

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## ČTEČKA OBLIČEJŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETER BUČKO

BRNO 2010



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## **ČTEČKA OBLIČEJŮ**

FACE READER

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**PETER BUČKO**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. VÍTĚZSLAV BERAN**

BRNO 2010

## **Abstrakt**

Tato práce se zabývá strojovým rozpoznáváním tváří. Budou zde popsány metody Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA) a Elastic Bunch Graph Matching (EBGM). Cílem této práce je vytvořit demonstrační aplikaci pro rozpoznávání tváří. Dále také otestuji metody PCA a LDA, abych zjistil, jak jsou přesné a jak je ovlivňují změny parametrů, jako velikost databáze a počet obrazů na osobu.

## **Abstract**

This thesis deals with computer face recognition. Methods of Components Analysis (PCA), Linear Discriminant Analysis (LDA) and Elastic Bunch Graph Matching (EBGM) are described here. Aim of this thesis is creation of a demonstration application for a face recognition. Moreover I test PCA and LDA methods to find out, how accurate it can be and how can be affected by changing of parameters, such as size of a database and picture count per person.

## **Klíčová slova**

Rozpoznávání, Rozpoznávání tváří, PCA, LDA, EBGM, porovnání metod.

## **Keywords**

Recognition, Face Recognition, PCA, LDA, EBGM, method comparison.

## **Citace**

Peter Bučko: Čtečka obličejů, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Čtečka obličejů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana

.....  
Peter Bučko  
19. května 2010

## Poděkování

Touto cestou sa chcem poďakovať vedúcemu diplomovej práce Ing. Vítězslavu Beranovi za odborné vedenie, konzultácie a pomoc pri vypracovaní mojej bakalárskej práce, svojim rodičom, blízkym a priateľom za trpezlivosť, podporu a porozumenie počas celého môjho štúdia. Tiež by som sa chcel poďakovať dobrovoľníkom, ktorí mi poskytli fotografie na testovanie.

© Peter Bučko, 2010.

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

# Obsah

<b>1</b>	<b>Rozpoznávanie tváre</b>	<b>3</b>
1.1	Využitie strojového rozpoznávania . . . . .	3
1.2	Čo treba vedieť . . . . .	4
<b>2</b>	<b>Prehľad metód strojového rozpoznávania</b>	<b>5</b>
2.1	PCA - Principle Components Analysis . . . . .	5
2.2	LDA - Linear Discriminant Analysis . . . . .	8
2.3	EBGM - Elastic Bunch Graph Matching . . . . .	9
<b>3</b>	<b>Návrh aplikácie</b>	<b>14</b>
3.1	Princíp činnosti . . . . .	14
<b>4</b>	<b>Implementácia</b>	<b>17</b>
4.1	Programové vybavenie . . . . .	17
4.2	Formát dát . . . . .	17
4.3	Princíp činnosti . . . . .	19
4.4	Normalizácia, tréning, výpočet vzdialeností . . . . .	21
<b>5</b>	<b>Testovanie</b>	<b>24</b>
5.1	Testovacie dáta . . . . .	25
5.2	Metodika testovania . . . . .	26
<b>6</b>	<b>Výsledky z testovania</b>	<b>29</b>
6.1	Referenčné testy . . . . .	29
6.2	Vplyv veľkosti databáze . . . . .	29
6.3	Vplyv počtu obrazov na osobu . . . . .	30
<b>7</b>	<b>Záver</b>	<b>34</b>
<b>A</b>	<b>Stromová štruktúra testovacích dát</b>	<b>37</b>
<b>B</b>	<b>Manual</b>	<b>38</b>
<b>C</b>	<b>Obsah DVD</b>	<b>39</b>

# Úvod

V tejto práci sa budem zaoberať rozpoznávaním tváre v obraze. V prvej kapitole uvediem, čo vlastne rozpoznávanie tváří znamená a načo sa dá využívať strojové rozpoznávanie. Do svojej práce som vybral metódy Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA) a Elastic Bunch Graph Matching (EBGM, ktoré v kapitole 2 teoreticky popíšem.

Cieľom mojej práce je vytvoriť demo aplikáciu na rozpoznávanie tváří. V kapitole 3 sa budem venovať návrhu aplikácie, ktorá má fungovať na princípe podobnom dohľadovému systému, kedy v obraze detekuje tváre a snaží sa ich rozpoznať, ale zároveň dokáže spracovávať anotované dáta, u ktorých počíta štatistiku úspešnosti rozpoznávania. V kapitole 4 popíšem detaily implementácie vytvorenej aplikácie.

V kapitole 5 popíšem vytvorené testovacie dáta a použitú metodiku na otestovanie presnosti implementovaných metód a vplyvu zmeny parametrov na presnosť rozpoznávania. V kapitole 6 uvediem výsledky testov a zhodnotím ich.

# Kapitola 1

## Rozpoznávanie tváre

Rozpoznávanie osôb na základe tváre je ľudská vlastnosť, ktorou disponujeme už niekoľko tisícročí. Vďaka tomu vieme relatívne ľahko a presne rozpoznať rodinných príslušníkov, kolegov, priateľov a tiež aj nepriateľov, čo nám v minulosti pomáhalo prežiť. O rozpoznanie tváre sa stará ľudský mozog, ktorý podvedomým, automatickým procesom za približne 20 milisekúnd rozpozna známú tvár a za asi sekundu tvár neznámu. S touto, pre nás samozrejmomou schopnosťou, sa už rodíme a napríklad novorodenec dokáže už na druhý deň rozpoznať tvár svojej matky.[8]

Niektoré tváre sú pre nás typickejšie, než iné, vďaka čomu sa nám lepšie pamätajú a rozpoznávajú. Toto je typické pre osoby rovnakej rasy. V mozgu je zakódované, čo je pre každú rasu typické a preto keď žijeme dlhú dobu v komunite ľudí rovnakého výzoru, môže sa nám zdať, že osoby inej rasy vyzerajú všetky rovnako. V našom prípade europoidná rasa môže vnímať aziatov (mongoloidná rasa) a černochoch (negroidná rasa), že všetci vyzerajú rovnako. Ak sa však budeme stretávať s osobami iných tvárových črt častejšie, je možné sa ich naučiť rozoznávať.[8]

Identifikácia ľudí na základe tváre patrí do širšej skupiny biometrickej identifikácie, kde sa radia napríklad odtlačky prstov, hlasová identifikácia, ale aj krvné riečisko ruky alebo sietnica. Tie sú však považované za aktívne systémy, nakoľko vyžadujú spoluprácu identifikovanej osoby – priloženie ruky na skener, priblížiť sa na malú vzdialenosť k skeneru sietnice alebo k mikrofónu pre nahranie hlasu skúmanej osoby. Naopak, rozpoznávanie na základe ľudskej tváre patrí medzi pasívne systémy. To znamená, že pre identifikáciu nie je potrebná spolupráca alebo vedomie osoby. Vďaka tomu je tento systém možné použiť na verifikáciu alebo identifikáciu osôb pre potreby dohľadu, monitorovania, ale aj v prípade použitia v inteligentných domoch.[10]

### 1.1 Využitie strojového rozpoznávania

Strojové rozpoznávanie tváre môžeme nájsť v rôznych oblastiach každodenného života, kedy nahrádzajú alebo dopĺňajú rozpoznávanie ľuďmi.

- *Dohľadový systém* – Porovnávanie osoby zachytenej na videu s databázou tvárí (nežadúce osoby), ktoré by mohli predstavovať ohrozenie bezpečnosti. Takéto uplatnenie môžeme nájsť napríklad na letiskách alebo vlakových a autobusových staniaciach.
- *Identifikácia a verifikácia osoby* – Uplatnenie nájde v prístupových termináloch, napríklad v dochádzkových systémoch alebo prístupových bodoch do oblasti s obmedzeným prístupom.

- *Personalizované zariadenia* – Prispôsobenie nastavení a správania zariadení v inteligentnom dome alebo ako súčasť zariadení v oblasti zábavy.

## 1.2 Čo treba vedieť

Každý spôsob použitia však vyžaduje splnenie rôznych požiadaviek. Jedným z kritérií je veľkosť databázy, s ktorou dokáže pracovať – väčšia databáza bude v prípade prístupového terminálu alebo databázy nežiadúcich osôb, než v prípade osôb bývajúcich v inteligentnom dome. Ďalším kritériom, ktoré požadujeme pri rozpoznávaní osôb na základe tváre, je presnosť, s akou je identita osoby určená. V prípade dohľadového systému je akceptovateľné, ak systém určí ako podozrivú aj osobu, ktorá ňou vôbec nie je, naopak by bolo nežiadúce, aby prípadný terorista nebol rozpoznatý. Opačné kritéria platia v prípade prístupových terminálov, kde budeme radšej, ak sa osoba, ktorá síce má povolenie, ďalej nedostane, ako by mal systém vpustiť osobu, ktorá nemá oprávnenie na vstup.

Už bola spomenutá výhoda systému, že nevyžaduje pozornosť identifikovaných osôb. Ďalšou výhodou je možnosť použitia stávajúcej techniky. Nie je potrebné kupovať špeciálne a drahé zariadenia. Je možné použiť takmer akúkoľvek kameru na snímanie obrazu. Ako bude ukázané neskôr, zachytená tvár nemusí byť vo veľkom rozlíšení. Medzi nevýhody však patrí hlavne menšia presnosť pri identifikácii oproti iným spôsobom, ako je napríklad odtlačok prsta alebo obraz sietnice. Na správne rozpoznanie má vplyv osvetlenie a tiež zmeny, ktoré sa môžu prejaviť na tvári a dokážu ju zmeniť – vek, make up, okuliare, zarastenie a podobne.[10]



## Kapitola 2

# Prehľad metód strojového rozpoznávania

Metód na rozpoznávanie tvári existuje niekoľko a ich stručný prehľad a odkazy na ďalšiu literatúru je možné nájsť na web stránke venovanej rozpoznávaniu tváre [1]. Do svojej práce som vybral tri – Principle Components Analysis (PCA), Linear Discriminant Analysis (LDA) kombinovanú s PCA, Elastic Bunch Graph Mathing (EBGM), ktoré v tejto kapitole teoreticky popíšem.

### 2.1 PCA - Principle Components Analysis

PCA – Principle Components Analysis alebo Analýza hlavných komponentov, je názov metódy, ktorá je používaná pre odvodenie menejdimenzionálnej reprezentácie z viacdimenzionálnej reprezentácie vstupného obrázka. [10]

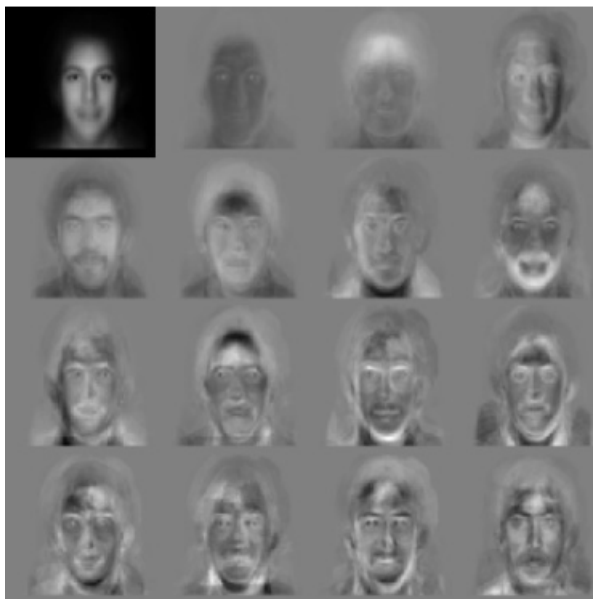
Na obraz sa môžeme pozeráť ako na vektor hodnôt – pixelov. To znamená, že obraz šírky  $m$  riadkov a výšky  $n$  stĺpcov môže byť reprezentovaný ako vektor o  $m \times n$  hodnotách, čo sa tiež nazýva pôvodný priestor obrázku, ktorý je jedným z priestorov, v ktorých môžeme obraz skúmať. Ďalším, ktorý nás zaujíma, je podpriestor vlastných vektorov (Eigenvectors) kovariančnej matice vytvorenej z tréningových dát.[7]

Principle Components Analysis pracuje tak, že najskôr vyberieme podpriestor, do ktorého sa budú premietáť všetky obrazy. Do tohto podpriestoru sú premietnuté tréningové obrazy. Každý testovaný obraz je tiež premietnutý do tohto podpriestoru a následne porovnaný s každým tréningovým obrazom. Výsledkom je vektor vzdialeností pre každý obraz, z ktorého môžeme určiť, ako sa ktorý obraz podobá.[7] Na obrázku 2.1 je ukážka získania vektoru príznakov z obrazu tváre pomocou metódy eigenfaces, ktorú PCA využíva.

Nevýhodou metódy PCA je, že na správne fungovanie potrebuje, aby bola tvár zosnímaná spredu. Výhodou je, že dokáže zredukovať pôvodný obsah dát na 1/1000.[10]

#### 2.1.1 PCA algoritmus

Podľa [9] PCA algoritmus pozostáva z 2 fáz: (1) tréningová fáza a (2) rozpoznávací fáza. V tréningovej fáze sa vzorka dát, na ktorej sa systém učí, použije na vytvorenie Eigen Matrix, ktorá transformuje vzory v obraze do bodov v Eigenspace. Tréningové obrazy sú konvertované na obrázky v stupňoch šedi a sú transformované z 2D matice na 1D stĺpcový vektor o veľkosti  $N^2 \times 1$  umiestnením stĺpcov z obrazovej matice za sebou. Tieto stĺpcové



Obrázek 2.1: Eigenfaces a získanie príznakov z obrazu tváre.[5]

vektory z  $n$  obrázkov sú uložené po stĺpcoch do dátovej matice  $X$  dimenzie  $N^2 \times n$ . Majme  $m$ , ktoré je strednou hodnotou vektoru dátových vektorov v matici  $X$ , získané z

$$m = \frac{1}{n} \sum_{i=0}^n x^i \quad (2.1)$$

Vektor dátovej matice  $X$  získame odčítaním vektora so strednými hodnotami  $m$  od všetkých stĺpcov vektorov  $X$ , aby sme dostali kovariančnú maticu stĺpcových vektorov  $\Omega$

$$\Omega = \overline{XX^t} \quad (2.2)$$

Eigenvalues a odpovedajúce Eigenvectors sú vypočítané pre kovariančnú maticu

$$\Omega V = \Lambda V \quad (2.3)$$

Kde  $V$  je matica Eigenvectors (vlastných vektorov) asociovaný s maticou  $\Lambda$  Eigenvalues (vlastných hodnôt).

Nasleduje zoradenie vlastných vektorov  $vi \in V$ , pričom sa zachováajú len tie prislúchajúce nenulovým vlastným číslam. Táto matica vlastných vektorov je vlastným priestorom  $V$ . Dátova matica  $X$  je zobrazená do vlastného priestoru, čo získame transformáciou

$$P = V^t X \quad (2.4)$$

V rozpoznávacej fáze je rozpoznávaný obraz  $I$ , konvertovaný na 1D vektor a zobrazený do rovnakého priestoru, ako trénovacie obrázky, aby sme dostali  $Z$

$$Z = X^t J \quad (2.5)$$

**Euklidova vzdialenosť**, kedy vzdialenosť  $d$  medzi  $Z$  a ostatnými testovacími obrázkami v  $P$  je merateľná použitím L2 normy (Euklidova vzdialenosť) z obrázkov  $A$  a  $B$  vzťahom

$$L_2(A, B) = \sqrt{\sum_{i=1}^N (A_i - B_i)^2} \quad (2.6)$$

Potom každý testovaný obrázok je porovnaný s každým tréningovým obrázkom a nájdený tréningový obrázok je považovaný za najbližší k testovanému obrázku.[9]

Ďalšou možnosťou, ako získať vzdialenosť dvoch obrazov je použitie metriky **Mahalinobis Cosine**, skrátene *MahCosine*. Podľa [2] a [7] je definovaný ako priestor, kde rozptyl každej pozdĺžnej dimenzie je rovný jednej. To znamená, že ak máme dva vektory  $m$  a  $n$  z MahCosine priestoru, ktoré korešpondujú vektorom  $u$  a  $v$  z neznámenho PCA priestoru, tak si musíme najskôr definovať  $\lambda_i = \sigma_i^2$ , kde  $\lambda_i$  sú vlastné čísla z PCA,  $\sigma_i^2$  je rozptyl pozdĺž ich dimenzií a  $\sigma_i$  je štandardná odchýlka. Vzťah medzi vektormi je

$$m_i = \frac{u_i}{\sigma_i} \quad (2.7)$$

a

$$n_i = \frac{v_i}{\sigma_i} \quad (2.8)$$

Mahalinobis Cosine je kosínus uhla medzi obrázkami po projekcii do rozpoznávacieho priestoru. Takže pre obrázky s vektormi  $u$  a  $v$ , ktoré korešpondujú projekciami  $m$  a  $n$  z Mahalinobisovho priestoru, je Mahalinobis Cosine:

$$D_{MahCosine}(u, v) = \cos(\theta_{mn}) = \frac{|m||n| \cos(\theta_{mn})}{|m||n|} = \frac{m \cdot n}{|m||n|} \quad (2.9)$$

### 2.1.2 Selekcia vlastných vektorov

Podľa [7] môžeme algoritmus PCA upraviť a to z dôvodu zníženia výpočtových nárokov. Ako bolo spomenuté vyššie, pri vytváraní PCA podpriestoru sa používali všetky vlastné vektory zodpovedajúce nenulovým číslam. Keďže je náročnosť priamo úmerná počtu vlastných vektorov, z ktorých tento podpriestor vytvárame, odobratím niektorých vlastných vektorov, najmä tých, ktoré ku klasifikácii obrázka neprispievajú alebo len veľmi málo, môžeme rýchlosť výpočtov znateľne zlepšiť. Okrem zvýšenia rýchlosti výpočtu sa môže zlepšiť aj samotná schopnosť správneho rozpoznania. Vlastné vektory môžeme vybrať nasledujúcimi spôsobmi: *Odrezanie posledných niekoľko percent vlastných vektorov* – Vlastné vektory máme zoradené podľa veľkosti v kovariančnej matici a odoberiem z konca napríklad 40 % vektorov. *Objem energie?* Využíva sa prah, typicky nastavený na 0,9 a my orežeme vektory tak, aby sa celková energia zachovala nad týmto prahom. Energia  $i$ -tého vlastného vektora je pomer súm prvých  $i$  vlastných čísel a sumy všetkých vlastných čísel:

$$e_i = \frac{\sum_{j=1}^i \lambda_j}{\sum_{j=1}^k \lambda_j} \quad (2.10)$$

*Prekračovanie percentuálnej hodnoty najväčšieho vlastného čísla* – je to ďalšia metóda založená na informáciách, ktoré nám poskytujú vlastné čísla. Vektory orezávame podľa toho, či dosahujú určitý pomer veľkosti ich vlastného čísla k najväčšiemu vlastnému číslu. Bežne sa používa pomer 0,01.

$$s_i = \frac{\lambda_i}{\lambda_1} \quad (2.11)$$

*Orezanie prvého vlastného vektora* – na rozdiel od predchádzajúcich spôsobov, tento predpokladá, že aj vektory zodpovedajúce najväčšiemu vlastnému číslu môžu pôsobiť nepriaznivo na presnosť rozpoznávania, pretože premietajú do seba napríklad osvetlenie a iné rušivé vplyvy.

## 2.2 LDA - Linear Discriminant Analysis

Linear Discriminant Analysis – štatistický prístup pre klasifikáciu vzoriek neznámych tried podľa tréningových vzoriek so známou triedou (viď. obrázok 2.2). Táto technika sa zameriava na maximalizáciu variácie medzi triedami (medzi rôznymi užívateľmi) a minimalizáciu variácie v rámci triedy (medzi rôznymi obrázkami jedného užívateľa).[10]



Obrázok 2.2: Príklad šiestich tried používajúcich LDA.[5]

V ďalšom texte budem vychádzať z [6] a [14]. Ak je súbor tréningových dát popísaný pre každú osobu, technika učenia s učiteľom ako LDA je výhodnejšia pre extrakciu príznačkov tváre v porovnaní s metódou učenia bez učiteľa. Keď použijeme učenie s učiteľom, môžeme zanedbať variácie spôsobené osvetlením. LDA si stále udrží informáciu o identite. LDA alebo Fisher Linear Discriminant je triedne špecifickou metódou v tom zmysle, že reprezentuje dáta užitočné pre vytvorenie klasifikácie. Majme súbor z  $N$  obrázkov  $\{x_1, x_2, \dots, x_N\}$  s každým obrázkom patriacim do jednej z  $C$  tried  $\{X_1, X_2, \dots, X_C\}$ . LDA vyberie takú lineárnu transformačnú maticu  $W$ , že pomer rozptylu medzi triedami a rozptylu v rámci tried je maximálny. Teda rozptyl v rámci triedy je čo najmenší a medzi rôznymi triedami je čo najväčší. Matematicky, medzitriedny rozptyl matice  $S_B$  a rozptyl matice v rámci triedy  $S_W$  je definovaný ako

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (2.12)$$

a

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} N_i (x_k - \mu_i)(x_k - \mu_i)^T \quad (2.13)$$

kde  $\mu_i$  značí stredový obraz z triedy  $X_i$ ,  $\mu$  znamená stredový obraz celého dátového súboru a  $N_i$  značí číslo obrázku v triede  $X_i$ . Ak nie je matica  $S_W$  rozptylov v rámci triedy jedinečná, LDA nájde ortonormálnu maticu  $W_{opt}$  hľadajúc najvyšší pomer determinantu z medzitriedneho rozptylu matice k determinantu z matice rozptylov v rámci triedy. To znamená, že LDA projekčná matica je reprezentovaná vzťahom

$$W_{opt} = \underset{W}{\operatorname{argmax}} \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 w_2 \dots w_m] \quad (2.14)$$

Súborom riešení  $\{w_i | i = 1, 2, \dots, m\}$  je zovšeobecnený vlastný vektor z  $S_B$  a  $S_W$  zodpovedajúci  $m$  najväčšej vlastnej hodnote  $\{\lambda_i | i = 1, 2, \dots, m\}$ , t.j.,  $S_B w_i = \lambda_i S_W w_i, i = 1, 2, \dots, m$ . Vo všeobecnosti, na prekonanie jedinečnosti  $S_W$ , sa použije najskôr PCA na zredukovanie vektora dimenzií. Každý LDA vektor príznakov je reprezentovaný vektorom zobrazení  $Y_k = W_{opt}^T x_k, k = 1, 2, \dots, N$ .

Pre určenie vzdialeností obrazov získaných pomocou tejto metódy, je možné použiť metriku Euclidean.

## 2.3 EBGM - Elastic Bunch Graph Matching

Táto metóda využíva to, že všetky ľudské tváre majú podobnú topologickú štruktúru. Tvár môžeme reprezentovať grafmi, uzlami umiestnenými na hlavných častiach tváre (hrany, nos,...) a označenými hranami s 2D vzdialenostnými vektormi. Každý uzol obsahuje súbor štyridsiaticich koeficientov Gaborových vlniek rôznych váh a orientácií (fáza, amplitúda). Tie sú nazývané Jet-y. Rozpoznanie je založené na pomenovaných grafoch. Pomenovaný graf je súborom uzlov spojených hranami, uzly sú označené jetmi, hrany sú označené vzdialenostmi. [1]

V ďalšom texte budem vychádzať z [11] a [12].

### 2.3.1 Jet-y

Jet-y sú založené na vlnkovej transformácii, definovanej ako konvolúcia obrázka s Gaborým jadrom (Gabor kernel)

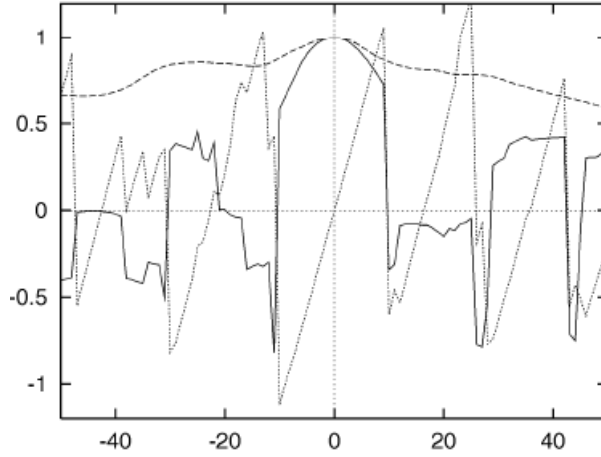
$$\Psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2} \exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right) \left[ \exp(i\vec{k}_j \vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right) \right] \quad (2.15)$$

vo forme rovinných vlniek s vlnkovým vektorom  $\vec{k}_j$ , obmedzenú Gaussovskou funkciou s relatívnou šírkou  $\sigma = 2\pi$ . Použijeme diskretný súbor z piatich rôznych priestorových kmitočtov a ôsmich orientácií. Pre obrázok veľkosti 128x128 pixelov, najnižšiu a najvyššiu frekvenciu majú vlnové dĺžky zo šestnástich a štyroch pixelov. Posledný člen z 2.15 vytvorí jadrá DC-free, t.j. integrál  $\int \Psi_j(\vec{x}) d^2\vec{x}$  vypadne. Toto je známe ako vlnková transformácia, pretože rodina jadier si je podobná, všetky jadrá sa generujú z jednej hlavnej vlnky rozťahnutím a rotáciou. Jet  $J$  je definovaný ako súbor  $\{J_j\}$  štyridsiaticich komplexných koeficientov Gaborových vlniek získaných pre jeden obrazový bod. To môžeme zapísať ako  $J_j = a_j \exp(i\Phi_j)$  s magnitudami  $a_j(\vec{x})$ , ktoré pomaly kolíše s pozíciou a fázami  $\Phi_j(\vec{x})$ , ktoré rotuje s rýchlosťou danou priestorovou frekvenciou vlnového vektora jadier  $\vec{k}_j$ . Počas tejto fázovej rotácie, jety vyberú z bodov z obrazu len pár pixelov majúcich rôzne koeficienty, aj keď reprezentujú takmer rovnaký lokálny príznak. Toto môže spôsobiť viaceré problémy pri

porovnávaní. Preto ignorujeme jej fázu alebo kompenzujeme jej variácie explicitne. Funkcia podobnosti

$$S_a(J, J') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j a_j a'^2_j}} \quad (2.16)$$

ignoruje fázu. S jetom  $J'$  zobraným z fixnej pozície obrazu a jety  $J = J(\vec{x})$  zobratým z rôznych pozícií  $\vec{x}$ ,  $S_a(J(\vec{x}), J')$  je hladkou funkciou s lokálnym *optima*, ktoré tvorí veľké dno attractorovej vane obrázka 2.3, vedúcej k prudkej a osvedčenej konvergencii s jednoduchým hľadaním motód ako postupného sklonu alebo difúzie. Použitie fázy má dve výhody.



Obrázok 2.3: Podobnosť  $S_a = (J, J')$  (čiarkovane) a  $S_\Phi = (J, J')$  (plnou čiarou) jetu  $J'$  z ľavého oka tváre a jetu  $J$  z pravého oka tváre na rovnakej horizontálnej úrovni. Bodkovaná čiara ukazuje odhadovaný posun  $\vec{d}(J, J')$  (deleného ôsmimi pre nastavenie rozsahu súradníc). Pravé oko je vzdialené 24 pixelov od ľavého oka, generujúc lokálne maximum pre obe podobnostné funkcie a nulový posun v blízkosti  $d_x = -24$  [11]

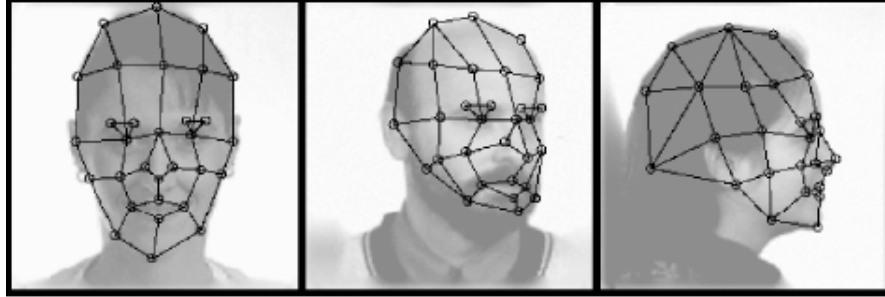
Prvou, fázova informácia je potrebná na rozlíšenie vzorov s podobnou magnitúdou, ktorá môže nastať. Druhou, pretože sa fáza rýchlo mení s pozíciou, poskytuje určenie stredovej hodnoty pre vhodnú lokalizáciu jetu v obraze. Predpokladajme, že jety  $J$  a  $J'$  odkazujú na umiestnenie objektu s malým relatívnym posunom  $\vec{d}$ , môžu byť fázovým posunom približne kompenzované pre podmienky  $\vec{d}\vec{k}_j$ , vedúcu k funkcii fázovej citlivosti podobnostnej funkcie

$$S_\Phi(J, J') = \frac{\sum_j a_j a'_j \cos(\Phi_j - \Phi'_j - \vec{d}\vec{k}_j)}{\sqrt{\sum_j a_j^2 \sum_j a_j a'^2_j}} \quad (2.17)$$

V poradí počítania, posun  $\vec{d}$  musí byť odhadovaný. Toto môže byť spravené maximalizáciou  $S_\Phi$  v Taylorovom rozvoji okolo  $\vec{d} = 0$ , ktoré je nútené padnúť z dvojdimenzionálneho  $\vec{d}$  do 40 fázového rozdielu  $\Phi_j - \Phi'_j$ . Veľký posun z ôsmich pixelov môže byť odhadovaný, ak sú fázove koeficienty vyšších frekvencií upravené násobením  $2\pi$  závisjúcom na disparite odhadovanej z nižších koeficientov frekvencie. Je veľkou výhodou tejto druhej podobnostnej funkcie, že ukazuje túto informáciu o posune. Profily podobností a odhadovaného posunu sú ukázané na obrázku 2.3.

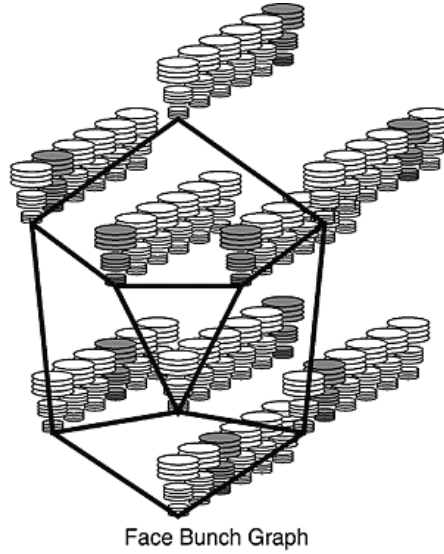
### 2.3.2 Grafy

Pomenovaný graf  $G$  reprezentuje tvár pozostávajúcu z  $N$  uzlov spojenými  $E$  hranami. Uzly sú umiestnené na význačných bodoch tváre  $x_n, n = 1, \dots, N$ , nazývané dôveryhodnostné body (fiducial points), napríklad zreničky, rohy úst, špička nosa, vrch a spodok uší, atď. Tento tvárový graf je objektovo prispôsobivý, pretože jeho geometrická štruktúra je adaptovateľná podľa štruktúry objektu 2.4. Uzly sú pomenované jetmy  $J_n$ . Hrany sú pomenované



Obrázok 2.4: Objektovo adaptovaná sieť pre rôzne pózy. Uzly sú pozicované automaticky pomocou EBGM. (Zjednodušená verzia, chýbajú nejaké uzly kvôli prehľadnosti). Môžeme vidieť, vo všeobecnosti, porovnávanie hľadá dôveryhodnostné body celkom presne. Ale netafenie sa do pozície sa stáva, napríklad, pre tvár v strede. Brada nebola nájdená správne v poradí; najľavejší uzol a uzol pod ním by mal byť na vrchu a pod uchom. [11]

dvojdimenziálnymi vzdialenostnými vektormi  $\Delta x_e = x_n - x_n', e = 1, \dots, E$ , kde hrany  $e$  spájajú uzly  $n'$  s  $n$ . (Odvolávame sa na geometrickú štruktúru grafu, neoznačeného jetom, ako sieť.) Grafy pre rôzne hlavné pozície sa rozlišujú v geometrii a lokálnych príznakov (jetov). Hoci dôveryhodnostné body odkazujú na odpovedajúce umiestnenia objektov, niektoré môžu byť okludované, a jety, podobne ako vzdialenosti, môžu kolísať v závislosti na rotácii. Aby sme boli schopní porovnať grafy z rôznych pozícií, manuálne definujeme ukazatele, aby sme dokázali asociovať odpovedajúce uzly v rozličných grafoch. Aby sme mohli automaticky získať grafy z obrazu pre nové tváre, potrebujeme skôr jednu všeobecnú reprezentáciu, než mnoho konkrétnych. Táto reprezentácia by mala pokrývať široký rozsah možných variácií príslušnej tváre, ako napríklad rôzne tvary očí, úst alebo nosov, rôzne typy brád, variácie pohlavia, veku, rasy, atď. Je jasné, že by bolo príliš drahé pokryť každú význačnú časť tváre kombináciou oddelených grafov. Namiesto toho kombinujeme reprezentatívnu vzorku  $M$  individuálnych grafových modelov  $G^{B_m} (m = 1, \dots, M)$  do štruktúry podobnej hromade, nazývanej face bunch graph (FBG) (viď obrázok 2.5). Každý model grafu má rovnakú štruktúru. Siete a uzly odkazujú na rovnaké dôveryhodnostné body (fiducial points). Súbor jetov odkazujúcich na dôveryhodnostné body sa nazýva zväzok (bunch). Očný zväzok môže na pokrytie týchto lokálnych variácií zahŕňať napríklad jety zo zatvorených, otvorených, mužských a ženských očí atď. Odpovedajúci FBG  $B$  je potom daný rovnakou sieťovou štruktúrou ako individuálne grafy, jeho uzly sú pomenované zhlukmi jetov  $J_n^{B_m}$  a ich hrany sú označené priemernými vzdialenostami  $\Delta x_e^B = \frac{\sum \Delta x_e^{B_m}}{M}$ . Počas hľadania dôveryhodnostných bodov v novom obraze tváre, nižšie popísaná procedúra vyberá najlepšie sa hodiaci jet, nazývaný lokálny expert (local expert), zo zväzku určeného každému dôveryhodnému bodu. Teda, úplná kombinácia jetov v zväzkovom grafe je dostupná, pokrývajú väčší rozsah variácií tváří, ako predstavujú grafy konstitučných modelov.



Obrázek 2.5: Face Bunch Graph slúži ako všeobecný popis tváří. Každá halda diskov reprezentuje jet. Z hľuku jetov sú vybrané a pripojené do jednotlivých uzlov len najlepšie sediace a tie sú potom vybrané na porovnanie. Je to znázornené šedým zafarbením.[11]

### 2.3.3 Porovnávanie Elastic Bunch Graph

Prvý súbor grafov je generovaný manuálne. Uzly sú umiestnené na dôveryhodnostné body, ako aj hrany medzi uzlami, takisto pre rôzne natočenia. Stačí, aby raz mal systém FBG (možné založiť len jeden manuálne definovaný model), grafy pre nové obrázky môžu byť generované automaticky pomocou porovnávania elastic bunch graphov. Na začiatku, keď FBG obsahuje len niekoľko tváří, je potrebné vykonať korekciu výsledného porovnania, ale keď je už raz FBG dostatočne obsiahle (približne 70 grafov), môže sa porovnávanie a generovanie veľkých modelových grafov robiť automaticky. Porovnávanie FBG na novom obrázku je dokončené hľadaním najväčšej hodnoty grafov podobnosti medzi obrazovým grafom a FBG identickej pozície. To závisí na podobnosti jetov a podmienkach topografie, ktoré berú do úvahy zmenu tvaru siete obrazu relatívne k FBG sieti. Pre graf obrázku  $G^I$  s uzlami  $n = 1, \dots, N$  a hranami  $e = 1, \dots, E$  a FBG  $B$  s modelovými grafmi  $m = 1, \dots, M$  je podobnosť definovaná ako

$$S_B(G^I, B) = \frac{1}{N} \sum_n \max_m (S_\Phi(J_n^I, J_n^{Bm})) - \frac{\lambda}{E} \sum_e \frac{(\Delta x_e^I - \Delta x_e^B)^2}{(\Delta x_e^B)^2} \quad (2.18)$$

kde  $\lambda$  určuje relatívnu dôležitosť podobnosti jetu a podmienku topografie.  $J_n$  sú jety na uzloch  $n$  a  $\delta x_e$  sú vektory vzdialeností použité ako označenie hrán  $e$ . Pretože FBG poskytuje viacero jetov pre každý dôveryhodnostný bod, vyberie sa jeden najlepší a použije sa pre porovnanie. Tieto najlepšie vyhovujúce jety slúžia ako lokálny expert pre obraz tváre. A heuristický algoritmus je použitý na nájdenie obrazového grafu, ktorý maximalizuje funkciu grafovej podobnosti. Najskôr sa nájde umiestnenie tváre riedkym skenovaním FBG cez obraz. Potom je FBG striedavo menený na veľkosť a pomer strán prispôbovaný správne formátu tváre. Tieto kroky nemajú žiadnu cenu v topografickom člene podobnostnej funkcie, pretože rohové označenia sú adekvátne transformované. Nakoniec sú všetky uzly



posunuté lokálne a relatívne k ostatným, aby sa ďalej graf podobnosti optimalizoval. Len uzly umiestnené s malou odhadovanou chybou sú brané do úvahy a len lokálna deformácia je viazaná topografickou podmienkou.

#### **2.3.4 Rozpoznávanie**

Potom, ako sme získali grafový model z galérie obrázkov a grafy obrázkov z testovaných obrázkov, je možné rozpoznanie s relatívne malým výpočtovým úsilím pri porovnávaní obrazového grafu so všetkými grafmi modelov a vybratím jedného s najvyššou hodnotou podobnosti. Podobnostná funkcia, ktorú používame pre porovnanie grafov je priemerom nad podobnosťami medzi párami odpovedajúcich si jetov. Niektoré jety v jednej pozícii nemusia mať odpovedajúce jety v inej pozícii, pretože sú skryté.

## Kapitola 3

# Návrh aplikácie

Aplikácia bola navrhnutá ako testovací nástroj na získanie údajov o presnosti implementovaných metód rozpoznávania tvárí a tiež ako demo pracujúce podobne ako software pre dohľadový systém. To znamená, že má k dispozícii databázu osôb s obrazmi tvárí a ako vstup dostáva obraz, v ktorom sa môžu nachádzať osoby. Tieto obrazy prehľadáva a hľadá tváre, ktoré následne porovnáva s osobami v databáze a hľadá zhodu. Uplatnenie takýchto systémov je napríklad na letiskách, autobusových a vlakových staniách a tiež na športových štadiónoch, kde sa hľadajú nežiadúce osoby.

### 3.1 Princíp činnosti

Aplikácia je ovládaná parametrami pri spustení a ako vstup vyžaduje cestu k adresáru s dátami, ktoré musia byť v požadovanom tvare popísanom v časti 4.2. Jednotlivé parametre ktorými je možné zmeniť chovanie a výsledky spracovania, ich formát a popis je v prílohe B.

Beh programu by sa dal rozdeliť do piatich blokov znázornených na obrázku 3.1. Každý blok predstavuje jednu etapu spracovania dát, pričom spracovanie každého ďalšieho kroku je závislé na spracovaných dátach v predchádzajúcom bloku.

#### 3.1.1 Inicializácia

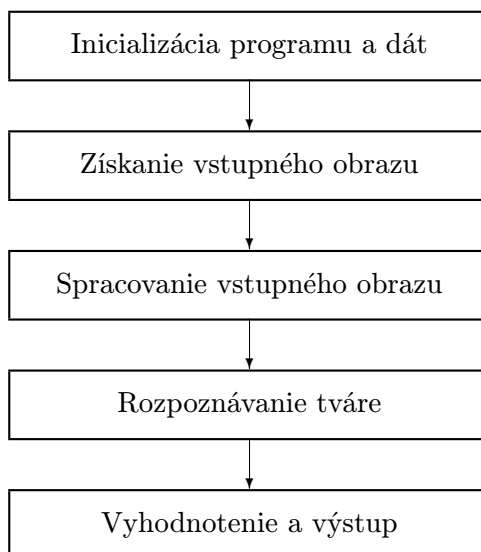
V rámci inicializácie sa kontrolujú vstupné parametre programu, nastavujú sa premenné prostredia na predvolené hodnoty alebo na hodnoty nastavené užívateľom. V adresári zadanom ako vstupný adresár sa kontroluje existencia a prístupnosť súborov potrebných pre správny beh programu. O používaných adresároch a súboroch je písané v časti 4.2.

Databázové obrazy uložené v rôznych podporovaných formátoch sú konvertované do `pgm` súborov, ktoré sa budú ďalej spracovávať a do súborov `png`, aby bolo možné zobrazovať obrazy vo výstupe.

Ďalej sa skontroluje a prípadne podľa potreby vykoná normalizácia databázových dát a tréning databázy. Tiež sa načíta zoznam databázových súborov zo súboru `database.srt`.

#### 3.1.2 Získanie vstupného obrazu

Vstupné dáta môžu byť vo formáte obrazových súborov, obrazových súborov s definovanými koordinátami očí, video súboru uloženého na disku a videa obrazu z webkamery.



Obrázek 3.1: Bloková schéma behu programu

Okrem vstupu z kamery, musia mať všetky typy vstupných médií k dispozícii súbor so zoznamom testovaných súborov, pričom jeden riadok obsahuje meno jediného súboru. V prípade obrazových súborov s definovanými koordinátami očí, musí tento súbor obsahovať vedľa názvu súboru aj koordináty očí v rovnakom formáte, ako je štandardný súbor s koordinátami očí (viď 4.2).

### 3.1.3 Spracovanie vstupného obrazu

V získanom obraze sa najskôr detekujú tváre a pre každú detekovanú tvár sa hľadajú oči. V prípade korektného nájdenia je tvár uložená vo formáte `pgm` a `png`, vytvorí sa súbor `coords.txt` s názvom testovaného súboru a nájdenými koordinátami očí. K zoznamu databázových dát sa pripojí názov aktuálneho testovaného záznamu. Testovaný obraz sa ešte znormalizuje, pričom výsledok normalizácie sa uloží do pracovného adresára.

Spracovanie pre obrazové súbory s definovanými koordinátami sa od ostatných líši v tom, že nie je potrebné detekovať tváre, nakoľko je od takéhoto súboru požadované, aby bola na jednom obraze iba jedna tvár. Tiež sa nerobí detekcia očí, pretože tieto údaje získame pri načítavaní súboru. Oproti ostatným typom médií je rozdiel aj v názvoch, pod ktorými sa ukládajú dáta. Tie nemajú názov generovaný, ale je preberaný z názvu pôvodného súboru.

### 3.1.4 Rozpoznávanie tváre

Pri rozpoznávaní sa vezme aktuálne spracovávaný obraz a porovná sa s každým obrazom v databáze. Výsledkom je súbor so vzdialenosťami testovaného súboru k ostatným obrazom. Nájdenie odpovedajúceho obrazu spočíva v prechádzaní tohto súboru a hľadaní záznamov, ktorých vzdialenosť je menšia ako určitý nastavený prah.

V prípade obrazov s definovanou pozíciou očí sa uchováva iba jeden záznam a ten sa považuje za jediný nájdený, aby bolo možné vyhodnotiť úspešnosť rozpoznávania.

### 3.1.5 Vyhodnotenie a výstup

Pri obraze z kamery, z videa alebo z obyčajného obrazu, musí správnosť rozpoznania vyhodnotiť sám užívateľ skontrolovaním výstupného súboru. Automatické vyhodnotenie rozpoznávania je možné vykonať iba v prípade obrazov s definovanými súradnicami očí, kedy sa vyhodnocuje pre každý testovaný obraz, či bol rozpoznaný/nerozpoznaný správne/nesprávne. Tiež sa uchovávaajú počty spracovávaných súborov, ktoré mali byť rozpoznané, nemali byť rozpoznané a celkový počet spracovaných obrazov, aby bolo možné určiť úspešnosť rozpoznávania. Výsledky rozpoznania a prípadne štatistika rozpoznávania sa nakoniec zapíše do výstupného súboru.

# Kapitola 4

## Implementácia

V tejto kapitole popíšem implementáciu demonštračnej aplikácie, ktorú som vytvoril na testovanie. O využitom programovom vybavení a použitých technológiach bude popísané nižšie. Ďalej detailnejšie popíšem princíp činnosti a dáta, s ktorými aplikácia pracuje, hlavne ich formát a štruktúru. Nakoniec popíšem vlastnosti aplikácie a uvediem jej výhody a nevýhody.

### 4.1 Programové vybavenie

Aplikácia bola písaná objektovo v jazyku C++ vo vývojovom prostredí MS Visual Studio 2008 spolu s knižnicou OpenCV 2.0. Ďalej som využíval programy z projektu The CSU Face Identification Evaluation System. Výstupné súbory boli vo forme html kódu.

#### 4.1.1 OpenCV 2.0

OpenCV je knižnica funkcií pre počítačové videnie napísaná v jazyku C a C++ a spustiteľná na všetkých hlavných platformách. Obsahuje viac ako 500 optimalizovaných funkcií so zameraním na spracovanie obrazu.[4]

#### 4.1.2 The CSU Face Identification Evaluation System

The CSU Face Identification Evaluation System, ďalej len CSU systém, bol vytvorený na ohodnocovanie výkonnosti systémov rozpoznávajúcich tváre. Celý projekt je písaný v jazyku C a obsahuje implementované štyri metódy. Konkrétne Principle Components Analysis (PCA), Linead Discriminant Analysis (LDA), Bayesian Intrapersonal/Extrapersonal Classifier (BIC) a Elastic Bunch Graph Matching (EBGM). Systém pôvodne vytvorený pre OS Linux som musel upraviť, aby fungoval na OS Windows doplnením zdrojových kódov, ktoré sú štandardné pre Linux, ale nie pre Windows.

### 4.2 Formát dát

Pred popisom samotnej aplikácie je vhodné najskôr uviesť a popísať dáta, ktoré aplikácia používa, pričom sa budem odvolávať na prílohu A, kde je uvedená stromová štruktúra, v ktorej sa dáta uchovávajú. Pre správny chod aplikácie je nevyhnutné, aby dáta dodržiavali túto hierarchiu a formát.

Aplikácia využíva štyri základné adresáre:

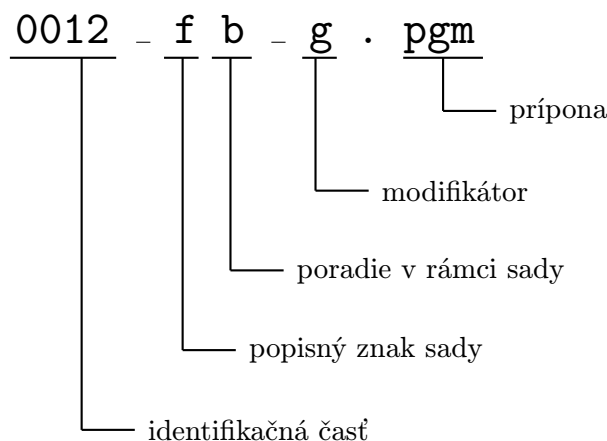
- **database** – Adresár s databázovými dátami. Musí obsahovať obrazy rozpoznávaných osôb v podporovanom formáte. Podporované formáty obrazov sú `jpg`, `png`, `ppm` a `pgm`. Ďalej musí databáza obsahovať súbory `eye_coords.txt` a `database.srt`. Samotná aplikácia si následne vytvorí konvertované obrazové súbory vo formáte `pgm` a `png`, pomocný súbor `normalized.txt` a tréningový súbor s názvom podľa použitej metódy a príponou `train`.
- **probe** – Adresár obsahujúci testovacie dáta, kde musia byť uložené všetky súbory, ktoré chceme testovať (ak netestujeme obraz z kamery). Adresár musí obsahovať súbor `list.txt` so zoznamom testovaných súborov. Aplikácia si sem ukladá obrazy vo formáte `pgm` a `png` z testovaných obrazov a výsledky pre konkrétne testy. Každý záznam sa nachádza vo vlastnom adresári, ktorého meno je buď generované alebo pre obrazy s definovanými koordinátami očí prebraté podľa pôvodného súboru.
- **wd** – je takzvaný pracovný adresár, kde sa ukladajú dáta z normalizácie databázy, a aj testovacích dát.
- **output** – obsahuje súbory s výstupnými html súbormi.

V nich má uložené alebo si do nich ukladá súbory typu:

- **eye\_coords.txt** – obsahuje zoznam súborov a koordináty očí. Každý riadok obsahuje vždy len jeden názov súboru bez prípony, za ktorým nasledujú štyri čísla, ktoré sú súradnicami očí. Súradnice sa zadávajú v poradí x-ová súradnica, y-ová súradnica, postupne pre ľavé oko a pravé oko. Na fotke je ľavé oko s vyššou x-ovou súradnicou, pretože sa berie skutočné ľavé a pravé oko osoby.
- **database.srt**, **list.srt** – obsahuje zoznam súborov rozdelených podľa príslušnosti k jednotlivým osobám. Na každý riadok sú umiestnené názvy súborov s príponou `sfi` patriace jednej osobe. Takto aplikácia vie, ktoré súbory patria ktorej osobe. Prvý súbor musí byť prítomný pred spustením aplikácie, druhý sa generuje počas behu programu k testovacím dátam.
- **súbor so vzdialenosťami** – takzvaný distance file, ktorý aplikácia generuje do adresára s testovacími súbormi a obsahuje zoznam databázových súborov a vzdialenosti medzi nimi. To znamená, že čím sú si obrazy bližšie, tým je vzdialenosť menšia. Tu si treba dať pozor na rôzne metriky, ktoré metódy používajú.
- **list.txt** – sa používa v spojení s testovacími súbormi. V podstate obsahuje zoznam všetkých súborov, ktoré sa budú testovať, pričom na jednom riadku je jeden súbor – názov súboru s príponou. Ak sa majú použiť obrazy s definovanými koordinátami očí, musia byť za názvom súboru súradnice očí vo formáte, v akom sú v súbore `eye_coords.txt`.
- **\*.pgm** – typ obrazových súborov v stupňoch šedi. Do týchto súborov sa konvertujú všetky súbory, ktoré sa budú spracovávať a tiež sú do tohto formátu generované normalizované obrazy.
- **\*.sfi** – používa CSU systém a ide o reprezentáciu obrazu pomocou čísel s desatinnou čiarkou.[2].
- **\*.train** – vygenerovaný súbor, ktorý je výsledkom tréningovania databázy. Názov sa určuje podľa použitej metódy.

Jednotlivé názvy obrazových súborov v databázy sú ukladané vo formáte podľa obrázku 4.1, kde

- *identifikačná časť* – pozostáva zo štvorčíslicia a je pre každú osobu jedinečná. Obrazy v rámci jednej osoby majú rovnaké číslo
- *popisný znak sady* – určuje, ako je tvár odfotená. Toto značenie vychádza zo značenia FERET obrázkov. V prípade implementovaných metód PCA a LDA, ktoré pracujú iba s obrazmi tváre spredu, je tento znak vždy *f*
- *poradie v rámci sady* – určuje, koľký v poradí je obraz pre danú osobu a spôsob zachytenia. Označuje sa písmenami *a* až *z*
- *modifikátor* – určuje, či je tvár zachytená s nejakou význačnou zmenou tváre, ako sú napríklad okuliare, ktoré sa označujú písmenom *b*



Obrázek 4.1: Vysvetlenie jednotlivých častí názvu databázového súboru.

## 4.3 Princíp činnosti

Aplikácia je postavená na triede `faceRec`, ktorá vyžaduje cestu k dátam, ktoré inicializuje a zadaný vstupný testovací obraz spracuje a vytvorí výstupný súbor s výsledkami rozpoznávania. Princíp činnosti celej aplikácie vychádza z návrhu popísanom v predchádzajúcej časti. Detaily implementácie popíšem z pohľadu jednotlivých blokov z blokovej schémy uvedenej na obrázku 3.1.

### 4.3.1 Inicialiázia

Na začiatku programu sa vytvorí objekt triedy `faceRec`, ktorého konštruktor zabezpečí vnútorné nainicializovanie premenných a ciest. Volaním metódy `argProcess` sa spracujú predané parametre, v tomto prípade parametre celého programu, s ktorými bol spustený. Parametre sú rozparsované a uložené do príslušných premenných objektu. Metóda `checkAll`

skontroluje stav povinných parametrov, testovacích a databázových súborov, načíta klasifikátory pre detekciu a načíta zoznam databázových súborov zo súboru `database.srt`.

Ak všetko prebehlo v poriadku, volá sa metóda `normalizeDatabase`, ktorá zabezpečí vytvorenie potrebných adresárov na ukladanie generovaných súborov, spraví konverziu databázových súborov uložených v rôznych formátoch v adresári `input` na súbory `pgm` v adresári `source`. V prípade potreby je databáza normalizovaná volaním programu `csuPreprocessNormalize` s príslušnými parametrami z CSU systému. Detaily samotnej normalizácie sú uvedené v kapitole 4.4. Normalizované obrazy sú ukladané do pracovného adresára.

Po normalizácii metóda `trainDatabase`, ak je potrebné, zariadi natrénovanie databázy správnym volaním programu `csuSubspaceTrain` z CSU systému. Výstupom je súbor s príponou `train`. Meno súboru je odvodené od použitej metódy.

### 4.3.2 Získanie vstupného obrazu

Získanie vstupného obrazu je oddelené od triedy `faceRec`, aby ostala táto trieda prehľadnejšia. Takto je možné objektu predať iba samotný obraz v premennej typu `IplImage*`. Obraz je získavaný podľa typu vstupného média. Využívajú sa funkcie z knižnice OpenCV na získanie obrazu zo súboru (`cvLoadImage`), z videa a z webkamery (`cvQueryFrame`).

### 4.3.3 Spracovanie vstupného obrazu

Vo vstupnom obraze sa metódou `detectFaces` detekujú všetky tváre. Pre každú nájdenú tvár sa musí zavolať metóda `processProbe`, ktorá v nájdenej tvári detekuje oči volaním metódy `detectEyes`, ktorá vráti vektor ľavých očí a vektor pravých očí. V tejto metóde môže dôjsť k odfiltraniu niektorých falošne detekovaných tvárí. Môže sa však stať, že odfiltruje aj správne nájdené tváre, keď nenájde odpovedajúce pozície očí. Ďalšou nechcenou vlastnosťou, s ktorou som sa musel vyrovnáť, bola detekcia viac potencionálnych očí, pričom nie je možné rozoznať správne detekované oči od nesprávnych. Z tohto dôvodu sa vyberajú také páry očí, ktoré by polohou mohli odpovedať skutočným očiam v obraze. Pre každý takto nájdený pár sa vytvorí záznam, v ktorom sa aplikácia následne pokúsi rozpoznať osobu. Pokiaľ je vstupom obraz s definovanými súradnicami očí, detekcia tváre ani detekcia očí sa nevykonáva.

Pre obraz tváre v adresári `source` vytvorí príslušný adresár s vygenerovaným názvom podľa pôvodného názvu súboru, respektíve pri normálnych obrazoch, podľa poradového čísla aktuálne spracovávaného obrazu. Do tohto adresára je vygenerovaný súbor `eye_coords.txt` s pozíciou očí a súbor `list.srt` so zoznamom databázových súborov a vloženým názvom aktuálne spracovávaného súboru. Vytvorí sa `pgm` a `png` súbor. Testovaný súbor je následne ešte normalizovaný.

### 4.3.4 Rozpoznávanie tváre

Pri rozpoznávaní spustí metóda `project` program `csuSubspaceProject` (viď 4.4) so správnymi parametrami. Výsledkom tohto programu sú súbory so vzdialenosťami medzi jednotlivými súbormi databázy a testovaného obrázku uložené v adresári aktuálne spracovávaného obrazu.

Metóda `recognize` prejde súbor so vzdialenosťami a nájde databázové obrazy, ktoré majú vzdialenosť nižšiu ako je nastavený prah a uloží ich na vyhodnotenie.



### 4.3.5 Vyhodnotenie a výstup

Ako už bolo spomínané v návrhu, verifikovať nájdenú osobu k testovanej musí užívateľ sám. Pre obrazy so zadanými pozíciami očí sa použije metóda `recognitionAccuracy`, ktorá zistí stav rozpoznania a to tak, že porovná začiatky názvov súborov, ktoré odlišujú jednotlivé osoby (napríklad súbor `0012_fa.pgm` patrí rovnakej osobe ako súbor `0012_fc_g.pgm`), pričom tieto úspešnosti si uchováva, aby bolo možné do výstupného súboru zobrazíť štatistiky pomocou metódy `generateFerretStats`.

Každý rozpoznávaný obraz a výsledok jeho rozpoznania je vložený do výstupného súboru pomocou metódy `toOutput`. Táto metóda zariadi prepočet hodnoty vzdialenosti na percentuálnu hodnotu (pomocou metódy `getRecLevelFromRate`) a v prípade, že je k dispozícii aj hodnota z `recognitionAccuracy`, zobrazí okolo testovaného obrazu aj farebný rámik a popis podľa výsledku rozpoznania. Výstupný súbor používa na zorbazenie súborov vo formáte `png`, na ktoré sú všetky databázové aj testované orbazy prevádzané.

Nakoniec metóda `generateOutput` vygeneruje výsledky do výstupného súboru.

## 4.4 Normalizácia, tréovanie, výpočet vzdialeností

V tejto časti popíšem princíp troch programov, ktoré využívam vo svojej aplikácii. Tie sú prebraté z projektu The CSU Face Identification Evaluation System z oddelenia Computer Science na Colorado State University.

### 4.4.1 Normalizácia – `csuPreprocessNormalize`

Normalizácia redukuje nežiadúce obrazové hodnoty zarovnaním tváre, vyvažuje hodnoty jednotlivých pixelov a normalizuje kontrast a jas. Celý proces pozostáva z piatich krokov:

1. *Konverzia na čísla s desatinnou čiarkou* – Konvertuje 256 úrovní šedej z celých čísel na odpovedajúce hodnoty v číslach s desatinnou čiarkou.
2. *Geometrická normalizácia* – Oči sa umiestnia na predvolené súradnice. Zvyšok tváre je potom prispôsobený podľa ich umiestnenia posunom a rotáciou, zrotovaním a zmenou rozmerov.
3. *Maskovanie* – Aby sa odstránili z obrazu pixely, ktoré pre rozpoznávanie nenesú žiadnu informáciu, vymaskuje sa oblasť tváre a ostatné pixely sú zmenené na neutrálnu hodnotu.
4. *Vyrovnanie histogramu* – Vyrovnanie histogramu pre nemaskovanú časť tváre.
5. *Normalizácia pixelov* – Nastavenie strednej hodnoty na nulu a štandardnej odchýlky na hodnotu jedna.

Pri normalizácii je možné nastaviť oblasť, ktorá sa bude maskovať. Pôvodné nastavenie spôsobovalo orezanie aj časti tváre a neponechávalo sa žiadne okolie tváre. Podľa výsledkov uvedených v [7] som sa rozhodol robiť testy s normalizovaním tváre na širšie okolie. Takto normalizovanú tvár spolu s pôvodným obrazom je vidieť na obrázku 4.2.



Obrázek 4.2: Ukážka pôvodného obrazu a normalizovaného obrazu.

#### 4.4.2 Trénovanie – csuSubspaceTrain

Trénovanie databázových obrazov sa vykoná načítaním obrazov do matice  $M$  a spočítaním vlastného priestoru (eigenspace). Výstupom programu je trénovací súbor, ktorý obsahuje popis parametrov tréovania, strednú hodnotu tréovacích obrazov, vlasné hodnoty (eigenvalues) a bázu pre podpriestor. [2] Metóda na vytvorenie vlastných priestorov a vlastných vektorov popísaná v kapitole 2.1 môže viesť k príliš veľkým kovariančným maticiam. Preto sa v implementácii tohto programu používa mierne odlišný prístup.

Nasledujúce kroky popisujú vytvorenie vlastného priestoru podľa [13] využívaného v CSU systéme.

1. *Centrovanie obrazov* – Každý trénovací obraz z databázy musí byť vycentrovaný. Odčítanie stredového obrazu od každého z tréovacích obrazov vycentruje tréovacie obrazy podľa

$$\bar{x}^i = x^i \quad \text{kde} \quad m = \frac{1}{P} \sum_{i=1}^P x^i \quad (4.1)$$

Stredový obraz je stĺpcový vektor rovnako ako každá položka je stredom všetkých prislúchajúcich pixelov tréovacích obrazov.

2. *Vytvorenie datovej matice* – Keď sú tréovacie obrazy vycentrované, tak ich spojíme do matice o veľkosti  $N \times P$ , kde  $P$  je počet tréovacích obrazov a každý stĺpec je jeden obraz.
3. *Vytvorenie kovariančnej matice* – Dátová matica je vynásobená transponovanou maticou

$$\Omega = \overline{X^T X} \quad (4.2)$$

4. *Výpočet vlastných hodnôt a vlasných vektorov* – Vlastné hodnoty a korešpondujúce vlastné vektory sú spočítané pre kovariančnú maticu podľa 2.3.
5. *Výpočet vlastných vektorov z  $\overline{X X^T}$*  – Vynásobi sa dátová matica vlastnými vektormi

$$\hat{V} = \overline{X T'} \quad (4.3)$$

Vydedia sa vlastné vektory normovanými vektormi

$$v_i = \frac{\hat{v}_i}{\|\hat{v}_i\|} \quad (4.4)$$

6. *Zoradenie vlastných vektorov* – Vlastné vektory sa zoradia podľa odpovedajúcich vlastných hodnôt zostupne, pričom sa ponechávajú iba také, ktoré majú nenulové vlastné hodnoty. Matica vlastných vektorov je vlastným priestorom, kde každý stĺpec je vlastným vektorom.

Pri trénovaní databázových obrazov je možné nastaviť parametre `dropNVectors` a `cutOff`, ktoré odstránia zo začiatku a z konca vlastné vektory, ktoré nesú najmenšiu relevantnú informáciu. Podľa výsledkov z [7] som pri všetkých testoch používal nastavenie `dropNVectors = 2` a `cutOff = 40%`

#### 4.4.3 Výpočet vzdialeností – `csuSubspaceProject`

Program najskôr premietne každý centrovaný tréningový obraz do vlastného priestoru. Na to musí vypočítať skalárny súčin obrazu a s každým zo zoradených vlastných vektorov.

$$\tilde{x}^i = V^T \bar{i}^i \quad (4.5)$$

Hodnota získaná skalárnym súčinom obrazu a prvého vlastného vektora bude prvá vlastná hodnota v novom vektore. Nový vektor premietnutých obrazov bude obsahovať toľko hodnôt, koľko je vlastných vektorov. Každý testovací obraz je najskôr stredovo centrovaný odpočítaním stredového obrazu a potom premietnutý do rovnakého vlastného priestoru  $V$ . Premietnutý testovací obraz je porovnávaný s každým premietnutým tréningovým obrazom a ten, ktorý je nájdený ako najbližší je braný ako odpovedajúci k testovaciemu. Podobnosť obrazov môže byť určovaná rôznymi metrikami. [13]

## Kapitola 5

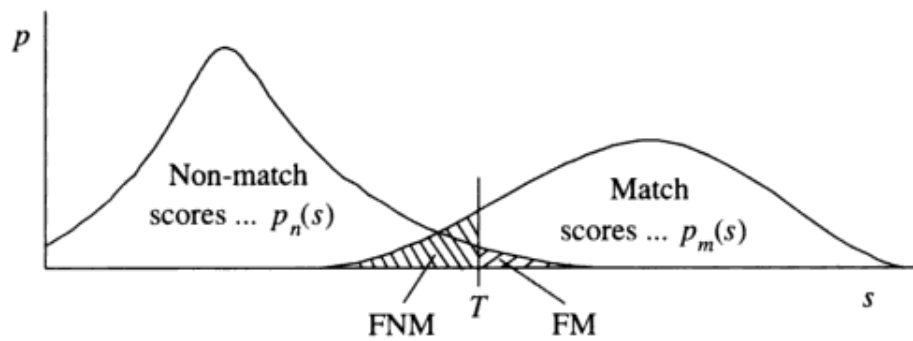
# Testovanie

Pri rozpoznávaní tváre sa porovnávajú 2 obrazy. Jeden je obrazom tváre osoby, ktorá sa identifikuje, označme si ho ako  $\mathcal{B}$  a druhý obraz je obraz tváre z databázy, označme si ho ako  $\mathcal{B}'$ . Pri porovnaní týchto dvoch obrazov vzniká problém, pretože úplnú zhodu dostaneme, iba ak by sa  $\mathcal{B}$  a  $\mathcal{B}'$  rovnali, čo v praxi je takmer nemožné, vzhľadom na to, že človek sa postupom času mení – starnutie, odraz nálady na výraze tváre, ale tiež svetelné podmienky a natočenie hlavy spôsobí, že medzi obrazmi je nejaká odlišnosť. To znamená, že o obrazoch  $\mathcal{B}$  a  $\mathcal{B}'$  môžeme povedať, že sa podobajú. Práve hodnota podobnosti určuje, či vyšetrovanú osobu označíme za referenčnú osobu, alebo nie.[3]

Aby bolo možné jednotlivé metódy porovnávať, je potrebné mať ukazatele, pomocou ktorých sa dá určiť úspešnosť rozpoznávania, chyba s akou rozpozná nesprávne a podobne. Medzi zaužívané ukazatele podľa [3] patrí:

- *False Match (FM)* – Nesprávne rozpoznanie: Ak máme 2 obrazy a biometrické rozpoznanie označí, že patria rovnakej osobe, zatiaľ čo v skutočnosti sú to obrazy rôznych osôb; frekvencia, s akou toto nastáva sa nazýva False Match Rate – stupeň nesprávneho rozpoznanie.
- *False Non-Match (FNM)* – Nesprávne nerozpoznanie: Ak máme 2 obrazy a biometrické rozpoznanie označí, že nepatria rovnakej osobe, zatiaľ čo v skutočnosti sú to obrazy rovnakých osôb; frekvencia, s akou toto nastáva sa nazýva False Non-Match Rate – stupeň nesprávneho nerozpoznanie.
- *Correct Match (CM)* – Správne rozpoznanie: Ak máme 2 obrazy a biometrické rozpoznanie označí, že patria rovnakej osobe a v skutočnosti sú to obrazy rovnakých osôb; frekvencia, s akou toto nastáva sa nazýva Correct Match Rate – stupeň správneho rozpoznanie.
- *Correct Non-Match (CNM)* – Správne nerozpoznanie: Ak máme 2 obrazy a biometrické rozpoznanie označí, že nepatria rovnakej osobe a v skutočnosti sú to obrazy rôznych osôb; frekvencia, s akou toto nastáva sa nazýva False Non-Match Rate – stupeň správneho nerozpoznanie.

Na obrázku 5.1 je vidieť, ako môže vyzeráť graf hodnôt FM a FNM. Je vidieť, že sa časť oboch kriviek prekrýva, čo znamená, že nie je možné nastaviť hranicu rozhodujúcu o rozpoznaní a nerozpoznaní tak, aby bola hodnota FM a FNM rovná nule. Znižovaním hranice budeme mať síce nízku úroveň nesprávnych nerozpoznaní, ale zvýši sa počet nesprávnych rozpoznaní. Pre rôzne aplikácie (viď kapitola 1) je žiadúca rôzna úroveň hodnôt



Obrázek 5.1: Graf hodnôt False Match a False Non-Match.[3]

FM a FNM. Napríklad pre systém verifikujúci osobu vyžaduje z pohľadu bezpečnosti, aby bola úroveň False Match Rate čo najnižšia. Opak by mohol byť bezpečnostným problémom. Naopak, pre dohľadový systém je dôležitejšie, aby bola nastavená taka hranica, pri ktorej bude úroveň False Non-Match čo najvyššia. Zvýšená hodnota False Match môže spôsobiť falošné poplachy.

## 5.1 Testovacie dáta

Na testy som zozbieral vlastné dáta, pričom dobrovoľníci boli odfotení spredu pri rôznych variáciách výrazu tváre (príklad na obrázku 5.2). Rozmery všetkých fotiek boli upravené na šírku 128 pixelov a výšku 192 pixelov.

Celkovo bolo zozbieraných 409 fotiek od 56 osôb. Z toho je 46 mužov a 10 žien. Zo všetkých osôb má 14 okuliare a počet fotiek s okuliarmi je 70. Zo všetkých osôb má 53 aspoň 4 fotky a 47 osôb aspoň 5 fotiek, pričom sa nezaratávajú fotky s okuliarmi.



Obrázek 5.2: Ukážka použitých dát s rôznymi variáciami tváre

Všetky testovacie obrázky je možné nájsť v adresári *data*. Každá osoba má samostatný adresár, ktorého názov pozostáva zo štvormiestneho identifikačného čísla. V adresári sa nachádza aj súbor *all\_coords.txt*, čo je inak pomenovaný súbor *eye\_coords.txt*, o ktorom som písal v kapitole 4.2. V ňom sú umiestnené názvy súborov a súradnice očí pre všetky obrázky.

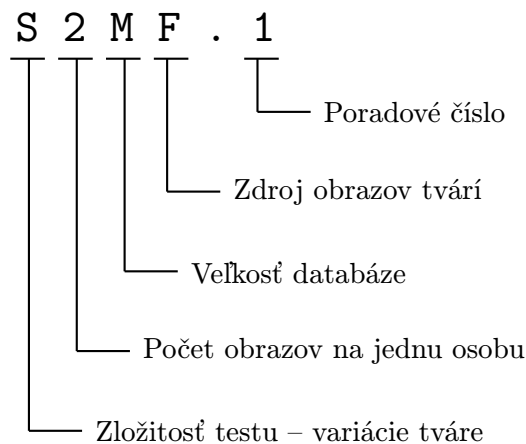
### 5.1.1 Systém označovania testov

V tejto časti popíšem, akým spôsobom sú označované jednotlivé testy. Na tie budem odkazovať v kapitole 6 a podľa označenia je možné určiť informácie o veľkosti databáze, variáciách

tváre, počtu obrazov na osobu a zdroj testovacích obrázkov.

Na obrázku 5.3 je ukážka a vysvetlenie, čo znamená ktorá pozícia v názve testu. Na jednotlivých pozíciách môžu byť tieto hodnoty:

- *Zložitosť testu* – určuje, aké rôzne variácie môže tvár nadobúdať. Možné hodnoty sú **S** – **simple** pre tváre s malou úrovňou variácií; **M** – **medium** – rôzne variácie výrazu tváre, zmena účesu, doplnkov (napríklad náušnice), pridané okuliare; **D** – **difficult** – natočenie tváre. To bolo testované iba s metódou EBGm.
- *Počet obrazov na jednu osobu* – môže nadobúdať hodnoty od nula po štyri, pričom nula znamená, že každá osoba má rôzny počet odpovedajúcich obrazov. Pre jedna až štyri platí, že číslo vyjadruje počet obrazov pre každú osobu.
- *Veľkosť databáze* – určuje, koľko osôb je v databáze. Možné hodnoty sú **T** – **tiny** približne 5 osôb; **S** – **small** – 20 osôb; **M** – **medium** – 35 osôb; **L** – **large** – 50 a viac osôb.
- *Zdroj obrazov tváří* – určuje typ testovacích dát a ich zdroj. Možné hodnoty sú **F** – **feret** pre feret databázu a vlastné obrazové súbory s definovanými koordinátami očí; **I** – **images** – obrázky a fotky aj s viac osobami; **C** – **camera** – pre dáta z web kamery; **V** – **video** – dáta sa získavajú z video súboru uloženom na disku
- *Poradové číslo* – rozlíšenie v prípade rovnako označených testov.



Obrázek 5.3: Vysvetlenie, čo znamenajú jednotlivé časti v názve testu.

## 5.2 Metodika testovania

Testovaním som chcel zistiť, ako sú metódy PCA a LDA presné v porovnaní medzi sebou a ako na ich presnosť vplyva veľkosť databázy a počet obrazov v databáze na osobu. Pre každý prípad som zostavil sériu testov, ktoré v tejto časti popíšem, uvediem a popíšem zoznam testov, ktoré som na daný prípad použil a v kapitole 6 uvediem získané hodnoty a

výsledky. Pri tvorbe testov som vybral 40 osôb, ktoré mali aspoň 5 obrazov. Všetky testy používajú tie isté osoby, aby boli maximálne minimalizované dopady spôsobené variáciou dát.

Testy boli vykonávané s minimálne dvomi obrazmi na osobu kvôli metóde LDA, ktorá potrebuje z princípu svojho fungovania viac obrazov jednej osoby.

Všetky testy boli vykonané s nastaveniami podľa tabuľky 5.1. Význam parametrov a ich hodnôt bol vysvetlený v kapitole 4.4.2 a 4.3.4.

Parameter	Metóda	
	PCA	LDA
dropNVector	2	2
cutOff	40%	40%
threshold	-0.9	2.0e+003

Tabuľka 5.1: Hodnoty parametrov, s ktorými boli vykonané testy

### 5.2.1 Referenčné testy

Na ziačiatok som spravil referenčné testy. Výsledky boli potom započítavané aj v každom bloku testov.

Parametre testov:

- *Veľkosť databázy* – 20 osôb so zachovaným pomerom muži-ženy podľa pomeru z celkového počtu osôb.
- *Počet obrazov na osobu* – 2, pričom boli vyberané tak, aby rozdiely výrazu tváre neboli príliš veľké. Každý test potom mal takú databázu, ktorá neobsahovala osoby z databázy druhého testu.
- *Počet testovacích obrazov* – 40 a polovica osôb mala odpovedajúci záznam v databáze.
- *Počet testov* – 2, pričom obidva testy mali rovnaké testovacie dáta, menil sa obsah databázy.
- *Výsledne určenie presnosti* – spriemerovaním úspešnosti zo všetkých zrealizovaných testov.

### 5.2.2 Vplyv veľkosti databáze

V tejto sade testov bolo cieľom zistiť, či má počet osôb v databáze vplyv na úspešnosť rozpoznávania. Testované boli veľkosti so 6, 20 (referenčný test) a 30 osobami. Parametre testov:

- *Veľkosť databázy* – podľa konkrétneho testu.
- *Počet obrazov na osobu* – 2, rovnako ako v referenčných testoch.
- *Počet testovacích obrazov* – podľa veľkosti databázy. Pre 6 a 20 osôb bol počet testovacích obrazov dvojnásobný oproti počtu osôb v databáze. V prípade testu s 30 osobami bolo testovacích obrazov 20, pričom 10 z databázy.

- *Počet testov* – 6 pre najmenšiu veľkosť, 2 pre referenčnú veľkosť a 1 v prípade najväčšej databázy.
- *Výsledne určenie presnosti* – spriemerovaním úspešnosti zo všetkých zrealizovaných testov danej veľkosti.

### **5.2.3 Vplyv počtu obrazov na osobu**

V tejto sade testov ostávali testovacie obrazy rovnaké ako v prvom, referenčnom bloku testov. Menil sa počet obrazov na osobu, pričom ich počet bol v jednotlivých testoch 2, 3 a 4. Navyše som PCA testoval aj len s jedným obrazom na osobu.



## Kapitola 6

# Výsledky z testovania

V tejto kapitole uvediem výsledky získané z jednotlivých testov. Metodika testovania, obsah testov a podmienky, za ktorých som testoval boli popísané v kapitole 5.2.

### 6.1 Referenčné testy

V tabuľke 6.1 sú uvedené výsledky po testovaní referenčných dát. Výsledné hodnoty CMR a FNMR sú na úrovni, kedy je možné predpokladať určitý posun jedným alebo druhým smerom pri zmene niektorých z parametrov.

	PCA				LDA			
	CMR	CNMR	FMR	FrnNMR	CMR	CNMR	FMR	FNMR
<b>M2SF.1</b>	85	95	2,5	15	80	100	0	20
<b>M2SF.2</b>	90	100	0	0	90	100	0	10
<i>priemer [%]</i>	87,5	97,5	1,25	7,5	85	100	0	15

Tabuľka 6.1: Výsledné hodnoty po referenčnom testovaní

### 6.2 Vplyv veľkosti databáze

Výsledné hodnoty z testovania vplyvu veľkosti databázy sú v tabuľke 6.2. Vplyv na Correct Match Rate je v grafe 6.1, na False Match Rate v 6.2 a na False Non-Mactch Rate v 6.3.

Ako je z grafov vidieť, metódy reagujú na väčšiu databázu negatívnym spôsobom, keď sa zhoršila hodnota CMR aj FNMR. Za zlepšenie hodnoty FMR môže fakt, že celkovo klesla schopnosť systému rozpoznávať tváre. Ďalšími testami som zistil, že pri zväčšení databázy je vhodné, aby sa znížila hodnota prahu, pri ktorom sa dve tváre považujú za rozpoznané. Keď som prah znížil na hodnotu -0.85 pre PCA, zlepšili sa všetky hodnoty rozpoznania pre databázy s 20 a 30 osobami a to až na úroveň CMR = 97,5 %, respektíve 90 % a úroveň FNMR = 2,5 % respektíve 10 % bez zvýšenia hodnôt FRM. Naopak, v prípade nižšieho prahu pre menšiu databázu bolo zvýšenie hodnôt FRM výrazné.

Z výsledkov testov vyplýva, že rozpoznávanie je závislé aj na veľkosti databáze, pričom so zväčšujúcou sa databázou dochádza k zhoršeniu rozpoznávania. Toto je však možné kompenzovať nastavením prahu rozpoznávania.

	PCA				LDA			
	CMR	CNMR	FMR	FNMR	CMR	CNMR	FMR	FNMR
<b>M2TF.1</b>	100	50	25	0	66,67	50	25	33,33
<b>M2TF.2</b>	100	50	25	0	100	66,67	16,67	0
<b>M2TF.3</b>	100	50	25	0	83,33	66,67	16,67	16,67
<b>M2TF.4</b>	83,33	50	25	16,67	100	50	50	0
<b>M2TF.5</b>	100	83,33	8,33	0	100	33,33	33,33	0
<b>M2TF.6</b>	100	66,67	16,67	0	83,33	100	0	16,67
<i>priemer [%]</i>	97,22	58,33	20,83	2,78	88,89	61,11	23,61	11,11

<b>M2SF.1</b>	85	95	2,5	15	80	100	0	20
<b>M2SF.2</b>	90	100	0	0	90	100	0	10
<i>priemer [%]</i>	87,5	97,5	1,25	7,5	85	100	0	15

<b>M2MF.1</b>	60	100	0	40	80	100	0	20
<i>priemer [%]</i>	60	100	0	40	80	100	0	20

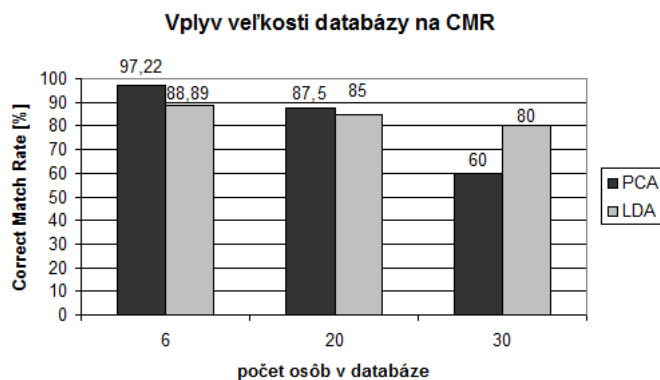
Tabuľka 6.2: Výsledné hodnoty po testovaní vplyvu veľkosti databázy

### 6.3 Vplyv počtu obrazov na osobu

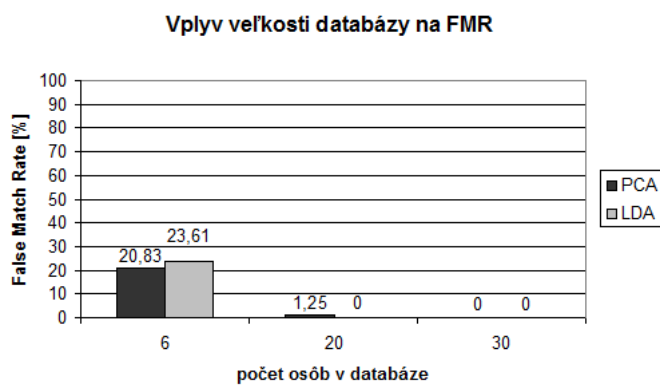
Výsledné hodnoty z testovania vplyvu počtu obrazov na osobu sú v tabuľke 6.3. Vplyv na Correct Match Rate je v grafe 6.4, na False Match Rate v 6.5 a na False Non-Mactch Rate v 6.6.

Ako je vidieť na grafoch, počet obrazov na osobu má veľký vplyv na rozpoznávanie pri použití oboch metód. Čím viac obrazov jednej osoby sa použije v prípade PCA, tým sú výsledky horšie. Naopak, LDA na túto zmenu reaguje veľmi priaznivo a všetky ukazatele vykazujú zlepšenie. Treba však poznamenať, že princíp LDA je postavený práve na tom, že znižuje rozdiely medzi rovnakými obrazmi (obrazmi rovnakej osoby) a medzi obrazmi rôznych osôb, čím má dosahovať lepšie výsledky. Pre PCA je viac podobných obrazov rušivých.

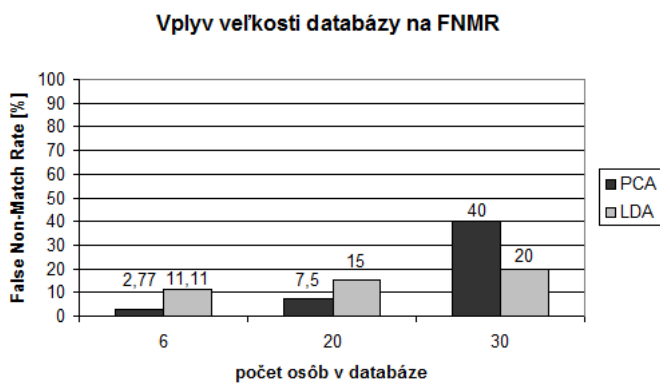
Z výsledkov testov teda vyplýva, že počet obrazov na osobu má vplyv na rozpoznávanie. V prípade PCA s väčším počtom obrazov negatívny, v prípade LDA, podľa očakávania, pozitívny.



Obrázek 6.1: Vplyv veľkosti databázy na hodnotu Correct Match Rate



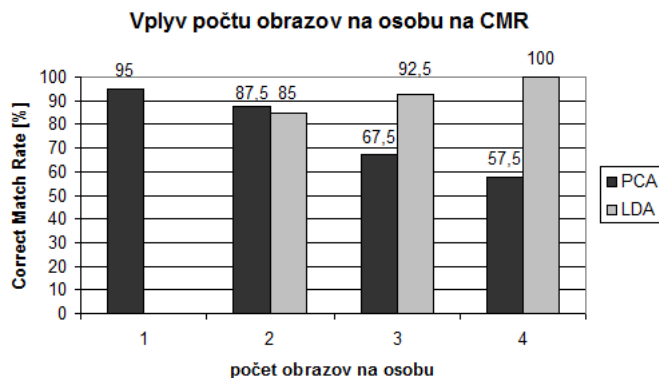
Obrázek 6.2: Vplyv veľkosti databázy na hodnotu False Match Rate



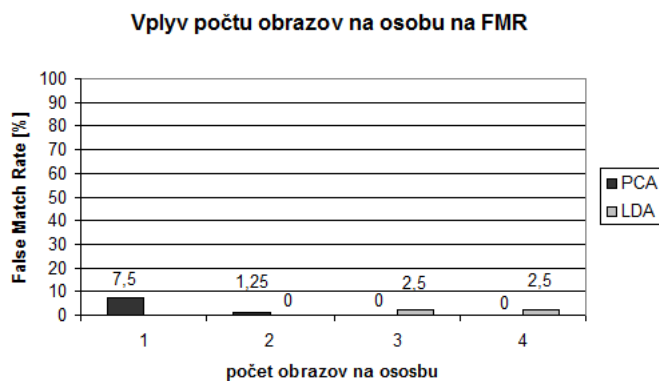
Obrázek 6.3: Vplyv veľkosti databázy na hodnotu False Non-Match Rate

	PCA				LDA			
	CMR	CNMR	FMR	FNMR	CMR	CNMR	FMR	FNMR
<b>M1SF.1</b>	100	90	5	0				
<b>M1SF.2</b>	90	85	10	5				
<i>priemer [%]</i>	95	87,5	7,5	2,5				
<b>M2SF.1</b>	85	95	2,5	15	80	100	0	20
<b>M2SF.2</b>	90	100	0	0	90	100	0	10
<i>priemer [%]</i>	87,5	97,5	1,25	7,5	85	100	0	15
<b>M3SF.1</b>	50	100	0	50	95	90	5	5
<b>M3SF.2</b>	85	100	0	15	90	100	0	10
<i>priemer [%]</i>	67,5	100	0	32,5	92,5	95	2,5	7,5
<b>M4SF.1</b>	40	100	0	60	100	95	2,5	0
<b>M4SF.2</b>	55	100	0	45	100	95	2,5	0
<i>priemer [%]</i>	47,5	100	0	52,5	100	95	2,5	0

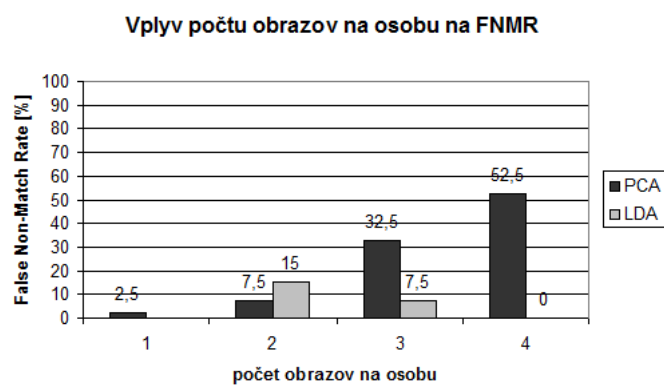
Tabulka 6.3: Výsledné hodnoty po testování vplyvu počtu obrazov na osobu



Obrázek 6.4: Vplyv počtu obrazov na osobu na hodnotu Correct Match Rate



Obrázek 6.5: Vplyv počtu obrazov na osobu na hodnotu False Match Rate



Obrázek 6.6: Vplyv počtu obrazov na osobu na hodnotu False Non-Match Rate

# Kapitola 7

## Záver

Cieľom mojej práce bolo naštudovať niektoré z používaných metód na popis a rozpoznávanie tváre. Vybral a popísal som metódy Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA) a metódu Elastic Bunch Graph Matching (EBGM).

Ďalej bolo treba vytvoriť testovaciu sadu dát, na ktorých by som metódy otestoval. Tie som vytvoril z frontálnych fotiek ľudí. Z fotiek som vyextrahoval pozíciu očí. Z upravených fotiek som vytvoril niekoľko testovacích sád.

Požadovanú experimentálnu aplikáciu som realizoval ako demo aplikáciu dohľadového systému. Aplikácia dokáže spracovávať obraz z rôznych vstupov. Vo vstupnom obraze detekuje tvár, ktorú sa pokúsi rozpoznať. Ďalej som implementoval spracovanie anotovaných dát. Aplikácia potom dokáže vypočítať štatistické údaje o úspešnosti rozpoznávania.

Aplikácia však nedokáže spracovávať obraz v reálnom čase. Je to spôsobené najmä tým, že na vykonanie niektorých operácií sa používa ďalší program. Použitie funkcií by značne zrýchlilo spracovávanie. Tiež by prospelo, keby boli dáta, ktoré sa budú ďalej spracovávať, ukladané do pamäti a nie na disk.

V rámci experimentov som testoval presnosť metód a aký vplyv na ich presnosť má veľkosť databázy a počet obrazov na osobu. Z testov som zistil, že obidve metódy dosahujú, pri ponechaní rovnakých nastavení, horšie výsledky pre väčšie databázy. Zlepšenie nastalo pri znížení prahu, ktorý určuje rozpoznanie. Toto však neplatilo pre malé databázy, u ktorých malý prah viedol k horším výsledkom.

Ďalej som testami potvrdil, že metóda LDA je podľa predpokladov so zväčšujúcim sa počtom obrazov na osobu presnejšia, zatiaľ čo metóda PCA sa stáva nepresnejšou.

O metóde LDA sa dá povedať, že dosahuje lepšie výsledky než PCA, ale nevýhodou je, že aby ju bolo možné použiť, potrebuje viac obrazov na jednu osobu.

# Literatura

- [1] FACE RECOGNITION HOMEPAGE. online.  
URL <http://www.face-rec.org/algorithms/>
- [2] Beveridge, R.; Bolme, D.; Teixeira, M.; aj.: The CSU Face Identification Evaluation System User's Guide. *Computer Science Department Colorado State University*, 2003, online.  
URL <http://www.cs.colostate.edu/evalfacerec/algorithms/version5/faceIdUsersGuide.pdf>
- [3] Bolle, R. M.; Connell, J. H.; Pankanti, S.; aj.: *Guide to Biometrics*. Springer, 2008, ISBN 0-387-40089-3.
- [4] Bradski, G.; Kaehler, A.: *Learning OpenCV*, ročník 1. O'Reilly Media, Inc., 2008, ISBN 978-0-596-51613-0, 555 s.
- [5] <http://www.biometrics.gov/>: Face Recognition. 2006, online.  
URL <http://www.biometrics.gov/>
- [6] Kim, T.-K.; Kim, H.; Hwang, W.; aj.: Component-based LDA Face Descriptor for Image Retrieval. 2005.
- [7] MAZANEC, J.: *Rozpoznávanie tváří v biometrii založené na metodách PCA a SVM*. Diplomová práce, Slovenská technická univerzita v Bratislave, 2008.
- [8] RAK, R.: *Biometrie a identita člověka ve forenzních a komerčních aplikacích*, ročník 1. Grada, 2008, ISBN 978-80-247-2365-5, 631 s.
- [9] RAYAT, N.; WALIA, E.: Face Recognition Using Improved Fast PCA Algorithm. In *2008 Congress on Image and Signal Processing*, IEEEExplore, 2008.
- [10] WAYMEN, J.; JAIN, A.; MALTONI, D.; aj.: *Biometric systems: technology, design and performance*. Springer, 2005, ISBN 1-85233-596-3, 370 s.
- [11] Wiskott, L.; Fellous, J.-M.; Krüger, N.; aj.: Face Recognition by Elastic Bunch Graph Matching. In *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, ročník 19.
- [12] WISKOTT, L.; FELLOUS, J.-M.; KRÜGER, N.; aj.: *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, kapitola Face Recognition by Elastic Bunch Graph. L. C. JAIN and U. HALICI and I. HAYASHI and S. B. LEE and S. TSUTSUI, 1999, ISBN 0-8493-2055-0.

- [13] Yambor, W. S.: ANALYSIS OF PCA-BASED AND FISHER DISCRIMINANT-BASED IMAGE RECOGNITION ALGORITHMS. Technická zpráva, Computer Science Department Colorado State University, 2000.  
URL [http://www.face-rec.org/interesting-papers/General/tr\\_00-103.pdf](http://www.face-rec.org/interesting-papers/General/tr_00-103.pdf)
- [14] Zhao, W.; Chellappa, R.; Krishnaswamy, A.: Discriminant Analysis of Principal Components for Face Recognition. Online.  
URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=670971>



## Příloha A

# Stromová štruktúra testovacích dát

- **data**
  - **database** Obsahuje databázové dáta
  - + **input** Obrazy osôb v databáze rôznych formátov (jpg, png, pgm, ppm)
  - + **source** Konvertované obrazy osôb z input vo formáte .pgm
  - **database.srt** Zoznam obrazov s rozdelením podľa príslušnosti k osobe
  - **eye\_coords.txt** Koordináty očí pre databázové súbory
  - **<nazov\_metody>.train** Vygenerovaný tréningový súbor
  - **normalized.txt** Informačný súbor, že databáza je už normalizovaná
  - **output** Obsahuje súbory s výstupmi
    - **<nazov\_metody> <pouzita\_metrika>.html** Výstupný súbor
  - **probe** Obsahuje dáta k testovaniu
    - **source** Uložené všetky spracovávané obrazy
      - **<nazov\_aktualne\_spracovavaneho\_obrazu>**
        - **distances** Obsahuje vzdialenostné súbory
          - **<nazov\_pouzitej\_metriky>**
          - **<databazove\_subory>.sfi**
          - **<nazov\_aktualne\_spracovavaneho\_obrazu>.sfi** Vzdialenostný súbor
        - **source** Obsahuje aktuálne spracovávaný súbor – len obraz tváre
          - **<nazov\_aktualne\_spracovavaneho\_suboru>.pgm** Obraz tváre
          - **<nazov\_aktualne\_spracovavaneho\_obrazu>.png** Obraz pre výstup
      - **<nazov\_suboru>.<jpg|png|pgm|ppm>** Testovací súbor pôvodný
      - **list.txt** Súbor so zoznamom testovaných súborov
    - **wd** Pracovný adresár
      - + **pgm** obsahuje súbory normalizovanými obrazmi tváří
      - + **sfi** obsahuje normalizované obrazy tváří vo variante s reálnymi číslami

### legenda

- **názov** – Adresár. Musí existovať pred spustením aplikácie.
- **názov** – Adresár generovaný aplikáciou.
- **názov** – Súbor. Musí existovať pred spustením aplikácie.
- **názov** – Súbor. Generovaný aplikáciou

# Příloha B

## Manual

Aplikácia je konzolovou aplikáciou vyvíjanou pod operačným systémom Microsoft Windows. Testovaná bola na verzii Windows 7. Aplikácia je písaná v jazyku C/C++ vo vývojovom prostredí Microsoft Visual Studio 2008 a využíva knižnicu OpenCV 2.0, ktorá musí byť pri preklade aplikácie k dispozícii.

Aplikácia musí byť spustená z hlavného adresára aplikácie kvôli ďalším závislostiam.

### Parametre programu

- inputDir** *cesta\_k\_adresaru* Povinný parameter. Cesta k adresáru s testovacími dátami.
- use** <hodnota> Ktorá rozpoznávací metóda sa použije. Možné hodnoty PCA, LDA; predvolená hodnota PCA.
- media** <hodnota> Aký sa použije typ vstupného média, z ktorého sa získajú testovacie obrázky. Možné hodnoty image, video, feret, camera, predvolená hodnota camera.
- print** <hodnota> Úroveň vypisovaných správ. Hodnota sa zadáva ako logický súčet hodnôt úrovní, ktoré chceme vypisovať. Možné hodnoty 0 – chybové výpisy, 1 – upozornenia, 2 – informačné výpisy, 4 – informácie o priebehu, 8 – debugovacie výpisy, 16 – podrobné debugovacie výpisy. Predvolená hodnota 1|2|4|8.
- threshold** <hodnota> Hranica, pre ktorú sa berie vzdialenosť medzi dvoma obrazmi ako rozpoznanie. Predvolená hodnota je pre PCA -0.85 a pre LDA 2.2e+001.

### Príklad použitia

```
bin\program.exe --inputDir data/TEST1 --use PCA --media feret --print 3  
--threshold -0.95
```

# Příloha C

## Obsah DVD

- + **csu** Obsahuje pôvodný projekt The CSU Face Identification Evaluation System
- **data** Testovacie dáta
  - + **all** Všetky zozbierané testovacie dáta
  - + **tests** Jednotlivé testy, s ktorými sa testovalo
- **document** Obsahuje technickú správu
  - **src** L<sup>A</sup>T<sub>E</sub>Xovské zdrojové kódy
  - **dokumentace.pdf** Technická správa
- **program** Súbory programu
  - + **bin** Obsahuje spustiteľný program
    - **src** Obsahuje všetky súbory potrebné na preklad
      - + **data** Predvolený adresár pre testovacie dáta
      - + **haarcascades** Súbory potrebné pri detekcii
      - + **opencvx** Zdrojové súbory OpenCV
      - + **src** Zdrojové súbory CSU systému
        - **faceRec.cpp** Súbor s definíciou triedy **faceRec**
        - **faceRec.h** Súbor s deklaráciou triedy **faceRec**
        - **getopt.c** Kvôli kompatibilite zdrojových kódov na windows
        - **getopt.h** Kvôli kompatibilite zdrojových kódov na windows
        - **program.cpp** Súbor so zdrojovým kódom aplikácie
        - **program.sln** Súbor pre Visual Studio
        - **ReadMe.txt** Súbor so základnými informáciami o aplikácii
        - **stdafx.cpp** Kvôli kompatibilite zdrojových kódov na windows
        - **stdafx.h** Kvôli kompatibilite zdrojových kódov na windows
        - **strcasecmp.c** Kvôli kompatibilite zdrojových kódov na windows
        - **strcasecmp.h** Kvôli kompatibilite zdrojových kódov na windows
        - **targetver.h** kvôli kompatibilite zdrojových kódov na windows
        - **unistd.h** Kvôli kompatibilite zdrojových kódov na windows
    - **plagat.png** Plagát prezentujúci moju prácu