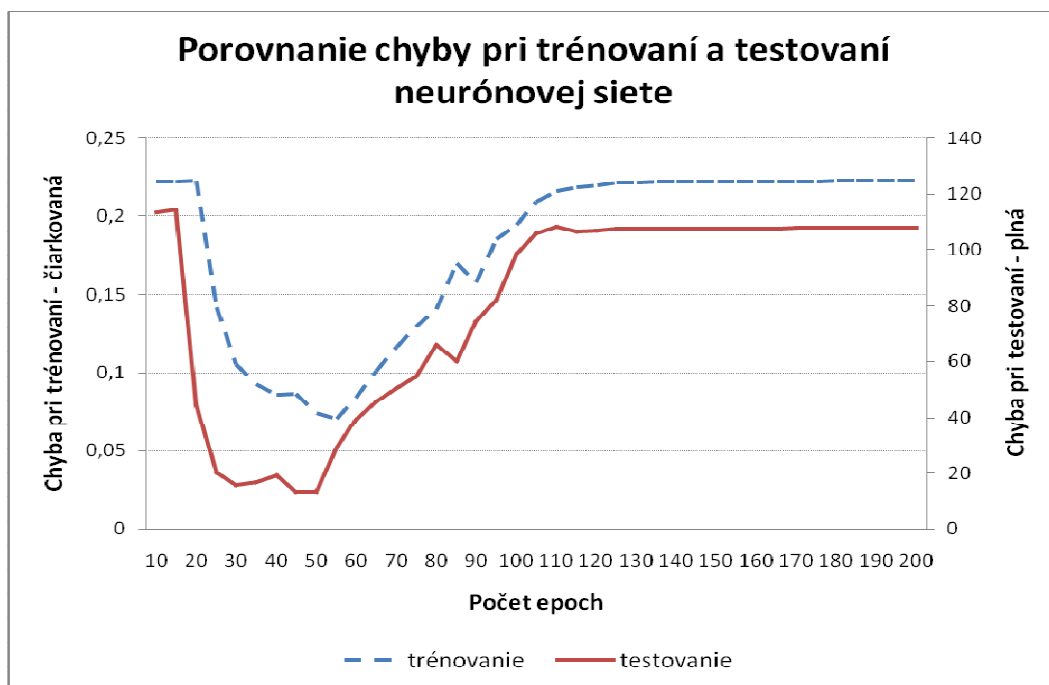
Obrázok 7.1: Malá trénovacia množina so zapnutou ekvalizáciou. Minimálna chyba pri testovaní v 105. Epoche



Obrázok 7.2: Malá trénovacia množina s vypnutou ekvalizáciou. Minimálna chyba pri testovaní v 50. epoche



Obrázok 7.3: Veľká trénovacia množina so zapnutou ekvalizáciou. Minimálna chyba pri testovaní v 100. Epoche



Obrázok 7.4: Veľká trénovacia množina s vypnutou ekvalizáciou. Minimálna chyba pri trénovaní v 50. Epoche

Merania so zapnutou ekvalizáciou ukázali rovnakých vývoj chyby pri trénovaní. Chyba na trénovacej množine začínala okolo hodnoty 0,25 (25%) a postupne klesala až na hodnoty menšie ako 0,005 (0,5%). Jej vývoj nebol významne ovplyvnený veľkosťou množiny.

Pri meraní s vypnutou ekvalizáciou došlo k nepredpokladanému javu. Chyba na trénovacej množine po istom čase začala narastať a ustálila sa na hodnotách okolo 0,22 (22%). Minimálna hodnota chyby bola oproti meraniu s ekvalizáciou niekoľko násobne vyššia. Neklesla pod hodnotu 0,05 (5%).

Vývoj chyby pri testovaní ukazuje väčšiu variabilitu. Pri použití *veľkej* množiny dosahuje chyba pri trénovaní minimálne hodnoty okolo 7 s ekvalizáciou a 13 bez ekvalizácie, pri *malej* množine minimálne hodnoty okolo 28 s ekvalizáciou a až 40 bez ekvalizácie.

Z experimentu vyplýva:

- ekvalizácia má veľký vplyv pri trénovaní bez ohľadu na veľkosť trénovacej množiny
- závislosť na veľkosti databáze a chybe pri testovaní je nepriamo úmerná
- potvrdenie predpokladov, že proces trénovania treba kontrolovať a získať sieť v momente minimálnej chyby na testovacej množine

7.1.2 Testovanie detekcie siete

Ak máme natrénovanú neurónovú sieť, je potrebné overiť jej detekčnú schopnosť. Budeme skúmať vplyv počtu neurónov v skrytej vrstve na úspešnosť siete. Nasleduje popis podmienok experimentu.

Testovacia množina obsahuje 200 tvári a 200 netvári, je vybraná z trénovacej skupiny databáze CBCL. Dôležité je, že na týchto obrázkoch neprebíha učenie neurónovej siete. CBCL množina obsahuje 472 tvári a 2500 netvári z testovacej skupiny databáze CBCL. Neurónová sieť bola trénovaná na tzv. *veľkej* trénovacej množine (popísaná v 7.1.1). Detekcia prebiehala s rozhodovacím prahom nastaveným na 0,5.

P.č.	Počet neurónov	Chyba na testovacej množine	Detekcia tvári	Falošná detekcia	Chyba na CBCL množine	Detekcia tvári	Falošná detekcia
1.	10	9,58581	97,5	2,5	384,330	29,45	2,72
2.		7,60410	97,0	3,5	309,220	24,15	1,08
3.		5,57713	98,5	1,5	315,520	24,79	1,00
1.	30	9,05142	97,0	2,5	382,040	19,49	1,80
2.		8,98082	96,5	3,0	290,804	37,29	2,16
3.		5,80000	98,0	2,5	308,553	29,66	1,08
1.	50	5,04080	98,0	2,5	310,333	30,51	1,08
2.		5,71905	98,0	2,5	283,077	34,75	1,12
3.		6,99317	97,5	1,5	336,348	23,73	1,20
1.	80	3,81052	98,5	0,5	320,705	29,24	0,92
2.		5,37629	98,0	1,0	339,038	23,31	0,60
3.		5,78254	97,5	1,5	323,468	26,48	1,08

Tabuľka 7.1: Výsledky detekcie so zapnutou ekvalizáciou

P.č.	Počet neurónov	Chyba na testovacej množine	Detekcia tvári	Falošná detekcia	Chyba na CBCL množine	Detekcia tvári	Falošná detekcia
1.	10	21,1233	98,0	7,5	425,167	27,5424	9,32
2.		15,8435	95,5	5,0	402,523	20,7600	5,28
3.		19,4828	98,5	9,0	403,856	34,1102	10,72
1.	30	19,6631	95,5	4,5	403,636	25,2119	7,80
2.		18,7437	98,5	7,0	394,093	28,6017	8,52
3.		18,4543	96,5	5,5	370,166	21,6102	6,28
1.	50	19,2129	97,0	8,5	457,222	33,0508	9,92
2.		23,2076	90,5	7,0	397,972	28,1780	7,32
3.		22,2644	97,0	7,0	412,952	24,5763	7,08
1.	80	15,4610	97,5	2,5	436,505	21,1864	5,96
2.		17,1873	92,0	6,0	389,017	20,1271	4,80
3.		17,7253	99,5	12,0	401,303	35,8051	12,28

Tabuľka 7.2: Výsledky detekcie s vypnutou ekvalizáciou

Merania so zapnutou ekvalizáciou dosahovali detekčnú schopnosť priemerne 97,67% a bez ekvalizácie 96,33% na testovacej množine. Na množine CBCL priemerne 27,74% s ekvalizáciou a 26,73% bez nej. Ekvalizácia mala väčší vplyv na falošné detekcie. Pri použití ekvalizácie histogramu testovaných obrázkov sme namerali priemerné hodnoty falošnej detekcie 2,08% pre tréningovú a 1,32% pre CBCL množinu, čo možno považovať sa úspešný výsledok. Keď sme ekvalizáciu vyppli, tak narástla početnosť falošných detekcií. Na tréningovej množine 6,79%, na CBCL množine 7,94% priemerne, čo je 3 až 6 násobný nárast.

Počet neurónov v skrytej vrstve jemne zlepšil detekčnú presnosť siete, no podstatne spomalil proces učenia.

7.2 Neurónová sieť ako lokalizátor

Dostávame sa do finále tejto práce, lokalizácie tvári v obraze. Na testovanie lokalizácie použijeme jeden vlastný obrázok a obrázky z CMU/ VASC databáze⁷, ktorá kombinuje obrázky zhromaždené na CMU a MIT. Neurónová sieť (uložená v /siete/network.net) aplikovaná na klasifikáciu má 30 neurónov v skrytej vrstve a je použitá ekvalizácia histogramu. Zároveň ukážeme použitie algoritmu na odstránenie násobných detekcií. Viac klasifikovaných obrázkov sa nachádza v prílohách 1 a 2, mi si prezentujeme len niektoré z nich.

⁷ Dostupná na adrese http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html

7.2.1 Obrázky jednotlivcov

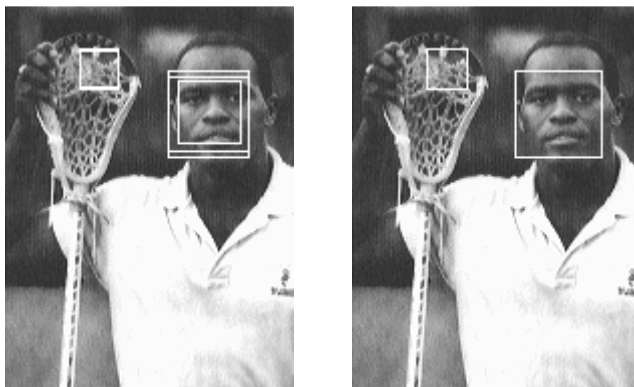
Najprv začneme s lokalizáciou tváre v obrazoch, kde je zastúpená raz, maximálne dva krát. Na ukážku som si dovoľil vybrať moju tvár, tvár autora tejto práce a obrázok z databáze CMU/VASC.

Pri spracovaní obrázku bolo klasifikovaných cez 250 000 okien. Prah rozhodovania bol nastavený na 0,9 a neurónová sieť označila 10 okien ako tváre. Detekcie boli výsledne zredukované na jednu úspešnú.



Obrázok 7.5: Klasifikácia obrázku /lokalizacia/jednotlivci/tom.pgm

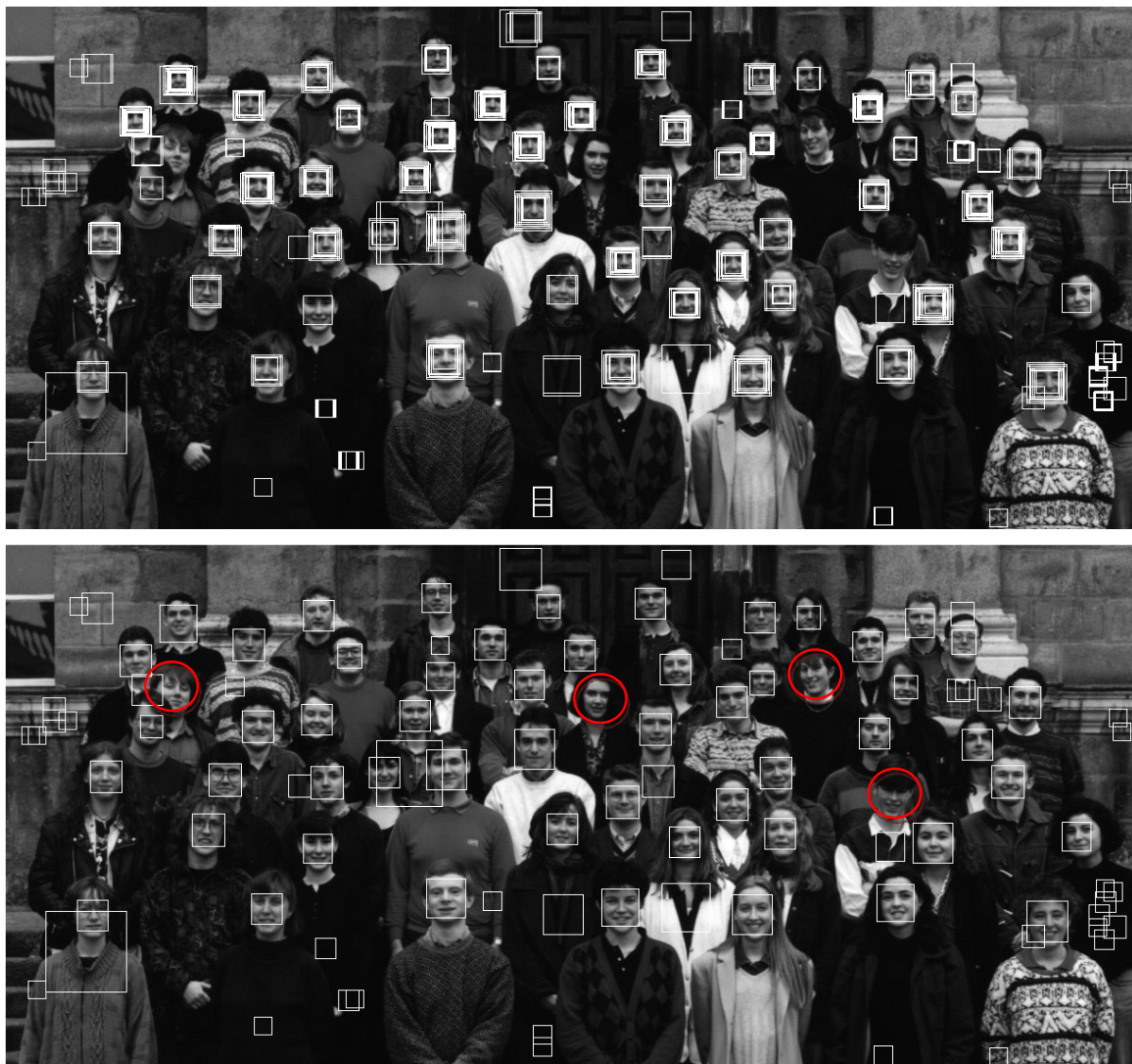
Nasledujúca klasifikácia nebola bezchybná, vyskytla sa v nej jedna falošná detekcia. Z 45 000 klasifikovaných okien určila sieť 5 ako tváre. Tie boli zlúčené na výsledne 2 lokalizácie. Prah rozhodovania bol nastavený na 0,7.



Obrázok 7.6: Klasifikácia obrázku /lokalizacia/jednotlivci/lacrosse.pgm

7.2.2 Obrázky skupín ľudí

Na ukážku lokalizácie tváří v obrazoch, kde sa nachádza viac ako jeden človek som zámerné vybral početnejšiu skupinu ľudí. Uvidíme ako to dopadlo.



Obrázok 7.7: Klasifikácia obrázku /lokalizacia/skupiny/class.pgm

Klasifikovaných bolo vyše 1 500 000 okien, z ktorých bolo vybraných 622 tváří. Odstránením násobných respektíve prekrývajúcich detekcií sme zredukovali počet lokalizácií na 97 s prahom rozhodovania 0,9.

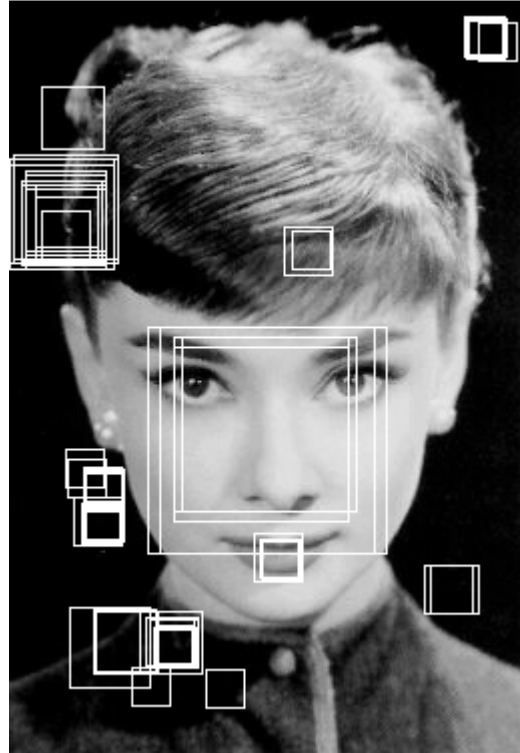
Aplikácia úspešne lokalizovala 53 z 57 tváří (93%) nachádzajúcich sa v obraze a zvyšných 44 lokalizácií bolo falošných. Niektoré z týchto falošných detekcií by bolo možné odstrániť použitím iného algoritmu spracovania lokalizácií. To je motiváciou k budúcej práci na tomto projekte.

7.2.3 Zlyhania aplikácie

Predchádzajúce kapitoly prezentovali úspešnosť lokalizácie tváre. Existujú obrázky, na ktorých táto aplikácia zlyháva. Aj napriek prahu nastavenému pomerne vysoko (na hodnotu 0,95) obsahuje obraz viacero falošných detekcií.



Obrázok 7.8: Klasifikovaný obrázok
/lokalizacia/zlyhanie/harvard.pgm



Obrázok 7.9: Klasifikovaný obrázok
/lokalizacia/zlyhanie/audrey2.pgm

Na obrázku Obrázok 7.8 bolo klasifikovaných 270 000 okien a 5 určených ako tváre. Sklamaním je výsledok na obrázku Obrázok 7.9, kde bolo klasifikovaných 190 000 okien a až 236 oblastí označených ako tváre.

8 Záver

Cieľom tejto práce bolo zoznámiť sa so základmi spracovania obrazu a s možnosťami detekcie objektov v obraze. Objektom detekcie bola zvolená ľudská tvár. Existuje veľa odborných prác, ktoré sa zaoberajú detekovaním, či lokalizovaním tvári v obraze. Práce sa dajú rozdeliť do základných kategórií podľa používania metódy klasifikácie obrazu. Rozdielnosť nastáva prínosom autorov, ktorý metódy klasifikácie upravujú, kombinujú prípadne rozširujú podľa svojho výskumu. V tejto práci sme na riešenie problému detekcie tvári použili metódu z kategórie Metódy založené na vzhľade, neurónové siete. Vzhľadom k rozsahu bakalárskej práce a náročnosti problematiky neurónových sietí sme túto metódu nijak nerozširovali. Motiváciu pre výber práve tejto metódy boli jej výsledky odpozorované z iných prác napr. [4], [5] a [9].

Prínos tejto práce bol najmä osobné zorientovanie sa v danej problematike. Osvojenie si práce na projekte väčších rozmerov ako sú bežné školské projekty. Téma ma zaujala natoľko, že som odhodlaný na nej pokračovať aj v budúcnosti. Práca na projekte bola náročná, z dôvodu pretrvávajúcich problémoch pri učení neurónovej siete. Po ich prekonaní sa dostavili potešujúce výsledky. Tie sú k nahliadnutiu v prílohách 1 a 2. Výsledkom práce je základný systém na detekciu a lokalizáciu tvári v obraze.

Ďalší vývoj projektu sa môže uberať viacerými smermi. Mňa osobne najviac zaujíma skúmanie vplyvu architektúry neurónovej siete na klasifikáciu. Knižnica FANN k tomuto nepostačuje a v rámci ďalšieho vývoja so zameraním na hlbšie skúmanie architektúry neurónových sietí je vhodné zvoliť inú knižnicu pre prácu s nimi. Ďalej ma zaujal fakt, že aj tá najlepšie nastavená neurónová sieť použitá bez iných algoritmov, ako je napríklad odstránenie duplicít, bude produkovať chybné (násobné, prekryvajúce sa) detekcie. Preto vývoj spracovania lokalizácii po použití neurónovej siete je taktiež zaujímavou oblasťou výskumu.

Literatura

- [1] Kolektiv autorů: *Histogram equalization* [online]. aktualizováno 2009-02-26 [cit. 2009-04-17]. Dostupné na URL: <http://en.wikipedia.org/wiki/Histogram_equalization>
- [2] McHugh, S.: *Digital Image Interpolation* [online]. [cit. 2009-04-26] Dostupné na URL: <<http://www.cambridgeincolour.com/tutorials/image-interpolation.htm>>
- [3] Svoboda, T., Kybic, J., a Hlavac, V.: *Image Processing, Analysis, and Machine Vision: A MATLAB Companion*. Thomson Engineering, 2008. s. 380-445. ISBN 0-495-08525-X.
- [4] Yang, M. H., Kriegman, D., Ahuja, N.: *Detecting Faces in Images – A Survey*. IEEE Transaction on Pattern Analysis and Machine Intelligence, leden 2002.
- [5] Rowley, H., Baluja, S., Kanade, S.: *Neural Network-Based Face Detection*. PAMI, 1998.
- [6] Sinčák, P., Andrejková, G.: *Neurónové siete Inžiniersky prístup* [online]. Košice: aktualizováno 1998-09-09 [cit. 2009-04-20]. Dostupné na URL: <<http://www.ai-cit.sk/source/publications/books/NS1/html/all.html>>
- [7] Švub, M.: *Segmentace obrazových a objemových dát pomocí neuronových sítí*, diplomová práce. Brno, FIT VUT v Brně, 2006.
- [8] Rojas, R.: *Neural Networks: A Systematic Introduction*. Springer-Verlag, Berlin, 1996. s. 159-192. ISBN 3-540-60505-3.
- [9] Hendrych, P.: *Lokalizace obličeje pomocí neuronové sítě*, diplomová práce. Brno, FIT VUT v Brně, 2008.

Seznam příloh

Príloha 1. Ukážky lokalizácie tvári v obraze, kde sa nachádza jedna maximálne dve tváre.

Príloha 2. Ukážky lokalizácie tvári v obraze skupinových fotografií.

Príloha 3. Popis inštalácie a práce so systémom.

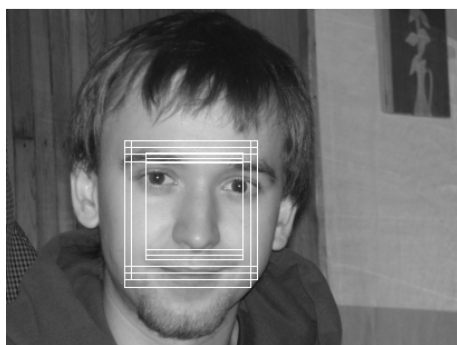
Príloha 4. Elektronická príloha na CD, obsahuje:

- Zdrojové súbory systému
- Databázu obrázkov na tréning a testovanie neurónovej siete
- Obrázky určené na lokalizáciu tvári
- Tento dokument vo formáte doc a pdf

Príloha 5. Plagát demonštrujúci vlastnosti a činnosť implementovaného systému.

Príloha 1

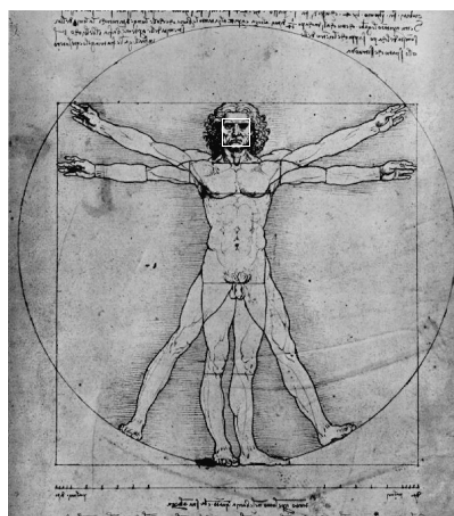
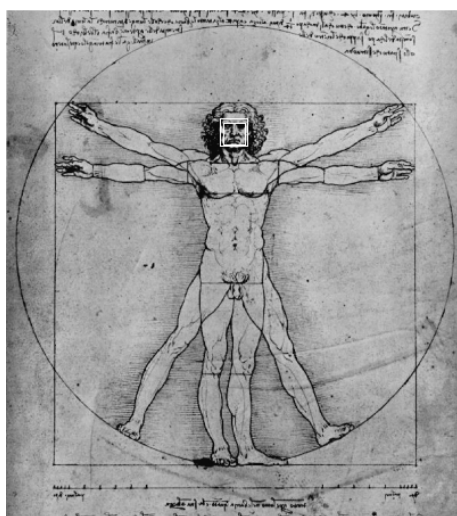
Ukážky lokalizácií tváre v obraze, kde sa nachádza jedna, maximálne dva tváre. Použitá neurónová sieť /siete/network.net s 30 neurónmi v skrytej vrstve. V ľavom stĺpci po lokalizácii v pravom po odstránení prekrývajúcich sa okien.



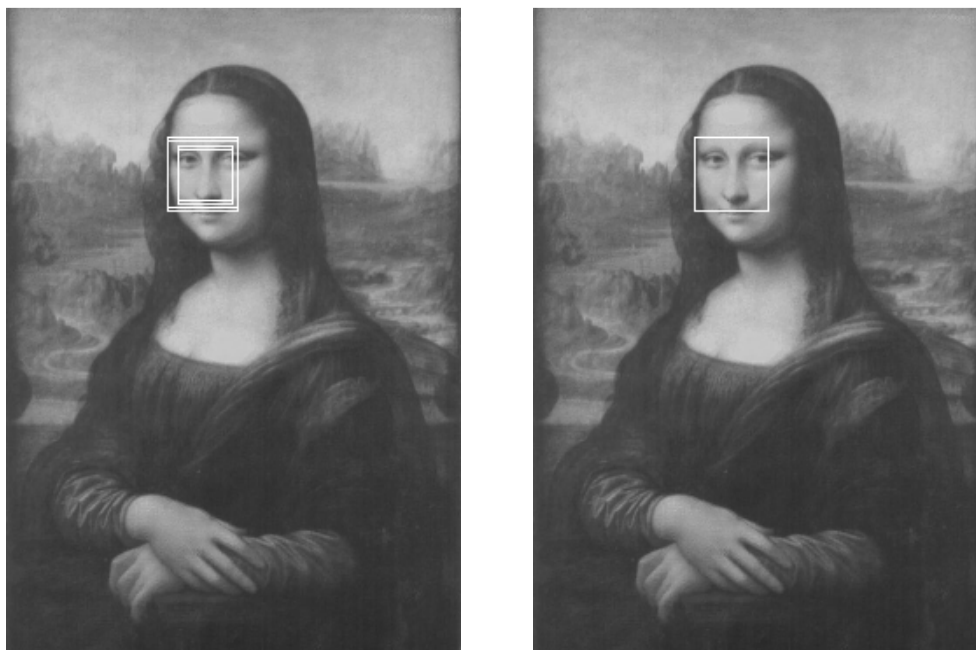
Obrázok 1.1: /lokalizacia/jednotlivci/tom.pgm



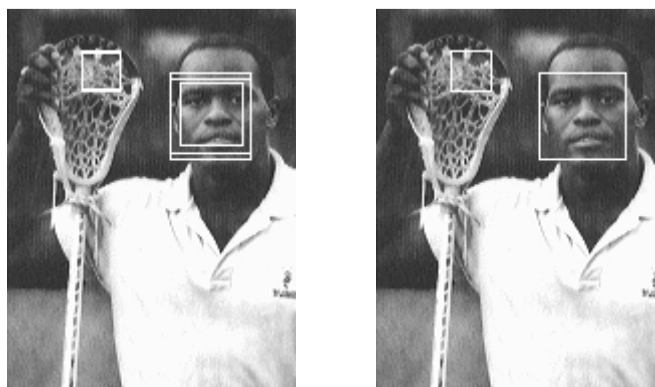
Obrázok 1.2: /loklizacia/jednotlivci/tor.pgm



Obrázok 1.3: /loklizacia/jednotlivci/divinci-man1.pgm



Obrázok 1.4: /loklizacia/jednotlivci/mona.pgm



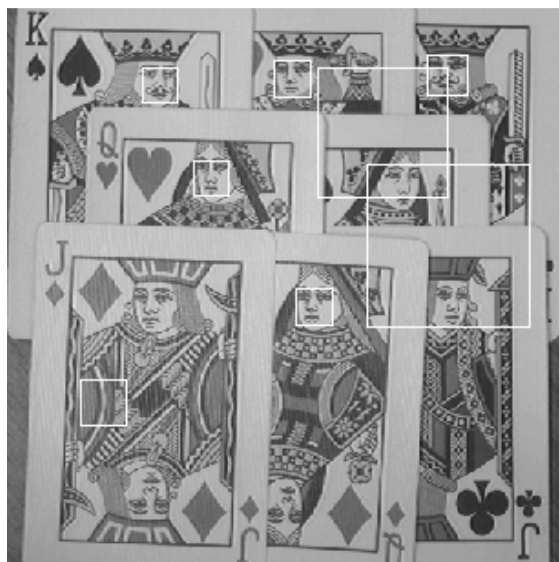
Obrázok 1.5: /loklizacia/jednotlivci/lacrosse.pgm

Príloha 2

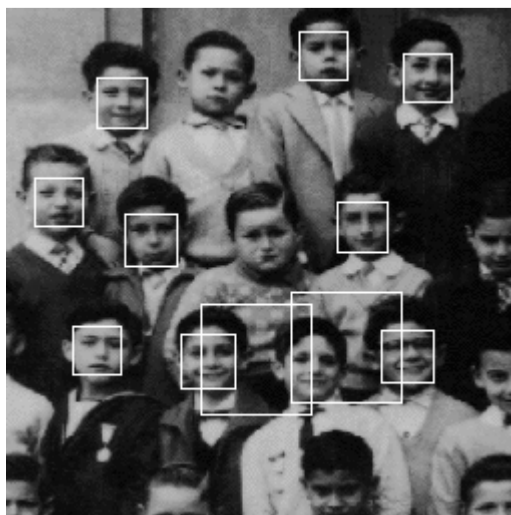
Ukážky lokalizácií tvári v obraze, kde sa nachádza väčší počet ľudí. Použitá neurónová sieť /siete/network.net s 30 neurónmi v skrytej vrstve. Kvôli veľkosti obrázkov uvedené výsledné klasifikácie s redukovanými násobnými lokalizáciami.



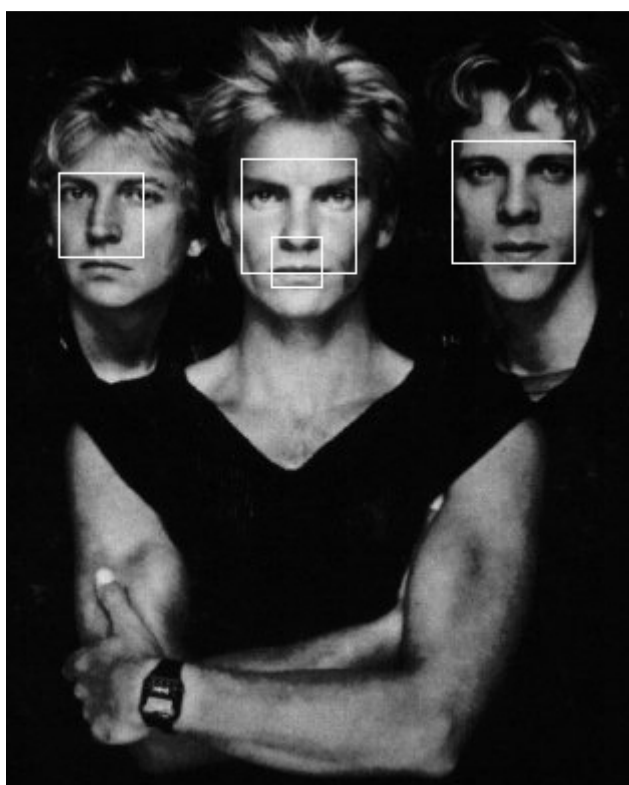
Obrázok 2.1: /lokalizacie/skupiny/Argentina.pgm, detekovaných 8 tvári z 11 (73%),
prah 0,875



Obrázok 2.1: /lokalizacie/skupiny/cards-perp-sml.pgm, detekovaných 5 tvári z 8 (63%),
prah 0,8



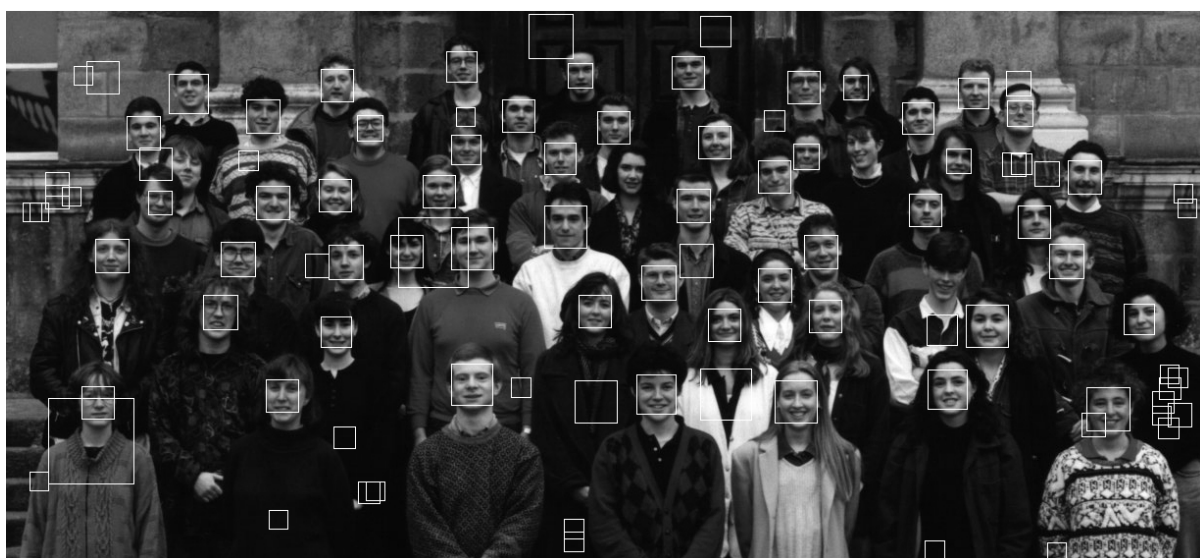
Obrázok 2.1: /lokalizacie/skupiny/nens.pgm, detekovaných 9 tvári z 12 (75%), prah 0,9



Obrázok 2.1: /lokalizacie/skupiny/police.pgm, detekovaných 3 tvári z 3 (100%), prah 0,9



Obrázok 2.1: /lokalizacie/skupiny/soccer.pgm, detekovaných 12 tvári z 15 (80%), prah 0,9



Obrázok 2.1: /lokalizacie/skupiny/class.pgm, detekovaných 53 tvári z 57 (93%), prah 0,9

Príloha 3

Popis adresárovej štruktúry priloženého CD

- **doc/** priečinok obsahuje technickú správu
- **face_database/** priečinok obsahuje množiny obrázkov tvári a netvári na tréovanie a testovanie
- **inst/** priečinok obsahuje inštalačný súbor ImageMagick potrebný na spracovanie formátu PGM
- **lib/** priečinok obsahuje použité knižnice CImg a Fann
- **lokalizacia/** priečinok obsahuje obrázky použité pri lokalizácii
- **siete/** priečinok obsahuje natrénované neurónové siete

Preklad systému

V koreňovom priečinku je Makefile, ktorý slúži na preklad pomocou programu make (mingw32-make). Je možné skompilovať jednotlivé moduly osobitne (*make train*, *make test*, *make main*) alebo celý projekt naraz (*make* prípadne *mingw32-make*). Ďalej je potrebné nainštalovať program ImageMagick z priečinka **inst/**, ktorý potrebuje knižnica CImg na korektnú prácu s formátom PGM.

Spustenie systému

Bližší popis v kapitolách 6.3, 6.4 a 6.5.

```
train.exe  [<neurónová sieť>] [<počet neurónov>] [<ekvalizácia on/off>]
test.exe   [<neurónová sieť>] [<prah>]           [<ekvalizácia on/off>]
main.exe   <neurónová sieť>  <obrázok>          <prah>           [<ekvalizácia on/off>]
```