

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
ŠTUDIJNÝ ODBOR: TELEKOMUNIKÁCIE

Bc. Marián Beszédeš

Detekcia ľudských tvárí v obraze

Diplomová práca

Vedúci diplomovej práce: Doc. Dr. Ing. Miloš Oravec.

December 2002

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný odbor: TELEKOMUNIKÁCIE

Autor: Marián Beszédeš, Bc.

Názov práce: Detekcia ľudských tvárí v obraze

Vedúci práce: Doc. Dr. Ing. Miloš Oravec

Popisujem systém na detekciu ľudských tvárí v neupravenom vstupnom obraze pomocou neurónových sietí.

Systém spracováva a normalizuje malé okná extrahované zo vstupného obrazu. Pri tomto sú použité vybrané techniky číslicového spracovania obrazu týkajúce sa normalizácie rozmerov, polohy, natočenia, vyrovnaní svetelných podmienok a kontrastu.

Neurónové siete, sa využívajú v dvoch častiach systému. Najprv ide o detekciu natočenia obsahu spracovávaného okna a následne o rozhodnutie či okno obsahuje tvár, alebo nie. V oboch prípadoch je ako typ siete zvolený viacvrstvový perceptrón. V práci je riešená problematika najvhodnejšej topológie a spôsobu učenia neurónovej siete. Pri sieti určenej na klasifikáciu obsahu vzhľadom na prítomnosť tváre je použitá metodika rozhodovania distribuovaného do viacerých podsietí. Na tréovanie tejto siete je aplikovaný špeciálny bootstrap algoritmus.

Výstupom systému je množina lokalizovaných ľudských tvárí vo vstupnom obraze.

Annotation

Slovak University of Technology, Bratislava

Faculty of Electrical Engineering and Information Technology

Degree Course: Telecommunications

Author: Marián Beszédeš, Bc.

Thesis: Neural Network-Based Face Detection

Supervisor: Doc. Dr. Ing. Miloš Oravec

In my thesis is presented a neural network-based face detection system. Faces are detected in an unprocessed input image.

The system processes and normalizes small windows extracted from the input image. Digital image processing techniques, as normalization of size, position and rotation, improvement of light conditions and contrast are used here.

Neural network is applied in two parts of the system. At first it detects rotation of the input window and afterwards it decides whether the window contains a face or not. In both of these cases multilayer perceptrone as a type of neural network is used. A problematic of the best topology and training method is discussed here, too. Face detection neural network uses method of distributing decision between multiple sub networks. A special bootstrap algorithm is used to train this network.

The face detection system result is in the form of set containing locations of human faces.

Čestné prehlásenie

Čestne prehlasujem, že diplomovú prácu som vypracoval samostatne pod odborným vedením Doc. Dr. Ing. Miloša Oravca, s použitím uvedenej literatúry

Marián Beszédeš

13. decembra 2002

Pod'akovanie

Touto cestou by som chcel vyjadriť poďakovanie môjmu diplomovému vedúcemu Doc. Dr. Ing. Milošovi Oravcovi za odborné vedenie diplomovej práce, a podnetné pripomienky pri jej vypracovaní.

Zároveň by som sa chcel poďakovať svojej rodine za pochopenie a podporu počas celej tvorby tejto diplomovej práce.

OBSAH

1	ÚVOD.....	3
1.1	ZÁKLADNÉ ÚLOHY PRI LOKALIZÁCII ĽUDSKEJ TVÁRE.....	3
1.2	ŠTRUKTÚRA PRÁCE.....	5
2	NEURÓNOVÉ SIETE	6
2.1	VIACVRSTVOVÝ PERCEPTRÓN	6
2.1.1	<i>Dôležité charakteristiky viacvrstvového perceptrónu:</i>	6
2.1.2	<i>Algoritmus spätného šírenia</i>	8
2.1.3	<i>Kritérium pre ukončenie učenia</i>	9
3	TEORETICKÝ POPIS SYSTÉMU	11
3.1	PREDSPRACOVANIE VSTUPNÉHO OBRAZU	12
3.2	VÝBER SEGMENTOV VSTUPNÉHO OBRAZU	12
3.3	PREDSPRACOVANIE VYBRANÉHO SEGMENTU.....	13
3.3.1	<i>Odstránenie nerovnomerného osvetlenia – lineárny model osvetlenia</i>	13
3.3.2	<i>Histogramová ekvalizácia</i>	15
3.4	DETEKCIA NATOČENIA	15
3.4.1	<i>Viacvrstvový perceptrón pre klasifikáciu natočenia vstupného obrazu</i>	15
3.4.2	<i>Učenie neurónovej siete</i>	17
3.4.3	<i>Získanie tvárových vzoriek s rôznym natočením</i>	17
3.4.4	<i>Proces učenia</i>	20
3.5	NORMALIZÁCIA	21
3.5.1	<i>Spätná rotácia</i>	22
3.5.2	<i>Aplikovanie masky</i>	22
3.6	DETEKCIA PRÍTOMNOSTI TVÁRE	22
3.6.1	<i>Viacvrstvový perceptrón pre klasifikáciu vstupného obrazu</i>	23
3.6.2	<i>Učenie neurónovej siete</i>	25
3.6.3	<i>Získanie frontálnych tvárových vzoriek:</i>	25
3.6.4	<i>Získanie netvárových vzoriek</i>	25
3.6.5	<i>Proces učenia</i>	26
3.6.6	<i>Vyhodnotenie</i>	27
3.7	FUNKČNÉ SCHÉMY NAVRHOVANÝCH SYSTÉMOV	27
3.7.1	<i>Funkčná schéma systému bez detekcie natočenia</i>	28
3.7.2	<i>Funkčná schéma systému s detekciou natočenia</i>	28
4	IMPLEMENTÁCIA SYSTÉMU A TEORETICKÝCH PRINCÍPOV	29
4.1	VÝVOJOVÝ DIAGRAM PROCESU LOKALIZÁCIE ĽUDSKÝCH TVÁRÍ.....	30
4.2	VSTUPY A VÝSTUPY SYSTÉMU	32

4.3	BLOK NEURÓNOVEJ SIETE PRE DETEKCIU NATOČENIA VSTUPNÉHO OBRAZU.....	34
4.3.1	<i>Vytvorenie neurónovej siete</i>	34
4.3.2	<i>Trénovanie neurónovej siete</i>	35
4.4	BLOK NEURÓNOVEJ SIETE PRE KLASIFIKÁCIU VSTUPNÉHO OBRAZU.....	37
4.4.1	<i>Vytvorenie neurónovej siete</i>	37
4.4.2	<i>Trénovanie neurónovej siete</i>	38
4.5	IMPLEMENTÁCIA NEURÓNOVÝCH SIETÍ.....	41
4.6	OSTATNÉ BLOKY SYSTÉMU.....	41
5	EXPERIMENTÁLNE VÝSLEDKY.....	43
5.1	VIACVRSTVOVÝ PERCEPTRÓN PRE KLASIFIKÁCIU NATOČENIA VSTUPNÉHO OBRAZU	43
5.1.1	<i>Počiatočná inicializácia váh</i>	43
5.1.2	<i>Parameter rýchlosti učenia</i>	43
5.1.3	<i>Počet cyklov tréovania a problematika premiešavania vzoriek</i>	44
5.1.4	<i>Veľkosti tréovacej a testovacej množiny</i>	45
5.1.5	<i>Parameter pre ohodnotenie naučenosti siete</i>	45
5.1.6	<i>Rozsah detekovaných otočení</i>	46
5.1.7	<i>Topológia siete</i>	47
5.2	VIACVRSTVOVÝ PERCEPTRÓN PRE KLASIFIKÁCIU VSTUPNÉHO OBRAZU	49
5.2.1	<i>Počet cyklov tréovania a problematika premiešavania vzoriek</i>	50
5.2.2	<i>Topológia siete</i>	52
5.3	VPLYV OBSAHU VZORKOVEJ MNOŽINY NOFACES.....	55
5.4	VPLYV VSTUPNÝCH PARAMETROV NA FUNKČNOSŤ SYSTÉMU.....	55
5.5	TESTY.....	56
5.5.1	<i>Test Audrey 1</i>	57
5.5.2	<i>Test Audrey 2</i>	60
5.5.3	<i>Test Školáci</i>	61
5.5.4	<i>Test Vysokoškoláci</i>	62
5.5.5	<i>Test Futbal Team</i>	64
5.5.6	<i>Ďalšie testy</i>	65
6	ZÁVER.....	66
7	POUŽITÁ LITERATÚRA	67
8	POUŽITÉ SKRATKY A SYMBOLY	69
9	PRÍLOHA	70
9.1	POPIS SPUSTENIA A PARAMETROV PROGRAMU FACELOCALIZER	70

1 Úvod

Automatická detekcia a rozpoznanie tváre je na poli výpočtovej techniky už veľa rokov veľmi zložitá úloha. Napriek tomu, že človek uskutočňuje túto činnosť bez zjavného úsilia ide o veľmi komplexný problém. Zdanlivo triviálna skutočnosť nájdenia a rozpoznania tváre je u človeka výsledok miliónov rokov evolúcie. Vedci sú ešte stále veľmi ďaleko od riešenia problému ako túto úlohu mozog rieši. Schopnosť vizuálneho nájdenia tváre v prostredí a jej rozpoznanie, je veľmi dôležitá pre mnohé oblasti. Preto automatizovanie tejto činnosti a jej realizácia prostredníctvom výpočtovej techniky je veľmi potrebná v mnohých aplikáciách, ako napr. ochrana bezpečnosti (letiská, prístup do budov), policajný dozor, video kompresia a animácia. Mojou snahou bolo v rámci diplomovej práce zhromaždiť dostatok teoretických informácií týkajúcich sa spomínanej problematiky a na ich základe vytvoriť systém, ktorý je schopný v neupravovanom vstupnom obraze lokalizovať objekty ľudských tvárí.

1.1 Základné úlohy pri lokalizácii ľudskej tváre

Systém na lokalizáciu ľudských tvárí musí byť schopný plniť svoju úlohu napriek mnohým variáciám výskytu tváre, ktoré táto v obraze môže mať. Navyše, musí byť schopný tolerovať rôzne variácie, ktoré sa objavujú vo vnútri tváre. Táto totiž nie je jednoducho popísateľná, konštantná, či nemenná. Je možný výskyt miliónov rôznych tvárí a každá z nich môže mať ešte ďalšie variácie. Môže ísť o tváre rôznych rás, identity, tvarov, polôh, nehovoriac o rôznych účesoch, bradách a farbách vlasov. Je to objekt, ktorý môže byť osvetlený z mnohých zdrojov a môže byť obklopený ľubovoľným počtom objektov v pozadí (napríklad aj ďalšími tvármi). Ak teda chceme vytvoriť systém na lokalizáciu tvárí, musíme so všetkými týmito faktami počítať, alebo ich ignorovať.

V súčasnosti existujú viaceré metodiky pre lokalizáciu ľudských tvárí. Vo všeobecnosti ich môžeme zaradiť do dvoch skupín [6].

V prvej z nich je použitá technika vyhľadávania tváre podobných vzorov vo vstupnom monochromatickom obraze pri jeho rôznych veľkostiach.

Vyhodnocované sú postupne všetky segmenty obrazu. Obraz môže prejsť najprv rôznymi krokmi predspracovania, ktoré zvýrazia dôležité črty na tvári, alebo jej obrysy (napr. Sobelov operátor [4]) Samotné vyhodnotenie či segment obsahuje, alebo neobsahuje tvár môže byť realizované pomocou porovnania segmentu so šablónou, ktorá predstavuje objekt tváre [5][11]. Nevýhodou tohto postupu je veľké množstvo chybných lokalizácií. Iným možným prístupom je transformácia obsahu segmentu do vlastného priestoru (*Eigenspace*) tváří [6][8][9]. Táto metóda predpokladá, že všetky obrazy tváří môžeme celkom úspešne aproximovať v nízkorozmernom podpriestore tváří, popísanom prostredníctvom vlastných vektorov tváří, vypočítaných z množiny testovacích („Ideálnych“) tváří. Obrazový segment je teda klasifikovaný ako tvár ak sa po transformácii dostane do prípustnej vzdialenosti k podpriestoru tváří (vzdialenosť je menšia ako určitý prah, pričom na výpočet vzdialenosti môžu byť použité rôzne metriky). Preto je potrebné všetky segmenty so vstupného obrazu transformovať do podpriestoru tváří a tu ich klasifikovať na základe hlavných zložiek transformovaného vektoru. Tento postup sa nazýva analýza hlavných komponentov (PCA - Principal Component Analysis) [1][2][3][12]. Táto metóda dáva dostatočne dobré výsledky, avšak jej výpočtová náročnosť je príliš vysoká.

V druhej skupine sa vychádza z techniky segmentácie obrazu na základe farebnostnej informácie (toto je samozrejme podmienené tým, že vstupný obraz je farebný). Farba je jeden z najpríznačnejších klasifikátorov oblasti tváre a preto je tento postup ako jeden z prvých krokov samotnej detekcie tváre vo viacerých algoritmoch [7]. Ďalšími krokmi pri detekcii je klasifikácia regiónov označených ako tváre na základe ich tvaru. Táto metóda však často zlyháva z dôvodu neúplného určenia celého regiónu tváre, alebo je za tvár označený úplne iný objekt pozadia s farbou približne podobnou ako tvár. Preto celý výsledok závisí prevažne od počiatočného určenia tvárových oblastí.

Ja som sa zameral na metódu, ktorú by sme mohli zaradiť do prvej skupiny. Postupne sa preverujú všetky segmenty vstupného obrazu, pričom samotné preverenie je realizované pomocou neurónovej siete. Táto štatisticky vyhodnocuje obsah segmentu a jej výstupom je pravdepodobnosť, s akou segment obsahuje, alebo neobsahuje ľudskú tvár. Jej výhodou je podstatne

vyššia výpočtová rýchlosť ako pri PCA (využitie je paralelné spracovanie neurónovou sieťou) a súčasne zanedbateľný počet chybných lokalizácií.

1.2 Štruktúra práce

Moja diplomová práca pozostáva z viacerých ucelených celkov týkajúcich sa danej problematiky. Uvádzam teda stručný popis jej jednotlivých častí.

Keďže nosnou témou mojej práce je využitie neurónových sietí, v prvej časti "**Neurónové siete**", uvádzam stručný teoretický prehľad týkajúci sa viacvrstvého perceptrónu. Ide o typ neurónovej siete, ktorý bol vyhodnotený pre moje potreby ako najvhodnejší.

V druhej časti "**Teoretický popis systému**", už prechádzam k návrhu samotného systému na automatizovanú lokalizáciu ľudských tvárí. Je v nej uvedený teoretický základ týkajúci sa danej problematiky. Postupne je vysvetlená funkcionálnosť jednotlivých blokov systému a dôvody prečo bola pre spracovanie zvolená práve daná technika.

V tretej časti "**Implementácia systému a teoretických princípov**" popisujem už konkrétnu implementáciu systému s dôrazom na popis tréningu a testovania neurónových sietí. Je tu uvedený aj popis jeho vstupných a výstupných parametrov.

Pokiaľ je v predchádzajúcich dvoch častiach uvedené, že sa nejaká časť systému, alebo postup ukázal ako najlepší, tak v štvrtej časti "**Experimentálne výsledky**" je vysvetlené prečo je tomu tak. Sú tu popísané čiastkové, ale aj koncové výsledky a možnosti voľby hodnôt vstupných parametrov systému. Súčasne tu sú popísané aj poznatky, ktoré ďalej ovplyvnili postup mojej práce. Táto časť je ukončená uvedením niekoľkých záverečných testov vykonaných pomocou systému.

V poslednej časti "**Záver**" sú zhodnotené poznatky a výsledky získané mojou prácou. Sú tu uvedené aj návrhy zlepšení, ktoré by v budúcnosti mohli viesť k lepším výsledkom. Navyše sú tu ešte načrtnuté ďalšie možné postupy, ktoré by mohli dotvárať celý systém na lokalizáciu a rozpoznanie ľudských tvárí.

2 Neurónové siete

Podrobným popisom problematiky neurónových sietí a ich využitia sa zaoberá množstvo literatúry [1][2][10][12][14][16]. V mojej práci som sa preto rozhodol uviesť iba teoretické poznatky bezprostredne súvisiace s riešenou problematikou.

2.1 Viacvrstvový perceptrón

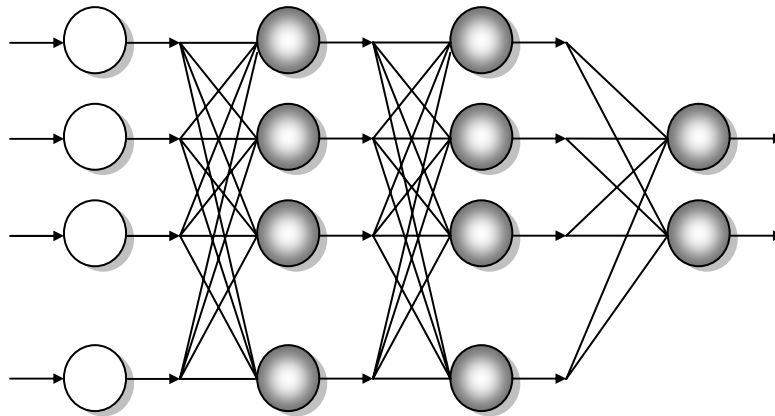
Jedná sa o viacvrstvovú doprednú sieť skladajúcu sa z vrstvy vstupných prvkov, tieto distribuujú vstupné signály do siete, z jednej alebo viacerých skrytých vrstiev výpočtových prvkov a z výstupnej vrstvy výpočtových prvkov [1][2]. Vstupné signály sa v sieti šíria dopredným smerom postupne od vrstvy k vrstve (Obr.2.1).

Viacvrstvový perceptrón sa trénuje s učiteľom – využíva sa algoritmus spätného šírenia, ktorý je založený na pravidle učenia korigujúcim chybu [1][2].

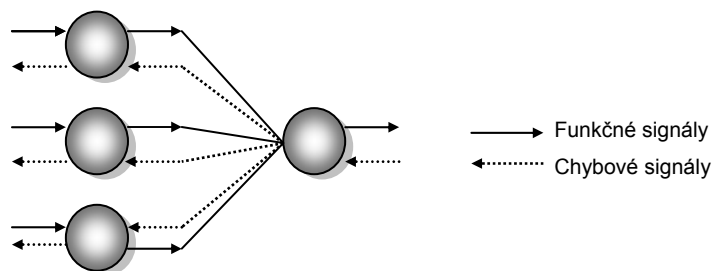
2.1.1 Dôležité charakteristiky viacvrstvého perceptrónu:

- Model každého výpočtového neurónu v sieti obsahuje na výstupe nelinearitu, ktorá musí byť hladká - t.j. všade diferencovateľná. Najpoužívanejšou formou nelinearity je sigmoidálna nelinearita [1][2]. Bez prítomnosti nelinearit by sa vstupno-výstupná charakteristika siete dala zredukovať na jednovrstvový perceptrón.
- Pre viacvrstvový perceptrón je charakteristická existencia jednej alebo viacerých skrytých vrstiev. Sú to práve skryté neuróny, ktoré umožňujú sieti naučiť sa komplexné úlohy, pretože vyberajú zo vstupných vzoriek najdôležitejšie príznaky.
- Viacvrstvový perceptrón má vysoký stupeň prepojenia siete, určený synapsami siete.

Spomenuté vlastnosti robia z viacvrstvého perceptrónu výpočtovo silný prostriedok.



Obr. 2.1: Viacvrstvový perceptrón s dvomi skrytými vrstvami



Obr. 2.2: Šírenie funkčných a chybových signálov vo viacvrstvovom perceptróne

Na Obr.2.2. je znázornená iba časť viacvrstvového perceptrónu. Je tu možné pozorovať dva druhy signálov: funkčné a chybové signály.

- Funkčný signál je signál, ktorý prichádza zo vstupu siete, šíri sa dopredne neurón po neuróne až sa objaví na výstupe siete ako výstupný signál.
- Chybový signál vzniká vo výstupných neurónoch siete počas tréningu a šíri sa späťne vrstvu po vrstve v sieti.

Každý skrytý a výstupný neurón viacvrstvového perceptrónu robí v čase tréningu dva výpočty:

- Výpočet funkčného signálu, ktorý je výstupom neurónu a je vyjadrený ako nelineárna funkcia vstupných signálov a synaptických váh spojených s týmto neurónom.
- Výpočet okamžitého odhadu gradientového vektora, t.j. gradientu chybového povrchu vzhľadom na váhy pripojené k danému neurónu. Gradientový vektor je potrebný pre fázu spätného šírenia chýb v sieti.

2.1.2 Algoritmus spätného šírenia

Viacvrstvový perceptrón sa trénuje s učiteľom – využíva sa algoritmus *spätného šírenia*. Tento algoritmus upravuje váhu $w_{ij}(n)$ o hodnotu $\Delta w_{ij}(n)$, ktorá je úmerná okamžitému gradientu

$$\partial \varepsilon / \partial w_{ji}(n)$$

Pre gradient platí nasledujúci vzťah:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (2.1)$$

Po diferencovaní a úpravách dostávame vzťah, ktorý poukazuje na významnú úlohu chybového signálu $e_j(n)$ pri výpočte úpravy váhy delta $\Delta w_{ij}(n)$

$$\Delta w_{ij}(n) = \eta \delta_j(n) y_i(n) \quad (2.2)$$

kde $\delta_j(n)$ je lokálny gradient definovaný ako

$$\delta_j(n) = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \cdot \varphi'_j(v_j(n)) \quad (2.3)$$

Môžu nastať dva prípady:

1. Neurón j je výstupným neurónom. Nevzniká problém, pretože žiadaná odpoveď je pre výstupné neuróny priamo dostupná. Pre výpočet chybového signálu $e_j(n)$ platí

$$e_j(n) = d_j(n) - y_j(n) \quad (2.4)$$

Po výpočte chybového signálu sa dá lokálny gradient určiť podľa (2.3).

2. Neurón j je skrytým neurónom, a teda žiadaná odpoveď je preň nedostupná. Hoci žiadanú odpoveď pre skrytý neurón nepoznáme, každý skrytý neurón nesie svoju časť zodpovednosti za chyby na výstupoch siete. Chybový signál musí byť vypočítaný rekurzívne podľa chybových signálov neurónov, s ktorými je priamo spojený.

Pre lokálny gradient $\delta_j(n)$ pre skrytý neurón j platí

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (2.5)$$

Člen $\varphi'_j(v_j(n))$ závisí iba od aktivačnej funkcie skrytého neurónu j . Členy v sumácii cez k závisia od dvoch faktorov. Členy $\delta_k(n)$ vyžadujú znalosť všetkých chybových signálov $e_k(n)$, ktoré ležia v najbližšej nasledujúcej skrytej vrstve neurónu j a sú s ním spojené. Členy $w_{kj}(n)$ opisujú synaptické váhy týchto spojení.

Pri použití algoritmu spätného šírenia rozlišujeme dve fázy výpočtu: doprednú a spätnú. Pri doprednej fáze sa synaptické váhy nemenia, funkčné signály siete sa počítajú postupne neurón po neuróne. Ak sa jedná o výstupný neurón j , funkčný signál na výstupe neurónu sa porovná so žiadanou odpoveďou, a rozdiel určuje chybový signál $e_j(n)$ pre výstupný neurón j . Dopredná fáza výpočtu teda začína v prvej skrytej vrstve privedením vstupného vektora a končí sa vo výstupnej vrstve určením chybových signálov pre všetky výstupné neuróny.

Spätná fáza výpočtu sa začína vo výstupnej vrstve, z ktorej sa chybové signály šíria spätne postupne vrstvu po vrstve a rekurzívne sa počíta lokálny gradient δ pre každý neurón. Tento rekurzívny proces umožní úpravu váh podľa (2.2). Pre neurón výstupnej vrstvy sa δ rovná súčinu chybového signálu a prvej derivácie jeho nelinearity. Keď sú už známe lokálne gradienty pre neuróny vo výstupnej vrstve, môžu sa použiť (2.5) pre výpočet lokálnych gradientov pre neuróny najbližšej skrytej vrstvy, a teda vypočítať aj zmeny všetkých váh spojení výstupnej a najbližšej skrytej vrstvy. Takýto rekurzívny výpočet pokračuje postupne pre ďalšie vrstvy, kde sú už vždy zohľadnené zmeny váh urobené v predchádzajúcich vrstvách.

Výpočet lokálnych gradientov δ pre neuróny viacvrstvového perceptrónu vyžaduje derivácie aktivačných funkcií neurónov, preto podmienkou na aktivačnú funkciu je jej diferencovateľnosť. Najčastejšie sa využíva sigmoidálna nelinearita.

2.1.3 Kritérium pre ukončenie učenia

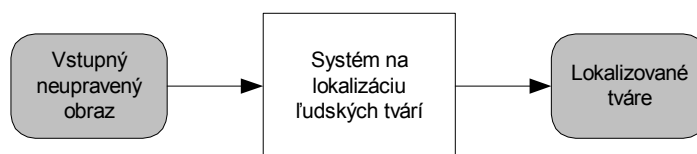
Pri použití algoritmu *spätného šírenia* sa nedá jednoznačne určiť či už skonvergoval. Často používaným kritériom je, že po každej iterácii, resp. po každej epoche sa sieť testuje z hľadiska generalizácie. Učenie sa skončí, keď

sieť už generalizuje primerane našim požiadavkám, prípadne schopnosť siete generalizovať dosiahla svoje maximum.

Viacvrstvový perceptrón reprezentuje mapovanie z p -rozmerného euklidovského priestoru vstupov do q -rozmerného euklidovského priestoru výstupov, kde p je počet vstupných a q počet výstupných neurónov viacvrstvého perceptrónu.

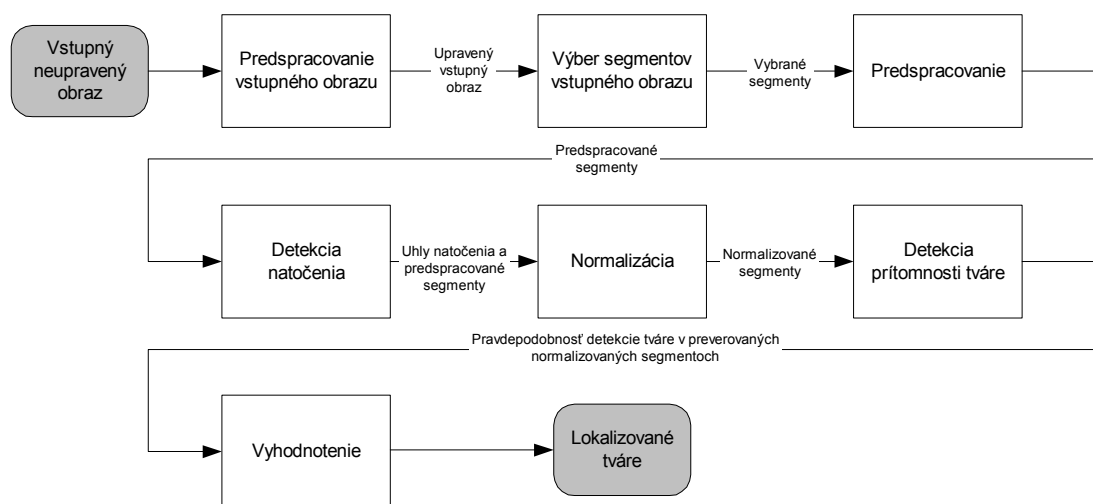
3 Teoretický popis systému

Všeobecný automatizovaný systém určený na lokalizáciu ľudských tvárí by sa dal zjednodušene popísať blokovou schémou na Obr.3.1.



Obr. 3.1: Blokova schéma všeobecného systému určeného na lokalizáciu ľudských tvárí

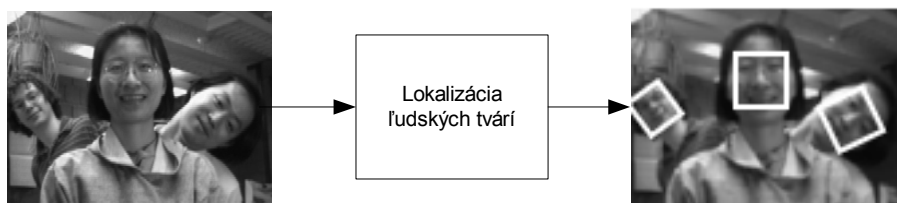
Systém, ktorý som sa rozhodol analyzovať, by sa dal popísať podrobnejšou blokovou schémou (Obr.3.2).



Obr. 3.2: Blokova schéma všeobecného systému určeného na lokalizáciu ľudských tvárí

Pri jeho návrhu som uvažoval s nasledujúcimi faktami:

1. Bude spracovávať statické, nepohybujúce sa obrázky.
2. Nemalo by záležať na tom či ide o obraz monochromatický, alebo farebný. Presne definovaná kvalita a veľkosť vstupného obrazu by tiež nemala byť nutnou podmienkou.
3. Systém bude schopný lokalizovať frontálny obraz tváre, pričom táto bude môcť byť ľubovoľne umiestnená, veľká, naklonená a osvetlená (Obr.3.3).
4. Systém by mal lokalizovať všetky tváre vyskytujúce sa na obraze.
5. Ako detekčný nástroj bude použitá neurónová sieť.



Obr. 3.3: Grafické znázornenie požiadaviek na systém

V ďalších častiach práce budem postupne popisovať jednotlivé systémové bloky.

3.1 Predspracovanie vstupného obrazu

Predspracovanie obrazu a vylepšovanie jeho kvality sa dá realizovať pomocou mnohých metód o ktorých je napísané množstvo literatúry [4][5][10]. Pre danú problematiku môže byť výber správnej metódy rozhodujúci. Pokiaľ ide o vylepšenie kvality môžeme spomenúť napríklad odstraňovanie šumov, zvyšovanie lokálneho a globálneho kontrastu, alebo vylepšenie svetelných pomerov v celom obraze [4]. Z techník na predspracovanie spomeniem len konverziu farebnosti obrazu (konverzia z farebného obrazu na monochromatický).

Aj keď dané techniky priamo súvisia s úlohou lokalizácie tvárí, ich použitie je veľmi závislé od konkrétnej aplikácie. Nezaoberám sa teda podrobne touto problematikou a predpokladám, že vstupný obraz neobsahuje veľké množstvo šumu. Problémy s jeho svetelnými pomermi a kontrastom by mali byť riešené v ďalších častiach systému pri spracovávaní vybraných segmentov.

Čo sa týka predspracovania vstupného obrazu, mala by byť použitá iba konverzia jeho farebnosti [4]. Systém by teda mal byť schopný spracovávať ako monochromatické, tak aj farebné vstupné obrazy.

3.2 Výber segmentov vstupného obrazu

Vstupný neupravovaný obraz je pred samotným spracovaním neurónovou sieťou potrebné upraviť. Prvým krokom je vybrať segment konštantnej veľkosti 20x20 bodov. Výber, realizovaný oknom rovnakej veľkosti, ktoré sa posúva krokom *STEP* po celej ploche obrazu (postupne po riadkoch a stĺpcoch), je poskytnutý ďalším častiam systému kde sa preveruje natočenie a

prítomnosť tváre. Systém je teda schopný nájsť tvár len spomínanej veľkosti – 20x20 bodov.

Po tom ako je spracovaná celá plocha obrazu nasleduje jeho zmenšenie. Hodnotu tohto zmenšenia označíme koeficientom K . Tento krok je realizovaný z dôvodu, že preverované okno má konštantnú veľkosť, ale veľkosť tváre na vstupnom obraze môže byť podstatne väčšia. Obraz zmenšujeme až dovtedy, kým jeden z jeho rozmerov (menší z dvojice) nadobudne veľkosť 20 bodov. Počet zmenšení c , ktoré takto dostaneme možno vyjadriť nasledovne :

$$c = \frac{\log\left(\frac{S}{20}\right)}{\log K} \quad (3.1)$$

pričom S je počet bodov menšieho z rozmerov vstupného obrazu, K je koeficient zmenšovania a c je výsledný počet zmenšení. Číslo 20 označuje veľkosť rozmeru jednej zo strán spracovávaného okna. Hodnotu c je samozrejme na záver nutné zaokrúhliť na jej celú časť.

3.3 Predspracovanie vybraného segmentu

Predpokladajme, že by vybraný segment obsahoval objekt ľudskej tváre. Táto môže byť osvetlená z rôznych strán, primálo, alebo priveľmi. Spôsob vyrovnania týchto nedostatkov budem popisovať v tejto časti.

3.3.1 Odstránenie nerovnomerného osvetlenia – lineárny model osvetlenia

Základnou myšlienkou je aproximovať hodnoty intenzity bodov v okne pomocou lineárnej funkcie a takto získať lineárny model osvetlenia pre spracovávané okno [10].

Postupoval som podľa nasledovných úvah. Ak by sme intenzitu v bode (x,y) označili $I(x,y)$, potom by lineárna funkcia (rovnica priamky) s parametrami a, b, c vyzerala nasledovne

$$(x \ y \ 1) \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = I(x,y) \quad (3.2)$$

Takúto rovnicu je možné vyjadriť pre každý bod v okne a následne z nich zostaviť sústavu rovníc. Výsledkom by bola sústava 400 rovníc (400 bodov v okne) o 1200 neznámych ($a_1, b_1, c_1, \dots, a_{400}, b_{400}, c_{400}$ – parametre priamky pre každý bod). Túto sústavu rovníc je však možné aproximovať a veľký počet neznámych nahradiť iba tromi (parametre priamky a, b, c)

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ \cdot & \cdot & \cdot \\ 20 & 20 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} \approx \begin{pmatrix} I(1,1) \\ I(1,2) \\ \cdot \\ I(20,20) \end{pmatrix} \quad (3.3)$$

čo môžeme zapísať ako

$$\mathbf{Q} \cdot \mathbf{p} \approx \mathbf{I} \quad (3.4)$$

Získanú „sústavu“ možno vyriešiť (aproximovať riešenie) pomocou pseudo-inverznej metódy

$$\mathbf{p} = (\mathbf{Q}^T \cdot \mathbf{Q})^{-1} \cdot \mathbf{Q}^T \cdot \mathbf{I} \quad (3.5)$$

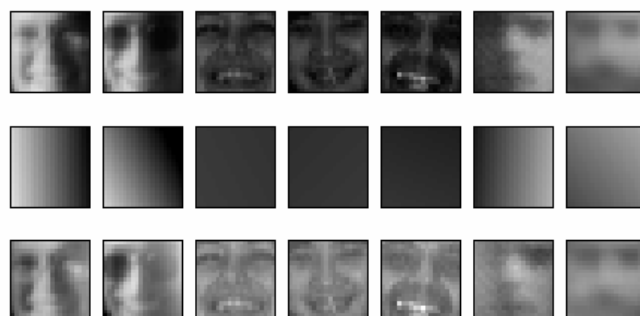
potom platí

$$\mathbf{Q} \cdot \mathbf{p} = \mathbf{I}' \quad (3.6)$$

pričom \mathbf{I}' sú lineárne aproximované intenzity. Ak určíme ich strednú hodnotu a túto od nich odčítame, dostávame maticu \mathbf{M} .

$$\mathbf{I}' - E(\mathbf{I}') = \mathbf{M} \quad (3.7)$$

Ak odčítame jej hodnoty od hodnôt intenzít preverovaného okna dostaneme maticu \mathbf{V} – obraz, v ktorom sú kompenzované zmeny v svetelných podmienkach (časti, ktoré sú prisvetlé sa stmavia a nopak (Obr.3.4)).



Obr. 3.4: Korekcia svetelných podmienok v obraze

3.3.2 Histogramová ekvalizácia

Pokiaľ by boli kontrastné podmienky na spracovávanom obraze nedostačujúce (obraz je príliš tmavý, alebo svetlý) je možné použiť *Histogramovú ekvalizáciu* [4] a odstrániť tieto nedostatky (Obr.3.5). Histogram príliš tmavého, alebo svetlého obrazu sústreďuje väčšinu svojich hodnôt (početností) iba v určitej oblasti farebného spektra. Histogramová ekvalizácia spočíva v rozložení týchto hodnôt po celej osi farebného spektra. Na zvýraznenie kontrastu je možné použiť mnoho ďalších techník., ja som sa však rozhodol použiť práve túto z dôvodu jednoduchosti, rýchlosti a kvality.



Obr. 3.5: Korekcia kontrastu pomocou histogramovej ekvalizácie

3.4 Detekcia natočenia

Blok zabezpečujúci klasifikáciu (detekciu) natočenia musí dokázať určiť otočenie (v stupňoch/radiánoch) vstupného obrazu oproti jeho štandardnej polohe. V tomto prípade ide o problematiku detekcie natočenia ľudskej tváre. Na jej vyriešenie je vhodné použiť **viacvrstvový perceptrón**, známy ako vhodný prostriedok pri klasifikácii vzorov [1].

3.4.1 Viacvrstvový perceptrón pre klasifikáciu natočenia vstupného obrazu

Klasifikáciu natočenia vstupného obrazu môžeme všeobecne popísať ako mapovanie z N rozmerného priestoru¹ do jednej z R výstupných hodnôt².

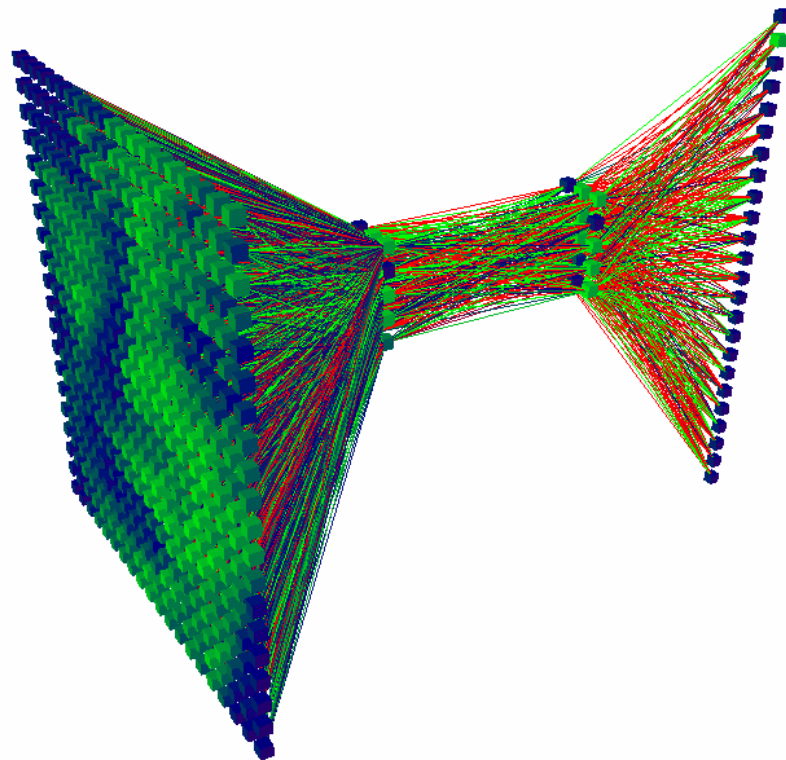
Táto neurónová sieť môže obsahovať rozličný počet skrytých vrstiev o rôznom počte neurónov. Od štruktúry siete závisia jej ďalšie vlastnosti, ako aj to ako kvalitne bude klasifikovať vstupný obraz. Ďalej bude popisovaná sieť,

¹ V našom prípade je $N=400$, pretože spracovávaný segment má rozmery 20×20 bodov

² R je počet natočení, ktoré má byť sieť schopná klasifikovať, teda počet výstupných neurónov

ktorá sa ukázala pri neskorších pokusoch ako najlepšia pre riešenie danej problematiky.

Vstupná vrstva má veľkosť 20×20 , čiže 400 neurónov. Neurónová sieť obsahuje dve skryté vrstvy, pričom každá z nich obsahuje 15 neurónov. Výstupná vrstva obsahuje toľko neurónov, koľko otočení (polôh) vstupného obrazu chceme klasifikovať. Označme tento počet R . Každý z výstupných neurónov klasifikuje jedno z R otočení, pričom každé otočenie je charakterizované počtom stupňov/radiánov. Pokiaľ je teda na vstup privedený obraz tváre s natočením S stupňov bude hodnota odpovede v ideálnom prípade 1 (v skutočnosti je hodnota blízka 1) iba pri neuróne klasifikujúcom toto natočenie. Ostatné neuróny budú mať hodnotu 0 (blízku 0). V každom prípade sa však za víťaza vyhodnotí ten ktorý má spomedzi všetkých neurónov najvyššiu hodnotu odpovede. Podľa toho ktorý neurón zvíťazil, vieme určiť natočenie vstupu.



Obr. 3.6: Priestorové zobrazenie viacvrstvého perceptrónu schopného klasifikovať 24 natočení vstupného obrazu ($R=24$). Klasifikácia tváre natočenej o 15° (Pre výstupné neuróny: Zelená farba – 1, Modrá - 0).

Pokiaľ je na vstup privedený obraz, ktorý neobsahuje vôbec tvár, sieť tomuto vstupu priradí také otočenie, pri ktorom má tvár vo všeobecnosti najpodobnejšie vlastnosti v porovnaní s uvažovaným vstupným obrazom.

Jednotlivé vrstvy tohto viacvrstvového perceptrónu sú úplne prepojené – každý neurón je prepojený so všetkými neurónmi z predchádzajúcej vrstvy (kompletný bigraf) (Obr.3.6).

3.4.2 Učenie neurónovej siete

Jedným z najdôležitejších faktorov pri tréňovaní je vytvoriť vhodnú tréningovú množinu. V našom prípade táto pozostávala z rôznych vzoriek tváre natočených v rôznych polohách (Obr.3.7). Tieto vzorky by mali byť predspracované a mali by spĺňať tieto podmienky :

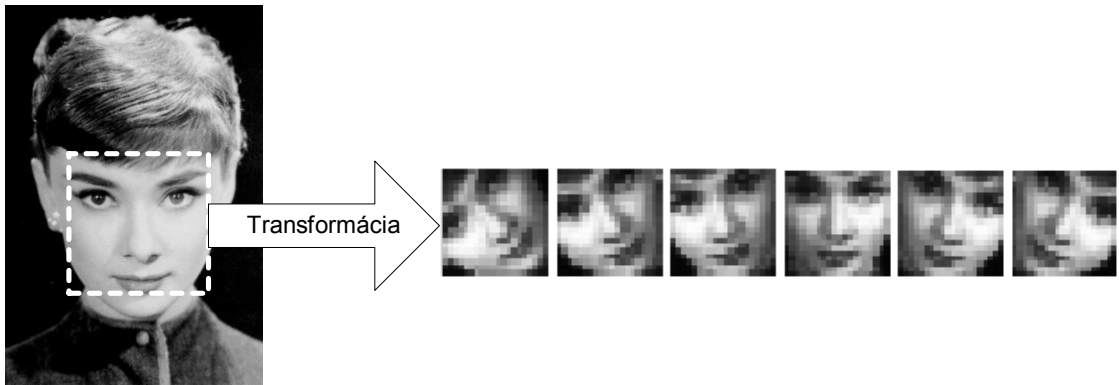
- Rozmer 20x20 bodov
- Umiestnenie nosu približne v strede obrazu
- Normalizované osvetlenie a kontrast



Obr. 3.7: 24 normalizovaných vzoriek vhodných na tréňovanie neurónovej siete pre klasifikáciu natočenia tváre ($R=24$). Ide o vzorky vytvorené rotovaním po 15° ($0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ, 105^\circ, 120^\circ, 135^\circ, 150^\circ, 165^\circ, 180^\circ, 195^\circ, 210^\circ, 225^\circ, 240^\circ, 255^\circ, 270^\circ, 285^\circ, 300^\circ, 315^\circ, 330^\circ, 345^\circ$)

3.4.3 Získanie tvárových vzoriek s rôznym natočením

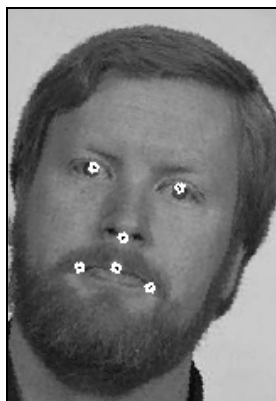
Vzorky tváří sú extrahované a normalizované z obrazov obsahujúcich tváre (Obr.3.8).



Obr. 3.8: Extrahovanie žiadaných normalizovaných výsledkov zo vstupného neupraveného obrazu

Toto je realizované nasledovným spôsobom :

1. Manuálne určíme súradnice dôležitých črt tváre (oči, nos, stred a kútiky úst) na vstupnom obraze (Obr.3.9). Tieto nám slúžia na rotáciu, a normalizáciu veľkosti spracovávaného obrazu do žiadanej normalizovanej formy.



Obr. 3.9: Umiestnenie dôležitých črt na tvári

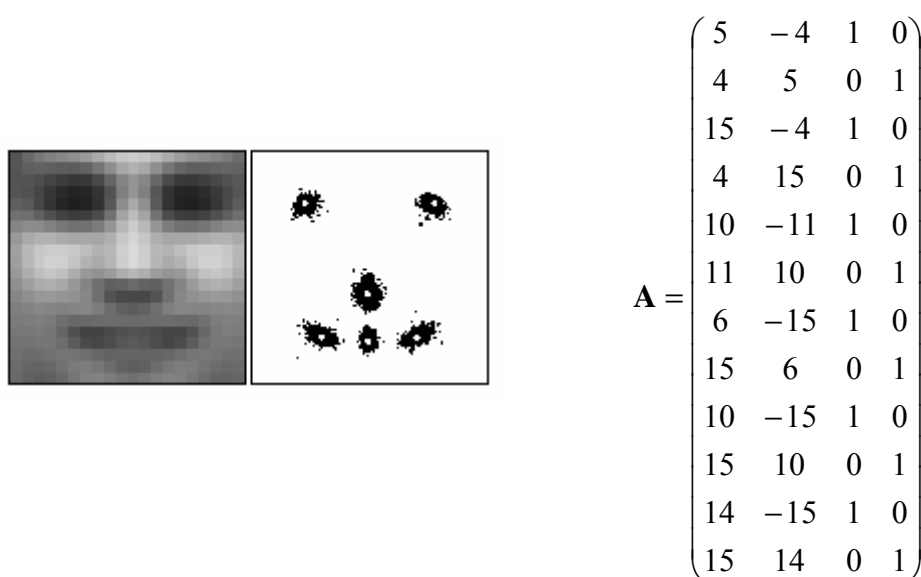
2. Vyjadríme si sústavu rovníc predstavujúcu nami požadovanú transformáciu

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ \vdots \end{pmatrix} \quad (3.8)$$

ktorú môžeme zapísať

$$\mathbf{A} \cdot \mathbf{t} = \mathbf{B} \quad (3.9)$$

kde matica \mathbf{A} obsahuje ideálne súradnice x_i, y_i dôležitých čŕt, \mathbf{t} je transformačný vektor a \mathbf{B} je vektor obsahujúci súradnice x_i', y_i' dôležitých čŕt na vstupnom obrázku. Ideálne súradnice dôležitých čŕt môžeme získať napríklad manuálne z už normovaného obrazu rozmeru 20x20 bodov.



Obr. 3.10: Znáozornenie súradníc dôležitých čŕt na tvári a ich hodnoty v matici \mathbf{A} pre základnú frontálnu polohu

Súradnice zobrazené na Obr.3.10 platia pre *základnú frontálnu polohu* (0°). Pre jednotlivé natočenia sa ideálne súradnice dôležitých čŕt samozrejme menia. Preto je nutné získať ich hodnoty pre všetky žiadané natočenia.

3. Vypočítame hodnotu transformačného vektora pomocou pseudo-inverznej metódy:

$$\mathbf{t} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot (\mathbf{A}^T \cdot \mathbf{I}) \quad (3.10)$$

4. Z prvkov transformačného vektora vyjadríme transformačnú maticu:

$$\mathbf{T} = \begin{pmatrix} a & -b & t_x \\ b & a & t_y \end{pmatrix} \quad (3.11)$$

5. Pomocou transformačnej matice vypočítame súradnice x',y' , ktoré určujú bod v spracovávanom obraze, ktorý zodpovedá súradniciam x,y výsledného obrazu:

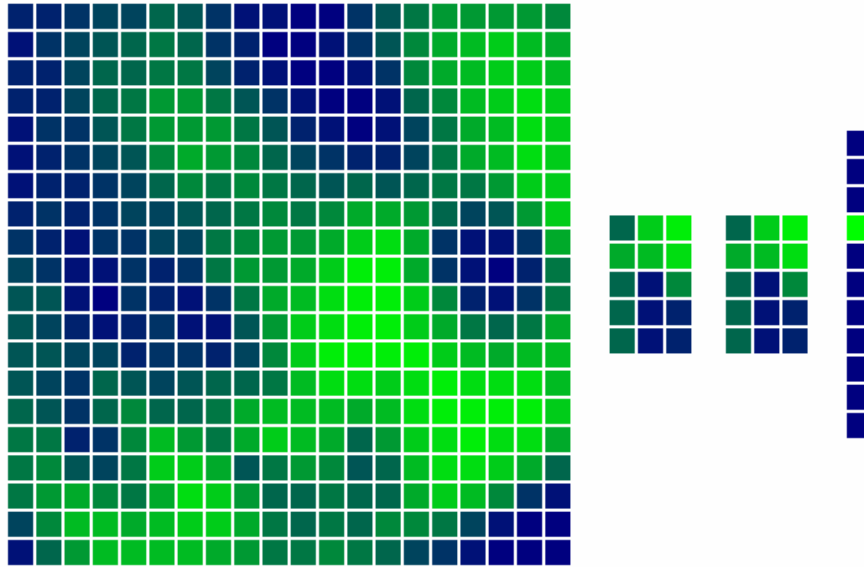
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & -b & t_x \\ b & a & t_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.12)$$

6. Intenzitám výsledného obrazu $I'(x,y)$ ($x = 1..20, y = 1..20$ - 20×20 bodov) priradíme hodnoty intenzít zo spracovávaného obrazu $I(x',y')$.
7. V takto získanom obraze s normovanou pozíciou, veľkosťou a natočením tváre ešte upravíme nerovnomerné osvetlenie a kvôli zvýrazneniu črt naň aplikujeme histogramovú ekvalizáciu.

3.4.4 Proces učenia

Učenie neurónovej siete klasifikujúcej natočenie prebieha v nasledujúcich krokoch:

1. Zostavenie trérovacej množiny P a testovacej množiny Q z čo možno najväčšieho počtu tvárí (X pre P , Y pre Q pričom $Y > X$) vo vhodných otočeniach. Nech počet neurónov vo výstupnej vrstve je R . Potom počet otočení jednej tváre vo vzorkovej množine musí byť tiež R . Jednotlivé tváre budú so vstupných obrazov extrahované na základe hodnôt dôležitých súradníc na vstupných obrazoch a hodnôt ideálnych súradníc pre jednotlivé otočenia postupom popísaným vyššie. Takto zostavená trérovacia množina P bude obsahovať $X.R$ vzoriek a podobne testovacia množina Q bude obsahovať $Y.R$ vzoriek.
2. Inicializácia váh siete náhodnými hodnotami.
3. Trérovanie neurónovej siete klasifikujúcej natočenie na množine P . Sieť bude trérovaná tak, aby hodnota odpovede bola 1 pre ten neurón, ktorý klasifikuje dané otočenie a 0 pre ostatné neuróny (Obr.3.11).



Obr. 3.11: Zobrazenie viacvrstvého perceptrónu schopného klasifikovať 11 natočení vstupného obrazu ($R=11$). Hodnoty výstupných neurónov pri vstupnom otočení 45° . (1. neurón detekcia otočenia 0° , 2. neurón detekcia otočenia 15° , 3. neurón detekcia otočenia 30° , 4. neurón detekcia otočenia 45° , atď)

Trénovanie prebieha na základe štandardného algoritmu spätného šírenia na celej množine P . Počet tréningových cyklov bude C .

4. Testovanie siete, natrérovanej na množine P v C cykloch, na množine Q . Pokiaľ natrérovaná sieť pri testovaní nepodáva dostatočne dobré výsledky, nastane návrat späť do bodu 3. Naopak, pokiaľ sú už výsledky dostatočné, učenie je ukončené a sieť prehlásená za natrérovanú.

3.5 Normalizácia

Vstupom do bloku normalizácie by mal byť predspracovaný segment (veľkosť 20×20 , odstránené nerovnomerné osvetlenie, aplikovaná histogramová ekvalizácia) a hodnota natočenia pre spracovávaný segment.

Hodnotu natočenia však nedostaneme priamo v stupňoch resp. radiánoch. Musíme ju vyjadriť vzhľadom na výstup daný neurónovou sieťou pre klasifikáciu natočenia. Celú túto transformáciu môžeme vyjadriť nasledovne:

1. Nech výstup neurónovej siete je vektor $\mathbf{v} = (v_1, v_2, \dots, v_R)$. Potom existuje $v_i \geq v_j$ pričom $j=1..R, j \neq i$. Potom môžeme vyjadriť vektor $\mathbf{v}' = (0, 0, \dots, x_i, \dots, 0)$, kde $x_i = 1$. Vektor \mathbf{v}' teda môže nadobudnúť práve R hodnôt.

2. Nech f je funkcia ktorá ku každej hodnote vektora \mathbf{v}' priradí dopredu definovanú hodnotu otočenia v stupňoch. Použitím funkcie f teda dostávame žiadané natočenie W .

V bloku normalizácie je zahrnutá aj spätná rotácia obrazu v predspracovanom segmente a odstránenie artefaktov pozadia nenesúcich informáciu o charaktere tváre.

3.5.1 Spätná rotácia

Pomocou techník známych z problematiky číslcového spracovania obrazu môžeme na základe uhla otočenia W „derotovať“ obraz v spracovávanom segmente okolo jeho približného stredu, teda bodu so súradnicami (10,10).

3.5.2 Aplikovanie masky

Štvorcový spracovávaný segment obsahuje okrem centrálne situovanej "derotovanej" tváre ešte aj artefakty pozadia v rohoch obrazu. Tieto artefakty môžu vnieť do okna nesúceho informáciu o tvári istú neurčitosť. Je preto dobré aj túto maličkosť odstrániť aplikovaním masky (Obr.3.12). Maska môže mať rozličný tvar, treba ju však zvoliť tak, aby potlačila iba nepotrebnú informáciu o pozadí.



Obr. 3.12: Aplikácia masky na obraz

3.6 Detekcia prítomnosti tváre

Blok detekcie prítomnosti tváre možno popísať ako klasifikáciu toho, či spracovávaný normalizovaný segment obsahuje tvár, alebo nie. Túto

problematiku možno riešiť prostredníctvom neurónovej siete. Ako typ siete bol opäť vybraný viacvrstvový perceptrón.

3.6.1 Viacvrstvový perceptrón pre klasifikáciu vstupného obrazu

Pri tejto klasifikácii ide o mapovanie z N rozmerného priestoru³ do jednorozmerného priestoru⁴. Podľa hodnoty odpovede siete vieme potom určiť s akou pravdepodobnosťou bol na vstup privedený obraz tváre.

Aj v tomto prípade môže neurónová sieť obsahovať rozličný počet skrytých vrstiev. Každá z týchto vrstiev však môže obsahovať podsiete s rozličnou štruktúrou. Opäť platí, že od štruktúry siete závisia jej vlastnosti. Popisovaná bude sieť, ktorá pri neskorších pokusoch vykazovala najlepšie vlastnosti.

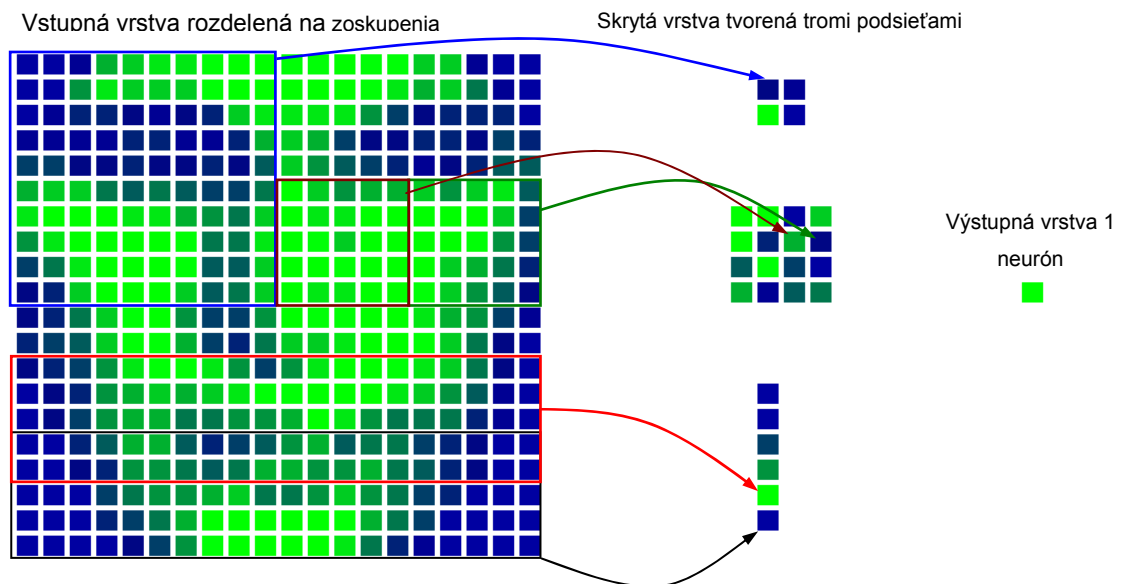
Táto neurónová sieť má jednu skrytú vrstvu, ktorá je zostavená z troch podsietí. Každá z nich analyzuje iné časti vstupného obrazu a ich výsledky sa navzájom dopĺňajú (Obr.3.14).

Vstupná vrstva veľkosti 20x20 neurónov (spracovávaný je obraz 20x20 bodov) je spoločná pre všetky podsiete. Je rozdelená na menšie zoskupenia neurónov (cluster, zhluk). Neuróny v jednom zoskupení sú úplne prepojené so zodpovedajúcim neurónom jednej z troch skrytých podsietí. V každom neuróne jednotlivých podsietí sa teda analyzuje iná časť vstupného obrazu (Obr.3.13):

- Štyri neprekrývajúce sa regióny veľkosti 10x10 - analýza prítomnosti väčších čŕt ako oči, lícne kosti atď.
- Šestnásť neprekrývajúcich sa regiónov veľkosti 5x5 - analýza prítomnosti menších čŕt ako kútiky úst, častí brady, nosa atď.
- Šesť prekrývajúcich sa regiónov veľkosti 20x5 – analýza prítomnosti páru očí, nosu, úst atď.

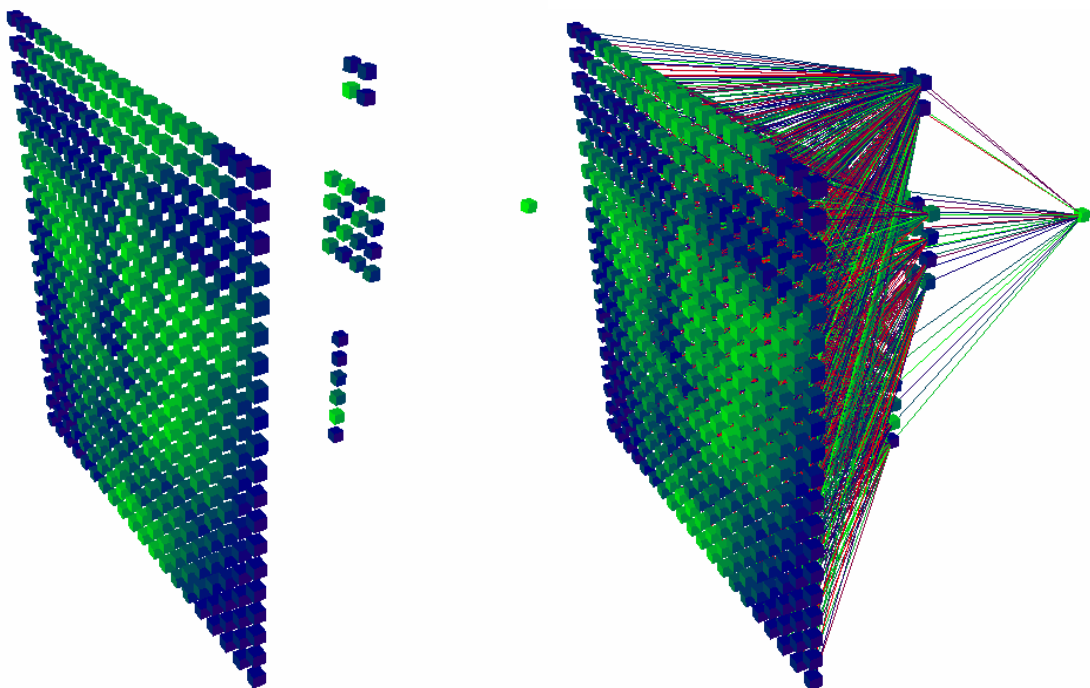
³V našom prípade je $N=400$, 20x20 vstupných neurónov.

⁴ 1 výstupný neurón.



Obr. 3.13: Znáznorenie regiónov vo vstupnej vrstve a ich spracovanie v skrytej vrstve

Výstupná vrstva je tvorená jedným neurónom a je spoločná pre všetky tri podsiete. Znamená to že neuróny všetkých podsietí sú úplne prepojené s neurónom výstupnej vrstvy. Tento neurón určuje či vstupný obraz obsahuje tvár, alebo nie (1 = obsahuje tvár, 0 = neobsahuje tvár).



Obr. 3.14: Priestorové zobrazenie viacvrstvého perceptrónu určeného na klasifikáciu obrazu bez spojení a so spojeními

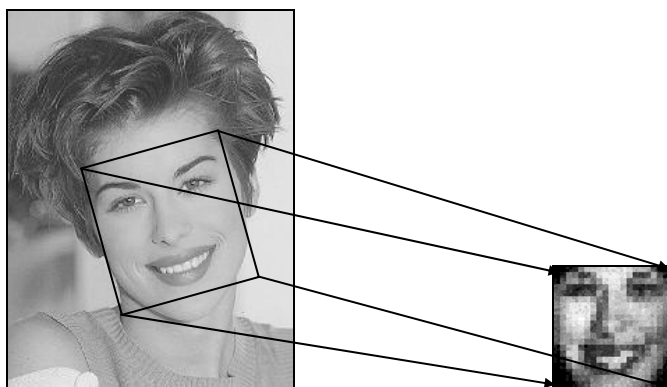
3.6.2 Učenie neurónovej siete

Pre potreby učenia je nutné znova vytvoriť vhodnú trénovaciu množinu. V našom prípade táto pozostáva z tvárových a netvárových vzoriek. Tvárové vzorky by mali byť normalizované a mali by spĺňať tieto podmienky :

- Rozmer 20x20 bodov
- Tvár by mala byť bez akéhokoľvek natočenia
- Umiestnenie nosu približne v strede obrazu
- Normalizované osvetlenie a kontrast
- Odstránenie zvyškov pozadia v rohoch obrázku

3.6.3 Získanie frontálnych tvárových vzoriek:

Vzorky tvárí sú extrahované a normalizované z obrazov obsahujúcich tváre podobným spôsobom ako v časti 3.4.3 (Obr.3.15). V tomto prípade sú potrebné len frontálne vzorky bez akéhokoľvek natočenia, takže použitá bude iba jedna matica ideálnych pozícií dôležitých črt. K celému postupu je nutné ešte na záver pridať aplikovanie masky odstraňujúcej nežiaduce elementy v rohoch obrazu.



Obr. 3.15: Extrahovanie žiadaného normalizovaného výsledku zo vstupného neupraveného obrazu

3.6.4 Získanie netvárových vzoriek

Prakticky ľubovoľný obraz neobsahujúci ľudské tváre môže slúžiť ako zdroj netvárových vzoriek. Z tohto vyplýva, že priestor netvárových vzoriek je mnohonásobne väčší ako priestor vzoriek tvárí. Tým pádom je zložité vybrať množinu vzoriek, ktorá bude reprezentovať čo najväčšiu časť netvárového priestoru. Situácia sa dá riešiť tak, že negatívne vzorky sa nevyberajú iba pred

trénovaním, ale aj priamo počas neho. Táto technika nazývaná aj *bootstrap algoritmus* [10] bude lepšie popísaná v časti týkajúcej sa tréovania neurónovej siete.

Tak ako pri tvárových vzorkách, tak aj pri netvárových musia byť splnené určité požiadavky :

- Rozmer vzorky je 20x20 bodov
- Pred tým ako vzorka vstúpi do neurónovej siete musí prejsť procesom normalizácie rovnakým ako pri tvárových vzorkách (vyrovnanie svetelných pomerov v obraze, histogramová ekvalizácia, nasadenie masky)

3.6.5 Proces učenia

Učenie neurónovej siete klasifikujúcej obsah prebieha v nasledujúcich krokoch:

1. Zostavenie inicializačnej trénovacej množiny I pozostávajúcej z X tvárových a X netvárových vzoriek. Tvárové vzory sú získané spôsobom popísaným vyššie. Netvárové vzorky sú náhodne generované (každá má 20x20 bodov, je tvorená náhodne generovanými intenzitami (rozsah 0..255), a sú na ňu aplikovane všetky kroky predspracovania).
2. Inicializácia váh siete náhodnými hodnotami.
3. Trénovanie siete na množine I , pričom učenie spočíva v tom, že hodnota odpovede siete na tvárovú vzorku je 1 a na netvárovú 0. Trénovanie prebieha na základe štandardného algoritmu spätného šírenia v C cykloch.
4. Testovanie takto natrénovanej siete na množine obsahujúcej iba netvárové vzorky získané výberom okien veľkosti 20x20 bodov z obrazov neobsahujúcich tváre⁵. Označme túto množinu ako P . Veľkosť tejto množiny je niekoľkonásobne väčšia ako veľkosť trénovacej množiny.
5. Zozbieranie tých vzoriek množiny P , na ktoré sieť zareagovala ako na tváre. Túto množinu označme Q .

⁵ Aj tu musia byť aplikované všetky kroky predpracovania

6. Náhodný výber vzoriek z množiny Q a ich zaradenie do trénovacej množiny I . Pokiaľ natrénovaná sieť ešte stále nepodáva dostatočne dobré výsledky návrat späť do bodu 3.

3.6.6 Vyhodnotenie

Posledný blok celého systému realizuje vyhodnotenie procesu lokalizácie tváří. Prvým krokom je vyhodnotiť odpoveď neurónovej siete klasifikujúcu obsah. Výstupná hodnota tejto siete (iba jeden výstupný neurón) je v rozsahu $\langle 0,1 \rangle$, čo možno vyjadriť aj percentuálne $\langle 0\%,100\% \rangle$. Je teda potrebné určiť prah T akceptovania toho, že na vstupe je skutočne tvár.

Na konci teda máme dostatok informácií na to, aby sme na vstupnom obraze mohli presne označiť všetky ľudské tváre:

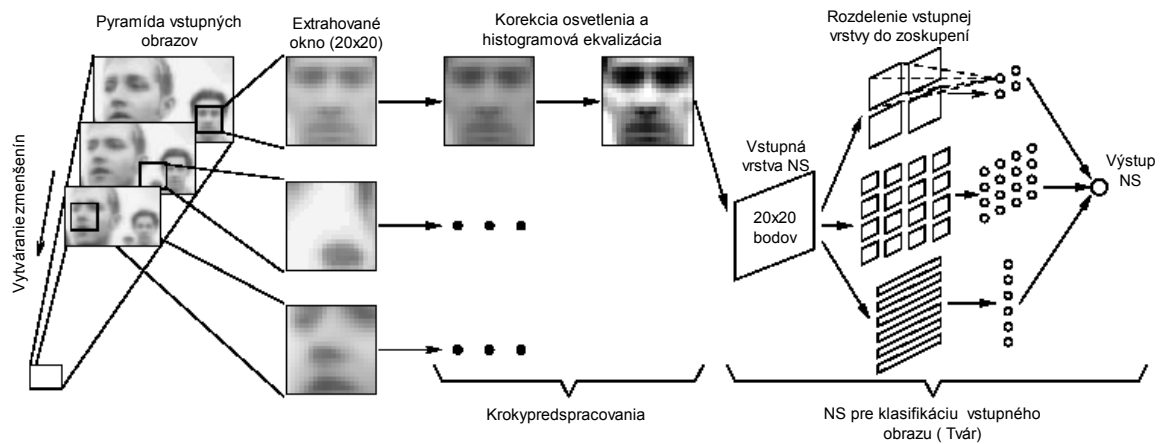
- Pozície segmentov, ktorých normalizovaný obsah bol v bloku detekcie prítomnosti tváre prehlásený za tvár
- Hodnoty natočenia tváre pre všetky tieto segmenty
- Hodnoty zmenšení vstupného obrazu pri ktorých boli segmenty vybrané

3.7 Funkčné schémy navrhovaných systémov

Na záver tejto kapitoly by som ešte chcel uviesť schematický obraz popisovaných systémov, v ktorých by bola znázornená aj funkčnosť jednotlivých častí navrhovaných systémov.

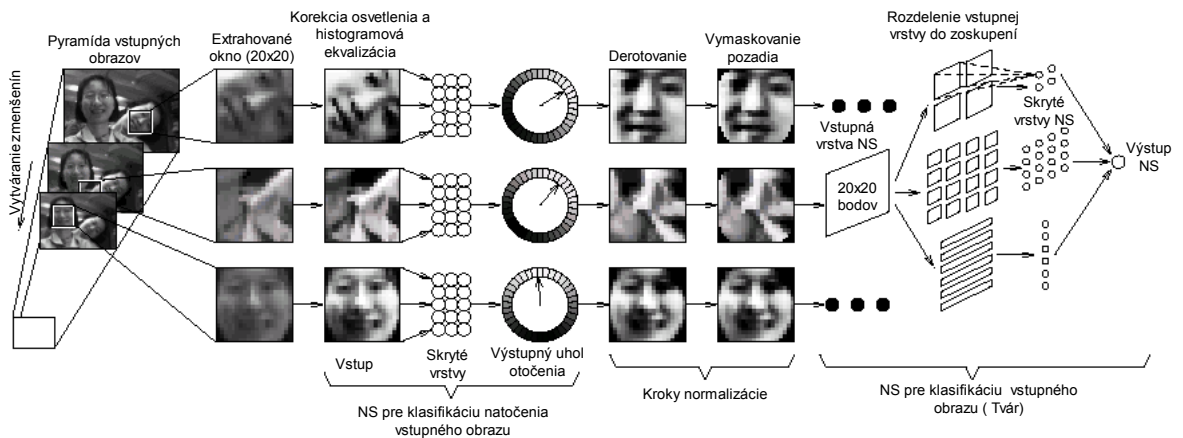
Pre názornosť uvediem dve funkčné schémy. V prvej je znázornený systém, ktorý dokáže lokalizovať iba frontálne tváre bez natočenia (Obr.3.16). Naopak v druhej je systém, ktorý dokáže lokalizovať aj natočené tváre (Obr.3.17).

3.7.1 Funkčná schéma systému bez detekcie natočenia



Obr. 3.16: Funkčná schéma systému bez detekcie natočenia

3.7.2 Funkčná schéma systému s detekciou natočenia



Obr. 3.17: Funkčná schéma systému s detekciou natočenia

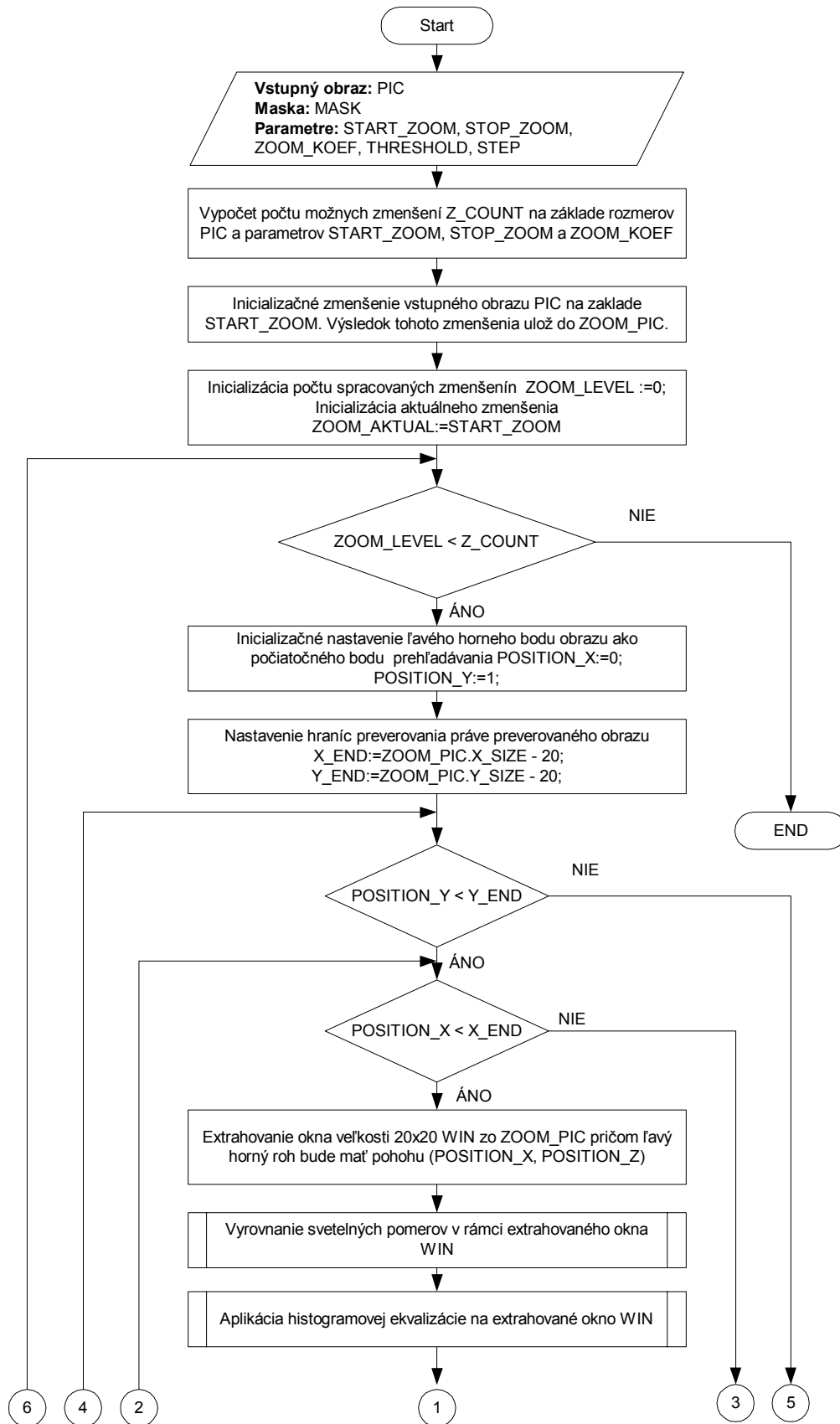
4 Implementácia systému a teoretických princípov

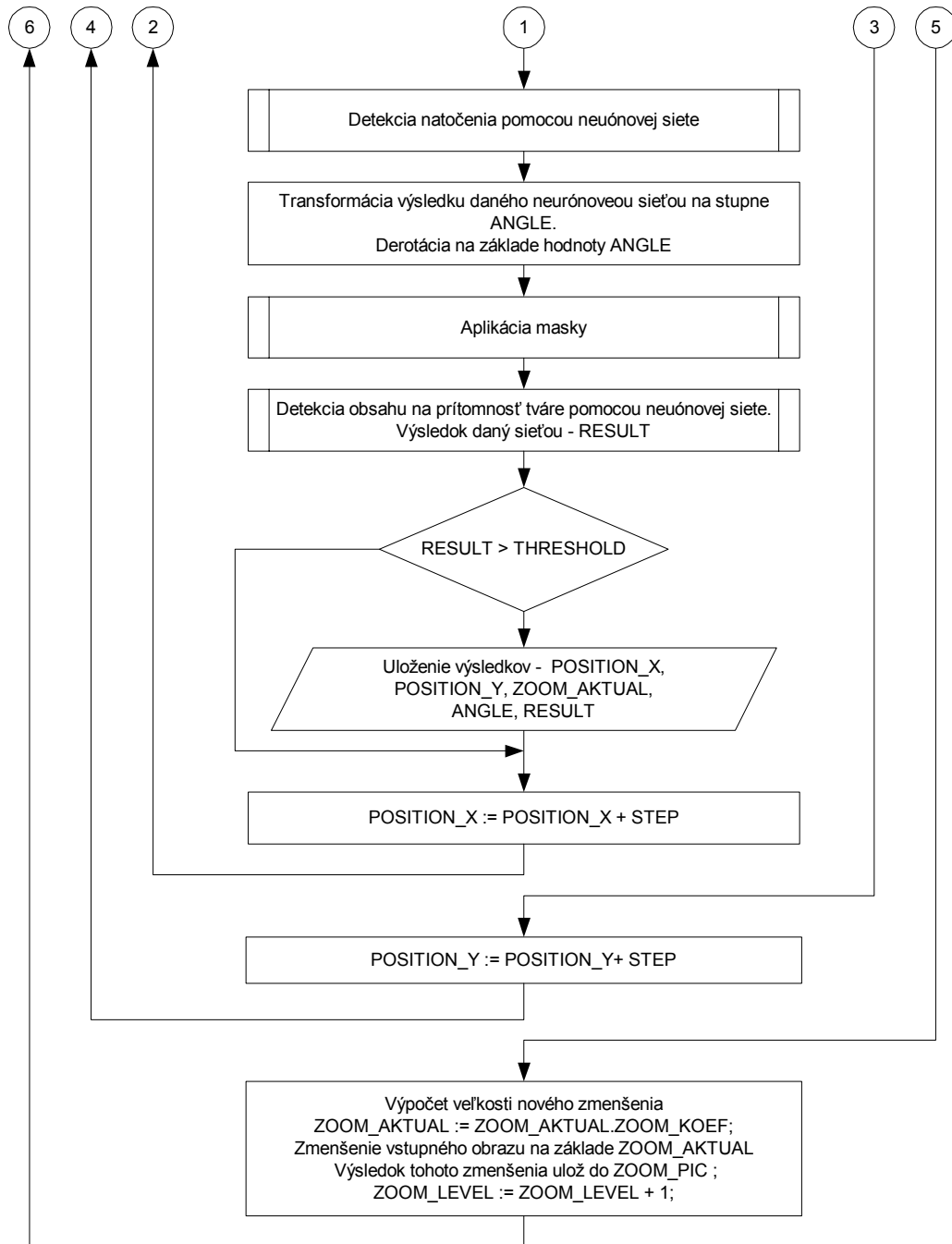
Systém na detekciu ľudských tvárí navrhnutý v teoretickej časti som sa rozhodol implementovať pomocou prostriedkov operačného systému Windows XP, Simulátoru Neurónových sietí SNNS 4.1 a SNNS 4.2 [16], programového prostredia Microsoft Visual C++ 6.0 a matematického programu Matlab 6.0.

Pomocou týchto nástrojov som postupne vytváral funkčné bloky (bloky neurónových sietí, blok na predspracovanie atď.). Na základe čiastkových výsledkov som sa rozhodoval a pokračoval v ďalších krokoch ich implementácie. Pre jednoduchosť a rýchlosť vývoja algoritmov som najprv väčšinu algoritmov preveril prostredníctvom programu MATLAB. Rýchlosť takto napísaných funkcií a procedúr však nebola dostatočujúca a preto som bol nútený všetky tieto algoritmy prepísať do jazyka C.

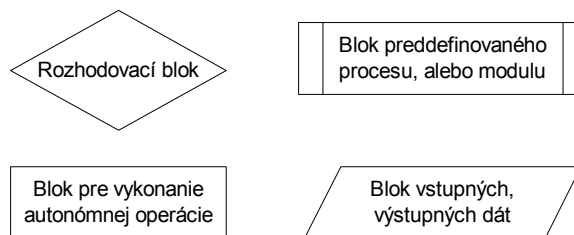
V tejto časti by som na úvod uviedol vývojový diagram implementácie celého systému na lokalizáciu ľudských tvárí (Obr.4.1). Následne by som podrobne popísal jeho najdôležitejšie časti - obe neurónové siete spolu s procesmi ich učenia. Nakoniec by som stručne popísal implementáciu ostatných častí systému

4.1 Vývojový diagram procesu lokalizácie ľudských tvárí





Obr. 4.1: Vývojový diagram procesu lokalizácie ľudských tvárí



Obr. 4.2: Popis funkčnosti jednotlivých blokov vo vývojovom diagrame

4.2 Vstupy a výstupy systému

Vstupné parametre

Vstupný obraz PIC:



Maska MASK:

Maska použitá pri normalizácii



Parameter START_ZOOM

Systém dokáže nájsť ľudské tváre rozmerov 20x20 obrazových bodov. Preto ak pri jeho použití vieme dopredu približne určiť veľkosť tváre na vstupnom obraze a teda aj zmenšenie ktoré transformuje vstupnú tvár približne do rozmerov 20x20 bodov, môžeme túto informáciu uviesť v tomto parametri. Podstatne to urýchli činnosť systému. Ak jeho hodnotu nevieme môžeme udať hodnotu 1. V tom prípade sa bude preverovať vstupný obraz počínajúc pôvodnými rozmermi.

Parameter STOP_ZOOM

Podobný význam ako predchádzajúci parameter, s tým rozdielom, že ide o zmenšenie pri ktorom preverovanie skončí. Ak jeho rozmer nevieme môžeme udať hodnotu 0. V tom prípade sa bude obraz preverovať až do najmenšej možnej veľkosti (20x20bodov).

Parameter ZOOM_KOEF

Koeficient, ktorý určuje hodnotu zmenšenia v pyramíde preverovaných obrazov

Parameter THRESHOLD	Prahová hodnota označenia vstupu za tvár.
Parameter STEP	Počet bodov posúvania okna na extrahovanie segmentov veľkosti 20x20.

Hodnoty parametrov START_ZOOM, STOP_ZOOM, ZOOM_KOEF, THRESHOLD a STEP závisia od konkrétnej aplikácie. Nemožno pre ne preto určiť všeobecne platné hodnoty.

Vstupné parametre

POSITION_X, POSITION_Y	X-ová a Y-ová pozícia ľavého horného rohu preverovaného segmentu WIN v aktuálnej zmenšenej vstupného obrazu ZOOM_PIC.
ZOOM_AKTUAL	Veľkosť zmenšenia vstupného obrazu PIC.
ANGLE	Uhol natočenia obsahu preverovaného segmentu WIN.
RESULT	Pravdepodobnosť že MASK_DEROT_WIN obsahuje tvár.
MASK_DEROT_WIN	Obsah preverovaného okna je uložený ako obraz vo formáte BMP.



Celkový výsledný obraz



4.3 Blok neurónovej siete pre detekciu natočenia vstupného obrazu

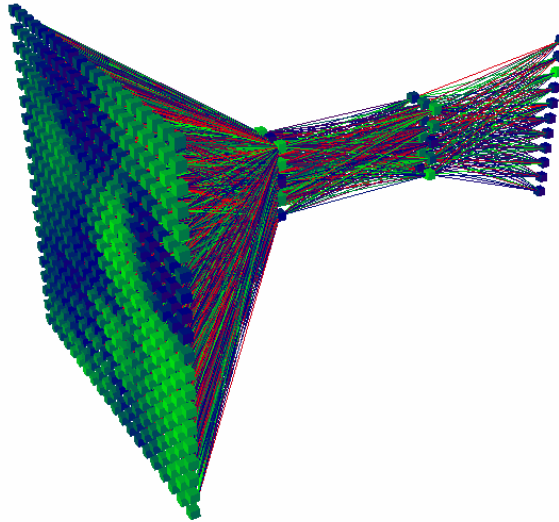
4.3.1 Vytvorenie neurónovej siete

Neurónovú sieť som vytvoril v prostredí simulátora SNNS. Na základe testov a výsledkov, ktoré budú ukázané v kapitole *Experimentálne výsledky* som sa dopracoval k neurónovej sieti, ktorú by som teraz chcel podrobne popísať (Obr.4.3).

Základné parametre siete:

Počet skrytých vrstiev:	2
Celkový počet neurónov:	441
Počet vstupných neurónov:	400
Počet neurónov v 1. skrytej vrstve:	15
Počet neurónov v 2. skrytej vrstve:	15
Počet výstupných neurónov:	11
Počet prepojení:	6390
Typ aktivačnej funkcie:	Sigmoidálna
Algoritmus učenia:	Štandardné učenie so spätným šírením
Parameter rýchlosti učenia:	0.2
Parameter μ^6 :	0.0
Typ inicializačnej funkcie váh:	Náhodne v rozmedzí $\langle -1, 1 \rangle$

⁶ Maximálna prípustná hodnota diferencie medzi žiadaným a obdržaným výsledkom



Obr. 4.3: Priestorové zobrazenie výslednej neurónovej siete pre detekciu natočenia

4.3.2 Trénovanie neurónovej siete

Na trénovanie som použil vnútorný skriptovací jazyk simulátoru SNNS. Skript vytvorený v tomto jazyku dokáže interpretovať interpreter príkazov SNNS **batchman** [16].

Trénovanie prebiehalo v nasledujúcich krokoch:

1. Načítanie nenatrénovanej siete.
2. Nastavenie parametrov siete (Typ inicializačnej funkcie váh, Algoritmus učenia).
3. Náhodná inicializácia váh siete.
4. Uloženie takto nainicializovanej siete.
5. Načítanie *trénovacej* a *testovacej množiny*.
6. Nastavenie režimu trénovania – s premiešavaním vzoriek⁷.
7. Inicializácia premenných potrebných pri učení:

$MinSSE = \infty$	Minimum SSE (stredná kvadratická odchýlka)
$MinSSECycles = 0$	Počet cyklov pri učení kedy nastalo minimum $MinSSE$

⁷ Premiešavanie vzoriek pri trénovaní znamená, že vzorky sú z trénovacej množiny v každom cykle vyberané v inom, náhodnom poradí. Pri režime trénovania s premiešavaním vzoriek sú dosiahnuté podstatne lepšie výsledky z hľadiska generalizujúcich vlastností siete.

$MinBadClass = \infty$ Minimum zlých klasifikácií

$MinBadClassCycles = 0$ Počet cyklov pri učení kedy nastalo minimum
 $MinBadClass$

8. Učenie siete na *trénovacej množine* v 150 cykloch, pričom po každom cykle (počítadlo počtu cyklov *CYCLE*) je sieť testovaná na *testovacej množine* a sú vykonané kroky číslo 9, 10, 11 a 12.
9. Pokiaľ je hodnota aktuálnej SEE (získanej pri testovaní) *AktualSSE* menšia ako hodnota premennej *MinSSE* aktualizuj obsah nasledujúcich premenných:
 $MinSSE = AktualSSE$
 $MinSSECycles = CYCLE$
10. Pokiaľ je hodnota aktuálneho počtu zlých klasifikácií *AktualBadClass* menšia ako hodnota premennej *MinBadClass* aktualizuj obsah nasledujúcich premenných:
 $MinBadClass = AktualBadClass$
 $MinBadClassCycles = CYCLE$
11. Zápis výsledkov testovania (Číslo cyklu tréovania, SSE, počet zle klasifikovaných vzoriek [%], počet dobre klasifikovaných vzoriek [%]).
12. Uloženie siete natrénovanej v *CYCLE* cykloch.
13. Po skončení učenia nasleduje zápis hodnôt *MinSSE*, *MinSSECycles*, *MinBadClass*, *MinBadClassCycles*.

V bodoch 9 a 10 je popísané hľadanie globálneho minima chybovej funkcie pre testovaciu množinu, jednak vzhľadom na parameter SSE a jednak vzhľadom na počet zlých kvalifikácií. Toto je realizované z dôvodu možného pretrénovania siete. Na konci učenia (po 150 cykloch⁸) teda stačí zobrať siete uložené po *MinSSECycles* a *MinBadClassCycles* cykloch a porovnať ich vlastnosti na základe výsledkov testovania uložených v bode 11. Výsledkom by mala sieť s najlepšimi generalizujúcimi vlastnosťami.

⁸Počet cyklov v ktorých sa hľadá globálne minimum je určený empiricky. Podľa spravených testov, nastane minimum väčšinou v prvých 70 cykloch. Následná rezerva 80 cyklov je preto postačujúca.

Trénovacia množina pozostávala zo 3850 vzoriek, pričom tieto boli extrahované z 350 tvárí v 11 pozíciách (Natočenie 0°, 15°, 30°, 45°, 60°, 75°, 285°, 300°, 315°, 330°, 345°).

Testovacia množina mala presne také isté zloženie, ale vzorky boli extrahované z inej množiny tvárí.

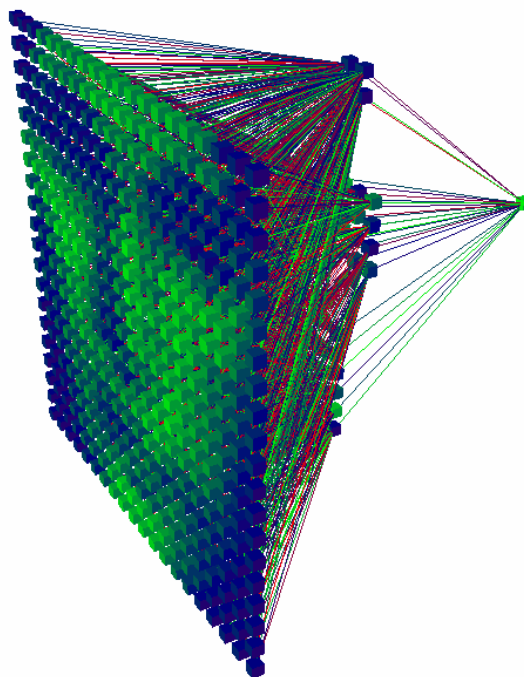
4.4 Blok neurónovej siete pre klasifikáciu vstupného obrazu

4.4.1 Vytvorenie neurónovej siete

Pri vytváraní tejto neurónovej siete (Obr.4.4) som postupoval obdobne ako v predchádzajúcom prípade.

Základné parametre siete:

Počet skrytých vrstiev:	1
Celkový počet neurónov:	427
Počet vstupných neurónov:	400
Počet neurónov v skrytej vrstve:	26
Počet výstupných neurónov:	1
Počet prepojení:	1426
Typ aktivačnej funkcie:	Sigmoidálna
Algoritmus učenia:	Štandardné učenie so spätným šírením
Parameter rýchlosti učenia:	0.2
Parameter μ :	0.0
Typ inicializačnej funkcie váh:	Náhodne v rozmedzí $\langle -1, 1 \rangle$



Obr. 4.4: Priestorové zobrazenie výslednej neurónovej siete pre klasifikáciu vstupného obrazu

4.4.2 Trénovanie neurónovej siete

Pri trénovaní boli použité rovnaké prostriedky ako v predchádzajúcom prípade avšak proces trénovania bol odlišný.

Trénovanie pozostáva z dvoch častí. **Prvá časť** prebiehala v nasledujúcich krokoch:

1. Načítanie nenatrénovanej siete.
2. Nastavenie parametrov siete (Typ inicializačnej funkcie váh, Algoritmus učenia).
3. Náhodná inicializácia váh siete.
4. Uloženie takto nainicializovanej siete.
5. Načítanie *základnej trénovacej a základnej testovacej množiny*.
6. Nastavenie režimu trénovania – bez premiešavania vzoriek⁹.

⁹ Pri trénovacom režime bez premiešavania vzoriek vykazuje chybová funkcia (Závislosť SSE od počtu cyklov trénovania) hladký priebeh (bez výrazných prekmitov). Preto sa najprv určí globálne minimum chybovej funkcie z tohto hladkého priebehu. Nasledovať bude trénovanie s premiešavaním vzoriek, ale iba v počte cyklov, kedy bolo dosiahnuté spomínané globálne minimum.

7. Inicializácia premenných potrebných pri učení¹⁰:

$MinSSE = \infty$ Minimum SSE (stredná kvadratická odchýlka)

$MinSSECycles = 0$ Počet cyklov pri učení kedy nastalo minimum
 $MinSSE$

8. Učenie siete na *základnej trénovacej množine* v 300 cykloch, pričom po každom cykle (počítadlo počtu cyklov *CYCLE*) je sieť testovaná na *základnej testovacej množine* a je vykonaný krok číslo 9.

9. Pokiaľ je hodnota aktuálnej SEE (získanej pri testovaní) *AktualSSE* menšia ako hodnota premennej *MinSSE* aktualizuj obsah nasledujúcich premenných:

$MinSSE = AktualSSE$

$MinSSECycles = CYCLE$

10. Načítanie nainicializovanej siete uloženej v bode 4.

11. Nastavenie režimu tréovania – s premiešavaním vzoriek.

12. Učenie siete na *základnej trénovacej množine* v počte cyklov *MinSSECycles*.

13. Uloženie takto natrénovanej siete – *NetBasicTrain*.

14. Testovanie siete *NetBasicTrain* na množine *NoFaces* obsahujúcej iba netvárové vzorky.

Aj v tomto prípade prebieha hľadanie globálneho minima chybovej funkcie pre *základnú testovaciu množinu*, vzhľadom na parameter SSE (9. bod). Globálne minimum sa teda hľadá v rámci prvých 300 trénovacích cyklov¹¹.

Zloženie *základnej trénovacej množiny*:

- 250 tvárových normalizovaných vzoriek
- 250 netvárových normalizovaných náhodne generovaných vzoriek

Zloženie *základnej testovacej množiny*:

¹⁰ V tomto prípade už nie je kontrolovaný počet nesprávne klasifikovaných vzoriek ako pri tréovaní predchádzajúcej siete. Toto je spôsobené tým, že teraz tento počet priamo závisí od SSE. Empiricky bolo totiž odpozorované, že v cykle kedy bolo najnižšie SSE bol aj najnižší počet nesprávne klasifikovaných vzoriek.

- 250 tvárových normalizovaných vzoriek
- 250 netvárových normalizovaných náhodne generovaných vzoriek¹²

Zloženie množiny *NoFaces* obsahujúcej iba netvárové vzorky:

- 4786 netvárových normalizovaných vzoriek. Tieto vzorky boli získané pri testovaní siete *NetBasicTrain* natrénovanej v 1. časti tréningu. Táto sieť bola totiž použitá v systéme na lokalizáciu ľudských tvárí a jej chybné detekcie boli vyhlásené za vhodné netvárové vzorky. Spomenuté nesprávne detekcie boli vybrané manuálne spomedzi všetkých výsledkov systému.

Na základe výsledku testov na množine *NoFaces* (14. bod) vieme určiť na ktoré netvárové vzorky zareagovala sieť opäť nekorektné - označila ich za tváre. Z týchto vzoriek náhodne vyberieme množiny, ktorými obohatíme základnú tréningovú množinu a základnú testovaciu množinu. Získame teda dve nové množiny vzoriek a to síce rozšírenú tréningovú množinu a rozšírenú testovaciu množinu.

Zloženie rozšírenej tréningovej množiny:

- 250 tvárových normalizovaných vzoriek
- 250 netvárových normalizovaných náhodne generovaných vzoriek
- 2000 vyselektovaných netvárových vzoriek

Zloženie rozšírenej testovacej množiny:

- 250 tvárových normalizovaných vzoriek
- 250 netvárových normalizovaných náhodne generovaných vzoriek.
- 2400 vyselektovaných netvárových vzoriek

Druhá časť tréningu prebiehala v nasledujúcich krokoch:

1. Načítanie natrénovanej siete *NetBasicTrain* z 1. časti tréningu
2. Načítanie rozšírenej tréningovej a rozšírenej testovacej množiny
3. Nastavenie režimu tréningu – bez premiešavania vzoriek.

¹¹ Opäť ide o mnou empiricky určenú hodnotu. Minimum je totiž dosiahnuté v prvých 150 cykloch. Rezerva 150 cyklov je preto dostatočná.

¹² Tvárové a netvárové vzorky sú iné ako pri základnej tréningovej množine

4. Inicializácia premenných potrebných pri učení:

$MinSSE = \infty$	Minimum SSE (stredná kvadratická odchýlka)
$MinSSECycles = 0$	Počet cyklov pri učení kedy nastalo minimum

$MinSSE$
5. Učenie siete na *rozšírenej trérovacej množine* v 1500 cykloch, pričom po každom cykle (počítadlo počtu cyklov *CYCLE*) je sieť testovaná na *rozšírenej testovacej množine* a je vykonaný krok číslo 6.
6. Pokiaľ je hodnota aktuálnej SEE (získanej pri testovaní) *AktualSSE* menšia ako hodnota premennej *MinSSE* aktualizuj obsah nasledujúcich premenných:

$MinSSE = AktualSSE$
$MinSSECycles = CYCLE$
7. Opätovné načítanie siete *NetBasicTrain*.
8. Nastavenie režimu trérovania – s premiešavaním vzoriek.
9. Učenie siete na *základnej trérovacej množine* v počte cyklov *MinSSECycles*.
10. Uloženie takto natrérovanej siete – *NetAdvancedTrain*.

Význam jednotlivých krokov je obdobný ako pri trérovaní v prvej časti. Opať sa hľadá globálne minimum chybovej funkcie SSE (6. bod). Sieť je potom natrérovaná iba v počte cyklov v ktorom nastalo toto minimum.

4.5 Implementácia neurónových sietí

Zo súboru obsahujúceho údaje o natrérovanej sieti je možné pomocou programu **snns2c** [16] vytvoriť implementáciu danej siete v prostredí programovacieho jazyka C. Táto sa dá prekompilovať na dynamickú knižnicu (dll) a používať ako vymeniteľná časť systému. Pokiaľ teda budú dodržané základné parametre pri použití sietí (počet vstupov a výstupov) je možné bez nutného zásahu do samotného kódu programu použiť sieť s ľubovoľnou vnútornou štruktúrou natrérovanú ľubovoľným spôsobom.

4.6 Ostatné bloky systému

Bloky systému zabezpečujúce extrakciu okien na preverovanie, vyrovnávanie svetelných pomerov, aplikáciu histogramovej ekvalizácie a

masky, zmenu veľkosti obrazu, jeho rotovanie, prípadne ďalšie výpočty súvisiace so spracovaním obrazu boli realizované presne podľa teoretických poznatkov popísaných v 3. kapitole.

Prostriedkom na ich implementáciu bol jazyk C v spojení s knižnicami pre spracovanie obrazu [17][18]. Tieto knižnice boli vyvinuté a optimalizované firmou INTEL pre procesorovú radu INTEL X86. Vďaka tomuto faktoru a faktoru, že z pôvodného prostredia MATLAB som prešiel na nižší programovací jazyk C som dosiahol rádovo 10-násobné zrýchlenie celého procesu. Keď predtým trvalo spracovanie jedného obrazu rádovo hodiny, teraz sú to minúty.

Použité knižnice mi umožnili efektívne pracovať s obrazom (ZOOM, Rotácia, Konvertovanie hĺbky farieb atď.) a súčasne používať maticové operácie (výpočet inverznej matice, násobenie matíc a pod.).

5 Experimentálne výsledky

Počas návrhu a implementácie opísaného systému sa objavovali rozličné problémy na ktoré som odpoveď v odbornej literatúre nenašiel. Tieto som riešil na základe experimentov s hodnotami jednotlivých systémových parametrov. S ich zmenou priamo súviselo aj správanie systému.

V tejto kapitole popíšem experimenty týkajúce sa problémov súvisiacich so správaním a tréňovaním neurónových sietí. Ďalej tu budú popísané experimenty súvisiace s funkcionalitou celého systému. Na záver bude uvedených niekoľko ukázkových testov hovoriacich o vlastnostiach systému.

5.1 Viacvrstvý perceptrón pre klasifikáciu natočenia vstupného obrazu

V súvislosti s touto časťou systému som musel riešiť nasledovné problémy:

- Ako veľmi závisí správanie sa siete od počiatočnej inicializácie váh?
- Aká hodnotu zvoliť pre parameter rýchlosti učenia?
- Aký je optimálny počet cyklov tréňovania siete?
- Je vhodnejšie tréňovanie s premiešavaním vzoriek, alebo bez?
- Aká je optimálna veľkosť tréňovacej a testovacej množiny?
- Ktorý parameter je najvhodnejší pre ohodnotenie naučenosti siete?
- Aký je najvhodnejší rozsah klasifikovaných otočení?
- Aká je najvhodnejšia topológia siete?

5.1.1 Počiatočná inicializácia váh

Ako bude v uvedených experimentoch neskôr vidno (viď. Tabuľka 1, 2, 3, 4), sieť je silne závislá od inicializácie a výsledok sa nedá vopred odhadnúť, alebo ovplyvniť. Preto som navrhol riešenie otestovania väčšieho počtu inicializácií a výber tej, pri ktorej sieť dosiahla najlepšie výsledky.

5.1.2 Parameter rýchlosti učenia

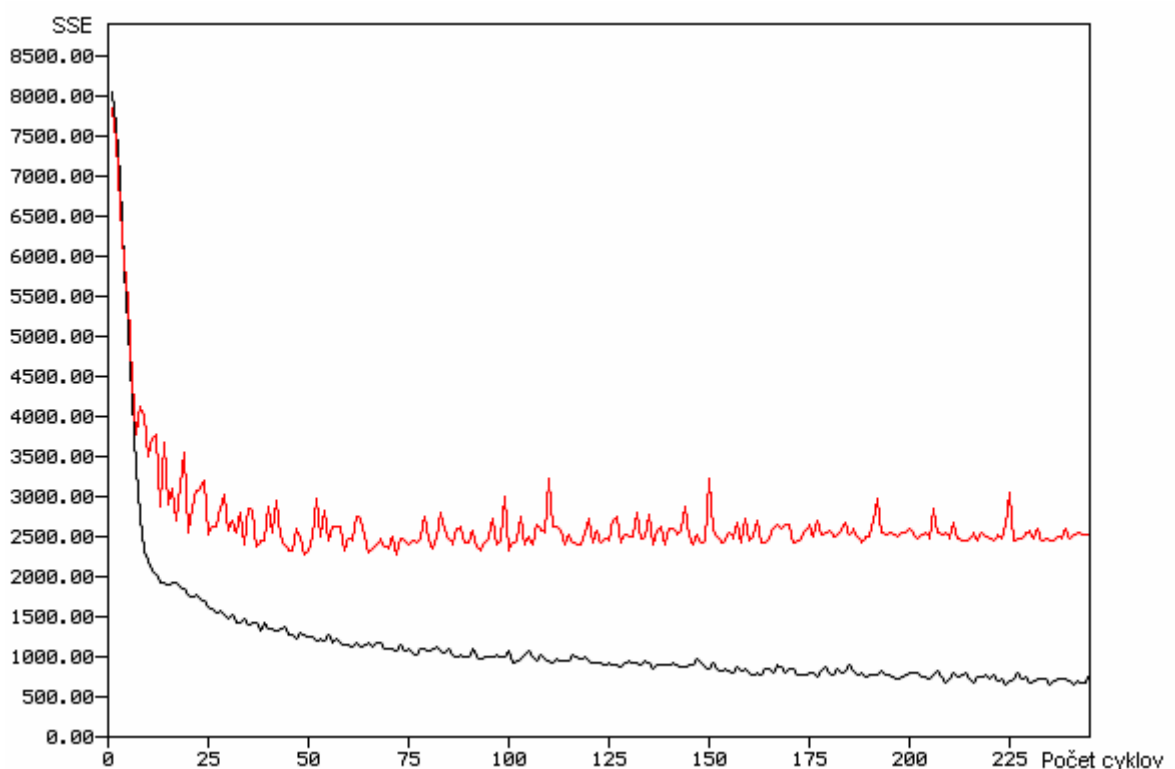
Pri voľbe hodnoty tohto parametra som postupoval postupným menením jeho hodnôt. Pokiaľ boli tieto hodnoty príliš veľké (väčšie ako 0,5), tréňovanie prebiehalo veľmi rýchlo, ale taktiež bolo možné pozorovať výrazné

pretrénovanie (SSE pre testovaciu množinu prudko stúpalo). Naopak, ak bola jeho hodnota príliš malá, (blízka 0) tréning prebiehalo veľmi pomaly. V mojom prípade sa ukázala najoptimálnejšou hodnota 0.2.

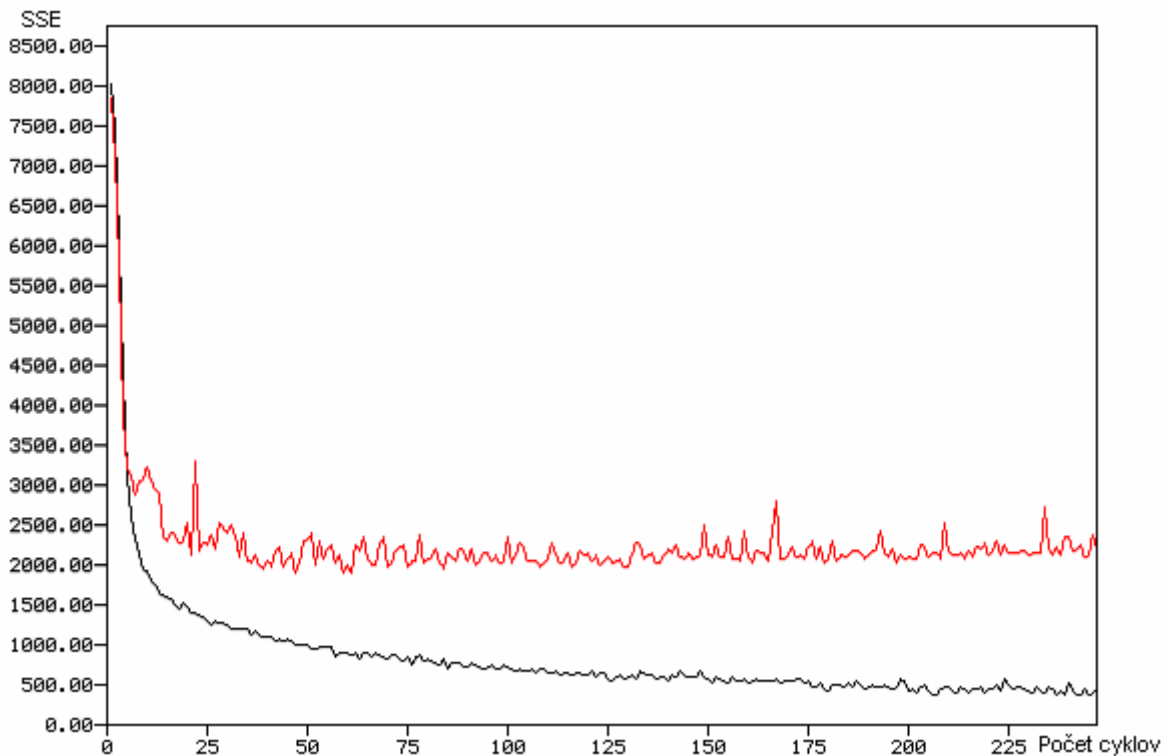
5.1.3 Počet cyklov tréningu a problematika premiešavania vzoriek

Z nižšie uvedených príkladov priebehov tréningu (Obr.5.1, Obr.5.2) je možné vidieť, že nezáleží na tom či je použité tréning s premiešaváním, alebo nie. V oboch prípadoch je priebeh závislosti SSE od počtu tréningových cyklov rozkmitaný a nadobúda približne rovnaké hodnoty. Preto som sa rozhodol pri tréningu použiť režim s premiešaváním vzoriek.

Aj keď pri procese tréningu nedochádza k výraznému pretrénovaniu, je možné pozorovať, že existuje počet cyklov pri ktorom je SSE pre testovaciu množinu najmenšie. Toto minimum nastane približne v prvých 100 cykloch. Preto som stanovil maximálny počet cyklov tréningu na 150 (zahrnutá je aj určitá rezerva). V týchto 150 cykloch sa teda hľadá globálne minimum pri ktorom je sieť vyhlásená za natréningovanú.



Obr. 5.1: Závislosť SSE od počtu tréningových cyklov. — Tréningová množina (350 tvárí v 24 pozíciách = 8400), — Testovacia množina (350 tvárí v 24 pozíciách = 8400 vzoriek). Tréning bez premiešavania vzoriek.



Obr. 5.2: Závislosť SSE od počtu tréningových cyklov. — Trénovacia množina (350 tvárí v 24 pozíciách = 8400), — Testovacia množina (350 tvárí v 24 pozíciách = 8400 vzoriek). Tréningovanie s premiešavaním vzoriek.

5.1.4 Veľkosti tréningovej a testovacej množiny

Rozhodol som sa zostaviť obe množiny z čo možno najväčšieho počtu vzoriek, pričom obe mali rovnakú veľkosť. K dispozícii som mal 700 tvárí. V oboch množinách sa teda nachádzalo 350 vzoriek v R otočeniach. Na takto koncipovaných množinách možno sieť dostatočne natréňovať a overiť jej generalizujúce vlastnosti.

5.1.5 Parameter pre ohodnotenie naučenosti siete

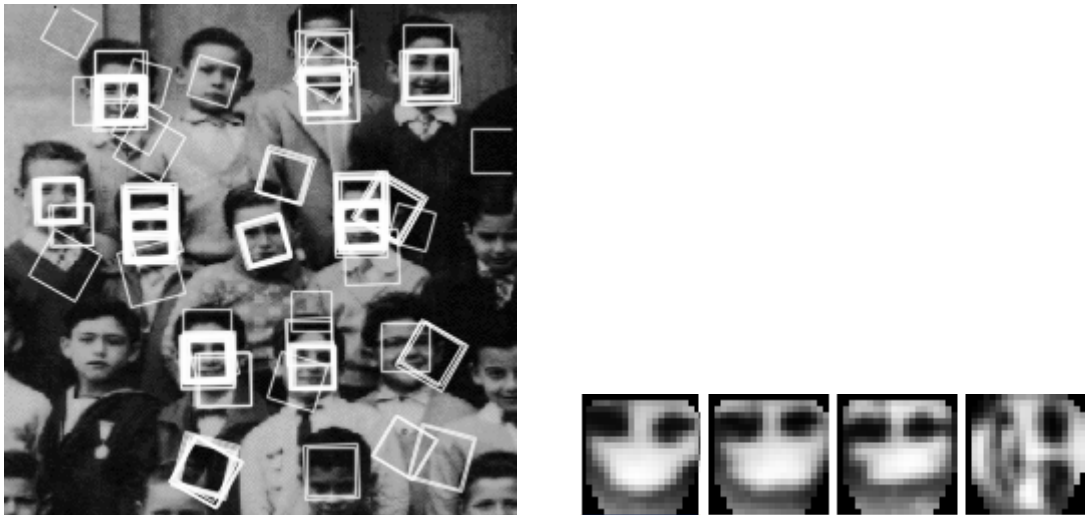
Do úvahy pripadajú dva parametre. Prvým bola hodnota SSE a druhým percentuálna úspešnosť ohodnocovania vzoriek v rámci tréningovej a testovacej množiny.

Pokiaľ sa porovnávajú procesy tréningovania sietí s rovnakými vlastnosťami, ale s inou počítačovou inicializáciou je jedno, ktorý parameter zoberieme do úvahy. Oba dokážu dostatočne vypovedať o naučenosti danej siete v rámci spomínanej množiny sietí.

Pokiaľ však porovnávame siete s rozličným počtom výstupných neurónov je zrejmé, že sieť s väčším počtom výstupných neurónov bude mať parameter SSE väčší. Preto som sa za signifikantnejší vybral druhý z týchto parametrov. Vo výsledných testoch však pre úplnosť uvádzam oba parametre.

5.1.6 Rozsah detekovaných otočení

Najprv som používal sieť, ktorá mala byť schopná rozpoznať 24 otočení tváří v rozsahu od 0° po 360° . Ako sa však ukázalo neskôr tak veľký rozsah otočení bolo zbytočné klasifikovať (Tváre sú vo väčšine prípadov vo frontálnej nenaklonenej, prípadne čiastočne (max $\pm 60^\circ$) naklonenej polohe) a použitie takejto siete viedlo k veľkému počtu chybné klasifikovaných tváří (pseudotváří) (Obr.5.3).



Obr. 5.3: Príklad nesprávnych klasifikácií pri použití siete rozpoznávajúcej 24 otočení tváre.

Problém som sa rozhodol vyriešiť obmedzením rozsahu rozpoznávaných otočení. Výsledný rozsah teda je $\langle -75^\circ, +75^\circ \rangle$. V tomto rozsahu je možné rozpoznať otočenia $-75^\circ, -60^\circ, -45^\circ, -30^\circ, -15^\circ, 0^\circ, +15^\circ, +30^\circ, +45^\circ, +60^\circ, +75^\circ$. Diferencia 15° medzi dvoma polohami by mala byť dostatočne malá na to, aby nebola vynechaná žiadna z možných polôh tváre.

5.1.7 Topológia siete

V literatúre [10] bolo možné nájsť, že najvhodnejšou je sieť s dvoma skrytými vrstvami po 15 neurónov. Rozhodol som sa však otestovať aj sieť s inou vnútornou štruktúrou a to síce sieť bez skrytých vrstiev (Obr.5.4). Výsledky testov je možné vidieť v nasledujúcich tabuľkách.

Trénovanie	Typ vzorkovej množiny	Nesprávne klasifikované[%]	Správne klasifikované[%]	SSE	Počet cyklov
1	Trénovacia	1,38	98,62	78,3724	55
	Testovacia	7,95	92,05	495,708	
2	Trénovacia	1,79	98,21	113,683	36
	Testovacia	8,75	91,25	506,533	
3	Trénovacia	1,4	98,6	86,3585	52
	Testovacia	8,96	91,04	537,136	
4	Trénovacia	1,3	98,7	74,5252	63
	Testovacia	8,52	91,48	509,244	
5	Trénovacia	1,25	98,75	71,3246	76
	Testovacia	9,04	90,96	553,839	
6	Trénovacia	1,58	98,42	100,238	37
	Testovacia	8,31	91,69	492,536	
7	Trénovacia	1,4	98,6	80,3495	56
	Testovacia	8,47	91,53	516,548	

Tabuľka 1: Trénovanie siete rozpoznávajúcej 11 natočení. Sieť má 2 skryté vrstvy po 15 neurónov. Trénovacia a testovacia množina: 3850 vzoriek (350 tváří v 11 pozíciách – Natočenia -75°, -60°, -45°, -30°, -15°, 0°, +15°, +30°, +45°, +60°, +75°). Trénovanie s podmienkou zastavenia pri najmenej chýbe. Žltou farbou je znázornená sieť s najlepšimi vlastnosťami.

Trénovanie	Typ vzorkovej množiny	Nesprávne klasifikované[%]	Správne klasifikované[%]	SSE	Počet cyklov
1	Trénovacia	1,38	98,62	78,3724	55
	Testovacia	7,95	92,05	495,708	
2	Trénovacia	1,3	98,7	72,477	69
	Testovacia	8,57	91,43	518,586	
3	Trénovacia	1,06	98,94	58,4303	111
	Testovacia	8,65	91,35	546,23	
4	Trénovacia	1,71	98,29	105,707	38
	Testovacia	8,47	91,53	515,397	
5	Trénovacia	1,25	98,75	71,3246	76
	Testovacia	9,04	90,96	553,839	
6	Trénovacia	1,64	98,36	101,006	36
	Testovacia	8,1	91,9	500,49	
7	Trénovacia	1,4	98,6	80,3495	56
	Testovacia	8,47	91,53	516,548	

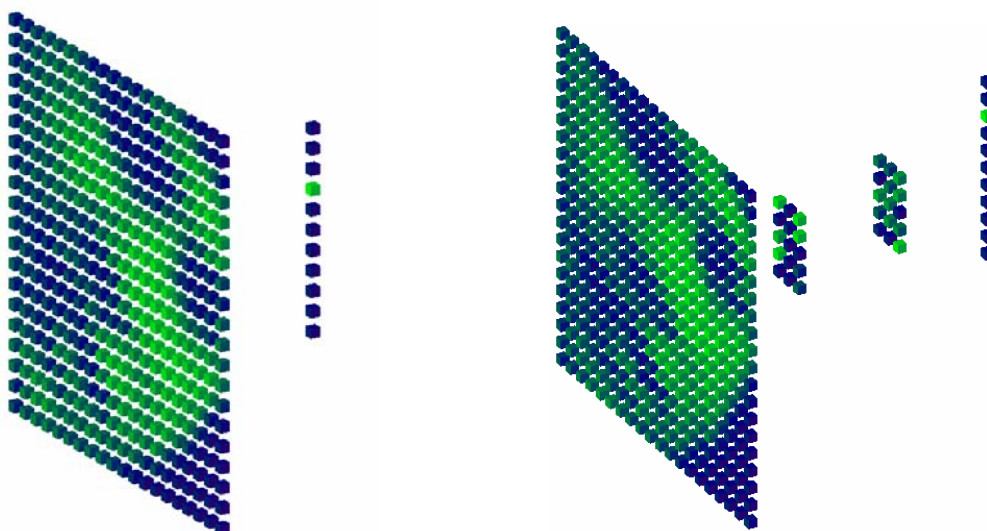
Tabuľka 2: Trénovanie rovnakej siete pri rovnakých podmienkach ako proti Tabuľke 1. Trénovanie s podmienkou zastavenia pri najmenšom počte nesprávne klasifikovaných. Žltou farbou je znázornená sieť s najlepšimi vlastnosťami.

Trénovanie	Typ vzorkovej množiny	Nesprávne klasifikované[%]	Správne klasifikované[%]	SSE	Počet cyklov
1	Trénovacia	11,56	88,44	586,982	37
	Testovacia	17,69	82,31	1181,76	
2	Trénovacia	1,97	89,03	159,194	106
	Testovacia	10,88	89,12	950,512	
3	Trénovacia	2,88	97,12	243,216	37
	Testovacia	10,26	89,74	908,342	
4	Trénovacia	3,04	96,96	266,322	32
	Testovacia	10,01	89,9	893,11	
5	Trénovacia	2,26	97,74	195,033	51
	Testovacia	10,03	89,97	909,352	
6	Trénovacia	11,15	88,06	531,933	48
	Testovacia	17,69	82,31	1145,05	
7	Trénovacia	2,62	97,38	215,112	38
	Testovacia	9,58	90,42	909,558	

Tabuľka 3: Trénovanie siete rozpoznávajúcej 11 natočení. Sieť nemá skryté vrstvy. Trénovacia a testovacia množina: 3850 vzoriek (350 tvárí v 11 pozíciách – Natočenia -75°, -60°, -45°, -30°, -15°, 0°, +15°, +30°, +45°, +60°, +75°). Trénovanie s podmienkou zastavenia pri najmenejšej chybe. Žltou farbou je znázornená sieť s najlepšimi vlastnosťami.

Trénovanie	Typ vzorkovej množiny	Nesprávne klasifikované[%]	Správne klasifikované[%]	SSE	Počet cyklov
1	Trénovacia	11,97	88,03	626,658	30
	Testovacia	17,43	82,57	1251,12	
2	Trénovacia	1,95	98,05	152,485	108
	Testovacia	10,47	89,53	966,872	
3	Trénovacia	3,48	96,52	348,691	20
	Testovacia	9,82	90,18	989,289	
4	Trénovacia	2,94	97,06	275,058	33
	Testovacia	9,4	90,6	965,193	
5	Trénovacia	2,75	97,25	270,635	37
	Testovacia	9,77	90,23	955,003	
6	Trénovacia	11,14	88,86	558,48	39
	Testovacia	17,06	82,94	1178,29	
7	Trénovacia	2,62	97,38	215,112	38
	Testovacia	9,58	90,42	909,558	

Tabuľka 4: Trénovanie rovnakej siete pri rovnakých podmienkach ako proti Tabuľke 3. Trénovanie s podmienkou zastavenia pri najmenšom počte nesprávne klasifikovaných. Žltou farbou je znázornená sieť s najlepšimi vlastnosťami.



Obr. 5.4: Priestorové zobrazenie oboch testovaných neurónových sietí bez znázornenia prepojení.

Ako je možné vidieť z výsledkov tréningov (viď Tabuľka 1, 2) lepšie výsledky dosiahla sieť z dvoma skrytými vrstvami. Je to vidieť na oboch parametroch hodnotenia kvality natréningovania siete. SSE a súčasne percentuálna úspešnosť siete pri klasifikácii je výrazne lepšia pri sieti so skrytými vrstvami.

Ako najlepšia bola vyhodnotená teda sieť s dvoma skrytými vrstvami pri tréningu č.1. Táto sieť bola potom začlenená do systému na lokalizáciu tváří a s ňou boli robené všetky ďalšie pokusy.

5.2 Viacvrstvový perceptrón pre klasifikáciu vstupného obrazu

Aj pri tejto časti systému vznikli určité problémy problémy:

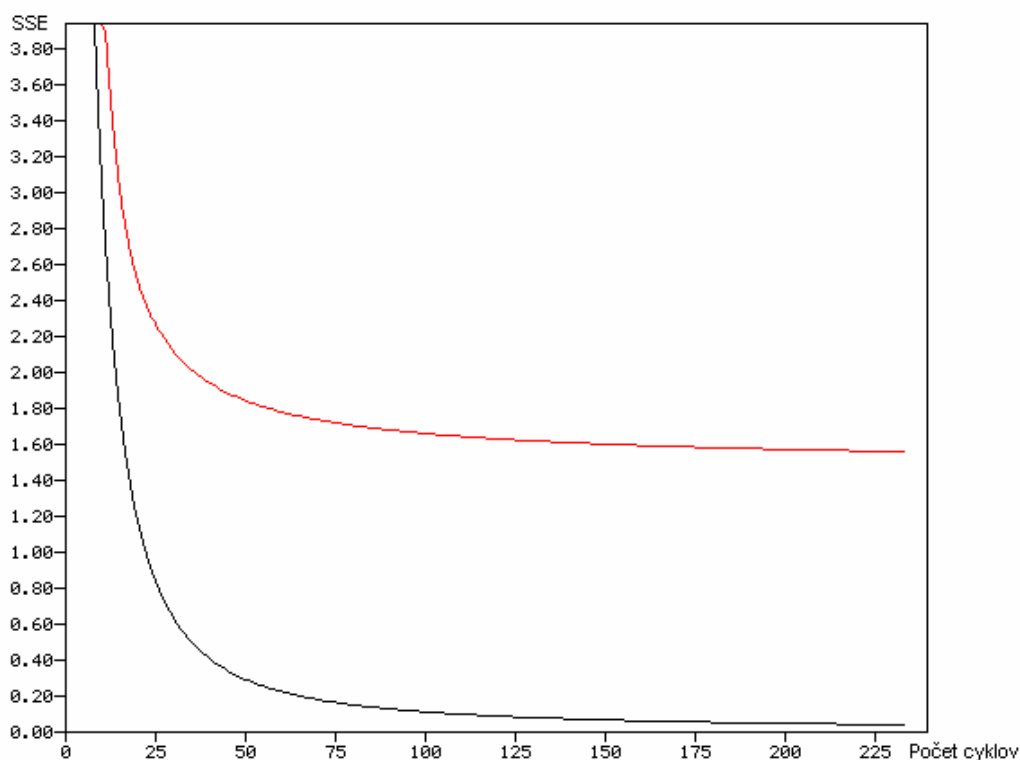
- Ako veľmi závisí správanie sa siete od počiatočnej inicializácie váh?
- Akú hodnotu zvoliť pre parameter rýchlosti učenia?
- Aký je optimálny počet cyklov tréningovania siete?
- Je vhodnejšie tréningovanie s premiešavaním vzoriek, alebo bez?
- Aká je najvhodnejšia topológia siete?

Odpoveď na otázky spojené s počiatočnou inicializáciou váh a parametrom rýchlosti učenia je rovnaká ako v predchádzajúcom prípade. Aj tu treba otestovať väčší počet rôzne nainicializovaných sietí a najvhodnejšou hodnotou pre parameter rýchlosti učenia sa ukázala 0,2. K ďalším otázkam sa budem vyjadrovať postupne.

5.2.1 Počet cyklov tréovania a problematika premiešavania vzoriek

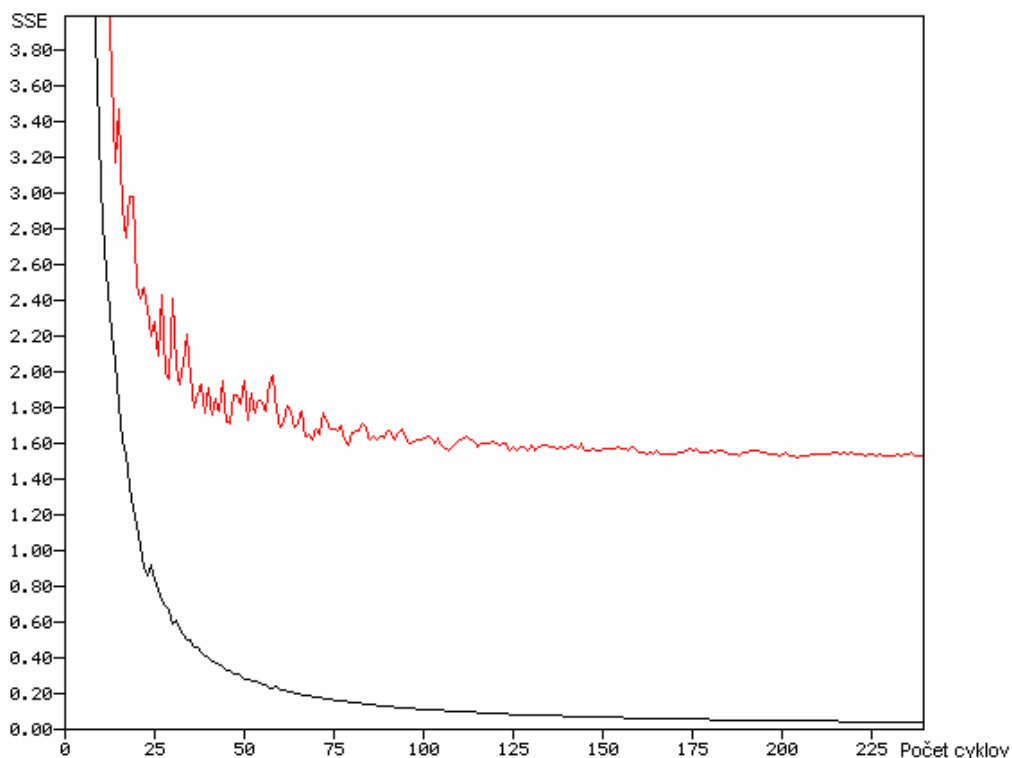
Keďže tréovanie tohto typu siete pozostáva z dvoch častí, musel som vlastnosti siete otestovať pre oba prípady. Najprv sa budem venovať prvej časti tréovania.

Nižšie možno vidieť priebehy závislosti SSE od počtu tréovacích cyklov (Obr.5.5, Obr.5.6). Je z nich zrejmé že počas tréovania v rámci prvých 225 cyklov nedochádza k pretrénovaniu (priebeh pre testovaciu množinu má stále klesajúci charakter). Tiež je možné sledovať fakt, že po prvých 100 cykloch je už klesanie hodnoty SSE menej výrazné. Z týchto dôvodov som sa rozhodol hľadať globálne minimum SSE v rámci prvých 300 cyklov na hladkom priebehu získanom pri tréovaní bez premiešavania vzoriek¹³. Následne na to by malo byť realizované tréovanie s premiešavaním vzoriek až do bodu, kedy bolo dosiahnuté spomínané globálne minimum.

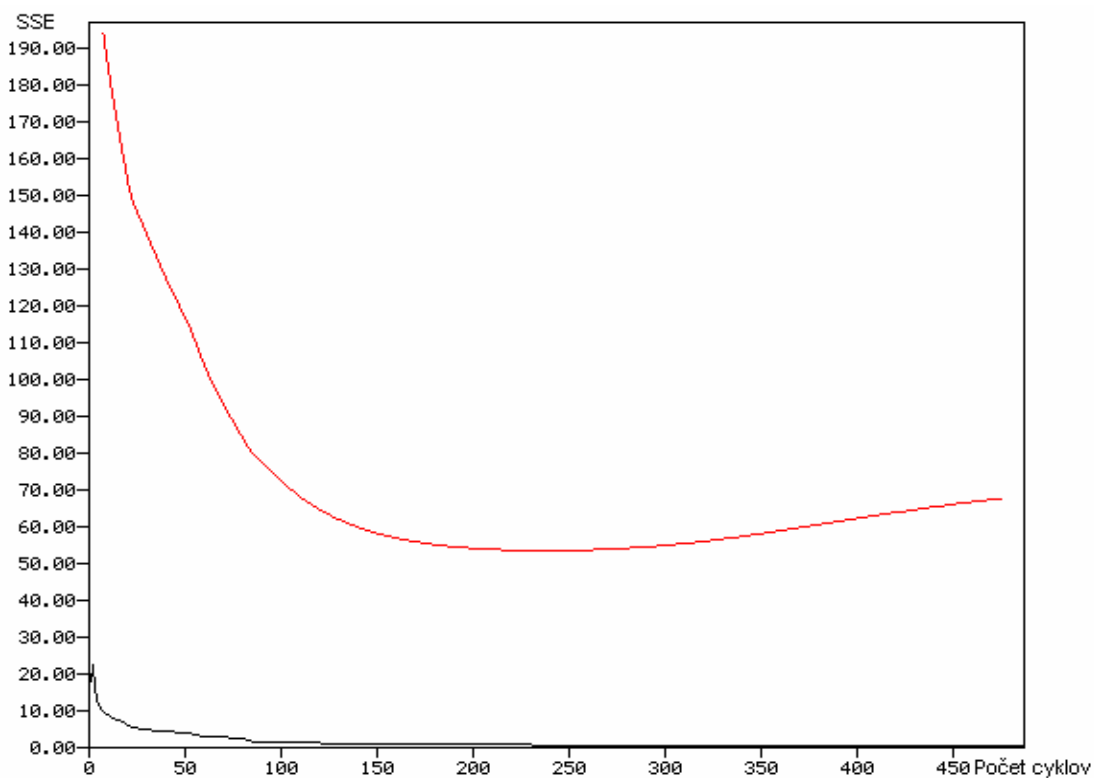


Obr. 5.5: Závislosť SSE od počtu tréovacích cyklov. — Tréovacia množina (250 tvárových vzoriek, 250 netvárových náhodne generovaných vzoriek), — Testovacia množina (250 tvárových vzoriek, 250 netvárových náhodne generovaných vzoriek). Tréovanie bez premiešavania vzoriek.

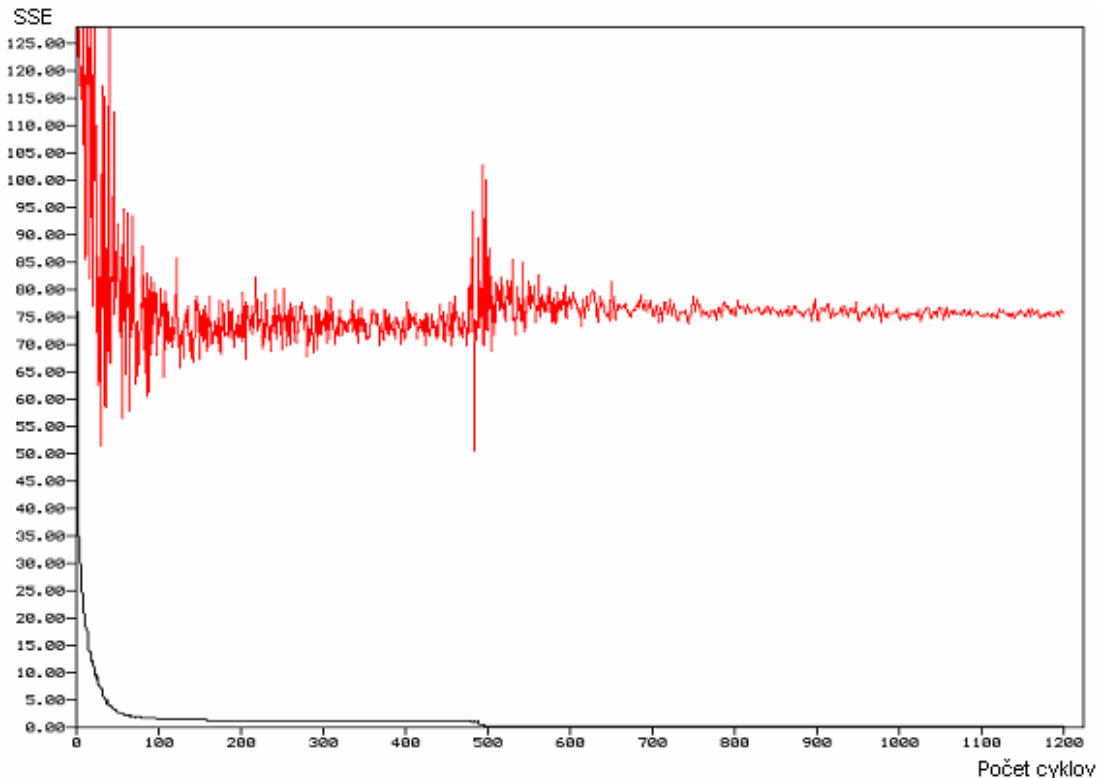
¹³ Z priebehov je zrejmé, že to globálne minimum bude získané na konci po 300 cykloch.



Obr. 5.6: Závislosť SSE od počtu tréningových cyklov. — Tréningová množina (250 tvárových vzoriek, 250 netvárových náhodne generovaných vzoriek), — Testovacia množina (250 tvárových vzoriek, 250 netvárových náhodne generovaných vzoriek). Tréningovanie s premiešavaním vzoriek.



Obr. 5.7: Závislosť SSE od počtu tréningových cyklov. Tréningovanie časť 2 . — Rozšírená tréningová množina, — Rozšírená testovacia množina. Tréningovanie bez premiešavania vzoriek.



Obr. 5.8: Závislosť SSE od počtu tréningových cyklov. Tréningovanie časť 2 . — Rozšírená tréningová množina, — Rozšírená testovacia množina. Tréningovanie s premiešavaním vzoriek.

V druhej časti tréningovania je možno v sledovaných priebehoch (Obr.5.7, Obr.5.8) pozorovať pretréningovanie po určitom počte cyklov. Aj tu platí, že priebeh pri tréningovaní bez premiešavania je hladký a naopak priebeh s premiešavaním vzoriek je rozkmitaný. Preto opäť hľadám globálne minimum v rámci hladkého priebehu a následne sieť trénujem pri premiešavaní vzoriek.

5.2.2 Topológia siete

V tomto prípade som sa rozhodol testovať vlastnosti troch typov sietí. Rozdiel medzi nimi bol v počte skrytých vrstiev. Testované boli siete s jednou dvomi a tromi skrytými vrstvami (Obr.5.9).

Pre každý typ siete bolo nainicializovaných 10 rôznych sietí a tieto boli potom podrobené prvej časti tréningovania opísanej vyššie. Výsledky pre jednotlivé typy sietí možno vidieť v nasledujúcich tabuľkách.

Číslo tréovania	Nesprávne klasifikované [%]	Správne klasifikované [%]	SSE
1	1,6	98,4	4,32919
2	1,6	98,4	3,77302
3	1,2	98,8	2,63035
4	1,8	98,2	3,16302
5	1,4	98,6	3,8866
6	1,4	98,6	2,65104
7	1,2	98,8	2,6577
8	1,6	98,4	2,63473
9	1	99	2,6861
10	1,2	98,8	2,58244

Tabuľka 5: 10 rôzne nainicializovaných neurónových sietí klasifikujúcich obraz. Siete majú jednu skrytú vrstvu. V tabuľke možno vidieť výsledky testovania sietí na testovacej množine popísanej v kap. 4.4.2. Žltou farbou sú znázornené dve siete s najlepšimi vlastnosťami.

Číslo tréovania	Nesprávne klasifikované [%]	Správne klasifikované [%]	SSE
1	0,8	99,2	1,57173
2	0,8	99,2	1,4185
3	2,2	97,8	2,33587
4	0,4	99,6	1,46667
5	1,8	98,2	3,39856
6	2,4	97,6	5,02191
7	1,2	98,8	3,13642
8	0,4	99,6	0,76703
9	1	99	2,48261
10	1,2	98,8	2,38818

Tabuľka 6: 10 rôzne nainicializovaných neurónových sietí klasifikujúcich obraz. Siete majú dve skryté vrstvy. V tabuľke možno vidieť výsledky testovania sietí na testovacej množine popísanej v kap. 4.4.2. Žltou farbou sú znázornené dve siete s najlepšimi vlastnosťami.

Číslo tréovania	Nesprávne klasifikované [%]	Správne klasifikované [%]	SSE
1	1,6	98,4	4,32919
2	1,6	98,4	3,77302
3	1,2	98,8	2,63035
4	1,8	98,2	3,16302
5	1,4	98,6	3,8866
6	1,4	98,6	2,65104
7	1,2	98,8	2,6577
8	1,6	98,4	2,63473
9	1	99	2,6861
10	1,2	98,8	2,58244

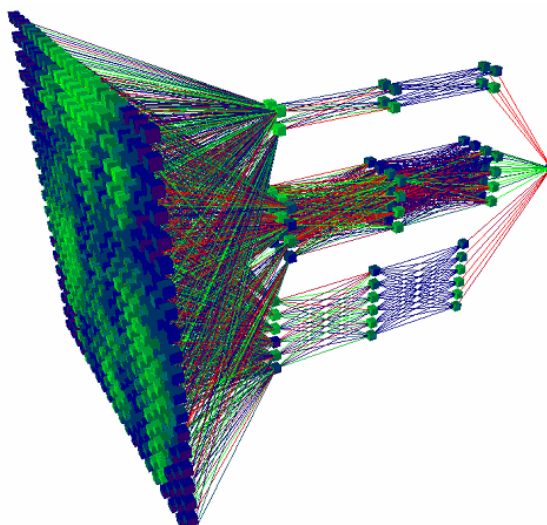
Tabuľka 7: 10 rôzne nainicializovaných neurónových sietí klasifikujúcich obraz. Siete majú tri skryté vrstvy. V tabuľke možno vidieť výsledky testovania sietí na testovacej množine popísanej v kap. 4.4.2. Žltou farbou sú znázornené dve siete s najlepšimi vlastnosťami.

Ako možno vidieť z výsledkov (viď Tabuľka 5, 6, 7), najlepšie výsledky po prvej časti učenia dosiahli siete s dvomi skrytými vrstvami. Do druhej časti tréningu boli však vybrané dve siete z každého typu. Týchto 6 vybraných sietí bolo potom podrobených druhej časti tréningu.

Číslo siete	Typ vzorkovej množiny	Nesprávne klasifikované[%]	Správne klasifikované[%]	SSE	Počet cyklov	Topológia siete
1	Rozšírená tréningová	0	100	0,0311	1382	1 skrytá vrstva
	Rozšírená testovacia	2,14	97,86	26,4592		
	NoFaces	0,69	99,31	11,0484		
2	Rozšírená tréningová	0	100	0,02887	1501	1 skrytá vrstva
	Rozšírená testovacia	2,62	97,38	43,655		
	NoFaces	0,88	99,12	19,1033		
3	Rozšírená tréningová	0,08	99,92	2,02914	628	2 skryté vrstvy
	Rozšírená testovacia	2,9	97,1	52,356		
	NoFaces	0,79	99,21	19,6894		
4	Rozšírená tréningová	0,12	99,88	3,0099	1501	2 skryté vrstvy
	Rozšírená testovacia	2,07	97,93	37,9199		
	NoFaces	0,52	99,48	13,6246		
5	Rozšírená tréningová	0,16	99,84	4,00312	1158	3 skryté vrstvy
	Rozšírená testovacia	2,52	97,48	52,984		
	NoFaces	0,75	99,25	22,3626		
6	Rozšírená tréningová	0,16	99,84	4,0033	906	3 skryté vrstvy
	Rozšírená testovacia	2,45	97,55	50,7297		
	NoFaces	0,75	99,25	19,0788		

Tabuľka 8: 2. časť tréningu sietí klasifikujúcich vstupný obraz. Zloženie vzorkových množín je zadané v kap. 4.3.2. Žltou farbou je znázornená sieť s najlepšimi vlastnosťami.

Ako je možné vidieť (viď Tabuľka 8) najlepšie vlastnosti dosiahla sieť č. 1 s jednou skrytou vrstvou. Táto sieť bola teda vybraná a začlenená do systému na lokalizáciu tváří.



Obr. 5.9: Priestorové zobrazenie viacvrstvového perceptrónu na klasifikáciu vstupného obrazu s tromi skrytými vrstvami.

5.3 Vplyv obsahu vzorkovej množiny NoFaces

Pri testovaní natrénovanej siete sa neustále objavoval problém nesprávnych klasifikácií. Nájdené síce boli všetky prítomné tváre, ale okrem nich tu bolo ešte stále určité množstvo chybné označených oblastí.

Jedným z možných spôsobov ako tomuto problému predísť, je získať množinu všetkých nesprávnych klasifikácií. Toto je vo všeobecnosti nemožné. Nedá sa totiž vytvoriť systém so 100% úspešnosťou pre ľubovoľný vstupný obraz. Situácia sa však mení ak ohraničíme množinu vstupných obrazov.

Systém by sa teda najprv mal otestovať na uzavretej konečnej množine obrazov scenérií neobsahujúcich tváre. Takto získané chybné klasifikácie by tvorili obsah množiny NoFaces. Ak by potom boli do týchto obrazov pridané objekty ľudských tvárí nehrozilo by nebezpečie týkajúce sa tohoto problému.










Príkladom by mohla byť konkrétna aplikácia. Ak by bol opisovaný lokalizátor zaradený ako súčasť systému na rozpoznanie osôb pri vstupe do budovy, stačilo by ho natrénovať na scenériu snímanú priemyselnou kamerou pri vstupných dverách.

5.4 Vplyv vstupných parametrov na funkčnosť systému

Veľkosť vstupného obrazu je jedným z najdôležitejších parametrov, pretože výrazne ovplyvňuje rýchlosť celej lokalizácie tvárí. Ak by mal byť systém aplikovaný v praxi jeho použitie je podmienené fungovaním v reálnom čase. Konkrétna aplikácia však môže znamenať aj konkretizáciu vstupného obrazu. Potom by bolo možné vedieť aj približné rozmery tvárí na vstupe. Z tohto údaju je potom možné vypočítať počiatočné a koncové rozlíšenie čím môžeme eliminovať podstatné množstvo preverovaných segmentov.

Hodnota prahu podľa ktorej rozhodujem či ide o tvár, alebo nie hrá tiež dôležitú úlohu. Ak je príliš nízka hrozí nebezpečenstvo nájdania veľkého množstva chybných výsledkov. Ak je príliš veľká, hrozí zasa, že niektorú z tvárí môžeme vynechať. Ako som zistil pri testovaní v intervale $(0,95;1)$ sa nachádzajú prevažne tváre, ale aj tu je istá pravdepodobnosť nájdania netvárových obrazov. Ak chceme aj tieto chybné klasifikácie eliminovať, treba ich zaradiť do tréningovej množiny ako netvárové vzorky. Konkrétnu hodnotu prahu je treba špeciálne určiť pre každý typ aplikácie. Pri jeho určovaní som

najprv zvolil hodnotu prahu nižšiu a na základe získaných výsledkov som určil výslednú optimálnu prahovú hodnotu (viď Tabuľka 9).

Číslo Tváre	Pozícia tváre v zdrojovom obraze (Stred tváre, Ľavý horný)				Zmenšenie	Otočenie [°]	P[%]	Detekcia
1	71	85	44	57	0.362292	0	99.999988	
2	70	84	41	55	0.341396	0	99.999964	
3	71	83	41	53	0.334702	0	99.995786	
4	73	82	42	51	0.328139	0	99.980658	
5	71	86	43	57	0.348224	0	99.976861	
6	71	83	40	52	0.321705	0	99.976367	
7	72	82	41	50	0.315397	0	99.974984	
8	70	83	43	56	0.369538	75	99.968278	
9	70	87	42	59	0.355189	0	99.966329	

Tabuľka 9: Chybná detekcia nastala pri tvári č. 8. Predísť tomuto javu sa dá zvolením inej vyššej prahovej hodnoty a to v tomto prípade hodnote väčšej ako 0,99968278.

Hodnota kroku posúvania okna tiež nie je zanedbateľná. Ak má hodnotu napríklad 1, posúvanie okna je dostatočne jemné a vôbec neriskujeme, že vynecháme nejakú tvár. Pri tejto hodnote dostávame veľký počet duplicitných nálezov. Ak zvolíme hodnotu 10 vôbec neriskujeme možnosť viacnásobného výberu tej istej tváre a proces detekcie je 10-krát rýchlejší. Je tu však príliš veľké riziko nenájdenia tvárí. Najlepšie je teda zvoliť hodnotu 1, 2, alebo 3.

Pre koeficient zmenšovania platia podobné úvahy. Hodnota v intervale $\langle 1,02; 1,07 \rangle$ sa ukázala ako najlepšia.

5.5 Testy

V nasledujúcej časti bude uvedených niekoľko záverečných testov na ktorých bude možno pozorovať vlastnosti systému.

Budú preverované obrázky rôznych rozmerov, s rôznou kvalitou a rôznym počtom tvárí (viaceré spôsoby naklonenia) pri rozličných svetelných podmienkach.

V rámci každého testu budú uvedené a zhodnotené výsledky. Výsledky budú zapísané v nasledujúcom tvare:

Výsledky detekcie

Počet preverených okien	Počet okien extrahovaných so vstupného obrazu.
Počet správne klasifikovaných okien	Počet okien, ktoré boli klasifikované ako tváre a skutočne obsahovali objekt tváre.
Počet nesprávne klasifikovaných okien	Počet okien, ktoré boli klasifikované ako tváre a pritom neobsahovali objekt tváre.
Počet správne klasifikovaných oblastí	Oblasťou je nazvané miesto vo vstupnom obraze, ktoré obsahuje objekt ľudskej tváre. Jedna oblasť v obraze môže byť označená viacerými oknami klasifikovanými ako tváre.
Počet nesprávne klasifikovaných oblastí	V tomto prípade ide o miesto vo vstupnom obraze, ktoré neobsahuje objekt tváre a pritom bolo označené určitým počtom okien ako tvár.
Počet tvárí prítomných na vstup. obraze	Počet objektov ľudských tvárí vo vstupnom obraze.
Počet nedetekovaných tvárí	Počet tvárí, ktoré systémom vôbec neboli nájdené.

5.5.1 Test Audrey 1

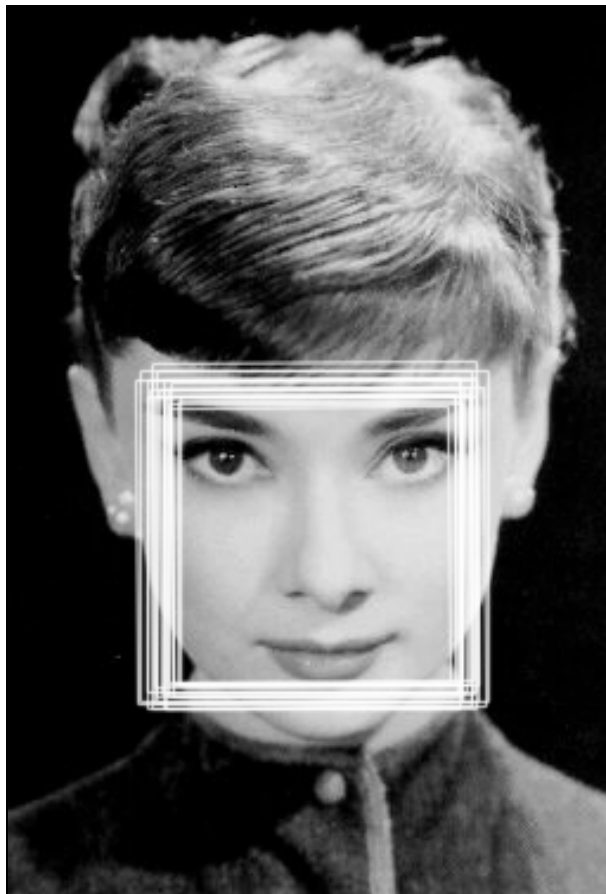
Parametre detekcie

Rozmery vstupného obrazu [Pixel]	256x377
Počiatkové zmenšenie [%]	20
Koncové zmenšenie [%]	14






















Koeficient zmenšenia	1,02
Počet zmenšení	17
Krok posúvania okna [Pixel]	1
Prah [%]	99.5

Výsledky detekcie

Počet preverených okien	18654
Počet správne klasifikovaných okien	21
Počet nesprávne klasifikovaných okien	0
Počet správne klasifikovaných oblastí	1
Počet nesprávne klasifikovaných oblastí	0
Počet tvárí prítomných na vstup. obraze	1
Počet nedetekovaných tvárí	0



Obr. 5.10: Výsledný obraz s naznačenými detekciami

Poradie	Pozícia tváre v zdrojovom obraze (Stred tváre, Ľavý horný bod)[Body]				Zmenšenie	Otočenie [°]	P[%]	Detekcia	
10	226	129	161	64	0.154607	0	99.999964		
20	224	126	154	56	0.142833	0	99.999815		
15	228	127	161	60	0.148603	0	99.999678		
8	228	126	164	63	0.157699	0	99.999595		
9	228	133	164	69	0.157699	0	99.999434		
4	225	134	164	73	0.164070	0	99.998891		
19	226	130	157	61	0.145689	0	99.998325		
16	228	134	161	67	0.148603	0	99.997538		
11	224	125	158	59	0.151575	0	99.992508		
13	230	125	164	59	0.151575	0	99.989498		
3	225	127	164	67	0.164070	0	99.985379		
5	223	130	161	68	0.160853	0	99.984896		
6	230	130	167	68	0.160853	0	99.967498		
12	224	131	158	65	0.151575	0	99.965775		
17	219	130	151	61	0.145689	0	99.959439		
7	221	133	158	69	0.157699	0	99.841440		
2	227	131	167	71	0.167351	0	99.808174		
18	226	123	157	54	0.145689	0	99.799389		
14	230	131	164	65	0.151575	0	99.776542		
21	224	133	154	63	0.142833	0	99.771702		
1	228	128	169	70	0.170698	0	99.589050		

Tabuľka 10: Získané detekcie usporiadané podľa pravdepodobnosti, že obsahujú objekt tváre

Zhodnotenie

Spracovávaný bol základný typ vstupného obrazu na ktorom je iba jedna tvár v základnej frontálnej polohe (Obr.5.10). Ako je možné vidieť táto bola viacnásobne nájdená pri rôznych zmenšeniach a pozíciách preverovacieho okna (viď Tabuľka 10). Všetky okná ohodnotené ako tvár spadajú do tej istej oblasti pokrývajúcej skutočnú tvár prítomnú na obraze. Úspešnosť systému bola teda v tomto prípade 100%.

Ak by sme ale chceli aby odpoveďou systému bola iba jedna presne ohraničená oblasť je nutné systém rozšíriť o ďalší systémový blok, ktorý bude spájať prekrývajúce sa okná - Blok spájania prekrývajúcich sa regiónov. Takýto blok by spracovával nájdené lokalizácie a spájaj by ich na základe polohy, veľkosti a otočenia do celkov.

5.5.2 Test Audrey 2

Parametre detekcie

Rozmery vstupného obrazu [Pixel]	600x377
Počiatkové zmenšenie [%]	20
Koncové zmenšenie [%]	14
Koeficient zmenšenia	1,02
Počet zmenšení	17
Krok posúvania okna [Pixel]	1
Prah [%]	99,5

Výsledky detekcie

Počet preverených okien	65099
Počet správne klasifikovaných okien	6
Počet nesprávne klasifikovaných okien	0
Počet správne klasifikovaných oblastí	2
Počet nesprávne klasifikovaných oblastí	0
Počet tvárí prítomných na vstup. obraze	2
Počet nedetekovaných tvárí	0



Obr. 5.11: Výsledný obraz s naznačenými detekciami

Zhodnotenie

Spracovávaný bol obraz na ktorom bola umelo umiestnená tá istá tvár v dvoch pozíciách a natočeníach (Obr.5.11). Aj tu systém dokázal rozpoznať ako umiestnenie tak aj otočenie tváre. Úspešnosť systému bola teda 100%.

Ako je vidieť na vstupných parametroch počítačové zmenšenie je 20% a koncové 14%. Znamená to, že práve v tomto rozmedzí veľkostí vstupného obrazu mal objekt tváre veľkosť približne 20x20 bodov a mohol byť pokrytý "preverovacím" oknom. Nastavenie týchto parametrov podstatne urýchlilo činnosť celého procesu. Opäť je vidno, že viacnásobné prekrývajúce sa okná by sa dali spojiť do dvoch presne ohraničených oblastí.

5.5.3 Test Školáci

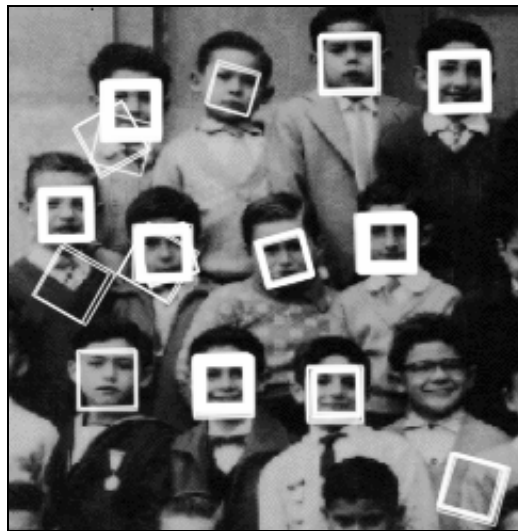
Parametre detekcie

Rozmery vstupného obrázku [Pixel]	256x256
Počiatkové zmenšenie [%]	100
Koncové zmenšenie [%]	70
Koeficient zmenšenia	1,02
Počet zmenšení	17
Krok posúvania okna [Pixel]	1
Prah [%]	99

Výsledky detekcie

Počet preverených okien	707574
-------------------------	--------

Počet správne klasifikovaných okien	257
Počet nesprávne klasifikovaných okien	11
Počet správne klasifikovaných oblastí	11
Počet nesprávne klasifikovaných oblastí	5
Počet tvárí prítomných na vstup. obraze	12
Počet nedetekovaných tvárí	1



Obr. 5.12: Výsledný obraz s naznačenými detekciami

Zhodnotenie

Iba 11 okien z celkového počtu 707574 bolo nesprávne klasifikovaných. Systém lokalizoval 90,9% prítomných tvárí. Čo sa počtu označených oblastí týka, 31,25% z nájdených oblastí bola nesprávna. Tento nedostatok možno odstrániť tým, že nesprávne detekcie budú zaradené medzi netvárové vzorky a sieť na klasifikáciu vstupného obrazu bude dotrénovaná. Iným spôsobom potlačenia nesprávnych detekcií by mohlo byť vyhodnotenie označenia oblastí. Na výslednom obraze (Obr.5.13) je možné vidieť že väčšina z nesprávne klasifikovaných oblastí je označená raz, maximálne dva krát, a že tvárové oblasti sú označené mnohonásobne. Spomínaný blok spájania prekrývajúcich sa regiónov by teda mohol byť obohatený o logiku potlačania zlých detekcií.

5.5.4 Test Vysokoškoláci

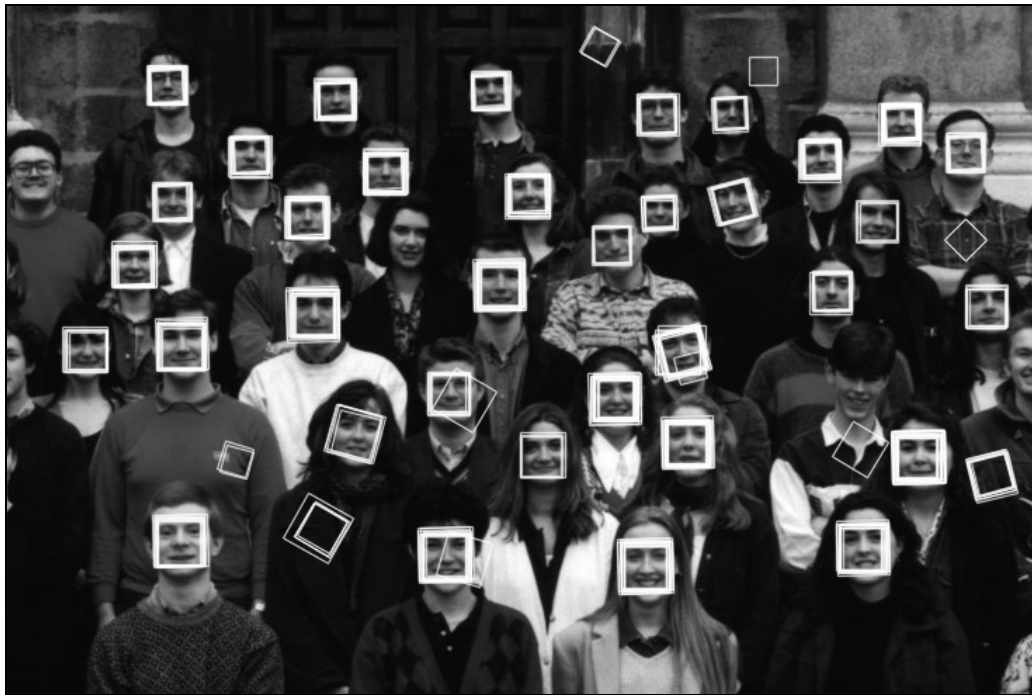
Parametre detekcie

Rozmery vstupného obrázku [Pixel]	732x490
Počiatkové zmenšenie [%]	100

Koncové zmenšenie [%]	50
Koeficient zmenšenia	1,07
Počet zmenšení	9
Krok posúvania okna [Pixel]	1
Prah [%]	99,9

Výsledky detekcie

Počet preverených okien	1917622
Počet správne klasifikovaných okien	321
Počet nesprávne klasifikovaných okien	18
Počet správne klasifikovaných oblastí	35
Počet nesprávne klasifikovaných oblastí	10
Počet tvárí prítomných na vstup. obraze	38
Počet nedetekovaných tvárí	3



Obr. 5.13: Výsledný obraz s naznačenými detekciami

Zhodnotenie

Z celkového počtu 1917622 bolo nesprávne klasifikovaných iba 18 okien. Úspešnosť nájdenia tvárí bola 92,10% a úspešnosť nájdenia tvárových oblastí 77,77%. Opäť sa ukazuje (Obr.5.14), že pridanie bloku spájania a vyhodnocovania prekrývajúcich sa regiónov by podstatne vylepšilo percentuálnu úspešnosť systému pri označovaní tvárových oblastí.

5.5.5 Test Futbal Team

Parametre detekcie

Rozmery vstupného obrázku [Pixel]	627x441
Počiatočné zmenšenie [%]	100
Koncové zmenšenie [%]	70
Koeficient zmenšenia	1,05
Počet zmenšení	6
Krok posúvania okna [Pixel]	2
Prah [%]	99

Výsledky detekcie

Počet preverených okien	336580
Počet správne klasifikovaných okien	52
Počet nesprávne klasifikovaných okien	10
Počet správne klasifikovaných oblastí	14
Počet nesprávne klasifikovaných oblastí	7
Počet tvárí prítomných na vstup. obraze	15
Počet nedetekovaných tvárí	1

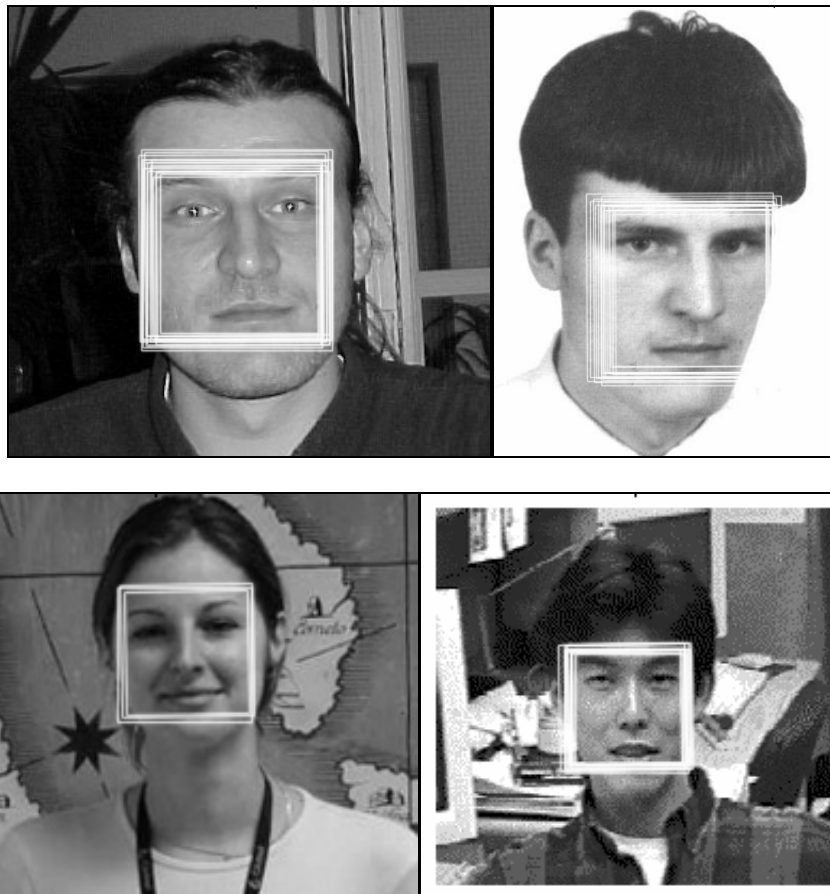


Obr. 5.14: Výsledný obraz s naznačenými detekciami

Zhodnotenie

Z výsledného počtu 336580 bolo nesprávne klasifikovaných 10 okien. Úspešnosť nájdenia tvárí bola 93,33% a úspešnosť nájdenia tvárových oblastí 66,66%. Závery sú podobné ako pri teste Vysokoškooláci.

5.5.6 Ďalšie testy



Obr. 5.15: Ďalšie príklady výsledkov testovania systému na lokalizáciu tvárí



Obr. 5.16: Príklad spracovania farebného obrazu

6 Záver

Úlohou mojej diplomovej práce bolo navrhnúť systém na lokalizáciu tvárí, ktorý by bol schopný vo vstupnom neupravenom obraze nájsť všetky ľudské tváre bez ohľadu na ich veľkosť, umiestnenie, natočenie a osvetlenie.

Systém bol podrobne navrhnutý na základe vyššie popísaných teoretických princípov, pričom jeho jadro tvorili dve neurónové siete. Pri spracovaní obrazu boli použité viaceré techniky z oblasti číslicového spracovania obrazu. Postupy týkajúce sa tréningu neurónových sietí, ako aj voľba ich topológie, či výber najvhodnejších parametrov pre jednotlivé časti systému boli určené na základe experimentálnych výsledkov.

Implementovaný systém (program napísaný v jazyku C) bol otestovaný na reálnych obrazoch a dokázal, že uspokojivo plní svoju funkciu. V závislosti od situácie bola úspešnosť nájdenia všetkých tvárí v rozmedzí 90% až 100%. Počty nesprávne klasifikovaných okien tvoria v pomere k celkovému počtu preverovaných okien skutočne zanedbateľné percento. V práci boli uvedené aj ďalšie návrhy na zlepšenie výsledkov systému.

V závislosti od vstupných parametrov a veľkosti vstupného obrazu dokázal systém plniť komplexnú úlohu lokalizácie tvárí v pomerne krátkom čase (na osobnom počítači AMD DURON 1.2 GHz to boli rádovo minúty). Tento výsledok možno pokladať za uspokojivý, no ešte stále nemožno hovoriť o použití systému v reálnom čase. Riešenie tohto problému sa ponúka vo forme hardwarovej realizácie systému (VLSI realizácia neurónových sietí) a použití signálových procesorov na spracovanie obrazu. Potom by sa už dalo s určitosťou hovoriť o funkčnosti systému v reálnom čase a jeho konkrétnej aplikácii.

Popisovaný systém by sa dal využiť ako súčasť systému na rozpoznávanie tvárí, alebo systému na kompresiu obrazu. Celá skúmaná problematika, nájdenia objektu s istými charakteristikami, by sa dala zovšeobecniť a ďalej používať v iných oblastiach (napr. rozoznávanie objektov a vzorov).

Ďalšie rozšírenia systému by sa mohli týkať bloku na spájanie a vyhodnotenie prekrývajúcich sa regiónov, prípadne lokalizácie tvárí v pohyblivom obraze.

7 Použitá literatúra

1. Oravec M., Polec J., Marchevský S.: Neurónové siete pre číslicové soracovanie signálov. Bratislava, 1998
2. Kvasnička V. et al.: Úvod do teórie neurónových sietí. Bratislava: IRIS, 1997
3. Polec, J., Pavlovičová, J., Oravec, M.: Metódy kompresie obrazu. Bratislava, 1998
4. Gonzales, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesey Publishing Comp., USA, 1992
5. Jebara T.S.: 3D Pose Estimation And Normalization For Face Recognition. Montreal: McGill University, 1996
[DOPLNIT LINKU](#)
6. Menser B., Brunig M.: Locating Human Faces in Color Images with Complex Background, Aachen: RWTH, 1999
[DOPLNIT LINKU](#)
7. Menser B., Wien M.: Segmentation an tracking of Facial Regions in Color Image Sequences. Aachen: RWTH, 1999
[DOPLNIT LINKU](#)
8. Menser B., Brunig M.: Locating Human Faces in Color Images using PCA. Aachen: RWTH, 1998
[DOPLNIT LINKU](#)
9. Sung K., Poggio T.: Example-based Learning for View-based Human Face Detection. MIT 1994, A.I.Memo No. 1521
10. Rowley H. A.: Neural Network-Based Face Detection. Pittsburgh: Carnegie Mellon University, 1999. PHD thesis
[DOPLNIT LINKU](#)
11. Jeng S. et al.: Facial Feature Detection Using Geometrical Face Model: An Efficient Approach. Patern Recognition, Vol. 31, No. 3, pp 273-282, 1998
12. Šošovička V.: Rozoznávane tvárí pomocou analýzy hlavných komponentov. Bratislava: FEI STU, 1998. Diplomová práca

13. Beszédeš M.: Predspracovanie obrazu pre účely rozpoznávania tvárí. Bratislava: FEI STU, 1998. Ročníková záverečná práca
 14. Beszédeš M.: Detekcia tvárí pomocou neurónovej siete. In: Zborník ku konferencii Elektrotechnika a Energetika 2002, s. 71-72
 15. Beszédeš M.: Detekcia tvárí pomocou neurónovej siete. Bratislava: FEI STU, 1998. Práca ŠVOČ
 16. Zell A. et al: The SNNS Neural Network Simulator. Stuttgart: UNIVERSITY OF STUTTGART 1993
- LINKA**
17. Open Source Computer Vision Library: Reference Manual. Intel Corporation, Documnet Number: A77028-004, USA 2001
<http://developer.intel.com>
 18. Intel Image processing Library: Reference Manual. Intel Corporation, Documnet Number: 663791-005, USA 2000
<http://developer.intel.com>

8 Použité skratky a symboly

<i>STEP</i>	Krok posúvania preverovacieho okna
<i>K</i>	Hodnota koeficientu zmenšovania vstupného obrazu
<i>S</i>	Počet bodov menšieho z rozmerov vstupného obrazu
<i>c</i>	Výsledný počet zmenšení
$I(x,y)$	Intenzita v bode (x,y)
I	Matica intenzít
$E(...)$	Stredná hodnota
<i>R</i>	Počet výstupov neurónovej siete pre klasifikáciu natočenia, počet natočení, ktoré dokáže sieť klasifikovať
A	Matica obsahujúca súradnice ideálnych črt
t	Transformačný vektor pre normalizáciu rozmerov a natočenia tváre
B	Matica obsahujúca súradnice dôležitých črt na tvári vo vstupnom obraze
T	Transformačná matica pre normalizáciu rozmerov a natočenia tváre
<i>P, Q, R</i>	Označenie pre tréningové a testovacie množiny
<i>C</i>	Počet tréningových cyklov
<i>W</i>	Otočenie vstupného obrazu
<i>T</i>	Threshold – Prah akceptovania toho, že na vstupe neurónovej siete je tvár
PCA	Principal Component Analysis – Analýza hlavných komponentov
SSE	Stredná kvadratická chyba

9 Príloha

9.1 Popis spustenia a parametrov programu FaceLocalizer

Tento program dokáže dávkovo spracovať sadu vstupných obrazov v závislosti od vstupných parametrov. Program sa spúšťa v príkazovom riadku nasledovným príkazom:

```
FaceLocalizer.exe input.par
```

Input.par (samozrejme, že názov môže byť aj iný) je súbor ktorý obsahuje parametre pre celý proces lokalizácie tváří.

Štruktúra súboru input.par:

```
Definícia výstupného adresára 1  
Definícia umiestnenia vstupného súboru 1  
Definícia mena výstupných obrazov 1  
Koeficient zmenšovania 1  
Počiatočné zmenšenie 1  
Koncové zmenšenie 1  
Krok preverovania 1  
Prah akceptovania 1  
.  
.  
Definícia výstupného adresára N  
Definícia umiestnenia vstupného súboru N  
Definícia mena výstupných obrazov N  
Koeficient zmenšovania N  
Počiatočné zmenšenie N  
Koncové zmenšenie N  
Krok preverovania N  
Prah akceptovania N
```

Príklad:

```
D:\DP\RESULT\CLASS  
D:\DP\PIC\class.bmp  
class_res  
1.02 0.5 0.2 1 0.995
```