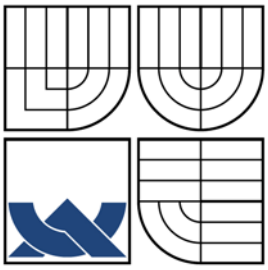


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

METODY A ALGORITMY PRO ROZPOZNÁVÁNÍ OBLIČEJŮ

METHODS AND ALGORITHMS FOR FACE RECOGNITION

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE — JIŘÍ SOUKUP
AUTHOR

VEDOUCÍ PRÁCE — doc. RNDr. Ing. JIŘÍ ŠŤASTNÝ, CSc.
SUPERVISOR

BRNO 2008

Vložené zadání

Cíle, kterých má být dosaženo:

Cílem projektu je provedení analýzy současného stavu problematiky identifikace obličejů s následnou implementací vhodných metod a algoritmů pro rozpoznávání obličejů z dané databáze snímků. Analýza bude zahrnovat mimo jiné také statistické metody a metodu shlukových grafů. Výsledky testování jednotlivých metod budou vhodným způsobem kvantifikovány.

Charakteristika problematiky úkolu:

Projekt zahrnuje analýzu klasifikačních algoritmů v oblasti počítačového rozpoznávání obličejů včetně výběru a implementace vhodného řešení pro danou databázi snímků. Vytvořený program bude založen na moderních prostředcích objektové technologie a bude umožňovat automatické vyhodnocení snímků včetně kvantifikovaného zhodnocení výsledků pro danou databázi.

ABSTRAKT

Práce se zabývá popisem základních metod pro problematiku rozpoznávání obličejů. Mezi popisované metody patří PCA, LDA, ICA, trasová transformace, technika shlukových grafů, genetické algoritmy a neuronové sítě. V praktické části se práce zabývá implementací algoritmu PCA a jeho kombinaci s neuronovou sítí s radiální bází a genetického algoritmu. Neuronová síť s radiální bází je použita v roli klasifikátoru a genetický algoritmus je v jednom případě použit pro trénink neuronové sítě a v druhém případě pro výběr vlastních vektorů vytvořených metodou PCA. Tato metoda, spojení PCA + GA, zvané EPCA, dosahuje na testované ORL databázi nejlepších výsledků v rámci porovnávaných algoritmů.

ABSTRACT

This work is describing basic methods of face recognition. The methods PCA, LDA, ICA, trace tranfsorm, elastic bunch graph map, genetic algorithm and neural network are described. In practical part, the PCA, PCA + RBF neural network and genetic algorithms are implemented. The RBF neural network is used in the way of clasificator and genetic algorithm is used for RBF NN training in one case and for selecting eigenvectors from PCA method in the other case. This method, PCA + GA, called EPCA, outperform other methods tested in this work on the ORL testing database.

KLÍČOVÁ SLOVA

PCA, ICA, LDA, vlastní vektory, trasová transformace, genetické algoritmy, neuronové sítě, RBF, rozpoznání obličeje

KEYWORDS

PCA, ICA, LDA, eigen vectors, trace transformation, genetic algorithms, neural network, RBF, face recognition

Bibliografická citace

SOUKUP, J. *Metody a algoritmy pro rozpoznávání obličejů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2008. 59 s. Vedoucí diplomové práce doc. RNDr. Ing. Jiří Šťastný, CSc.

Čestné prohlášení

Čestně prohlašuji, že jsem diplomovou práci na téma Metody a algoritmy pro rozpoznávání obličejů vypracoval samostatně pod vedením svého vedoucího diplomové práce pana doc. RNDr. Ing. Jiřího Šťastného, CSc. Všechnu použitou odbornou literaturu jsem citoval v seznamu literatury.

V Brně dne 17. 10. 2008

.....

Jiří Soukup

Poděkování

Na prvním místě bych chtěl poděkovat svému vedoucímu diplomové práce doc. RNDr. Ing. Jiřímu Šťastnému CSc. za přidělení této diplomové práce, protože se mi naskytla jedinečná možnost nahlédnout do zajímavého světa zpracování obrazu, který je dost odlišný od mé dosavadní práce na poli vývoje informačních systémů. Dále bych mu rád poděkoval za přívětivý přístup, dobré připomínky a nasměrování správným směrem.
Na dalším místě děkuji Ing. Petru Chmelaři za zasvěcení do technické části problematiky a dodání optimismu pro vypracování této práce. Dále Ing. Lukáši Strykovi za ochotnou pomoc při shánění odborné literatury a v neposlední řadě svému bratru Ing. Lukáši Soukupovi za kritiku, která mě hnala k usilovnější práci, a také za poskytnutí výpočetního výkonu pro urychlení náročných testů.
Také děkuji svým rodičům za podporu během celého studia na vysoké škole a své přítelkyni za péči během posledních měsíců.

1 Úvod

Už tisíce let se člověk zdokonaluje v rozpoznávání obličejů. Odpradávna potřeboval rozlišovat kdo je přítel a kdo nepřítel, a právě tato rozpoznávací schopnost je potřebná i v současnosti. Je dobré si pamatovat a rozlišovat důležité tváře. Asi by nebylo dobré poplést si pana ředitele s vrátným a tak podobně. Stejně tomu bylo i v minulosti, protože neexistovali žádné občanské průkazy a bylo třeba rozlišovat členy tlupy, kdo byl vůdce, kdo byli členové, a tak mít možnost zavčasu rozpoznat například vetřelce s nekalými úmysly. Bohužel pro techniku, ale naštěstí pro nás lidi, nefunguje rozpoznávání známých osobností pouze na obličeji, ale dokážeme lidi rozpoznat pomocí celé řady pomocných faktorů. Určitě si sami dokážete vybavit, jak jste svého známého/známou poznali už podle chůze nebo na 100 metrů podle barvy vlasů či oblečení. Lidský mozek má velkou výhodu že nedokonalosti vjemů dokáže nahradit kontextem. Je tomu stejně u zraku, sluchu a pravděpodobně i dalších lidských smyslů. U sluchu je to více než zřejmé. Pokud posloucháme rozhlas nebo něčí rozhovor a kvalita zvuku není příliš vysoká, či okolí způsobuje silné rušení, charakteristické části signálu jsou zkresleny a v ten okamžik mozek použije kontextový přístup k vyhodnocení vjemu. Pokud budeme poslouchat politickou debatu, určitě dáme při vyhodnocení slova přednost slovu „právo“ než například „prádlo“. Stejně tak pokud potkáme na ulici známého, který se podobá Karlovi nebo Romanovi a nejsme si tedy zprvu jisti, začne pracovat kontextová logika. Karel by měl být v práci, ale Roman jel na dovolenou do Egypta. Jaká je pravděpodobnost, že se Roman vrátil z Egypta a jaká je pravděpodobnost, že se Karel zrovna vydal z práce na oběd nebo si potřeboval něco zařídit. Nejzajímavější je, že tyto myšlenkové pochody vůbec nemusí probíhat uvědoměle. Je to stejné, jako když bydlíte v Olomouci, kde máte svůj okruh známých, a po příjezdu do Brna automaticky, aniž byste si to uvědomovali, přepínáte na jiný okruh známých a s tím mozek pracuje. Mozek je ale natolik propracovaný, že co nenajde rychle pomocí jednoho kontextu, pokusí se nalézt řešení pomocí dalšího. To znamená, že dokáže velice rychle a efektivně přepínat mezi kontexty, dokud zadaný problém nevyřeší.

Pro názornost si můžeme představit kontext jako obecnou rovinu. Další kontext bude další obecná rovina, která se s prvním kontextovou rovinou nemusí vůbec nikde potkat, takže ji můžeme označit za paralelní rovinu (kontext). V oblasti, kde se kontextové roviny protínají, bychom tedy měli hledat řešení. Například v průniku kontextové roviny „voda“ a „sport“ by se měli nacházet například: vodní lyžování, plavání, kanoistika, windsurfing, vodní pólo a mnoho dalších. Pokud se pří bádání po dalších sportech zamyslíte, zjistíte, že dost času zabere najít právě další kontext, který by pomohl nalézt další možné vodní sporty. Tak například kontext „loď“ pomohl nalézt během chvilky sporty typu windsurfing, kanoistiku, jízda v kajaku. Kontext „balón“ pomohl nalézt vodní pólo. Dá se z toho tedy usoudit, že čím více kontextu známe, tím lépe a rychleji dojdeme k řešení.

Tento předpoklad nám ale identifikaci obličeje poměrně stěžuje, protože nemůžeme pracovat v obecném kontextu, jako je výška postavy, váhová kategorie, pohlaví, oblečení atd. U pohlaví by mohl někdo namítat, že lze rozpoznat už z obličeje, ale ne vždy tomu tak je a mohlo by dojít pouze ke zbytečnému zmatení. Jediným vodítkem nám tedy zůstávají samotné rysy obličeje. Rysem obličeje v obecné rovině můžeme uvažovat tvar obličeje, velikost a tvar rtů, nosu, očí, obočí, už méně si všímáme např. uší, které bývají často skryty vlasy, a proto nemají příliš velkou váhu, i když v některých případech mohou být velmi charakteristickým rysem obličeje.

Samozřejmě i délka a barva vlasů hraje svou roli pro rychlé zaškatulkování posuzovaného obličeje. Bohužel pro technické zpracování se většina těchto rysů mění. Je pár rysů, které se nemění, jako je rozteč očí, velikost nosu a vzdálenost od očí, ale i tady metody identifikace obličejů mají problémy se změnou těchto vzdáleností při natočení hlavy jinam než přímo na snímací zařízení. Je to jako bychom se snažili měřit pravítkem vyfoceným pod jiným než pravým úhlem. Další problém je, že neznáme absolutní hodnoty vzdáleností, ale jenom poměr vzdáleností, takže do stejného pytle házíme všechny obličeje ať už mají rozteč očí 6cm a vzdálenost k nosu 5cm nebo rozteč očí 6,6cm a vzdálenost k nosu 5,5cm. Tyto metody jsou již předem odsouzené k nezdaru pro trošku větší databázi snímků. Lepším přístupem je snaha rozlišovat tvary. O tento přístup se pokouší například metoda shlukových grafů. Bohužel i tvary se zkreslují s natočením tváře a tak i tato metoda je odsouzena k nezdaru u snímků, které nejsou z dokonale čelního pohledu.

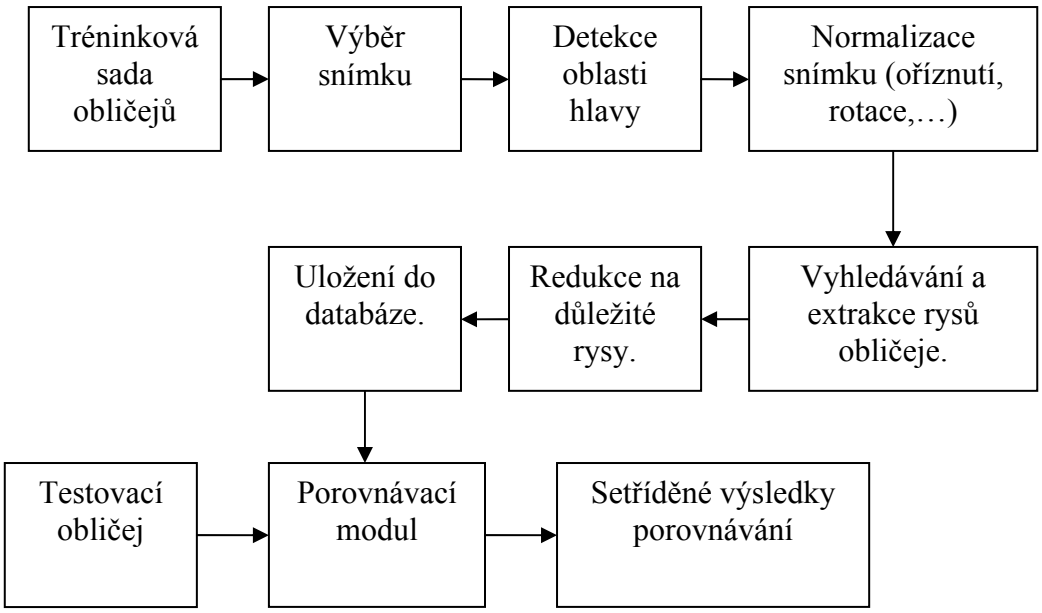
Zatím nejlepším způsobem je přenechat vyhledání význačných rysů samotnému počítači a pomocí statistických metod pak vybírat nejpodobnější tvář. Tyto metody jsou například PCA, LDA, ICA.

1.1 Cíl práce

Cílem práce je provedení analýzy současného stavu problematiky identifikace obličejů a implementování vybrané metody se snahou dosáhnout lepších výsledků. Při výběru metody k implementaci budou upřednostněny metody využívající neuronové sítě nebo genetické algoritmy. Nově implementovaný algoritmus bude vhodným způsobem porovnán se základní metodou nad stejnou databází snímků, aby se dal jednoznačně posoudit přínos či nevýhoda nové metody.

2 Proces rozpoznávání

Proces rozpoznávání má následující obecné schéma. Nejprve se ve snímku vyhledá obličej. Posléze se obličej vyřízne a normalizuje. Následně se mohou hledat a extrahovat význačné části obličeje jako jsou rty, oči, nos atd. Dále se tyto rysy musí ohodnotit a výsledky se uloží do databáze, ve které se přiřazují k určité osobě. Při rozpoznávání se totéž provede s testovanou fotkou a získané hodnoty se porovnávají s hodnotami v databázi a hledá se nejpodobnější obličej.



Obr. 1 Schéma obecného procesu identifikace obličejů.

Různé algoritmy nemusí obsahovat všechny bloky uvedeného schéma nebo mohou spojovat více bloků do jednoho.

2.1 Předzpracování obrazu

Před samotným procesem rozpoznávání se u snímků mohou provádět rozličné operace. Podle toho, jaký snímek dostaneme, volíme vhodný postup. Pokud získáme snímky obsahující šum, můžeme aplikovat konvoluční filtry na potlačení tohoto šumu, stejně tak můžeme převádět barevné snímky do odstínů šedi. V případě že nemáme snímky pouhého výřezu obličeje, musíme provést nejprve detekci obličeje a ten pokud možno transformovat do správné velikosti a úhlu natočení, tak aby oči byly v předem definované rovině. Tato problematika by si však svou rozsáhlostí zasloužila samostatnou práci, a proto se jí v této práci věnovat nebudeme.

3 Metody rozpoznávání

Přístupů a metod v problematice rozpoznávání obličejů je velmi mnoho. Mezi dvě hlavní skupiny se řadí metody strukturální a metody holistické. Strukturální metody se zaměřují na části obličeje, jako jsou oči, ústa a nos, a využívají měření antropometrických veličin s jejich následným porovnáním s databází naměřených hodnot u jiných snímků. Holistické metody naopak využívají celého obrazu chápaného jako signál a hledání charakteristických rysů v tomto signálu s následným porovnáváním. Holistické metody bývají častěji využívány, protože jejich proces může být plně automatizován. Častým zástupcem této skupiny bývá metoda PCA a kombinace metod často založené právě na této metodě.

3.1 Korelační metody

Pravděpodobně nejjednodušší klasifikační schéma je založeno na klasifikaci přes nejbližšího souseda v projekčním prostoru. Tedy, obrázek v testovací sadě je rozpoznán (klasifikován) přiřazením hodnoty nejbližšího bodu v tréninkové sadě, kde jsou vzdálenosti měřeny v projekčním prostoru. Pokud jsou všechny obrázky normalizovány na nulový průměr a jednotkový rozptyl, pak je tato procedura rovna výběru obrázku z tréninkové množiny, který nejlépe koreluje s testovaným obrázkem. Při použití normalizace jsou výsledky nezávislé na intenzitě světelného zdroje či na automatickém vyvážení jasu u snímací techniky.

Tato metoda, která bývá částečně označována jako korelační, má několik dobře známých nevýhod. První nevýhodou je, pokud snímek v učící se množině je získán za rozdílného osvětlení, pak korespondující body v projekčním prostoru nebudou blízko sebe. V důsledku toho bude zapotřebí husté vzorkovací množiny pro různé možnosti nasvícení. Druhou nevýhodou je výpočetní náročnost. Pro rozpoznání je zapotřebí provést korelaci testovacího snímku s každým snímkem v tréninkové množině. Pro pokusnou implementaci byl vyvinut i speciální VLSI hardware. Třetí zcela zřejmou nevýhodou je nutnost velkého místa pro uložení tréninkové množiny, která musí obsahovat velký počet snímků pro každou osobu. [1][2]

3.2 PCA (Principal Component Analysis)

Pro výpočetní a prostorovou nákladnost korelační metody bylo přirozenou snahou snížit rozměrnost této metody. Technikou pro snížení rozměrnosti úlohy v počítačovém vidění – zejména v identifikaci obličejů – je analýza hlavních složek.[2]

Metoda analýzy hlavních složek je odvozena z Karhunen-Loevovi transformace. Pro daný reprezentující s-rozměrný vektor pro každý obličej z množiny tréninkových obrázků se metoda analýzy hlavních složek snaží najít t-rozměrný podprostor, jehož základní vektory korespondují s maximy směrových rozdílů v originálním projekčním prostoru. Tento nový podprostor je obvykle méně rozměrný (t<<s). Pokud považujeme části obrazu jako náhodné proměnné, potom PCA základní vektor je definován jako vlastní vektory z rozptylné matice.[1][4]

PCA vybírá lineární projekci redukující rozměrnost, která maximalizuje rozptyl ze všech promítaných vzorků. Formálněji: uvažujme sadu *N* vzorových snímků $\{x_1, x_2,\ldots,x_N\}$ nesoucí hodnoty v *n*-rozměrném projekčním prostoru a předpokládejme, že každý snímek náleží do jedné z *c* tříd $\{X_1, X_2, \ldots, X_c\}$. Dále také uvažujme lineární transformační mapování originálního *n*-rozměrného projekčního prostoru do

m-rozměrného prostoru rysů, kde $m < n$. Nový vektor rysů $y_k \in \mathbf{R}^m$ je definován následující lineární transformací:

$$y_k = W^T x_k \qquad k = 1,2,\ldots,N$$

kde $W \in \mathbf{R}^{n \times m}$ je matice s ortogonálními sloupci.

Pokud je rozptylová matice $S\tau$ definována jako

$$S_\tau = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T$$

kde *n* je počet vzorových snímků a $\mu \in \mathbf{R}^n$ je průměrný snímek ze všech vzorků, pak po aplikaci lineární transformace W^T, je rozptylová matice z vektorů rysů $\{y_1,y_2,\ldots,y_N\}$ rovna $W^T S_T W$. Projekce W_{opt} u metody PCA je volena pro maximalizaci determinantu celkové rozptylové matice z promítaných vzorků, např.

$$W_{opt} = \arg\max_w \left| W^T S_T W \right|$$

$$= \left[w_1 \quad w_2 \quad \ldots \quad w_m \right]$$

kde $\{w_i | \ i = 1, 2,\ldots, m\}$ je sada *n*-rozměrných vlastních vektorů z S_T korespondující s *m* největších vlastních hodnot. V momentě, kdy vlastní vektory dosáhnou stejného rozměru jako originální obrázky, začneme je označovat jako *Charakteristické obrázky* a *Charakteristické obličeje*. Pokud provádíme klasifikaci za použití klasifikátoru přes nejbližší sousedy v redukovaném prostoru rysů a *m* je voleno jako počet *N* obrázků v tréninkové sadě, pak metoda charakteristického obličeje je rovnocenná s korelační metodou popisovanou v předchozí kapitole.

Nevýhodou tohoto přístupu je, že rozptyl, který se stává maximální, nenáleží pouze mezitřídě rozptylu, což je užitečné pro klasifikaci, ale i rozptylu uvnitř třídy, což je pro klasifikační účely nežádoucí informace. Hodně rozdílů mezi obrázky je zapříčiněno změnami osvětlení. Pokud je tedy PCA použita pro snímky obličejů v různorodém osvětlení, pak projekční matice W_{opt} bude obsahovat hlavní složky (tj. Charakteristické obličeje) které si zachová v projekčním prostoru díky různorodému osvětlení. V důsledku toho nebudou vzorky v projekčním prostoru dostatečně napěchovány, či se dokonce mohou promíchat s dalšími třídami.

Bylo zjištěno, že odstraněním 3 nejvýznačnějších základních složek, se rozlišitelnost v závislosti na osvětlení sníží. Snahou je, aby se např. odstraněním některých hlavních složek potlačila rozdílnost způsobená světlem. Pak ignorováním takové složky lze dosáhnout lepšího uspořádání vzorků v projekčním prostoru. S tím ale ještě souvisí fakt, že několik prvních hlavních složek výhradně souvisí s rozdílností v nasvícení, a tím pádem můžeme přijít i o užitečné informace potřebné k rozpoznávání. [2]

3.3 Lineární podprostory

Obojí, korelace i metoda charakteristické obličeje jsou předurčeny k nevhodnému chování při změnách směru osvětlení. Dokonce i zobrazení pro metody využívající pozorování snímků konkrétní tváře (za předpokladu Lambertianova povrchu beze stínů) leží v 3D lineárním podprostoru.

Uvažujme bod *p* na difúzním povrchu osvětleném světelným zdrojem umístěným v nekonečnu. Nechť $\mathbf{s} \in \mathbf{R}^3$ je sloupcový vektor vyznačující součin intenzity osvětlení

s jednotkovým vektorem pro směr zdroje světla. Když je povrch snímán kamerou, výsledná intenzita bodu obrázku je dána

$$E(p) = a(p)n(p)^{\tau} s$$

kde **n**(p) je vnitřní normálový vektor povrchu v bodě p a $a(p)$ je podíl dopadajícího a odrážejícího záření od povrchu v bodě p. To ukazuje, že intenzita snímku v bodě p je lineární v $\mathbf{s} \in \mathbf{R}^3$. Tedy při absenci stínování postačí 3 dané snímky difúzního povrchu ze stejného bodu pohledu pro 3 známé lineárně nezávislé směry zdroje světla k získání normál povrchu a podílu dopadajícího a odráženého záření. Tato metoda je dobře známá pro stereo fotografii. Alternativně lze obrázek získat z povrchu při libovolném směru osvětlení, při lineární kombinaci 3 originálních snímků.

Pro klasifikaci je tento fakt velmi důležitý. Poukazuje na to, že pro pevný bod pohledu leží snímky s difúzním povrchem v 3D lineárním podprostoru ve vysoce-rozměrovém zobrazovacím prostoru. Toto pozorování navrhuje jednoduchý klasifikační algoritmus pro rozpoznání difúzního povrchu – necitlivého na velký rozsah světelných podmínek.

Pro každý obličej se vezmou 3 nebo více snímků nasvícených z různých stran a vytvoří se 3D základ pro lineární podprostor. Povšimněte si, že tři základní vektory mají stejný rozměr jako snímky z tréninkové sady a můžou být chápány jako základní obrázky. Pro proces rozpoznávání už jen jednoduše dopočítáme vzdálenosti nového obrázku do každého z lineárních podprostorů a zvolíme obličej s nejmenší prostorovou vzdáleností. Tento proces nazýváme metodou *Lineárního podprostoru*.

Pokud snímky nejsou zašuměné či nejsou částečně zahaleny ve tmě, algoritmus lineárního podprostoru by měl dosahovat nulové úrovně chybového rozpoznávání při jakémkoliv osvětlení za předpokladu difuzního povrchu. Nicméně je zde několik pádných důvodů proč tomu tak není. První důvodem je vlastní stínění, zrcadlení a výraz tváře, díky čemuž se nacházejí v obličeji místa, která přes své rozdílnosti nezapadají do modelu lineárního podprostoru. Při dostatečném počtu snímků bychom měli být schopni rozlišit, která místa v obličeji těmto neduhům podléhají, a která jsou vhodná pro rozpoznávací proces. Druhým problémem je potřeba měřit vzdálenost k lineárnímu podprostoru pro každou osobu. I když jsme tímto dosáhly velkého pokroku od korelačního přístupu, stále potřebujeme velký počet snímků reprezentujících rozmanitosti každé třídy, a tím i velkou výpočtovou náročnost. A v poslední řadě, z pohledu ukládání dat, algoritmus lineárního podprostoru musí držet tři snímky v paměti pro každou osobu. [2]

3.4 Fisherfaces

Předchozí algoritmus těžil z faktu, že za nepochybně zidealizovaných podmínek, rozdílnosti třídy ležely v lineárním podprostoru v rámci projekčního prostoru. Čili jsou třídy konvexní a tedy i lineárně separovatelné. Šlo by provést redukci rozměrnosti prostřednictvím lineární projekce při současném dosažení lineární separovatelnosti. Toto je silný argument hovořící pro použití lineárních metod pro redukci rozměrnosti v procesu rozpoznávání obličejů, zejména když hledáme necitlivost k světelným podmínkám.

V momentě kdy máme tréninkovou sadu ohodnocenu, dává smysl použít tuto informaci k vytvoření více spolehlivé metody pro redukci rozměrnosti prostoru rysů. Tady se ukazuje, že užití specifické lineární metody třídy pro redukci rozměrnosti a jednoduchý klasifikátor v redukovaném prostoru rysů, může dosáhnout lepší

rozpoznávací schopnosti než dokonce metoda *Lineárních podprostorů* nebo metoda Charakteristického obličeje. Fisherův lineární diskriminant (FLD) je příkladem metody specifické třídy ve smyslu, že zkouší „zformovat“ rozptyl za účelem zlepšit spolehlivost při klasifikaci. Tato metoda vybírá W (více v [3]) v takovém smyslu, že poměr rozptylu mezi třídami a uvnitř třídy je maximalizován.

Nechť je matice rozptylu mezi třídami definována jako

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

a matice rozptylu uvnitř třídy je definována jako

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

kde μ_i je průměrný obrázek třídy X_i a N_i je počet vzorků v třídě X_i. Pokud je S_W nesingulární, pak optimální projekce W_{opt} je volena jako matice s ortogonálními sloupci, což maximalizuje poměr determinantu rozptylové matice mezi třídami z projekčních vzorků a determinant matice rozptylu uvnitř třídy z projekčních vzorků, tj.

$$W_{opt} = \arg\max_w \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \qquad (1)$$

$$= \left[w_1 \quad w_2 \quad \ldots \quad w_m\right]$$

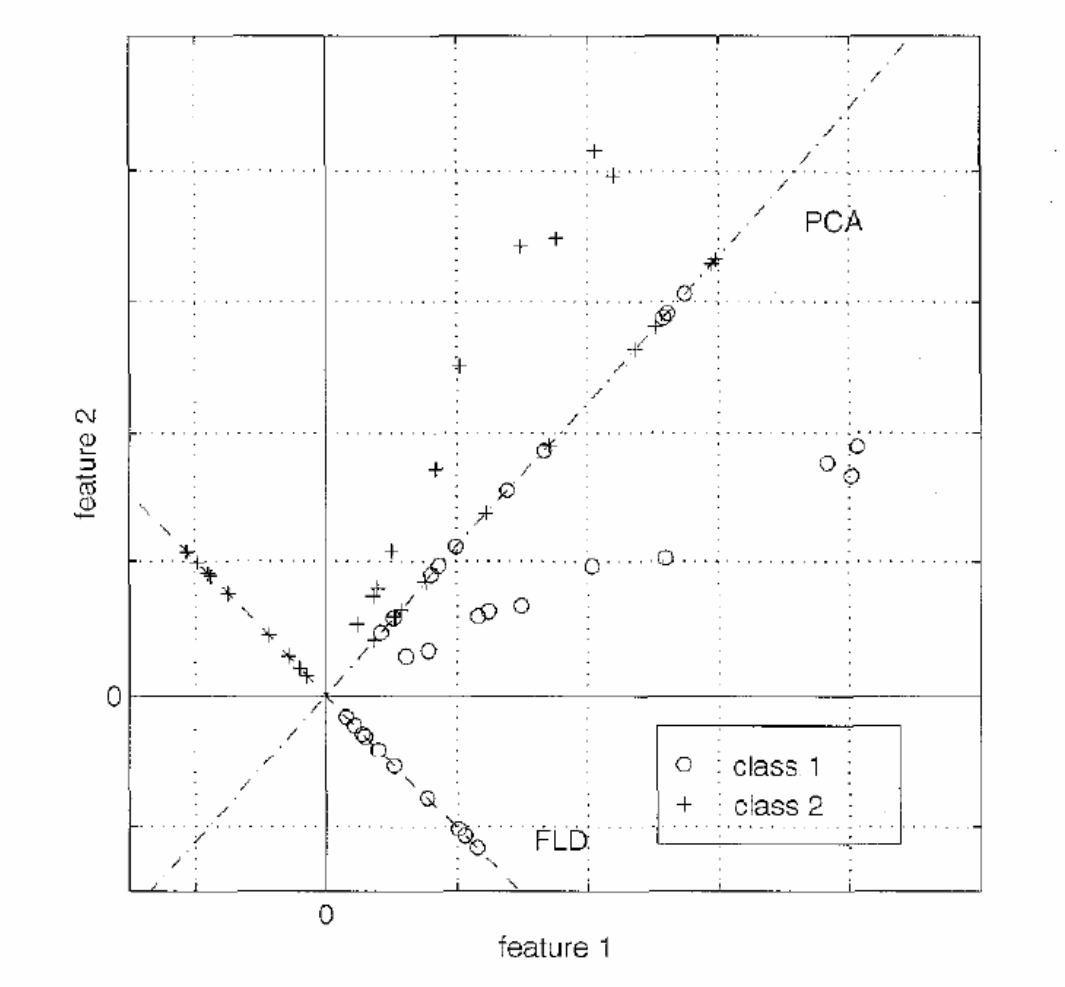
kde $\{\mathbf{w}_i | \ i = 1, 2, \ldots, m\ \}$ je sada zobecněných vlastních vektorů S_B a S_W korespondující s m největších zobecněných vlastních hodnot $\{\lambda i \mid i = 1,2,\ldots,m\}$ tj.,

$$S_B \mathbf{w}_i = \lambda i S_w \mathbf{w}_i, \qquad i = 1, 2, \ldots, m$$

Povšimněte si, že je zde nejvýše c - 1 nenulových zobecněných vlastních hodnot a horní mezí pro m je tedy c - 1, kde c je počet tříd. Více v [5].

Pro ilustraci výhod projekce lineární specifické třídy, se můžete podívat na obr. 2, na kterém najdete srovnání metody PCA a FLD pro problém dvou tříd, kde jsou vzorky náhodně rozmístněny ve směru kolmém k lineárnímu podprostoru. V tomto případě je $N = 20$, $n = 2$ a $m = 1$. Vzorky každé třídy leží blízko přímky procházející skrz originál v 2D prostoru rysů. PCA i FLD byly použity k projekci bodů z 2D do 1D. Porovnání těchto dvou projekcí na obrázku ukazuje, že PCA promíchá třídy do sebe, takže v projekčním prostoru už nejsou lineárně separabilní. Je tedy zřejmé, že ačkoliv PCA dosáhne většího celkového rozptylu, metoda FLD dosahuje většího rozptylu mezi třídami, a tím pádem zjednodušuje klasifikaci.

V problematice rozpoznávání je člověk vystaven komplikaci, že rozptylová matice vnitřní třídy $S_W \in R^{nxn}$ je vždy singulární. To pramení z faktu, že rozsah S_W je nejčastěji $N - c$, a obecně je počet obrázků v tréninkové množině N mnohem menší, než je počet bodů n (rozlišení) v každém obrázku. To znamená, že je možné zvolit matici W takovou, kde rozptyl uvnitř třídy projekčních vzorků může být vynulován.



Obr. 2 Porovnání metody analýzy hlavních složek (PCA) s metodou Fischerova lineárního diskriminantu (FLD) pro problém 2 tříd, kde data každé třídy leží blízko lineárního podprostoru.

Pro překonání komplikací se singulární S_W, je zapotřebí jiné měřítko než (1). Tato metoda, kterou nazýváme Fisherfaces, řeší tento problém promítáním obrázků do méně rozměrného prostoru, takže výsledná matice rozptylu vnitřní třídy S_W je nesingulární. Toho je dosaženo použitím metody PCA k redukci rozměrnosti prostoru rysů na N – c a posléze aplikování standardní FLD metody (1) k redukci rozměrnosti na c – 1. Formálněji

$$W_{opt}^T = W_{fld}^T W_{pca}^T \qquad (2)$$

kde

$$W_{pca} = \arg\max_W \left| W^T S_T W \right|$$

$$W_{fld} = \arg\max_W \frac{\left| W^T W_{pca}^T S_B W_{pca} W \right|}{\left| W^T W_{pca}^T S_W W_{pca} W \right|}$$

Povšimněte si, že optimalizace pro W_{pca} je provedena přes $n \times (N - c)$ matici s ortogonálními sloupci, zatímco optimalizace W_{fld} je provedena přes $(N - c) \times m$ matici s ortogonálními sloupci. Při počítání W_{pca}, jsme tedy zahodili pouze tu nejmenší $c - 1$ hlavní složku.

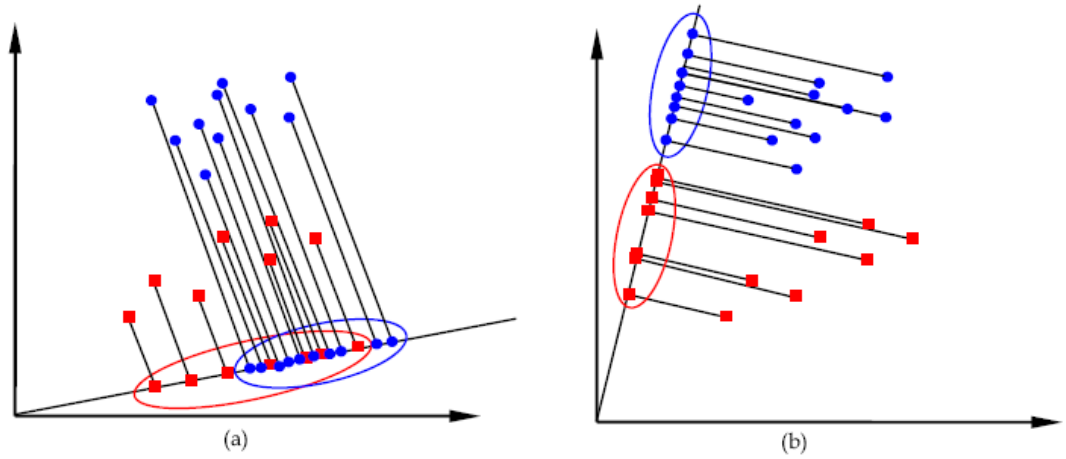
Samozřejmě jsou zde i jiné způsoby jak redukovat rozptyl uvnitř třídy, zatímco rozptyl mezi třídami zachováme. Další takovou možností je například volba W pro maximalizaci rozptylu mezi třídami promítaných vzorků až po ukončení redukce rozptylu vnitřní třídy. Vzato do extrému, můžeme maximalizovat rozptyl mezi třídami s omezením, že rozptyl uvnitř třídy bude nula, tj.

$$W_{opt} = \arg\max_{W \in W'} \left| W^T S_B W \right| \qquad (3)$$

kde W' je sada $n \times m$ matic s ortogonálními sloupci obsažených v jádře S_W. [2]

3.5 LDA (Linear Discriminant Analysis)

Lineární diskriminační analýza je zobecnění Fisherfaces, takže základ metody je stejný. Tyto metody překonávají metodu charakteristického obličeje použitím fischerova lineárního diskriminačního kritéria. Snaží se o maximalizaci determinantu matice rozptylu mezi třídami a matice rozptylu uvnitř třídy pro promítané vzorky. Fisherův diskriminant seskupuje snímky z jedné třídy a snímky z jiných tříd se snaží oddělit. Při promítnutí tříd ve 2D prostoru do pravoúhlých os, by mohlo dojít k promíchání těchto tříd, a tak se Fisherův diskriminant snaží najít takovou přímku, na které by tyto třídy byly odděleny, viz obr. 3. Na obr. 4 jsou vidět ukázky dobré a špatné separace tříd.[4][12]



Obr. 3 a) Body jsou při projekci na přímku smíchány. b) Stejné body jsou při projekci na jinou přímku odděleny.

Obr. 4 a) Dobře oddělené třídy. B) Špatně oddělené třídy.

Matice rozptylu uvnitř třídy S_w a matice rozptylu mezi třídami S_b jsou definovány jako:

$$S_W = \sum_{j=1}^{c} \sum_{i=1}^{N_j} \left(\Gamma_i^j - \mu_j\right)\left(\Gamma_i^j - \mu_j\right)^T$$

kde Γ_i^j je *i*-tý vzorek třídy *j*, μ_j je průměr třídy *j*, *C* je počet tříd, N_j je počet vzorků ve třídě *j*.

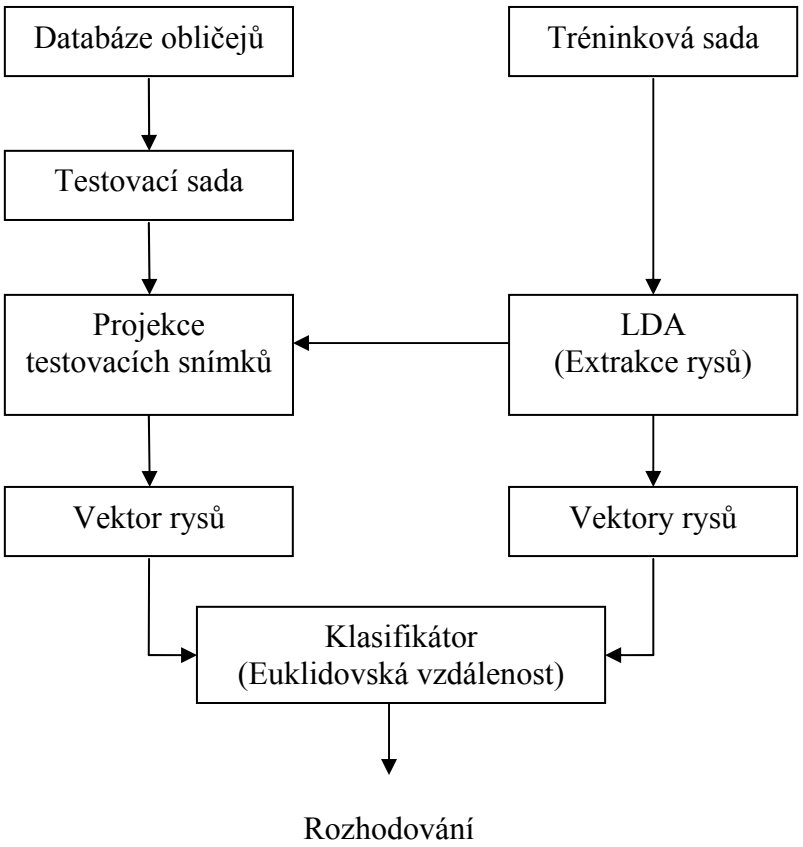
$$S_b = \sum_{j=1}^{c} \left(\mu_j - \mu\right)\left(\mu_j - \mu\right)^T$$

kde μ reprezentuje průměr všech tříd. Pak podprostor LDA je zastřešen sadou vektorů $W = [W_1, W_2, \ldots, W_d]$ vyhovující rovnici

$$W = \arg\max \left| \frac{W^T S_b W}{W^T S_W W} \right|$$

Matice rozptylu uvnitř třídy popisuje, jak blízko jsou snímky obličeje rozprostřeny uvnitř třídy, a matice rozptylu mezi třídami popisuje, jak jsou třídy odděleny mezi sebou navzájem.

Pro představu celkového průběhu rozpoznávání pomocí metody LDA se podívejte na následující schéma na obr. 5.



Obr. 5 Schéma rozpoznávacího procesu při použití LDA metody.

3.6 ICA (Independent Component Analysis)

Metoda analýzy nezávislých složek (ICA) hledá základní faktory nebo složky v mnohoproměnných statistických datech. Na rozdíl od spousty jiných metod, ICA hledá složky, které jsou, jak statisticky nezávislé, tak negausovské.[9]

V problematice rozpoznávání obličejů se využívají dvě architektury, které se liší ve směru porovnávání dat. Zjednodušeně řečeno je rozdíl v tom, zda se postupuje od obličejů k bodu obrazu nebo od bodů obrazu k obličeji.

Obecně se metoda ICA používá k separaci zdrojů signálu, či extrakci rysů a toho se právě využívá v problematice rozpoznávání obličejů.

3.6.1 Podrobnější popis ICA

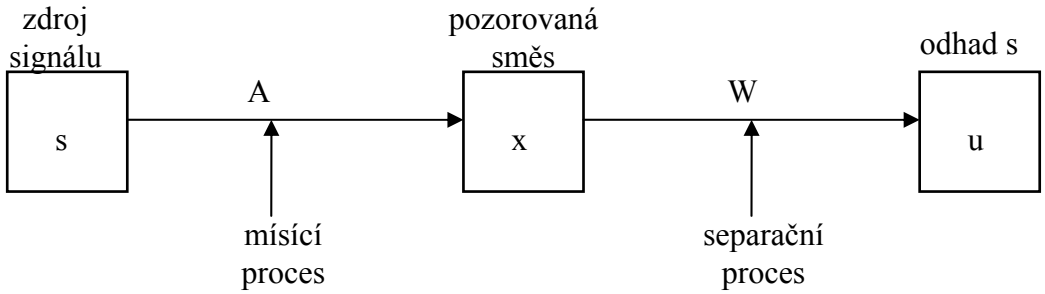
Zatímco PCA dekoreluje vstupní data pomocí statistiky druhého řádu a tím generuje kompresovaná data s minimální střední kvadratickou projekční chybou, ICA minimalizuje vstupní závislosti druhého i vyššího řádu. To je těsně spjato s problémem *rozdělování slepého zdroje* (blind source separation - BSS), kde je cílem rozložit sledovaný signál do lineární kombinace neznámých nezávislých signálů. Nechť s je vektor signálu neznámého zdroje a x je vektor pozorované směsice signálů. Pokud A je neznámá mísící matice, pak model míchání bude zapsán takto:

$$x = As$$

To předpokládá, že zdroj signálů je nezávislý na každém jiném zdroji a matice směsice A je regulární. ICA algoritmus, založený na těchto předpokladech a pozorování směsice, se snaží najít mísící matici A nebo separační matici W tak, že

$$u = Wx = WAs$$

je odhad nezávislého zdrojového signálu (obr. 6)



Obr. 6 Separační model slepého zdroje.

Na metodu ICA se dá nahlížet jako na zobecnění metody PCA. Metoda PCA se v tomto případě používá na předzpracování vstupního signálu. K předzpracování se často používá i bělení (whitening) [8]. Transformace bělení se používá například pro odstranění redundance zdrojů či odstranění adaptivního šumu a může být určena jako $D^{-1/2}R^T$ kde D je diagonální matice vlastních hodnot a R je matice ortogonálních vlastních vektorů z kovariantní vzorové matice. Použitím bělení na pozorované směsi, nicméně získáme výsledky na zdrojových signálech pouze do ortogonální transformace. ICA jde o krok dál, takže transformuje vybělená data na sadu statisticky nezávislých signálů.

Signály jsou statisticky nezávislé, když

$$f_u(u) = \Pi_i f_{ui}(u_i)$$

kde f_u je pravděpodobnost hustoty funkce u.(Jinými slovy, vektory u jsou rovnoměrně rozděleny) Bohužel nemusí existovat žádná matice W, která by splnila podmínku nezávislosti a není ani žádný ucelený způsob nalezení W. Namísto toho je několik algoritmů, které iterativně aproximují W a tím nepřímo maximalizují nezávislost.

Ačkoliv je obtížné maximalizovat podmínku nezávislosti přímo, všechny běžné ICA algoritmy převádějí problém na iterativní optimalizaci hladké funkce, jejíž globální optimum je v místě, kde vektory u jsou nezávislé. Nejběžnější ICA algoritmy jsou *InfoMax*, *FastICA* nebo např. *JADE*, ale i spousta dalších.

Tak například metoda InfoMax je založena na pozorování, že nezávislost je maximální když se maximalizuje entropie H(u)

$$H(u) \equiv -\int fu(u) \log fu(u) du$$

InfoMax provádí gradientní výstup(ascent) na elementech w_{ij} tak, aby maximalizovala H(u). (název vznikl pozorováním, že maximalizaci H(u) také maximalizuje společná informace I(u,x) mezi vstupními a výstupními vektory) Algoritmus *JADE* minimalizuje strmost funkce f(u) skrze společnou diagonalizaci kumulantů čtvrtého řádu, zatímco minimalizace strmosti zároveň maximalizuje statistickou nezávislost. Metoda *FastICA* je asi nejobecnější, při maximalizaci

$$J(y) \approx c[E\{G(y)\} - E\{G(v)\}]^2$$

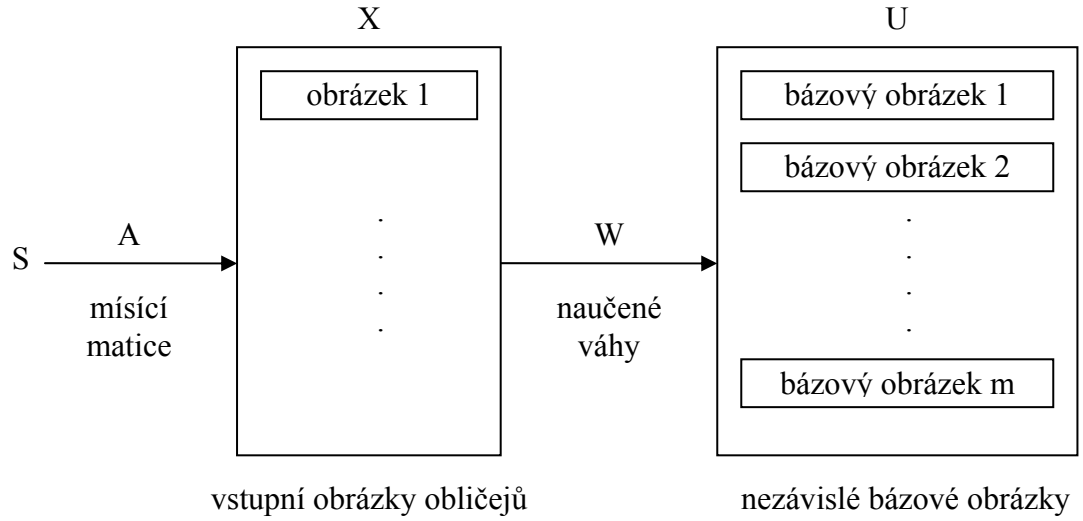
kde G je nekvadratická funkce, v je gausovská náhodná proměnná a c je jakákoliv kladná konstanta. Jde tedy dokázat, že maximalizací jakékoliv funkce z této rovnice, budeme maximalizovat i nezávislost.

InfoMax, JADE a FastICA jsou všechny maximalizující funkce se stejným globálním optimem. Jejich výsledky by měly vždy konvergovat ke stejným hodnotám, pokud je budeme porovnávat na stejné sadě dat. V praxi je testovali Zibulevsky a Pearlmutter a došli jenom k velmi malým rozdílům [10].

3.6.2 Architektura I – statisticky nezávislé základy obrázků

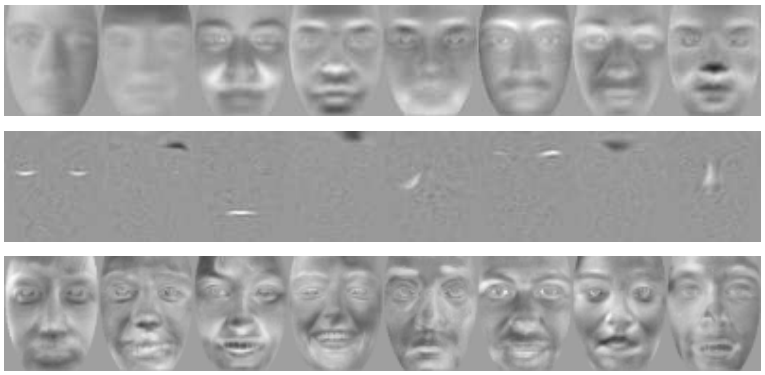
Nehledě na použitý algoritmus k výpočtu ICA, existují dva rozdílné přístupy jak metodu ICA aplikovat v problematice rozpoznávání. V architektuře I se vstupní snímky obličejů považují za lineární směs statisticky nezávislých základních obrázků S kombinovaných neznámou mísící maticí A. ICA algoritmus se naučí váhy matice W, která se používá k vytvoření sady nezávislých základních obrázků v řadách U obr. 7.

V této architektuře jsou snímky obličejů jako proměnné a hodnoty bodů obrázku poskytují pozorování pro tyto proměnné. Separace zdrojů tedy probíhá v prostoru obličejů. Promítnutím vstupních snímků na vážený tréninkový vektor produkuje nezávislé bázové obrázky. Zmenšenou reprezentací snímků obličejů jsou vektory koeficientů použitých pro lineární kombinování nezávislých bázových obrázků k vytvoření zobrazení. Prostřední řada Obr. 8 ukazuje osm bázových obrázků vytvořených touto architekturou. Jsou prostorově lokalizované na rozdíl od PCA bázových obrázků (horní řada) nebo těch co vytvořila ICA architektura II (spodní řada).



Obr. 7 Hledání statisticky nezávislých bázových obrázků.

Barlett a kolektiv poprvé použili PCA k projekci dat do *m*-rozměrného podprostoru, aby řídili počet nezávislých komponent produkovaných metodou ICA. Pak aplikovali algoritmus InofMax na charakteristické vektory na minimalizaci statistické závislosti v rámci výsledných bázových obrázků. Toto použití PCA, jako předzpracování ve dvou krokovém procesu, umožnilo ICA vytvořit podprostor o velikosti *m* pro jakoukoliv hodnotu *m*. Použití PCA na předzpracování je také dobré pro zvýšení výkonnosti ICA, zahozením malých stop charakteristických hodnot před bělením, a redukcí výpočtové komplexnosti minimalizací zdvojených závislostí. PCA dekoreluje vstupní data. Zbývající závislosti druhého řádu rozděluje ICA.



Obr. 8 Osm vektorů rysů pro techniky PCA, ICA I a ICA II.

Horní řádek obsahuje osm vlastních vektorů s nejvyššími vlastními hodnotami pro metodu PCA. Druhý řádek ukazuje osm lokalizovaných vektorů rysů pro ICA architekturu I a na posledním řádku je zobrazeno osm (nelokalizovaných) vektorů rysů za použití ICA architektury II.

Pokud se pokusíme vyjádřit architekturu I matematicky, vypadal by zápis následovně:

Nechť R je p x m matice obsahující prvních m charakteristických vektorů ze sady n obrázků obličejů. Pak p bude počet bodů v tréninkovém obrázku. Řádky vstupní matice ICA jsou proměnné a sloupce jsou pozorování, čili ICA je provedeno na R^T.

Počet m nezávislých bázových obrázků v řádcích U jsou počítány jako $U = W*R^T$. Pak $n \times m$ ICA matice koeficientů B pro lineární kombinaci nezávislých bázových obrázků v U je počítána následovně:

Nechť C je n x m matice z PCA koeficientů, pak

$$C = X * R \text{ a } X = C * R^T$$

Z $U = W * R^T$ a předpokladu, že W je invertibilní dostáváme

$$R^T = W^{-1} * U$$

Čili

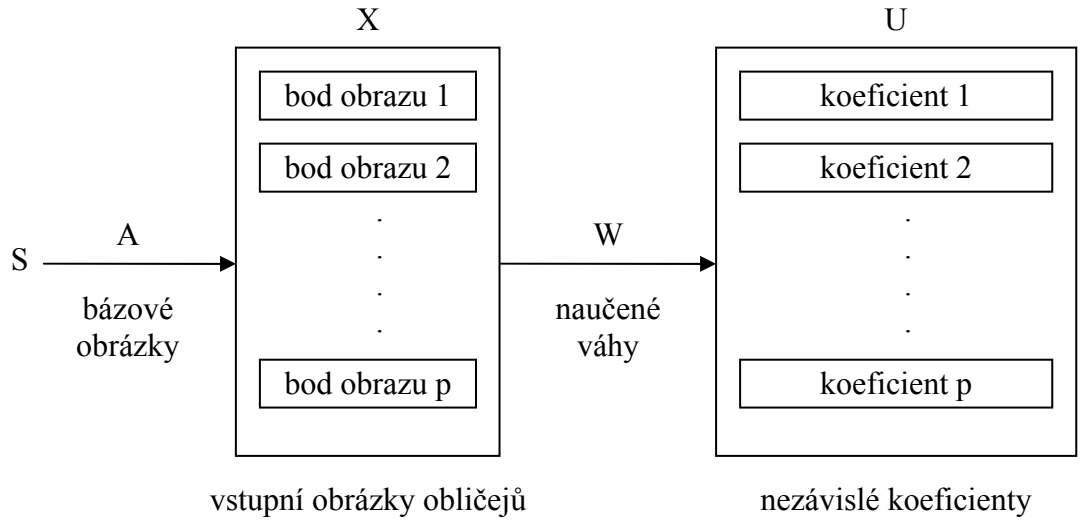
$$X = (C * W^{-1}) * U = B * U$$

Každý řádek matice B obsahuje koeficienty pro lineární kombinaci s bázovými obrázky k porovnání snímku obličeje s korespondujícím řádkem z matice X. Také platí, že X je rekonstrukcí originálních dat s minimální čtvercovou chybou stejně jako u PCA.[11]

3.6.3 Architektura II – Statisticky nezávislé koeficienty

Zatímco jsou bázové obrázky získané architekturou I statisticky nezávislé, koeficienty, reprezentující vstupní obrázky v podprostoru definovaném bázovými obrázky, nejsou. Cílem architektury II je najít statisticky nezávislé koeficienty pro vstupní data. V této architektuře je vstup transponován z arch. I, kde body obrazu jsou proměnné a obrázky jsou předmětem pozorování. Separace zdrojů je prováděna na bodech a každý řádek z naučené matice vah W je obrázek. Matice A (inverzní matice W) obsahuje bázové obrázky ve svých sloupcích. Statisticky nezávislé zdroje koeficientů v matici S, která zahrnuje vstupní obrázky, se dají získat ze sloupců matice U (obr. 9). Tato architektura byla použita k hledání filtrů obrázků, které vytvářeli statisticky nezávislé výstupy z přirozené scény. Osm bázových snímku ve spodní řadě na obr. 8 ukazují více globálních vlastností než základní obrázky produkované architekturou I (prostřední řádek).

Matematicky bychom mohli schéma popsat následovně. Statisticky nezávislé koeficienty jsou počítány jako $U = W * C^T$ a bázové obrázky ukázané na obr. 8 jsou obdrženy ze sloupců $R * A$.



Obr. 9 Hledání statisticky nezávislých koeficientů.

3.7 EP (Evolutionary Pursuit)

Adaptivní přístup založený na charakteristickém prostoru, který vyhledává nejlepší sadu projekčních os s ohledem na maximalizaci vhodnostní funkce, měřící u systému ve stejný čas přesnost a schopnost generalizace. Protože je rozměr dimenze řešeného prostoru tohoto problému příliš velká, využívá se specifického druhu genetického algoritmu nazývaného „Evoluční pronásledování“ (EP).

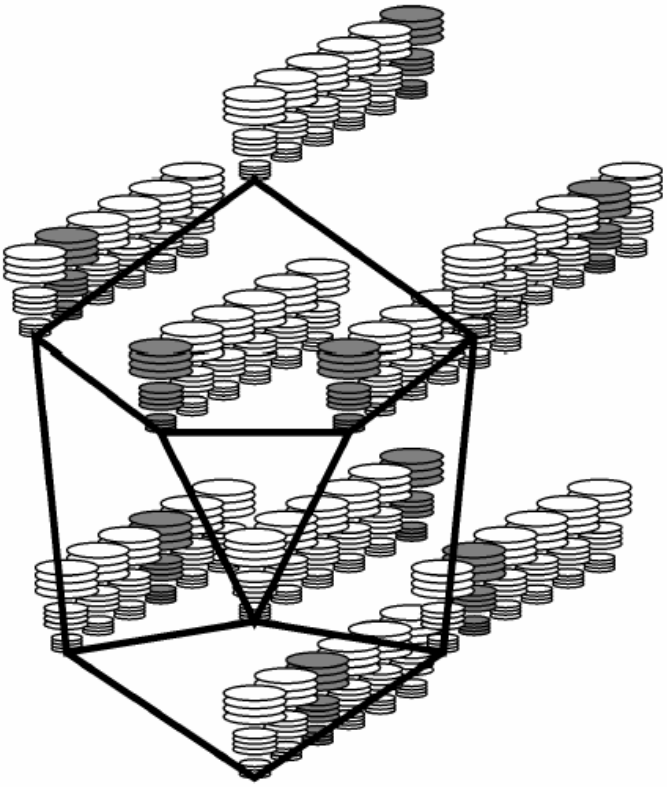
3.8 Evoluční algoritmus použitý pro výběr vlastních vektorů LDA

LDA je populární technikou pro extrakci rysů při rozpoznávání obličejů. Nicméně tato metoda často trpí malým počtem tréninkových vzorků v momentě, kdy pracujeme s mnohorozměrnými daty. Jsou jisté postupy navrhující překonání tohoto problému, ale většinou jenom využijí všech vlastních vektorů i neplatných nebo rozsahu podprostoru rozptylné matice uvnitř třídy (S_W). Nicméně experimentální výsledky ukazují, že ne všechny vlastní vektory v plném prostoru jsou přínosem pro klasifikační výkon. Některé z nich jsou na škodu. (tohoto poznatku bylo využito k vytvoření úspěšné metody viz EPCA).

Doposud nebyly vytvořeny konkrétní metody, jež by vyhodnotily, které vlastní vektory by se měly použít a které ne. Proto autoři článků [17] navrhují novou metodu EDA + Full-space LDA, která poskytuje plnou výhodu diskriminujících informací v podprostoru S_W výběrem optimální sady vlastních vektorů. Tato metoda úspěšně překonává ostatní LDA metody. [17]

3.9 EBGM (Elastic Bunch Graph Matching)

V českém názvosloví „Metoda shlukových grafů“. Všechny obličeje sdílejí podobnou topologickou strukturu. Obličeje se tak dají reprezentovat jako graf s uzly představující pozice význačných bodů tváře (oči, nos …) a hranami popisující 2D vzdálenosti. Každý uzel obsahuje sadu 40 komplexních Gaborových waveletových koeficientů pro různá měřítka a orientaci (fáze, amplitudy). Nazýváme je „jety“[džety]. Rozpoznávání je založeno na ohodnocených grafech. Ohodnocený graf je sada uzlů spojených hranami, uzly jsou ohodnoceny jety a hrany jsou ohodnoceny vzdálenostmi. Takový graf je schématicky znázorněn na obr. 10.

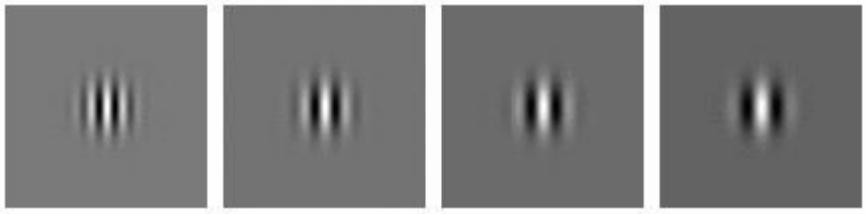


Obr. 10 Symbolické znázornění grafu s jety a význačnými body obličeje.

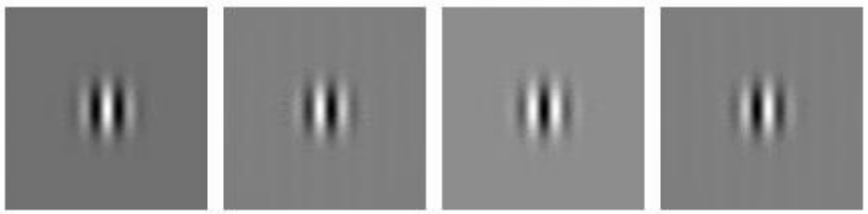
3.9.1 Technika shlukových grafů

Wiskott a kol. (1999) definuje tuto techniku jako základní proces pro porovnání grafů s obrázky a pro generaci nových grafů. V nejjednodušším případě je jeden ohodnocený graf napasován na obrázek. Ohodnocený graf má sadu uzlů umístěných na konkrétních místech v prostoru obličeje. Každý uzel se nazývá jet a popisuje malou oblast šedých hodnot v okolí bodu. Tento popis je založen na waveletové transformaci (pro tento případ zvaná Gabor-waveletová transformace) obrázku. Sada konvolučních koeficientů pro jádra různých pootočení a frekvencí v jednom bodě vytváří jet.

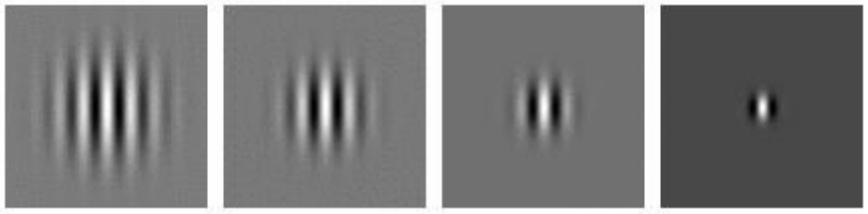
Obr. 11 Ukázka různých pootočení Gabor-waveletového filtru.



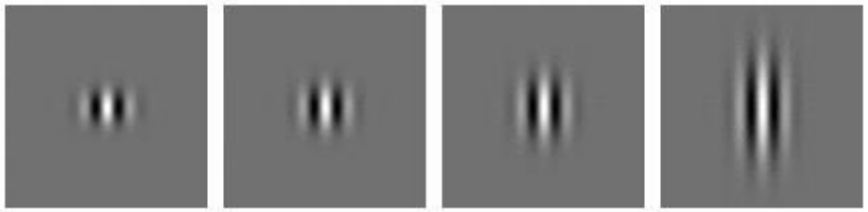
Obr. 12 Různá vlnová délka Gabor-waveletového filtru.



Obr. 13 Různá fáze Gabor-waveletového filtru.



Obr. 14 Různá velikost Gabor-waveletového filtru.



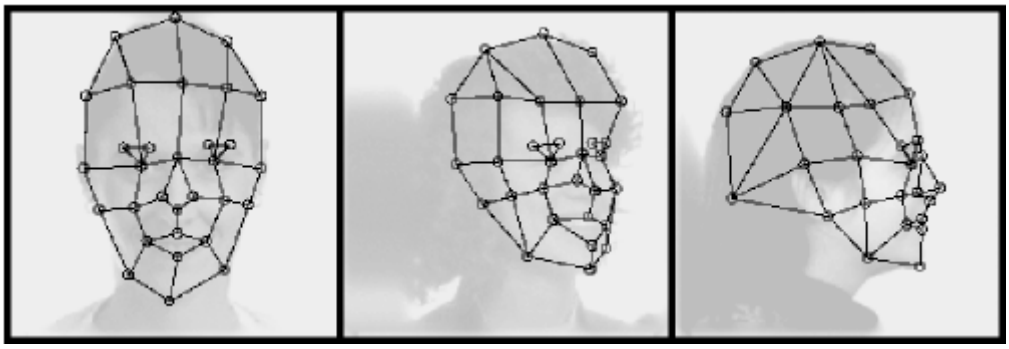
Obr. 15 Různý poměr stran Gabor-waveletového filtru

3.9.2 FGB – shlukový graf obličeje

Shlukový graf obličeje je obecnou reprezentací obličeje jako takového. Používá se pro automatické získávání grafů z dalších obličejů. FBG zahrnuje velký počet rozličných výskytů obličeje, jako jsou různé tvary očí, úst a rozdílnosti v rase, pohlaví, atd. Bylo by hodně nákladné pokrýt každý rys zvláštním grafem. Místo toho se využívá sada M individuálních modelových grafů G^{Bm} (m = 1, …M), kde B reprezentuje sadu jetů korespondujících s jedním základním bodem zvaným shluk (např. shluk očí) a jejich kombinací tak vzniká struktura podobná haldě, kterou pak nazýváme shlukový graf obličeje. Viz obr. 10.

3.9.3 Trénování shlukových grafů

Prvním krokem je natrénovat FBG. To se provádí manuálním umístěním význačných bodů tváře a definování vztahu rozdílných bodů a rozdílných póz (různé pózy jsou reprezentovány rozdílnými grafy)



Obr. 16 Ukázka grafů pro různé pózy obličeje.

V momentě kdy má systém FBG, může graf pro nový obrázek generovat automaticky. Tyto nové grafy označujeme jako grafové podpisy, abychom zabránili zmatení s grafy pro FBG, protože jsou jedinečným identifikátorem obrázku nebo slouží k rozpoznání obrázku. Wiskot a kol. (1997) varuje, že v momentě kdy FBG obsahuje pouze několik tváří, je nezbytné kontrolovat správné umístění výsledného grafového podpisu, ale v momentě, kdy FBG dosáhne dostatečného počtu (přibližně asi 70 grafů), můžeme se spolehnout na automatické generování a použít bez obav pro velké galerie snímků.

3.9.4 Rozpoznávání pomocí shlukových grafů

Vyextrahované grafové podpisy z tréninkové fáze jsou porovnávány s galerií snímků a pro každý snímek se počítá podobnost. Výsledná shoda se získá výběrem snímku s nejvyšší naměřenou podobností.

Při použití shlukového grafu pro porovnání se procedura trošku komplikuje. Krom jiného dochází k odchylkám při výběru význačných bodu obličeje, nicméně graf podobnosti se maximalizuje výběrem nejlépe sedícím jetům v každém shluku nezávisle na dalších shlucích. Bez této nezávislosti by příliš rostla výpočtová náročnost. (Wiskot a kol. 1997)

Nejlepších výsledků dosahuje metoda pro čelní pohledy, kdy se podařilo dosáhnout až 98% úspěšnosti rozpoznání [14]. Avšak při rozdílnosti póz se úspěšnost rapidně snižuje. V některých experimentech klesla úspěšnost až na 9%.

3.10 Trace Transform

Jinými slovy *trasová transformace* je zobecněním Radonovy transformace. Jedná se o nový nástroj pro zpracování obrazu, který může být použit pro rozpoznávání objektů, jež jsou nějakým způsobem pootočeny, posunuty nebo zvětšeny vzhledem k pozorovateli. K vytvoření transformace je zapotřebí provést výpočet funkce podél trasovacích přímek v obrázku. Užitím různých funkcionálů můžeme dosáhnout různých transformací. [1]

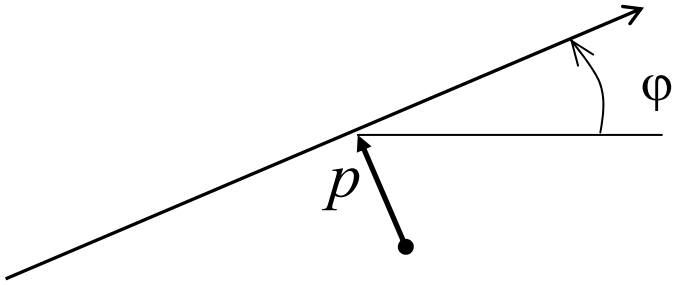
3.10.1 Trasová transformace v kostce

Trasová transformace se skládá z trasovacích přímek, podél kterých se počítají určité funkcionály. Ze stejného obrázku se mohou díky použití různých funkcionál získat různé transformace. Trasová transformace obrázku je 2D funkcí parametrů každé trasovací přímky. V rovině je přímka charakterizována dvěma parametry. Nejjednodušší příkladem trasové transformace je Houghova transformace, která se aplikuje na binární

hranové mapy a počítá počet hran na každé trasovací přímce. Tento počet se potom promítne jako funkce dvou přímkových parametrů k sestrojení Houghova prostoru.

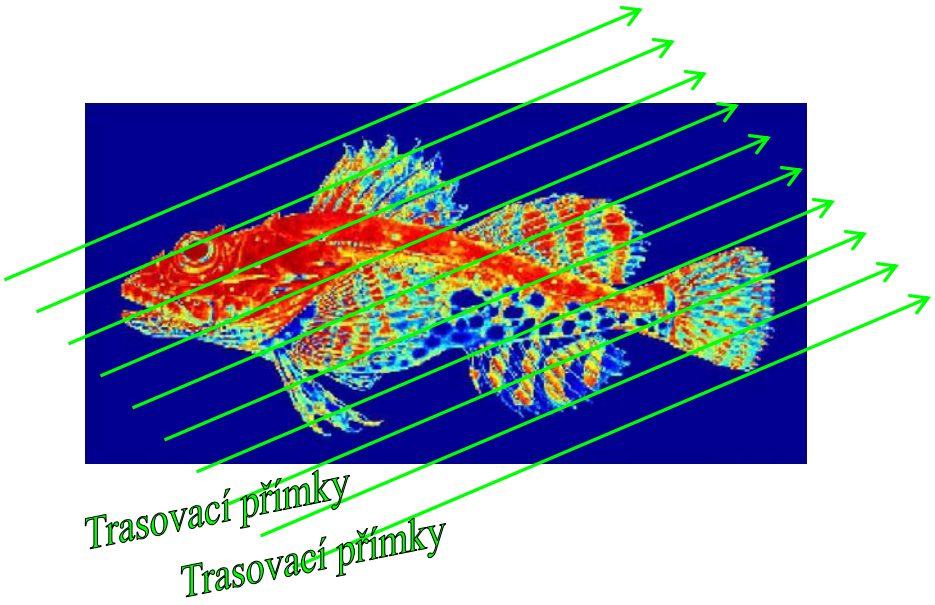
V následujících odstavcích si ukážeme tři funkcionály vhodné k vytvoření necitlivosti na natočení, posuvu a změně velikosti obrázku.

Radonova transformace obrázku je 2D reprezentace obrázku v souřadnicích φ a p s hodnotou *integrálu* vypočítaného z obrázku podél příslušné přímky vložené do buňky (φ,p).



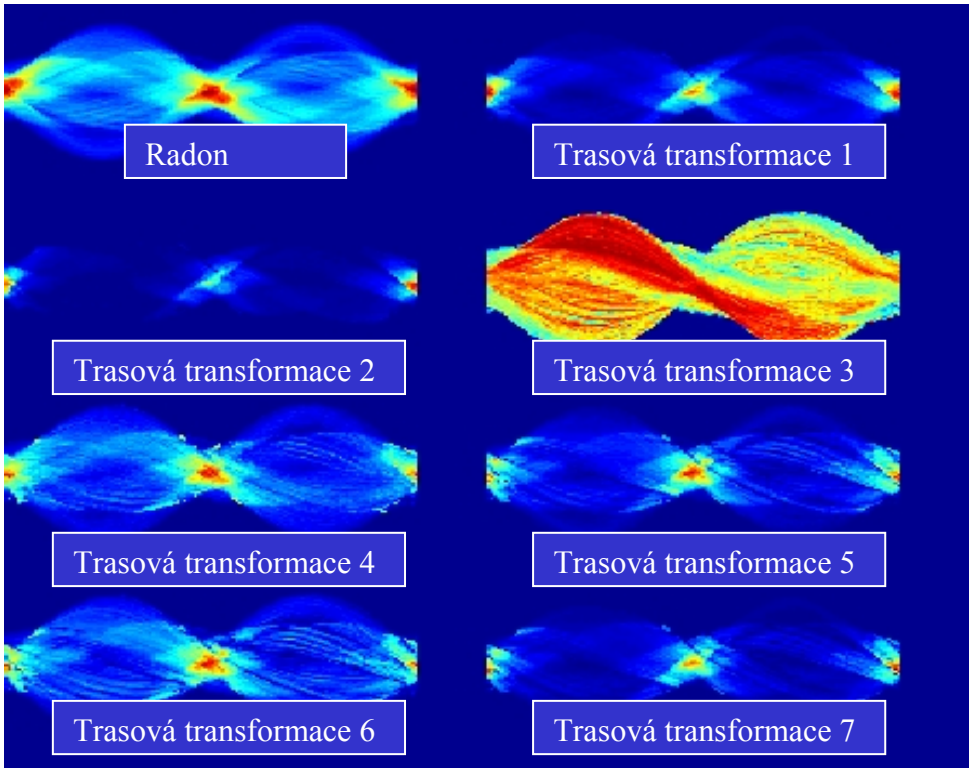
Obr. 17 Definice parametrů trasovací přímky.

Trasová transformace počítá *funkcionál T* přes parametr *t* podél přímky, ale nemusí jít o integrál.



Obr. 18 Ilustrační obrázek pro trasovou transformaci.

Na následujícím obrázku se můžete podívat, jaké výsledky se mohou různými transformacemi získat. První obrázek je Radonova transformace předešlého obrázku a ostatní jsou trasové transformace stejného obrázku užívající k výpočtu vzorce uvedené v následující tabulce.



Obr. 19. Příklady trasových transformací.

V tabulce použitý median{*x,w*} znamená vážený medián řady *x* s váhami v řadě *w*. Například median{{5,4,9,3},{3,2,1,1}} je medián čísel 5,4,9 a 3 s korespondujícími váhami 3,2,1 a 1. To znamená standardní medián z čísel 5, 5, 5, 4, 4, 9, 3 tj. medián uspořádané řady 3,4,4,5,5,5,9. Výsledkem je 5.

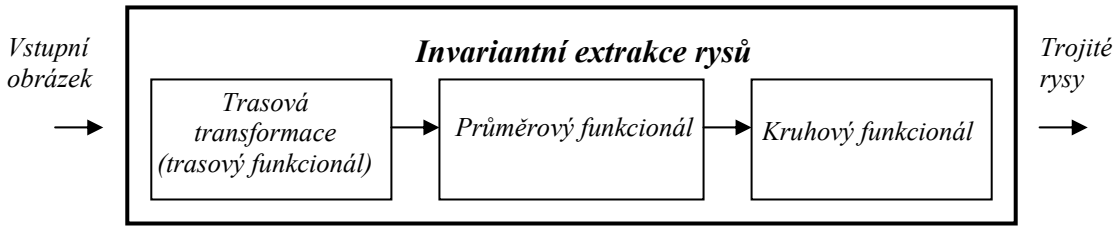
Trasová transformace	Použitý funkcionál
1	$T(f(x))=\int_{[0,\infty]} rf(r)dr$ kde $r = x–c$, a c=medián$_x\{x,f(x)\}$
2	$T(f(x))=\int_{[0,\infty]} r^2f(r)dr$ kde $r = x–c$, a c=medián$_x\{x,f(x)\}$
3	$T(f(x))=$ medián$_{r\geq 0}\{f(r), (f(r))^{1/2}\}$ kde $r = x–c$, a c=medián$_x\{x,f(x)\}$
4	$T(f(x))=$ medián$r_{\geq 0}\{rf(r), (f(r))^{1/2}\}$ kde $r = x–c$, a c=medián$_x\{x,f(x)\}$
5	$T(f(x))=\int_{[0,\infty]} e^{ik\log r} r^pf(r)dr$, ($p$=0.5, k=4) kde $r = x–c$, a c=medián$_x\{x,(f(x))^{1/2}\}$
6	$T(f(x))=\int_{[0,\infty]} e^{ik\log r} r^pf(r)dr$, ($p$=0, k=3) kde $r = x–c$, a c=medián$_x\{x,(f(x))^{1/2}\}$
7	$T(f(x))=\int_{[0,\infty]} e^{ik\log r} r^pf(r)dr$, ($p$=1, k=5) kde $r = x–c$, a c=medián$_x\{x,(f(x))^{1/2}\}$

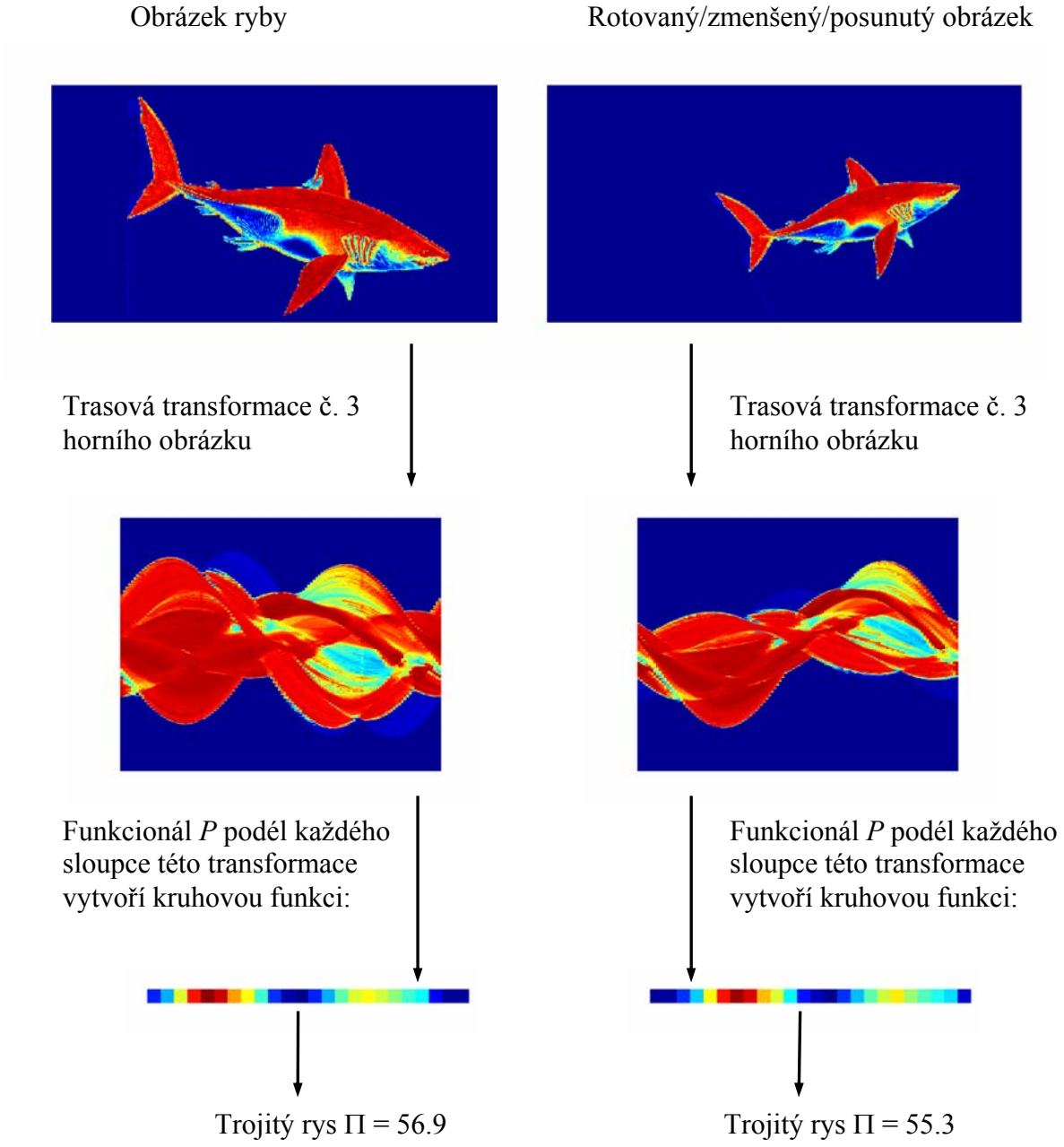
Tab. 1 Přehled použitých funkcionálů pro transformaci obr. 19

Konstrukce trojité rysu

1. Vytvořit trasovou transformaci z obrázku aplikací trasového funkcionálu *T* podél přímek procházejících obrázkem.
2. Vytvořit „kruhovou funkci“ z obrázku aplikací průměrového funkcionálu *P* podél sloupců trasové transformace.
3. Vytvořit trojitý rys aplikací kruhového funkcionálu Φ podél řetězu čísel vytvořených v kroku 2.

Blokové schéma:





Výběr funkcionálů závisí na tom, čeho chceme dosáhnout. Abychom věděli který zvolit, je dobré znát jejich vlastnosti. Smyslem funkcionálu je charakterizovat funkci číslem.

Funkcionál je operace Ξ definovaná na sadě funkcí a z toho plynoucích čísel. Nechť $\xi(x)$ vyjadřuje funkci proměnné $x \in \mathrm{R}$ (R zaujímá řadu reálných čísel). Pak výsledek aplikace funkcionálu Ξ na funkci $\xi(x)$ se značí $\Xi\xi$ nebo $\Xi(\xi(x))$ a jedná se o jediné číslo.

Homogenní funkcionál: Pro práci s obrázkem rozměrově změněným předpokládáme, že funkcionál má úsečkově-homogenní vlastnost.

$$\Xi(\xi(ax)) = a^{\kappa_\Xi}\,\Xi(\xi(x)) \quad \text{pro všechna } a > 0 \qquad \text{(vlastnost } i_1\text{)}$$

a může mít i jinou pořadově-homogenní vlastnost

$$\Xi(c\xi(x)) = c^{\lambda_\Xi}\,\Xi(\xi(x)) \quad \text{pro všechna } c > 0 \qquad \text{(vlastnost } i_2\text{)}$$

Konstanty κ_Ξ a λ_Ξ nazýváme konstanty homogenity funkcionálu Ξ.

Obecně se funkcionály nemusí řídit těmito vlastnostmi, nicméně často užívané funkcionály obvykle takovéto vlastnosti mají nebo můžou být snadno změněny, aby tyto vlastnosti získaly.

Invariantní funkcionál: Invariance posunutí znamená, že hodnota funkcionálu se nemění, pokud se funkce posune. Příkladem může být integrál, hodnota mediánu, maximální hodnota funkce atd. Jinými slovy se dá říct, že invariantní funkcionál si volí pořadí nezávisle na posunutí.

Funkcionál Ξ se nazývá *invariant posunutí*, pro jakoukoliv přípustnou funkci ξ

$\Xi(\xi(x+b)) = \Xi(\xi(x))$ pro všechna $b \in \mathrm{R}$ (vlastnost I_1)

Citlivý funkcionál: Funkcionál Z je citlivý na posunutí pro jakoukoliv přijatelnou funkci $\zeta(x)$ $(x \in \mathrm{R})$

$Z(\zeta(x+b)) = Z(\zeta(x)) - b$ pro všechna $b \in \mathrm{R}$ (vlastnost S_1)

Pro obrázky se změněnou velikostí je nezbytná i následující vlastnost:

$Z(\zeta(ax)) = (1/a)Z(\zeta(x))$ pro všechna $a > 0$ (vlastnost s_1)

Obě vlastnosti vyústí ve výslednou vlastnost:

$Z(\zeta(ax+b)) = (1/a)Z(\zeta(x))\text{-}b/a$ (vlastnost S_1s_1)

Další vlastností je

$Z(c\zeta(x)) = Z(\zeta(x))$ pro všechna $c > 0$ (vlastnost s_2)

Vlastnosti s_1 a s_2 ukazují funkcionály citlivé na posunutí. Ty by měli mít konstanty homogenity $\kappa_Z = -1$a $\lambda_Z = 0$. To může být dokázáno tím, že žádné další hodnoty konstant homogenity se nemůžou objevit v konjunkci s vlastností (S_1).

Kritickými parametry, které charakterizují funkcionál Ξ jsou κ_Ξ a λ_Ξ. Můžeme vytvořit funkcionál požadované hodnoty κ_Ξ a λ_Ξ kombinací jiných funkcionálů. V tabulce si ukážeme jak.

	κ	λ
$\Xi_1\Xi_2$	$\kappa_{\Xi_1} + \kappa_{\Xi_2}$	$\lambda_{\Xi_1} + \lambda_{\Xi_2}$
$(\Xi)^q$	$q\kappa_\Xi$	$q\lambda_\Xi$
Ξ_1/Ξ_2	$\kappa_{\Xi_1} - \kappa_{\Xi_2}$	$\lambda_{\Xi_1} - \lambda_{\Xi_2}$

K dosažení necitlivosti k posunutí, rotaci a změně velikosti budeme volit funkcionáli T, P a Φ s následujícími vlastnostmi:

Kombinace A:

T je invariantní k posunutí a má konstantu homogenity κ_T

P je invariantní k posunutí a má konstanty homogenity κ_P a λ_P

Φ je invariantní k posunutí a má konstanty homogenity κ_Φ a λ_Φ

Potom je trojitý rys spojován vzorcem

$$\Pi_2 = \mu^{-\lambda_\Phi(\kappa_T\lambda_P+\kappa_P)}\Pi_1 \qquad \text{(A)}$$

kde μ je faktor měřítka mezi dvěma obrázky.

Kombinace B:

T je invariantní k posunutí a má konstantu homogenity κ_T

P je citlivý k posunutí s vlast. (s_1) a (s_2) a má konstanty homogenity κ_P a λ_P

Φ je invariantní k posunutí a má konstantu homogenity λ_Φ a není citlivý na první harmonickou funkce. To znamená, že vrací stejný výsledek při použití jak na samotnou funkci ale i na funkci bez první harmonické.

Spojením těchto dvou kombinací je získáme vztahem

$$\Pi_2 = \mu^{\lambda_\Phi}\Pi_1 \qquad \text{(B)}$$

Obě rovnice (A) i (B) můžeme prezentovat jako

$$\Pi_2 = \mu^{\omega}\Pi_1$$

kde ω je definována rozdílně pro rovnici (A) a (B). Když $\omega = 0$, pak trojitý rys bude neměnný k ohledu na natočení, změnu rozměru a posunutí v obrázku. Nicméně podmínka $\omega = 0$ je příliš přísná. A je lepší zvážit rysy s $\omega \neq 0$. Toho se dá dosáhnout umocněním $1/\omega$ a dostat taky rysy $(\Pi_i)^{1/\omega}$ které jsou lineárně závislé na faktoru měřítka μ. Když zvážíme, že poměr dvou takových rysů $(\Pi_1)^{1/\omega}/(\Pi_2)^{1/\omega}$ může vytvořit invariantní rys na změnu měřítka, pak výsledný trojitý rys je nezávislý na změně rozměru, rotaci a posunutí. Více v [7].

Čím víc tříd objektů se musí identifikovat, tím víc rysů je potřeba, zejména pokud jsou objekty předmětem složitých transformací a jejích výskyt se může významně měnit. Metoda trojice rysů dovoluje vytvoření mnoha rysu velmi jednoduše. Pokud se pro každou fázi použije 10 funkcionálů (např. 10 *T* funkcionálů, 10 *P* funkcionálů a 10 *Φ* funkcionálů) pak lze zkonstruovat 10x10x10 = 1000 rysů. Tyto rysy nemají fyzicky nic do činění s lidským vnímáním, ale mají tu správnou matematickou vlastnost, která jim umožní rozlišit objekty pro určitou skupinu transformací.

3.10.2 Autentifikační systém využívající trasové transformace

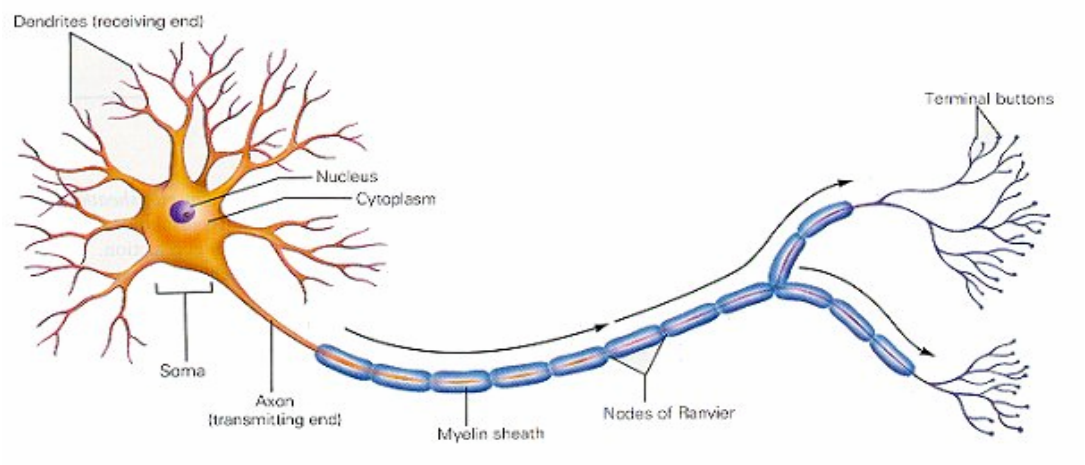
Velmi zajímá a slibná metoda (avšak výpočetně náročná) založena na 3 různých trasových transformacích. Využívá se maskovací trasové transformace (MTT), tvarové (shape) trasové transformace (STT) a vážené (weighted) trasové transformace (WTT) pro rozpoznávání obličejů v autentifikačním systému. Nejprve se transformuje prostor obrázku do prostoru trasové transformace pro vytvoření MTT. Dalším krokem je vytvoření WTT pomocí identifikace bodů MTT, které mají podobné hodnoty nezávislé na odchylkách vnitřní třídy. Dále se ořízne MTT na určitý práh a vyextrahují se hrany prahových oblastí k vytvoření určitých tvarů, které charakterizují osobu. Tuto charakteristiku značíme STT. Proto je každá osoba v databázi reprezentována svojí WTT a STT.

Autoři této metody odhadují odlišnost mezi dvěma tvary pomocí nové míry, Hausdorffovy vzdálenosti. Posílení učení je použito pro hledání optimálních parametrů pro algoritmus. Tvary a rysy z MTT jsou poté integrovány algoritmem kombinačního

klasifikátoru na rozhodovací úroveň. Tento systém byl testován na databází XM2VTS obsahující 2360 snímků obličejů a dosáhl celkového výskytu chyb (Total Error Rate – TER) 0.18%, což bylo do té doby nejlepší hodnocení ze všech publikovaných metod, které použili stejnou databázi a stejný způsob ohodnocení. [6]

3.11 Neuronové sítě

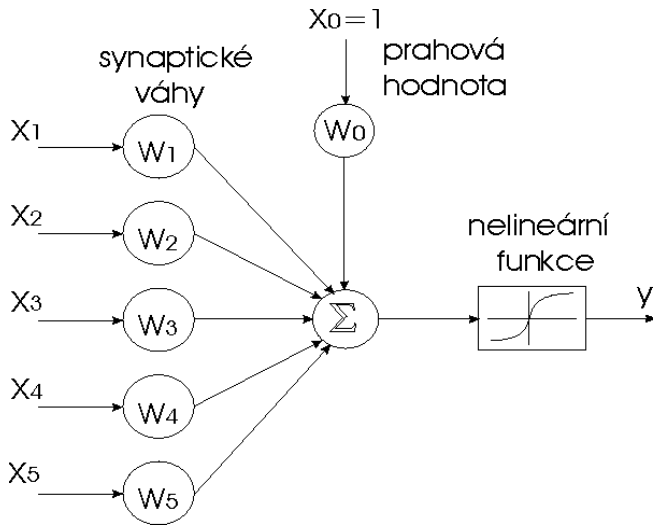
Neuronové sítě jsou inspirovány skutečnými neuronovými sítěmi v lidském organismu. Snahou je vytvořit model skutečného neuronu viz. obr. 20.



Obr. 20 Ukázka biologického neuronu.

Důležitým prvkem neuronových sítí je jednak jádro, které vyhodnocuje vstupní vzruchy a rozhoduje, jaký vzruch vyšle dál, ale i tzv. synapse, které zajišťují propojení s ostatními neurony. V počítačovém světě je jádro neuronu nahrazeno „přenosovou“ (aktivační) funkcí, která rozhoduje, jaký bude mít neuron výstup na základě hodnot vstupujících do neuronu. Výstup jednoho neuronu bývá často vstupem pro další neuron nebo může být přímo výstupní hodnotou. V počítačovém světě se používají zejména neuronové sítě s dopředným řetězením, pro svoji jednoduchost co do implementace, tak tréninku. Pokud ale vytvoříme propojení zpětné, realizuje se tím krátkodobá paměť.

Na následujícím obrázku (obr. 21) si můžeme prohlédnout matematický model počítačového neuronu, který obsahuje jak vážené vstupy, tak sumační člen s prahovou hodnotou a aktivační funkci v tomto případě označenou jako nelineární funkce.[4][20]

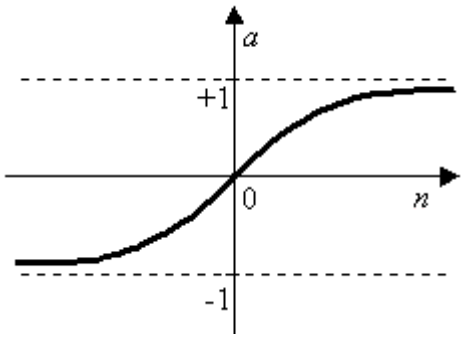


Obr. 21 Jednoduché schéma základního neuronu.

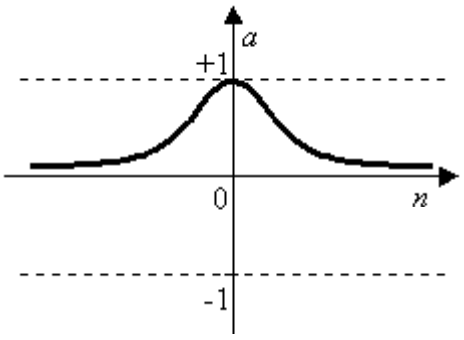
3.11.1 Aktivační funkce

Každý neuron má v jádře aktivační funkci, která rozhoduje o hodnotě výstupu na základě vstupních hodnot.

Aktivačních funkcí je celá řada. Patří sem například *silně omezená aktivační funkce, symetrická silně omezená aktivační funkce, logsigmoidální aktivační funkce, tansigmoidální aktivační funkce, lineární aktivační funkce, aktivační funkce radiálního základu,* atd.



Obr. 22 Tansigmoidální aktivační funkce.



Obr. 23 Aktivační funkce radiálního základu.

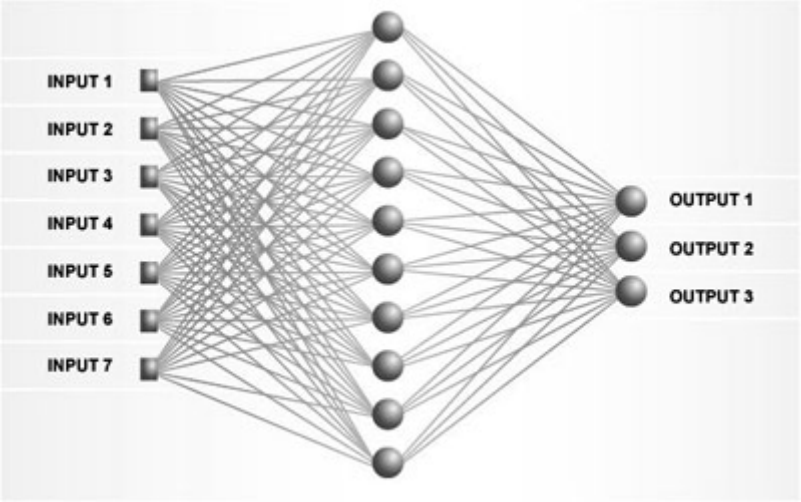
3.11.2 Topologie neuronových sítí

Neuronové sítě jsou tvořeny mnoha neurony, které tvoří jednotlivé vrstvy. Jednoduché neuronové sítě mají např. jen jednu skrytou vrstvu.

Vstupní vrstva – zajišťuje načtení vstupních dat do neuronové sítě.

Skrytá vrstva – obsahuje neurony se zvolenou aktivační funkcí. Zde se odehrává většina logiky. Skrytá vrstva může být tvořena z více vrstev. Tyto vrstvy nejsou z venku viditelné.

Výstupní vrstva – na výstupech této vrstvy čteme výsledek prováděné operace.



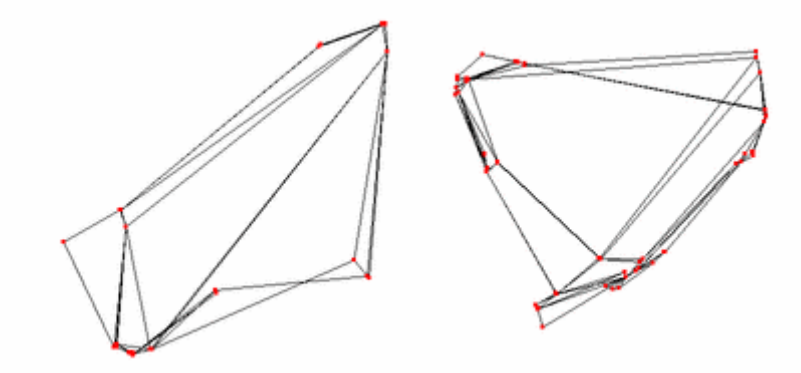
Obr. 24 Ilustrační obrázek topologie neuronové sítě.

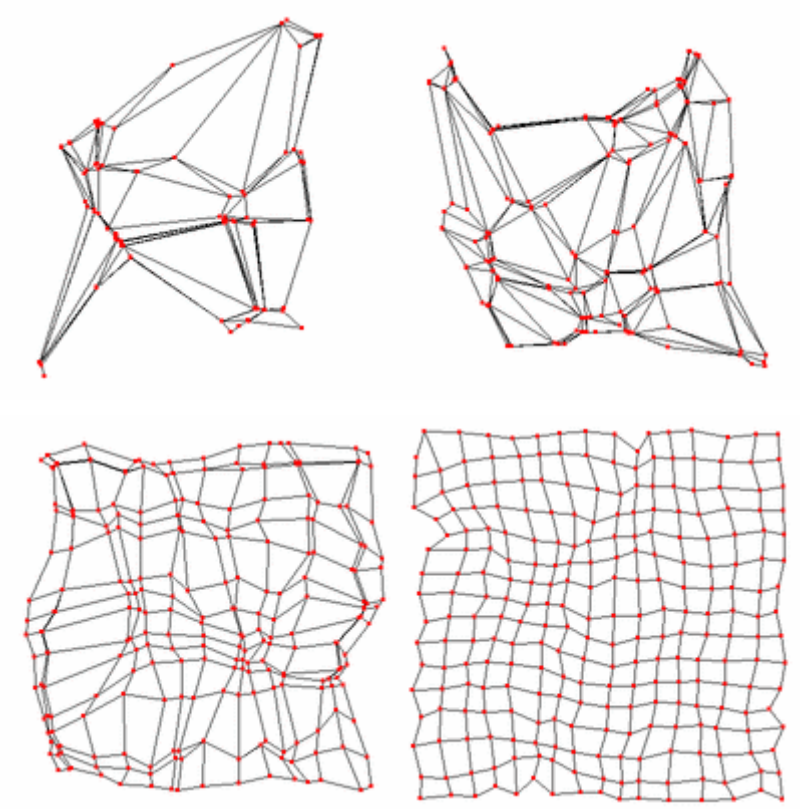
3.11.3 Druhy neuronových sítí

Podle topologie a typu aktivační vrstvy získáváme různé druhy neuronových sítí vhodných pro různé účely použití. Vznikají tak např. sítě *Adaline, Madaline*, *BPN*, *RBF*, *Hopfieldova, SOM,* a mnoho dalších.

RBF neuronová síť využívá ve skryté vrstvě neuronů s aktivační funkcí radiálního základu. Úspěšně tak řeší úlohy, které nejsou lineárně separabilní. Zejména se hodí na problematiku klasifikace, ale i řadu jiných funkcí.

SOM – z anglického Self Organizing Map jsou sítě, které se umí samostatně uspořádat, resp. uspořádat vstupní hodnoty.





Obr. 25 Příklad chování SOM sítě ve 2D prostoru při použití euklidovské vzdálenosti jako klasifikace.

3.11.4 Učení neuronové sítě

Neuronová síť se učí buď s učitelem nebo bez učitele. Učitel neuronové síti předhazuje vstupní data a k nim odpovídající data výstupní. Důležitým parametrem správného naučení je tedy tréninková množina, která nesmí být příliš malá, aby neuronová síť uměla vyhodnotit všechny požadované situace. Nejznámější metodou pro učení s učitelem je metoda „back-propagation“. Při učení bez učitele se síti předhazují pouze vstupy a síť se pro každý vstupní vzorek snaží aktivovat jeden výstup tak, aby nebyl aktivován jiným vstupním vzorkem.

3.12 Genetické algoritmy

Je heuristický přístup, využívající principů evoluční biologie k řešení složitých problémů, pro které nejde vytvořit přesný algoritmus řešení. Základními principy genetického algoritmu jsou dědičnost, mutace a křížení.[21]

3.12.1 Jedinec

Je nositelem chromozomu, prakticky řetězce jedniček a nul, které popisují nějaké řešení. Z většího počtu jedinců se skládá populace.

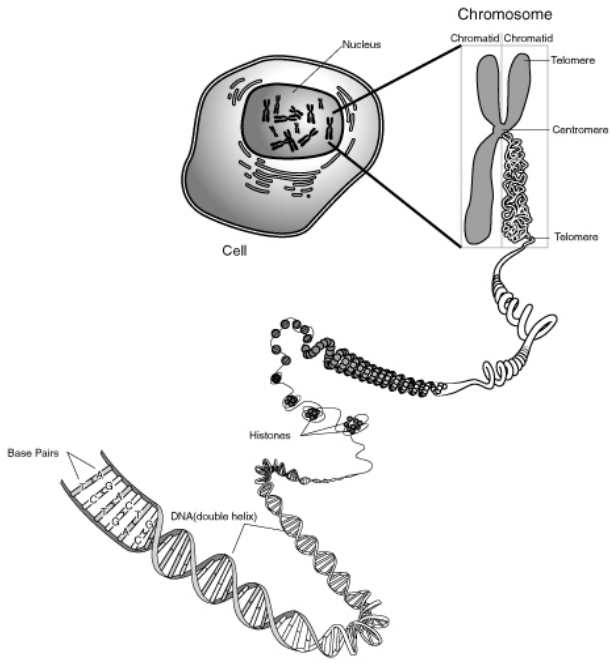
3.12.2 Populace

Aby byl genetický algoritmus úspěšný, musíme vytvořit populaci jedinců (možných řešení) tak, aby byl dostatečný prostor pro operace křížení. S rostoucí populací však rostou i nároky na paměť a tak populaci odhadujeme úměrně složitosti

problému. Populace je tedy množinou jedinců, kteří jsou ohodnoceni vhodnou fitnes funkcí. Jedinci s nejvyšším ohodnocením se dostávají do další generace.

3.12.3 Chromozom

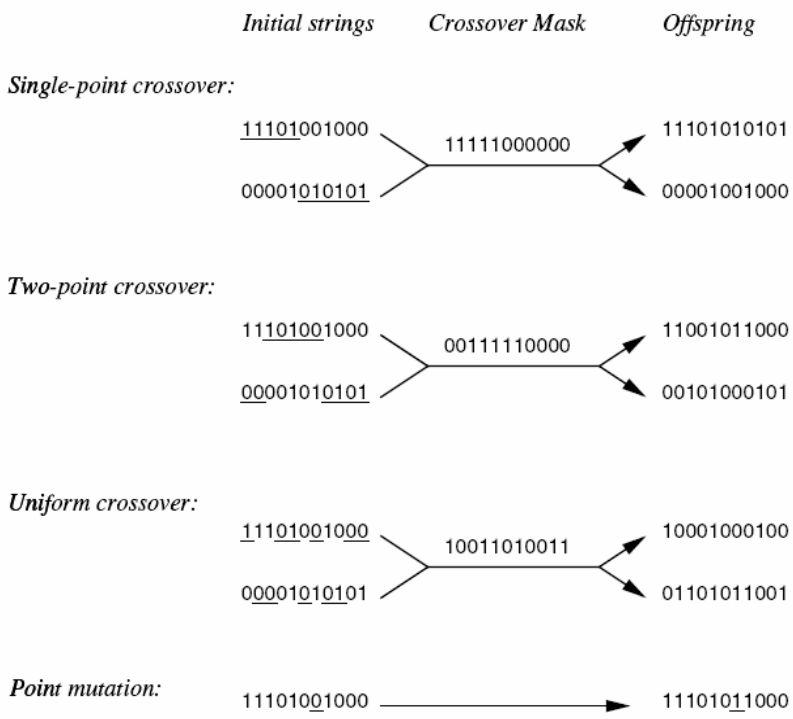
Může být vyjádřen řetězcem jedniček a nul, ale i seznamem čísel nebo grafem objektů. Záleží jenom na potřebě popisu hledaného řešení.



Obr. 26 Ukázka biologického chromozomu.

3.12.4 Křížení

Je jednoduchá operace nad chromozomy, při které se vyměňují informace v jednotlivých chromozomech. Implementace křížení může být různá. V podstatě může docházet křížení v jednom bodě, kdy se chromozom v tomto místě rozdělí a dojde k výměně jedné části za jinou. K takovému rozdělení může dojít i ve více bodech, pak se jedná o vícebodové křížení. Dalším typem křížení je způsob kdy se o jednotlivé části řetězce, zda dojde k prohození, rozhoduje náhodně a není tak daný přesný počet bodů křížení. Jednotlivý typy křížení jsou na následujícím obrázku (obr. 27).



Obr. 27 Přehled různých typů křížení.

3.12.5 Mutace

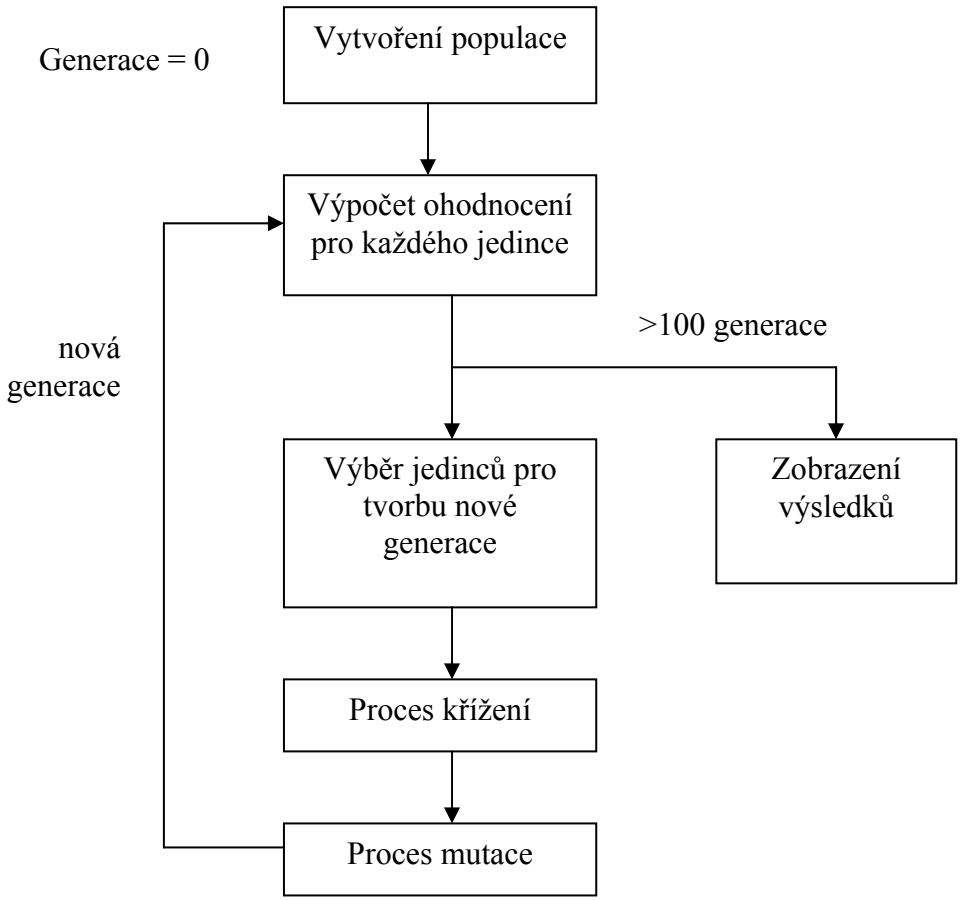
Při určité pravděpodobnosti může dojít v chromozomu k mutaci, což je náhodná změna hodnoty v řetězci.

Např. řetězec (11001**0**01) zmutuje na (11001**1**01).

Mutace je důležitá pro jemnou změnu řešení, kdy řešení je velmi blízko hledanému optimu a křížení s jiným jedincem toto řešení pouze degraduje. Záleží ovšem na způsobu, jakým křížení probíhá, a podle toho se nastavují různé pravděpodobnosti mutace.

3.12.6 Algoritmus

Začneme s vytvořením první generace, která je tvořena jedinci s náhodně generovanými chromosomy. Pro jedince v populaci vytvoříme tabulku s ohodnocením a vybíráme jedince, které křížíme a tvoříme tak novou generaci. Při výběru jedinců můžeme využít např. náhodný výběr, výběr ruletou nebo turnajový výběr. Po výběru se provede již zmíněné křížení a z rodičů vzniknou nový potomci. Po křížení dochází i s určitou pravděpodobností k mutaci. Nová generace se tak plní potomky předchozí generace dokud nedosáhne omezení velikosti zvolené populace. Tento proces se opakuje pořád dokola, dokud se nevyčerpá přednastavený počet generací k výpočtu nebo pokud dojde k nalezení řešení, které je velmi blízké optimálnímu. Většinou ale toto optimální řešení neznáme, a proto volíme strategii zastavení výpočtu po určitém počtu generací. Tento postup je znázorněn na následujícím schématu.



Obr. 28 Schéma genetického algoritmu

4 Projekt

V praktické části této diplomové práce bylo snahou implementovat základní metodu a k ní další metodu využívající nové pokrokové technologie pro srovnání. Při výběru nových metod jsem se zaměřil na využití základních znalostí neuronových sítí a genetických algoritmů a jejich použití na vylepšení právě základní metody, kterou je metoda PCA. Tato metoda má velmi slušnou rozpoznávací schopnost nad malou databází, a protože výsledné algoritmy budou právě testovány nad databází obsahující 400 fotografií pro 40 různých lidí, tak je použití této metody na místě.

Prvním srovnávací metodou bude metoda využívající neuronovou síť s radiální bází pro klasifikaci nad PCA daty [16]. Volba středů neuronů a šířek bude volena podle [16], váhy neuronové sítě budou trénovány genetickým algoritmem.

Druhá a poslední srovnávací metoda bude mít opět základ v metodě PCA, ale vlastní vektory nebudou vybírány pouze od největšího, ale k výběru bude použit genetický algoritmus. Jak se v průběhu testovaní dozvíte, tato metoda překonala obě předešlé.

4.1 Implementace algoritmů

Protože se jedná o studii algoritmů, rozhodl jsem se použít pro dobrou čitelnost a další použitelnost zdrojových kódů programovací jazyk c#. Jeho použití mi umožnilo snadnější a rychlejší implementaci a odladění algoritmů.

Původním plánem bylo použít již někým vytvořenou metodu PCA a pouze doimplementovat další metody vhodné k porovnání. Bohužel se mi tato metoda v jazyce c# nepodařila sehnat a tak jsem musel přistoupit k samostatné implementaci již u této metody. Implementace této metody a tím pádem i lepší porozumění, mi pak pomohla navrhnout další dvě metody.

4.1.1 PCA

Princip a popis této metody najdete v teoretické části, takže se zda zaměřím už na přímý popis implementace.

Základem je získání vektoru popisující obrázek. Tento vektor získáme čtením obrázku řádek po řádku a skládáním jednotlivých řádků za sebe. Tento vektor označíme x a bude mít velikost $p = m \times n$, kde m je počet bodů obrázku do šířky a n do výšky. Hodnoty vektorů získáme např. z jasu konkrétního obrazového bodu. Pokud získáme hodnoty 0-255, je dobré tyto hodnoty převést na rozsah 0 až 1. S tímto rozsahem se pracuje při dalších výpočtech a nedochází ke zbytečnému narůstání hodnot při násobení. Tento vektor vytvoříme pro každý snímek z tréninkové sady, která obsahuje různé osoby. Výskytům jedné osoby říkáme třída. Takže v tréninkové sadě může být například 5 osob a z toho každá osoba má v tréninkové sadě 3 fotky. Tzn., že máme tréninkovou sadu o 15 snímcích a 5 třídách. Odtud získáme sadu vektorů

$X = \{x_1, x_2, \ldots x_i\}$, kde $i = 15$.

Dále je potřeba vypočíst průměrný snímek (vektor), kterým získané vektory vycentrujeme. Vycentrování spočívá v jednoduchém odečtení tohoto průměrného snímku od každého snímku (vektoru) v sadě a získáme tak sadu vektoru X_0. Tedy vypočteme nejdřív průměrný vektor

$X_{avg} = \frac{1}{R}\sum_{i=1}^{R} x_i$, kde R je počet snímků v tréninkové sadě.

a dále provedeme vycentrování snímků

$x_{0i} = x_i - x_{avgi}$ pro i = 1,2,…R;

Dalším důležitým bodem je výpočet kovarianční matice *C*. Toho dosáhneme jednoduchým vynásobením matice vytvořené centrovanými vektory x_0 a její transponovanou podobou.

$C = X_0 \times X_0^{\mathrm{T}}$

,kde X_0 má rozměr $p \times R$, kde *p* je počet bodů obrázku a *R* je počet snímků v tréninkové sadě. Kovarianční matice pak bude mít rozměr $R \times R$, což je výhodné pokud $R << p$. Jinými slovy, pokud počet obrázků v tréninkové sadě je menší než počet bodů obrázku. To v našem případě bude vždy platit.

Kovarianční matice nám nyní poslouží k výpočtu vlastních vektorů a vlastních hodnot. Vlastní hodnoty nejsou nijak důležité, jenom se podle nich budou třídit vlastní vektory tak, abychom byli schopni vybrat nejvýznamnější vlastní vektory. Nejdůležitější jsou právě získané vlastní vektory, ze kterých si vytvoříme matici, kterou posléze použijeme k transformaci tréninkových snímků, ale při vyhodnocování i snímků testovacích. Tato matice nám totiž redukuje velikost vektoru snímku z *p* na $r < R$, tedy z počtu bodů snímku na velikost menší než počet snímků v tréninkové sadě. Volba r záleží jenom na nás a většinou při výběru postačí první 2/3 vlastních vektorů (seřazených podle velikosti vlastních hodnot). Tato redukce nám velmi usnadňuje následné prohledávání projekčního prostoru při procesu rozpoznávání. Který se zredukoval z *p*-dimenzionálního prostoru na *r*-dimenzionální prostor.

Právě výpočet vlastních vektorů není vůbec triviální záležitost a tady jsem použil volně dostupné matematické knihovny LAPACK [22], která optimalizovanou metodou dokáže z kovarianční matice získat a setřídit vlastní vektory. Z těchto vlastních vektorů si vybereme ty, které nás nejvíce zajímají, a vytvoříme si z nich tzv. transformační matici a označíme si ji jako T. Každý vlastní vektor má rozměr R, takže z 5 vlastních vektorů poskládáme transformační matici o velikosti R × 5.

Transformované vektory snímků (označíme *y*) pak získáme jednoduchým roznásobením centrovaného vektoru snímku s maticí obsahující vybrané vlastní vektory.

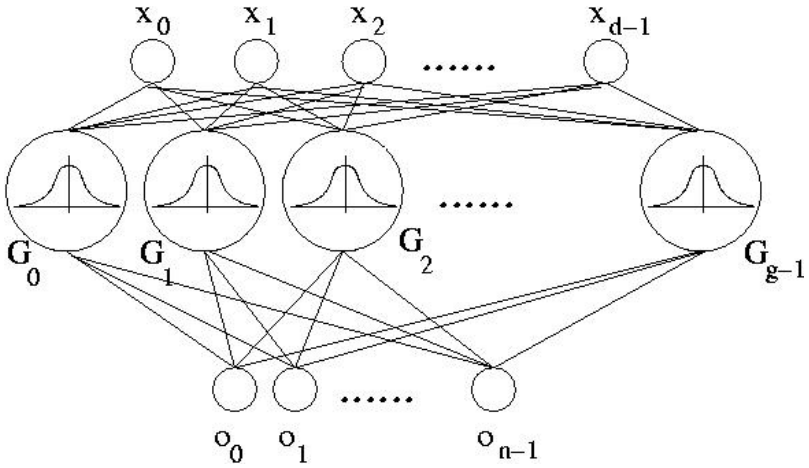
$y_i = x_i \times T$

získáme tak redukované vektory $Y=\{y_1, y_2, \ldots, y_i\}$, kde i = R.

Následný proces rozpoznávání spočívá v načtení testovaného snímku, vytvoření vektoru a jeho vycentrování odečtením průměrného snímku z tréninkové sady, transformací do *r*-dimenzionálního prostoru pomocí transformační matice T a následným výpočtem např. euklidovské vzdálenosti ke všem tréninkovým snímkům a vybrání toho nejbližšího a tím i nejpodobnějšího snímku. Podle třídy, ze které byl nejpodobnější tréninkový snímek vybrán, se určí osoba na snímku.

4.1.2 PCA + RBF

Metoda PCA se zde využívá pro redukci prostoru, pro rozpoznávání se využívá natrénovaná RBF neuronová síť. Po PCA transformaci se jednotlivé redukované vektory snímků přivádí na vstup neuronové sítě. Použitá neuronová síť se skládá ze vstupní vrstvy s počtem vstupů odpovídajících dimenzi redukovaných vektorů snímků *r*. Dále z jedné skryté vrstvy obsahující RBF neurony o počtu 120 neuronů pro 40 tříd, tzn. 3 neurony na jednu třídu. Poslední vrstvou je vrstva výstupní, která obsahuje 40 sumačních neuronů. Schéma takové sítě si ukážeme na obr. 30.



Obr. 29 Schématické znázornění RBF neuronové sítě.

Aktivační funkce skryté neuronové vrstvy mají následující tvar

$$G_j(x_i) = e^{-\frac{\|x_i - c_j\|^2}{2\sigma_j^2}}, j = 1, 2, \ldots, g; i = 1, 2, \ldots, d$$

kde c_j a σ_j jsou středy a šířky neuronů skryté vrstvy a g a d jsou počet neuronů a počet vstupů (dimenze). Výstupní vrstva má v našem případě tolik neuronů, kolik je tříd v trénovací množině snímků. Výstup k-tého výstupního neuronu je dán vztahem

$$Z_{ik} = \sum_{j=1}^{g} G_j(x_i) w_{kj} + b_k w_k \;, k = 1, 2, \ldots, n$$

kde w_{kj} je váha spojení mezi j-tým neuronem skryté vrstvy a k-tým výstupním neuronem a n je počet výstupních neuronů.

Po vytvoření neuronové sítě je nejdůležitější částí trénink. Podle vědeckého článku [16], kde se můžete dočíst přesnější postup, jsem nastavil středy a šířky neuronů. Ve zkratce je postup následující. Pro tréninkové snímky, které obsahují např. 5 snímků na třídu, se zvolí počet neuronů, které budou tuto třídu zastupovat. Poté se hledá nejbližší euklidovská vzdálenost mezi snímky jedné třídy v projekčním prostoru a po výběru dvou nejbližších, se jejich středy zprůměrňují, a ze dvou snímků vznikne jeden střed. To se opakuje, dokud není počet snímků ve třídě redukován na předem zvolený počet neuronů.

Dalším krokem je nastavení šířek těchto neuronů. Mezi jednotlivými třídami dochází k překryvu a právě míra tohoto překryvu je potřeba nastavit optimálně tak, abychom využili generalizační schopnosti neuronové sítě a zároveň dosáhli dobrého rozlišení tříd. Tzn. nastavit šíři neuronů na maximum v rámci třídy tak, aby vzdálenost

mezi jinými třídami byla dostatečná. Algoritmus ve své podstatě pracuje tak, že vypočítá průměrné středy jednotlivých tříd a získává jak maximální vzdálenosti uvnitř konkrétních tříd, ale i minimální vzdálenost od ostatních tříd. Z těchto vzdáleností vždy pro konkrétní třídu vybere tu větší a nastaví ji příslušným neuronům třídy.

Závěrečným krokem tréninku je nastavení vah neuronové sítě. Zde je využito genetického algoritmu s netradičním přednastavením hodnot vah. Standardně se v genetických algoritmech využívají náhodné hodnoty pro vytvoření populace, v tomto případě jsem využil heuristický přístup a ze znalosti postupu nastavení neuronů přednastavil hodnoty vah pro jednotlivé výstupy tak, aby upřednostňovali neurony ze skryté vrstvy náležící stejné třídě. Výhodou genetického algoritmu je, že s největší pravděpodobností neuvázne v lokálním maximu, takže pokud nastavení není optimální, tak GA toto nastavení nejméně vylepší.

4.1.3 EPCA

Evolutionary PCA je metoda inspirovaná EDA + Full LDA [17]. Jak bylo zjištěno, tak ne všechny vlastní vektory získané PCA metodou mají pozitivní vliv na rozpoznání obličejů a tak by bylo dobré zjistit, které z vlastních vektorů mají dobrý vliv a které mají vliv spíše negativní. Pro takto složitou úlohu je použití genetického algoritmu nasnadě. Využil jsem tedy genetického algoritmu, který zjišťuje nejvhodnější kombinaci vlastních vektorů pro transformaci do projekčního prostoru. Úkolem fitnes funkce GA bylo maximalizovat vzdálenosti mezi třídami a zároveň minimalizovat vzdálenosti uvnitř třídy. Toho bylo dosaženo maximalizací poměru

$$F = \frac{dist_{in}}{dist_b},$$

kde F je hodnota fitness funkce, $dist_{in}$ je průměrná vzdálenost tréninkových snímku od jejich průměrné hodnoty uvnitř třídy a $dist_b$ je průměrná nejbližší vzdálenost mezi třídami.

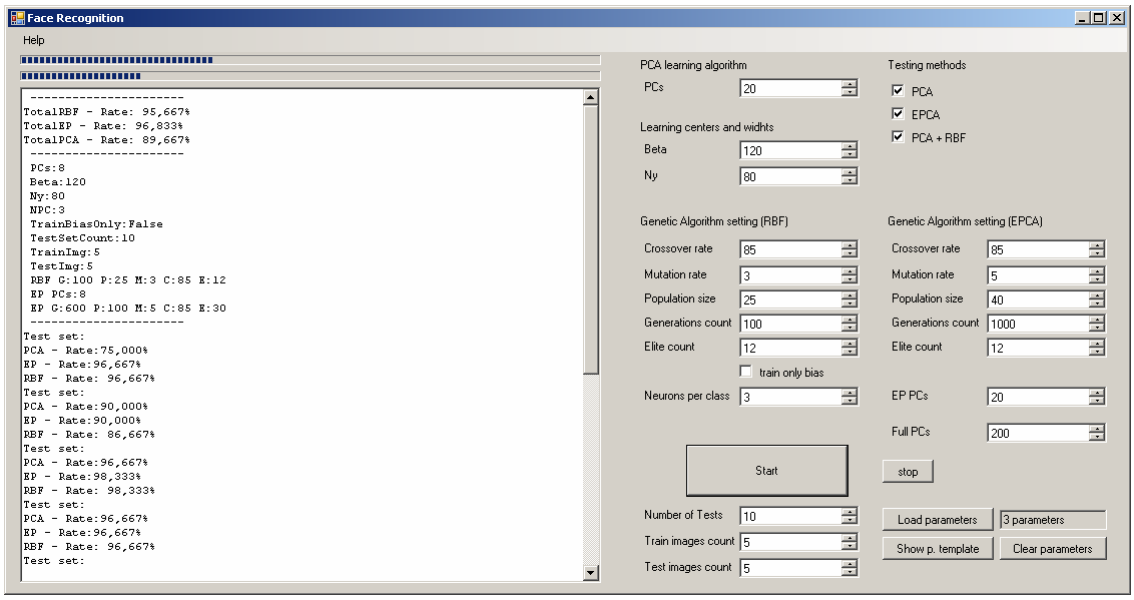
Přestože je tato metoda velmi jednoduchá, její výsledky jsou z testovaných metod nejlepší. Dalo by se říct, že krása této metody vychází z její jednoduchosti.

4.2 Aplikace

Pro možnosti testování byla vytvořena aplikace, která umožňuje spouštět testy s libovolným nastavením. Vstupní snímky jsou automaticky převedeny do odstínu šedi a zmenšeny na velikost 64 x 64 bodů. Tyto snímky však musí být vhodně označeny, aby aplikace byla schopná rozpoznat, které osoby patří do stejné třídy. Kromě srovnávacích testů všech metod lze vybrat pouze požadované metody a šetřit čas při testování citlivosti vstupních parametrů pro konkrétní metodu.

Pro funkční spuštění aplikace je potřeba mít na počítači nainstalovaný Microsoft .NET Framework ve verzi 3.5. Ke změnám velikosti obrázků byla použita volně dostupná knihovna AForge.Net [23].

Aplikace je zaměřena čistě na funkčnost, proto nečekejte žádné grafické excesy.



Obr. 30 Snímek aplikace.

4.3 Testování

Pro testování byla implementována metoda, která se stará o načtení testované databáze, rozřazení obrázků do jednotlivých tříd a generovaní testovacích sad podle zvolených parametrů. V programu je možno zvolit kolik testů má proběhnout a kolik obrázku ze třídy má být použito pro trénink a kolik pro testování.

Tato testovací sada je zastřešena dávkovým zpracováním, kdy se potřebné parametry mohou načíst ze seznamu a tak spustit různé testy za sebou, aniž by člověk musel přenastavovat ručně parametry výpočtu.

4.3.1 Databáze

Pro testování byla použita databáze snímků ORL, která se často používá pro porovnání experimentálních metod. Databáze obsahuje 400 fotografií 40 různých osob. Snímky jsou v odstínech šedi a mají rozlišení 92 x 112 bodů. Databáze je volně dostupná ke stažení na internetu [18].

Obr. 31 Ukázka snímků jedné třídy databáze ORL

Snímky jsou pořízeny v rozdílných časech a s různým osvětlením. Stejně tak pohled není zcela přímý. Lehce se střídají i emoční výrazy a některé obličeje obsahují varianty s brýlemi a bez brýlí.

4.3.2 Druhy testů

Nejprve byla provedena citlivostní analýza parametrů genetických algoritmů a parametrů pro RBF neuronovou síť. Následně byly provedeny srovnávací testy tří implementovaných metod pro různá nastavení počtu hlavních složek. Sady snímků byly pro trénink a pro testování vybírány náhodně podle zvolených poměrů. Poměry byly voleny 3:7, 5:5, 7:3, 9:1 (tréninkové snímky : testovací snímky). Všechny 3 testované

metody běžely nad stejnou sadou dat a tyto sady byly pro každý test 20x generovány pro potlačení vlivu náhodného výběru obličejů do tréninkové a testovací sady.

4.3.3 Výsledky testů citlivostní analýzy

Testy citlivostní analýzy GA probíhaly v některých případech pouze 10x a proto některé hodnoty dost oscilovaly. Vliv náhodnosti výběru tak nebyl příliš potlačen.

Jedním z mnoha testů bylo zjišťování vlivu velikosti populace a počtu generací pro výběr vlastních vektorů v metodě EPCA.

Při populaci o velikosti 100 nastává moment nasycení v okolí generace 200. To však nemusí být zcela přesná hodnota, protože další hodnoty u vyšších generací stále lehce oscilovaly. Ukázkou této oscilace byl rozdíl mezi 0. a 10. generací, kde rozpoznávací ohodnocení dosahovalo vyššího hodnocení pro menší počet generací. Neznamená to, že by genetický algoritmus fungoval špatně, ale testovací sady pro jednotlivé testy se náhodně vybraly s výhodnějším složením pro případ s 0. generací.

Při testu s populací 200 se zlomový bod nachází u 100. generace, což odpovídá 200. generaci u populace o velikosti 100 jedinců. Násobek 200 x 100 nebo 100 x 200 odpovídá stejnému „prohledávanému“ prostoru. Jeví se tedy jako nepodstatné, zda se použije menší populace s větším počtem generací nebo se zvolí populace větší a vystačíme si tak s menším počtem generací. Pro poslední test o populaci 500 jedinců by se tedy měla oblast nasycení objevit u 40. generace, což se také potvrdilo.

Dalším testovaným parametrem, který může sehrát svou roli při počtu potřebných generací na vyřešení optimalizačního úkolu, je pravděpodobnost mutace. V některých případech může nesprávný odhad vést ke zbytečnému zpomalení konvergence.

Podle provedených testů vyplynulo, že pokud se budeme držet v rozmezí 2 - 10%, neměla by mít pravděpodobnost mutace zásadní vliv na průběh hledání optima. Proto se u srovnávacích testů použijeme hodnotu 5%.

Stejně jako odzkoušení GA bylo zapotřebí odladit neuronovou síť. Zde je problematika podstatně složitější. Jednak se musí odladit samotné nastavení neuronové sítě, ale i nastavení GA který síť natrénuje. Bohužel přes různorodé experimentální testy, které jsem prováděl na částečných datech, se mi nepodařilo síť odladit tak, aby nad celou databází splňovala požadované předpoklady. Možná, že byla pouze chyba v předpokladech nebo se doopravdy nepodařilo nastavit všechny parametry tak, aby metoda dosáhla lepších výsledků. Za povšimnutí stojí parametr určující počet neuronů zastupující jednu třídu.

Je poměrně zajímavé, že podle testů (5 tréninkových a 5 testovacích snímků na třídu) jsou 3 neurony na třídu nejoptimálnějším nastavením. Menší počet neuronů již hůře popisuje danou třídu a u většího počtu neuronů se už jejich působiště asi příliš překrývá a procento úspěšnosti se tak zmenšuje. Toto nastavení bude využito také ve srovnávacích testech.

4.3.4 Výsledky srovnávacích testů

Jak už bylo zmíněno, srovnávací testy probíhaly pro každou sadu 20x pro různě generované tréninkové a testovací skupiny a s různým poměrem tréninkové sady a testovací sady. Testy zjišťovali citlivost na změnu počtu hlavních složek a maximální dosažitelnou průměrnou rozpoznávací hodnotu. Metoda EPCA by měla být výrazně méně citlivá na počet hlavních složek než metoda PCA, protože si do své množiny

hlavních složek vybírá jenom ty vhodné. Tento předpoklad se potvrdil na všech vykonaných testech.

Parametry pro všechny srovnávací testy byly stejné, kromě poměru snímků v testovací sadě.

Počet tříd testované databáze: 40

Počet redukovaných hlavních složek: 10-110

Počet neuronů na třídu: 3

Počet opakování testu: 20

Genetický algoritmus pro trénink RBF:

Počet generací: 100

Velikost populace: 25

Pravděpodobnost křížení 85%

Pravděpodobnost mutace: 3%

Genetický algoritmus EPCA:

Počet generací: 600

Velikost populace: 100

Pravděpodobnost křížení 85%

Pravděpodobnost mutace: 5%

Test 3:7

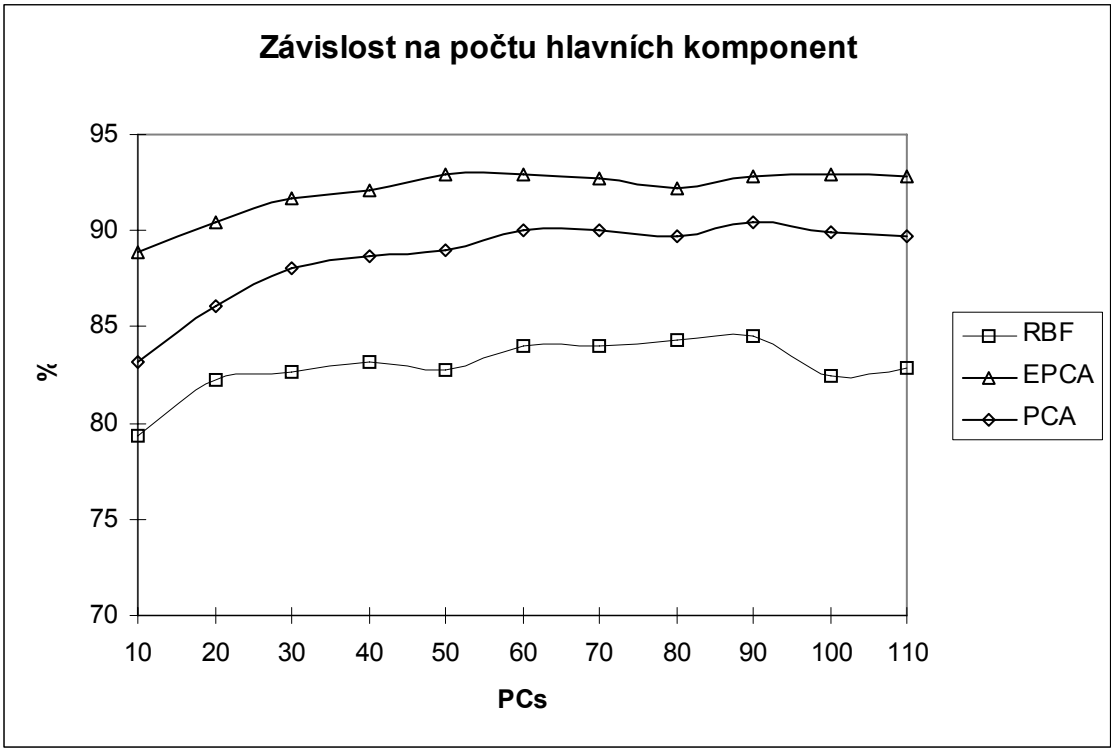
RBF – průměr 75,497 (max. 77,25)

EPCA – průměr 84,203 (max. 85,036)

PCA – průměr 80,603 (max. 82,518)

V tomto testu jde krásně vidět, jak algoritmus EPCA vybírá vhodné komponenty, které popisují rozdílnost tříd a při počtu 10 hlavních komponent překonává původní metodu PCA o více než 9%. Od počtu 30 komponent se situace pro metodu PCA zlepšuje, ale po celý průběh si metoda EPCA udržuje uctivý náskok a metodu PCA průměrně překonává o 4%. Metoda PCA + RBF od počátku ztrácí na obě metody a tak v průměru zaostává za EPCA o 9% a za PCA 5%.

Test 5:5



graf 1 Srovnávací test 5:5.

RBF – průměr 82,943 (max. 84,5)

EPCA – průměr 92,027 (max. 92,925)

PCA – průměr 88,627 (max. 90,475)

Tento test se pro srovnávání používá nejčastěji. Tvar průběhu grafu je stejný jak pro test 3:7, proto zde tento graf není uveden. Pouze nastal rozdíl v rozpoznávací úrovni, kde metoda EPCA dosahuje průměrné hodnoty 92% a překonává tak původní PCA o úctyhodné 3,4%. Metoda RBF si oproti předchozímu stavu polepšila o pěkných 7,5% přestože počet neuronů zůstal stejný. Je to způsobeno tím, že se našli lepší centra neuronů vypočítané z výskytů tréninkových snímků. Nadále však pro tento rozsah dat metoda RBF zůstává outsiderem, který by si zasloužil podrobnější prozkoumání.

Test 7:3

RBF – průměr 85,102 (max. 86,875)

EPCA – průměr 94,648 (max. 96,5)

PCA – průměr 92,878 (max. 94,833)

Vlivem zvětšujícího se poměru tréninkových snímků vůči testovacím, se prostor na vylepšení PCA začíná zmenšovat, ale přesto metoda EPCA ve všech případech stále překonává původní PCA v průměru o necelé 2%. Metoda RBF se už výrazně nezlepšila a pravděpodobně může hrát omezení počtu neuronů na třídu svou roli, kdy využívá pouze 3 projekčních bodů namísto 7 u metody PCA a EPCA v tomto případě.

Test 9:1

RBF – průměr 86,841 (max. 90,125)

EPCA – průměr 96,25 (max. 97,5)

PCA – průměr 95,227 (max. 97,5)

Při poměru 9:1 už je prostor pro vylepšení minimální. Výhodné vlastnosti EPCA lze pozorovat pouze na počátečních hodnotách PCs do 30, kdy chování PCA vylepšuje, nicméně zbytek testů už byl silně závislý na výběru kandidátů testovací množiny a tak je náskok EPCA nepatrný (průměrně o 1%).

Všechny zmiňované testy probíhaly nad celou databází, pokud jsme ale vybrali vzorek například o pouhých 12 třídách, situace se výrazně změnila, ale jenom u metody PCA + RBF kde se při testu 5:5 vyšplhala úspěšnost této metody na hodnoty kolem 96% oproti 90% u metody PCA. Metoda EPCA rovněž dosahovala na redukované množině průměrně kolem 95%.

4.4 Zhodnocení výsledků

Ze srovnávacích testů jasně vyplývá favorit, kterým je metoda EPCA. Ve všech testech překonává ostatní metody a patří jí zaslouženě vavříny slávy. Na druhou stranu i metoda PCA+RBF přinesla něco nového, když prokázala své generalizační schopnosti, kde za použití 3 neuronů, jako reprezentaci třídy, dosahovala v testu 7:3 v průměru 85%, což by se mohlo porovnat s metodami PCA a EPCE v testu 3:7, kde tyto metody rovněž byly reprezentovány pouze 3 výskyty a dosahovaly průměrného výsledku 80% resp. 84%. To naznačuje, že RBF síť skrývá daleko větší potenciál a uzpůsobením tréninku a volbou vhodnějších parametrů této sítě bychom se mohli dostat ještě o kus dál. Víceméně tomu napovídá test na redukované množině s výsledkem přes 95% při poměru 5:5. To však může být námětem další práce, kdy výkon hardwaru poroste a bude tak více prostoru pro experimentování.

Ve srovnání s ostatními metodami samotná PCA metoda podávala stabilní výsledky a na databázi tak prokázala poměrně dobré rozlišovací schopnosti, což byl pro tuto metodu předpoklad, který se potvrdil.

Pro praktické nasazení tak z testovaných metod rozhodně doporučuji použití metody EPCA, která vyšla v testování nejlépe s 92% rozpoznávací úspěšností a jejíž trénink není nikterak časově náročný, a přesto metodu PCA překonává ve všech provedených testech.

Metodu PCA s RBF neuronovou sítí bych doporučil prozatím pouze pro vědecké experimenty. Jednak z důvodu doposud nízké rozpoznávací schopnosti nad celou databází, kde je zapotřebí zapracovat na nastavení všech parametrů, ale zejména pro časovou náročnost tréninku této sítě, která velmi prudce roste s rostoucím počtem tříd. Po natrénování jsou však klasifikační časy srovnatelné s ostatními metodami.

5 Závěr

5.1 Shrnutí použitých metod

V průběhu vypracování práce bylo zapotřebí se seznámit s nepřeberným množstvím metod pro práci s obrazem a spoustou algoritmů, které mnohdy slibovaly nejisté výsledky. Často jsem narazil na zajímavé metody, které však neposkytovaly dostatek informací pro jejich implementaci a tak se okruh implementovaných metod velmi těžce rozrůstal. Příkladem může být metoda PCA + RBF, která slibovala zajímavé výsledky ve vědecké publikaci, ale po jejím naimplementování se dostavilo spíše zklamání. Na druhou stranu se dostavila radost z elegantního řešení metody EPCA, která vylepšení přinesla nekompromisně ve všech prováděných testech. Dále byla k implementaci zvažována metoda LDA, která se však hodí na větší databáze a na menších ji PCA překonává, proto by její srovnání nepřineslo nic pozitivního a v testech by pravděpodobně propadla[19].

5.2 Použití

Poměrně dobré výsledky již nabádají ke komerčnímu nasazení pro monitoring menších firem nebo domácností. Ve výsledku by tak měl např. vedoucí pracovník představu, jestli už například jeho podřízený, kterého shání, dorazil do práce nebo starostliví rodiče by měli přehled o tom, že jejich dcera nebo syn už jsou ze školy doma. V žádném případě by se nedalo doporučit použití jako bezpečnostního prvku, protože metody 2D snímání jsou velice snadno překonatelné. Nicméně jako informativní způsob použití, popřípadě do zábavního průmyslu, je nasazení předváděných metod možné. Dalším možným způsobem využití by mohlo být vylepšení interakce počítače s uživatelem. Pokud si k počítači sedne osoba, kterou počítač „zná“, tak jí sám nabídne její profil k přihlášení. Stejně tak moderní auta, která si dokáží pamatovat nastavení volantu, sedačky a zrcátek, by mohla automaticky nabídnout změnu profilu po rozpoznání osoby řídící auto. U fotoaparátu by se hodila funkce pro skupinové fotky, kdy fotoaparát spustí spoušť v momentě, kdy se na scéně objeví tvář fotografa. Jak se ukazuje, příležitostí na využití se zdá poměrně dost a bude záležet jenom na firmách, zda se toho chytí a poskytnou svým zákazníkům nové funkce či lepší pohodlí.

6 Použitá literatura

[1] CVL. Face recognition homepage – algorithms [online]. 2005, last modified: 8th of October 2008 [cit.2008-9-7]. Dostupné z: < http://www.face-rec.org/algorithms/ >

[2] BELHUMEUR N. Peter, HESPANHA P. João, KRIEGMAN J. David.Eigenfaces vs. Fisherfaces: Recognition using vlase specific linear projection.New Haven,1997. 10 s. IEEE transactions on pattern analysis and machine intelligence, vol. 19, no. 7, July 1997

[3] CHELLAPPA R., WILSON C., SIROHEY S. *Human and Machina Recognition of Faces: A Survey*. Proc. IEEE, vol. 83, no. 5, pp. 705-740, 1995.

[4] ELEYAN A., *DEMIREL H. PCA and LDA based Neural Network for Human Face Recognition*.2007 [pdf]. Dostupné z: < http://s.i-techonline.com/Book/Face-Recognition/ISBN978-3-902613-03-5- fr06.pdf>

[5] DUDA R., HART P. *Pattern Clasification and Scene Analysis*. New York: Wiley, 1973.

[6] SRISUK S., PETROU M., KURUTACH W., KADYROV A. *A face authentication using the trace transform*. 2003. ISSN: 1063-6919.

[7] KADYROV A., PETROU M., *The Trace transform and its applications*. IEEE Transactions on Pattern Analysis and Machina Inteligence, PAMI, vol 23, pp 811-828

[8] EKSLER VÁCLAV. *Analýza hlavních komponent v problematice separace naslepo*. 2005. Eektrorevue, číslo 29/2005. ISSN 1213-1539. Dostupné z: <http://www.elektrorevue.cz/clanky/05029/index.html>

[9] HYVÄRINEN, KARHUNEN, OJA. *Independent Komponent Analysis - Introduction*. 2001.PDF dostupné z: <http://www.cis.hut.fi/projects/ica/book/intro.pdf>

[10] ZIBULEVSKY M., PEARLMUTTER B. A. *Blind separation of sources with spase representation in a given signal dictionary*. Helsinky, 2000.

[11] DRAPER A. B., BEAK K., BARTLETT M. S., BEVERIDGE J. R. *Recognizing faces with PCA and ICA*. Computer Vision and Image Understanding, Volume 91, Issues 1-2, July-August 2003, Pages 115-137. Dostupný z <http://www.face-rec.org/algorithms/Comparisons/draper_cviu.pdf>

[12] ETEMAD K., CHELLAPPA R. *Discriminant analysis for recognition of human face images.1997.* J. Optical Society of America. A/Vol. 14, No. 8/August 1997. Dostupné z: < www.face-rec.org/algorithms/LDA/discriminant-analysis-for-recognition.pdf>

[13] ZHAO. W., CHELLAPPA R. PHILLIPS P. J. *Subspace Linear Discriminant Analysis for Face Recognition.*April [PDF] 1999. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.3721>

[14] WISKOTT L., FELLOUS J. M., KRÜGER N., MALSBURG CH. *Face Recognition by Elastic Bunch Graph Matching*, 1999. Inteligent Biometric Techniques in Fingerprint and Face Recognition, eds. Jain, L.C. et al., publ. CRC Press, chapter 11, ISBN 0-8493-2055-0

[15] DLANGISA B., *DRUBIS: A Distributed Face-Identification Experimentation framework*. 2003, thesis for degree of Master of Science Rhodes University South Africa.

[16] S. THAKUR, J.K. SING, D.K. BASU, M. NASIPURI, M. KUNDU, *Face Recognition Using Principal Component Analysis and RBF Neural Network.* icetet,pp.695-700, 2008 First International Conference on Emerging Trends in Engineering and Technology, 2008

[17] XIN LI, BIN LI, HONG CHEN, XIANJI WANG, ZHENGQUAN ZHUANG*: Full-Space LDA With Evolutionary Selection for Face Recognition*. CIS 2006: 1097-1105

[18] ORL face database. AT&T Laboratories, Cambridge, U. K. [Online] Dostupné z: <http://www.uk.research.att.com/facedatabase.html>

[19] A.M. MARTINEZ, A.C. KAK, *PCA versus LDA*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 228-233, Feb 2001

[20] HAYKIN S. *Neural Networks, A Comprehensive Foundation*. Macmillan, New York, NY, 1994.

[21] DAVIS L. D.,*Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991

[22] ANDERSON E., BAI Z., BISCHOF C., BLACKFORD S., DEMMEL J., DONGARRA J., DU CROZ J., GREENBAUM A., HAMMARLING S., MCKENNEY A.,SORENSEN D., *LAPACK User's Guide, Third edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA. 1999. ISBN 0-89871-447-8. Dostupné z: <http://www.netlib.org/lapack>

[23] AFORGE.NET [c# open source]. Dostupné z: <http://www.codeproject.com/KB/recipes/aforge.aspx>

Seznam obrázků