

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

# **Raspoznavanje ljudskih lica korištenjem linearne diskriminantne analize**

Toni Benussi  
Dragutin Hrenek  
Darko Jurić  
Nenad Mikša  
Robert Perica  
Pavle Prentašić  
Boris Trubić

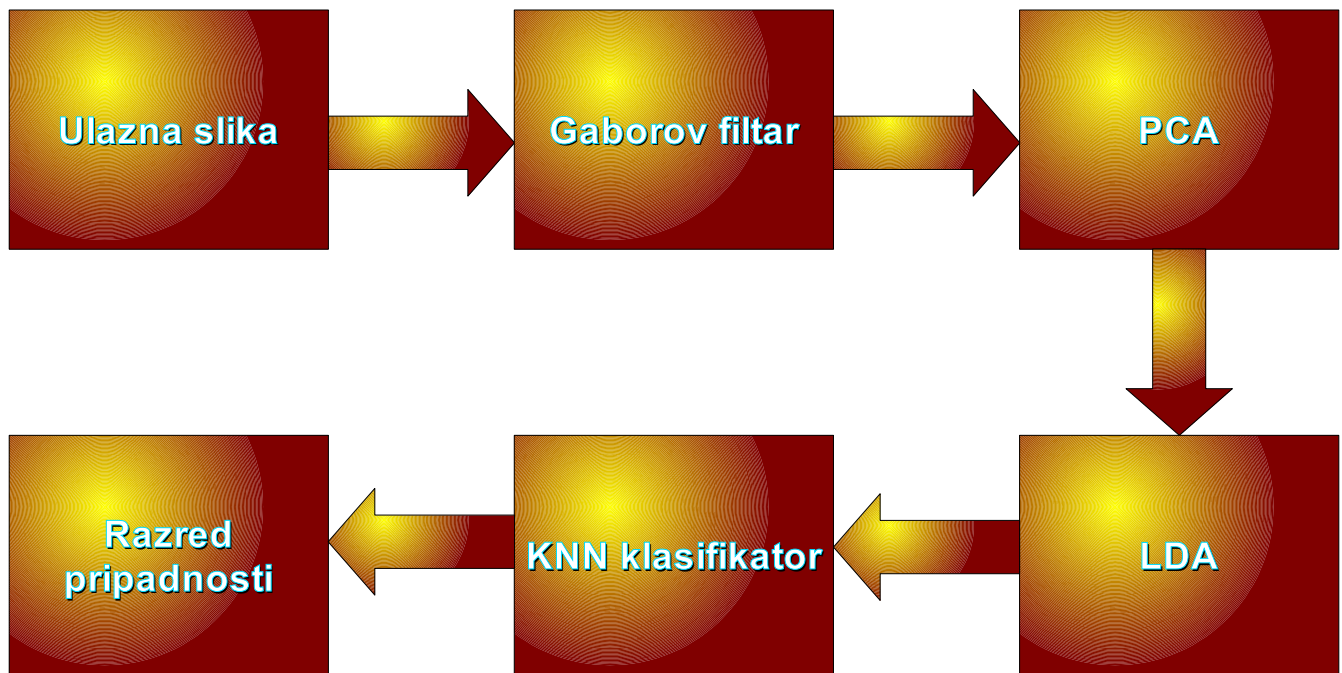
Zagreb, siječanj 2011.

# Sadržaj

1. Uvod.....	1
2. Višestruka diskriminantna analiza.....	2
3. Gaborov filter.....	6
3.1. Uvod.....	6
3.2. Analiza Gaborovog filtra u frekvencijskoj domeni.....	6
3.2.1. Ovisnost o frekvenciji.....	6
3.2.2. Ovisnost o veličini ovojnice.....	7
3.2.3. Ovisnost o orijentaciji.....	8
3.3. Primjenjivost filtra na izlučivanje značajki lica.....	8
4. Principal component analysis (PCA).....	11
4.1. Karhunen-Loeve transformacija.....	11
5. KNN algoritam.....	15
6. Struktura i implementacija sustava za raspoznavanje lica.....	16
6.1. Gaborovi filtri.....	16
6.1.1. Razred Gabor.....	16
6.1.2. Razred FilterBank.....	16
6.1.3. Ekstraktori značajki.....	16
6.2. Višestruka diskriminantna analiza.....	18
6.3. Klasifikator.....	19
6.4. Ostali razredi.....	19
6.5. Postupci učenja i klasifikacije.....	21
7. Rezultati rada sustava.....	22
8. Zaključak.....	24
9. Literatura.....	25

## 1. Uvod

U postupcima za raspoznavanje lica mogu se koristiti različite metode klasifikacije i ekstrakcije značajki. U našem slučaju odlučili smo se za kombinaciju nekoliko različitih metoda ekstrakcije značajki i klasifikacije kako bismo vidjeli što je najbolji pristup za klasifikaciju ljudskih lica. Prema našem planu koristimo Gaborov filter kako bismo izlučili značajke ljudskih lica. Primjenom Gaborovih filtara različitih parametara dobivamo vektor značajki relativno visoke dimenzionalnosti pa prema drugim radovima je poželjno prije uporabe linearne diskriminantne analize provesti redukciju dimenzionalnosti vektora značajki. Za to se koristi PCA kojom reduciramo dimenzionalnost na prihvatljivu veličinu. Tako reducirani vektor značajki dovodimo na ulaz LDA-a. Nakon provedene analize slijedi jednostavna klasifikacija koju provodimo korištenjem KNN algoritma. Cijeli sustav se jednostavno može prikazati pomoću slike. Teorijski aspekti nabrojanih metoda su objašnjeni u nastavku teksta, a grafički prikaz strukture sustava je prikazan na slici 1.

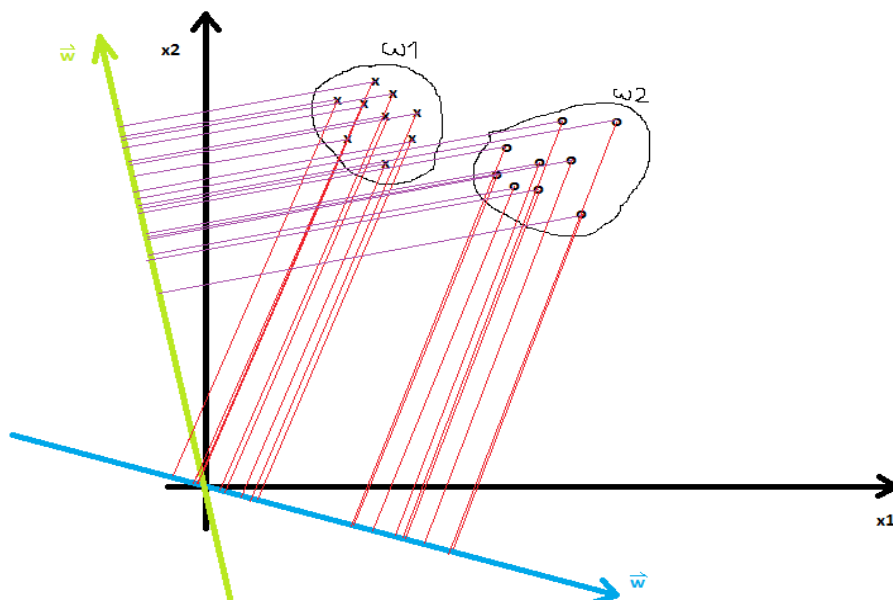


Slika 1. Arhitektura sustava

## 2. Višestruka diskriminantna analiza

Svaki sustav koji klasificira neke objekte, te objekte se svrstava u određene razrede. Tih razreda može biti proizvoljno mnogo, ovisno o prirodi problema. Promatrat ćemo opći slučaj kada imamo  $c$  razreda

$D_1, D_2, \dots, D_c$  odnosno  $\omega_1, \omega_2, \dots, \omega_c$ . Svaki uzorak  $\vec{x}$  iz nekog od razreda se predstavlja jednom točkom u  $n$ -dimenzionalnom prostoru. Generalizirana linearna diskriminantna analiza uključuje  $c-1$  diskriminantnih funkcija, kojima se ostvaruje preslikavanje iz  $d$ - u  $c-1$ -dimenzionalni prostor (uz uvjet  $d \geq c$ ). Slučaj dva razreda u dvodimenzionalnom prostoru možemo vidjeti na slici 2.



Slika 2. Ilustracija dvodimenzionalne LDA

Za određivanje tih diskriminantnih funkcija, prvo nam je potrebna generalizirana matrica raspršenosti unutar razreda (engl. *within-class scatter matrix*):

$$S_W = \sum_{i=1}^c S_i \quad (1)$$

gdje je:

$$S_i = \sum_{\vec{x} \in D_i} (\vec{x} - \vec{m}_i)(\vec{x} - \vec{m}_i)^T \quad (2)$$

i vektor srednje vrijednosti za razred  $\omega_i$

$$\vec{m}_i = \frac{1}{n_i} \sum_{\vec{x} \in D_i} \vec{x} \quad (3)$$

Da bi dobili matricu raspršenosti između razreda (engl. *between-class scatter matrix*)  $S_B$ , trebamo malo raspisati izraz za ukupnu raspršenost razreda, ali krećemo od formule za ukupan vektor srednjih vrijednosti, koji nam daje ukupno težište svih uzoraka:

$$\vec{m} = \frac{1}{n} \sum_{\vec{x}} \vec{x} = \frac{1}{n} \sum_{i=1}^c n_i \vec{m}_i \quad (4)$$

Ukupnu matricu raspršenosti svih razreda definiramo kao:

$$S_T = \sum_{\vec{x}} (\vec{x} - \vec{m})(\vec{x} - \vec{m})^T \quad (5)$$

Raspisom tog izraza i izlučivanjem dobije se:

$$S_T = S_W + S_B \quad (6)$$

pri čemu je

$$S_B = \sum_{i=1}^c n_i (\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T \quad (7)$$

Da bismo projicirali uzorke iz  $n$ -dimenzionalnog prostora u  $(c-1)$ -dimenzionalni, koristimo se s  $(c-1)$  diskriminantnih funkcija oblika:

$$y_i = \vec{w}_i^T \vec{x} \quad i=1,2,\dots,c-1 \quad (8)$$

Radi lepšeg zapisa,  $\vec{y}_i$  možemo promatrati kao komponente vektora  $\vec{Y}$ , a težinske vektore  $\vec{w}_i$  kao stupce matrice  $W$  dimenzija  $d \times (c-1)$ , pa tako možemo pisati projekciju:

$$\vec{Y} = \vec{W}^T \vec{x} \quad (9)$$

Naši uzorci  $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$  se projiciraju u odgovarajući skup uzoraka  $\vec{y}_1, \vec{y}_2, \dots, \vec{y}_n$ , a ti uzorci mogu biti opisani svojim srednjim vrijednostima i matricama raspršenosti, kao i originalni uzorci:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{\vec{y} \in Y_i} \vec{y} \quad (10)$$

$$\tilde{m} = \frac{1}{n} \sum_{i=1}^c n_i \tilde{m}_i \quad (11)$$

$$\tilde{S}_W = \sum_{i=1}^c \sum_{\vec{y} \in Y_i} (\vec{y} - \tilde{m}_i)(\vec{y} - \tilde{m}_i)^T \quad (12)$$

$$\tilde{S}_B = \sum_{i=1}^c n_i (\tilde{m}_i - \tilde{m})(\tilde{m}_i - \tilde{m})^T \quad (13)$$

što se nakon kraćeg raspisa može zapisati u obliku:

$$\tilde{S}_W = \vec{W}^T S_W \vec{W} \quad (14)$$

$$\tilde{S}_B = \vec{W}^T S_B \vec{W} \quad (15)$$

Da bismo riješili problem, trebamo pronaći matricu  $W$  takvu da se maksimizira kriterijska funkcija  $J(W)$  :

$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|} \quad (16)$$

Računamo determinante matrica raspršenosti jer je to jednostavna skalarna mjera raspršenosti. No, pronalazak takve pravokutne matrice  $W$  koja maksimizira gore navedenu kriterijsku funkciju je težak problem. Rješenje tog problema se nazire u traženju svojstvenih vrijednosti kao korijena karakterističnog polinoma

$$|S_B - \lambda_i S_W| = 0 \quad (17)$$

te zatim rješavanjem matrične jednadžbe

$$(S_B - \lambda_i S_W) \vec{w}_i = \vec{0} \quad (18)$$

Takav pristup je moguć jer su stupci optimalne matrice  $W$  generalizirani svojstveni vektori koji odgovaraju najvećim svojstvenim vrijednostima u izrazu

$$S_B \vec{w}_i = \lambda_i S_W \vec{w}_i \quad (19)$$

te ako je  $S_W$  nesingularna, umjesto traženja inverza te matrice, možemo pronaći korijene karakterističnog polinoma u formuli (17).

### 3. Gaborov filter

#### 3.1. Uvod

Motivacija za korištenje Gaborovih filtara najviše je potaknuta biološkim aspektom, jer su u vizualnom korteksu mozga živih bića pronađene stanice čiji je odziv na pobudu sličan odzivu Gaborovih filtara. Pozitivnim reagiranjem na određene značajke u slici kao što su rubovi i značajke lica te time i osjetljivošću na određene frekvencije bili su ključni pokretači za daljnje istraživanje njegovih mogućnosti.

Gaborov filter je definiran kao produkt Gaussove funkcije i kompleksnog sinusnog signala. Kako Gaussova funkcija množi sinusni signal, ograničava njegovo djelovanje. Gaborov filter je kompleksan filter. Za dvodimenzionalnu inačicu filtra prikazan je njegov odziv izrazom 20 .

$$g(\theta; \lambda; \sigma; x; y) = e^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)} e^{j \frac{\pi}{\lambda}(x \cos \theta + y \sin \theta)} \quad (20)$$

Parametri su: orijentacija filtra  $\Theta$  , valna duljina  $\lambda$ , veličina Gaussove ovojnice  $\sigma$ , te ulazne koordinate  $x$  i  $y$ . Faze nema odnosno njezina je vrijednost jednaka 0.

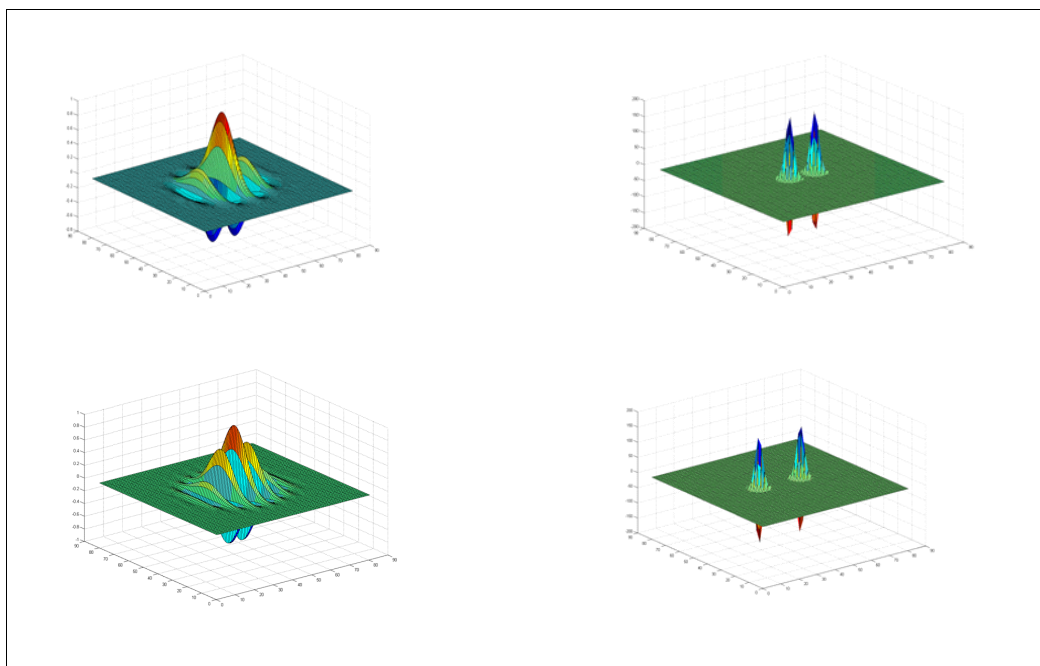
#### 3.2. Analiza Gaborovog filtra u frekvencijskoj domeni

Karakteristike Gaborovih filtara kao i ovisnost o parametrima je lakše razumjeti ako se izvrši Fourierova transformacija filtra te se takav promatra. Bit će prikazana ovisnost o promijeni frekvencije, veličini filtra te njegovoj orijentaciji. Kroz ova tri elementa moći ćemo pokriti cijelu frekvencijsku domenu. Iako je filter kompleksan zbog nemogućnosti prikaza oba dijela istovremeno prikazani su samo realni dijelovi filtra. Imaginarni dijelovi su samo pomaknuti u fazi.

##### 3.2.1. Ovisnost o frekvenciji

Mijenjanjem frekvencije postavljamo filter na odgovarajuće mjesto u frekvencijskoj domeni. Smanjivanjem frekvencije sinusni dijelovi filtra se približavaju dok njezinim povećanjem ti isti dijelovi se razmiču kako je i prikazano slikom.

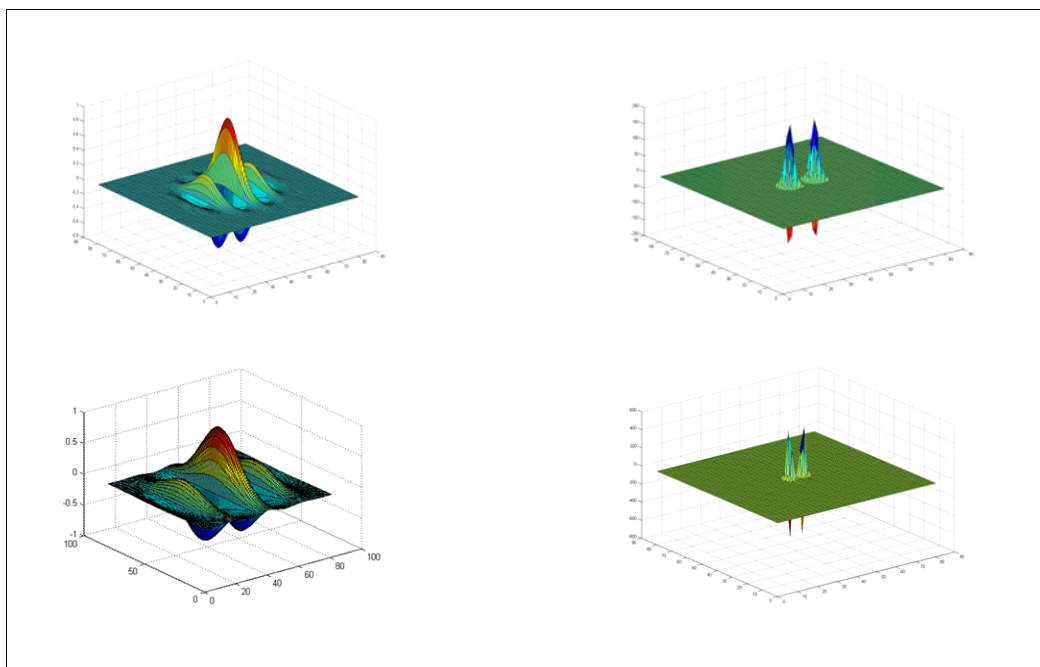




Slika 3. Ovisnost filtra o povećaju frekvencije

### 3.2.2. Ovisnost o veličini ovojnice

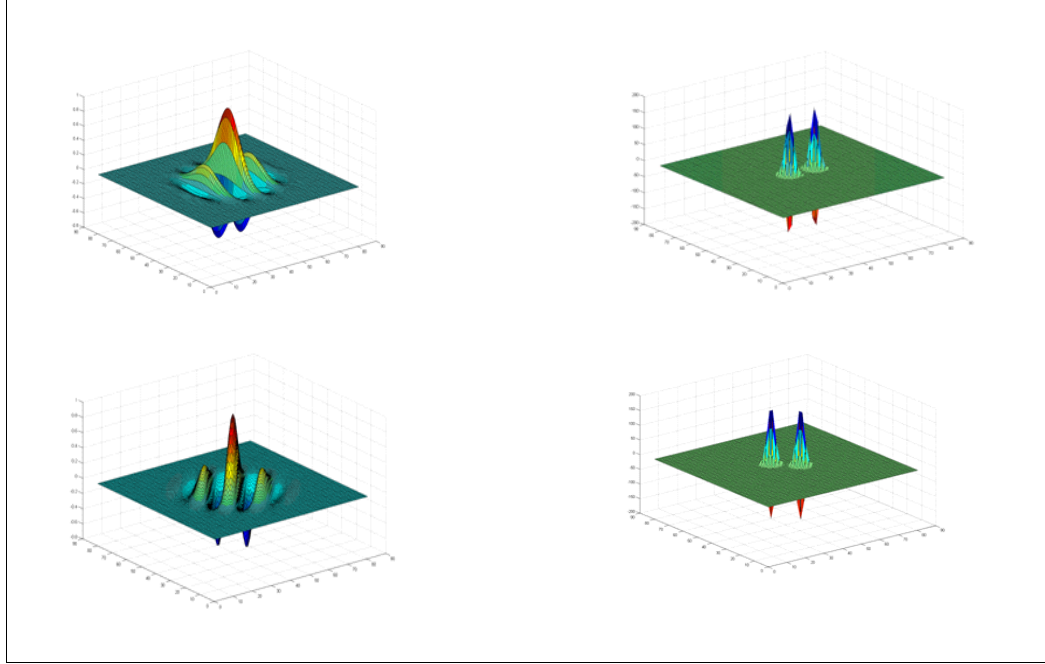
Mijenjanjem veličine filtra dobivamo obrnuto proporcionalno mijenjanje veličine filtra u frekvencijskoj domeni.



Slika 4. Ovisnost filtra o povećanju veličine Gaussove ovojnice

### 3.2.3. Ovisnost o orijentaciji

Promjena orijentacije utiče na obje domene jednako. Rotacijom filtra u prostornoj domeni zakrećemo filter i u frekvencijskoj domeni.



Slika 5. Ovisnost filtra o orijentaciji

### 3.3. Primjenjivost filtra na izlučivanje značajki lica

Kako bi se izlučilo što više značajki objekta na slici česta je uporaba filtara uz mijenjanje njegove veličine i frekvencije. Da bi se lakše kreirao set filtara (*eng. Gabor bank*) često se uvodi oblik koji ima takve parametre koji jednoliko omogućuju skaliranje filtra i mijenjanje frekvencije. Jedan takav oblik prikazan je dolje. Parametri  $\sigma$  i  $\Lambda$  zamijenjeni su jedinstvenim parametrom  $\gamma$ . Filtar je i rotiran za  $\pi/2$  radijana.

$$g(\theta; \gamma; \sigma; x; y) = e^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)} e^{j \frac{\pi}{\gamma\sigma} (x \sin \theta - y \cos \theta)}, \gamma = \frac{\lambda}{\sigma} \quad (21)$$

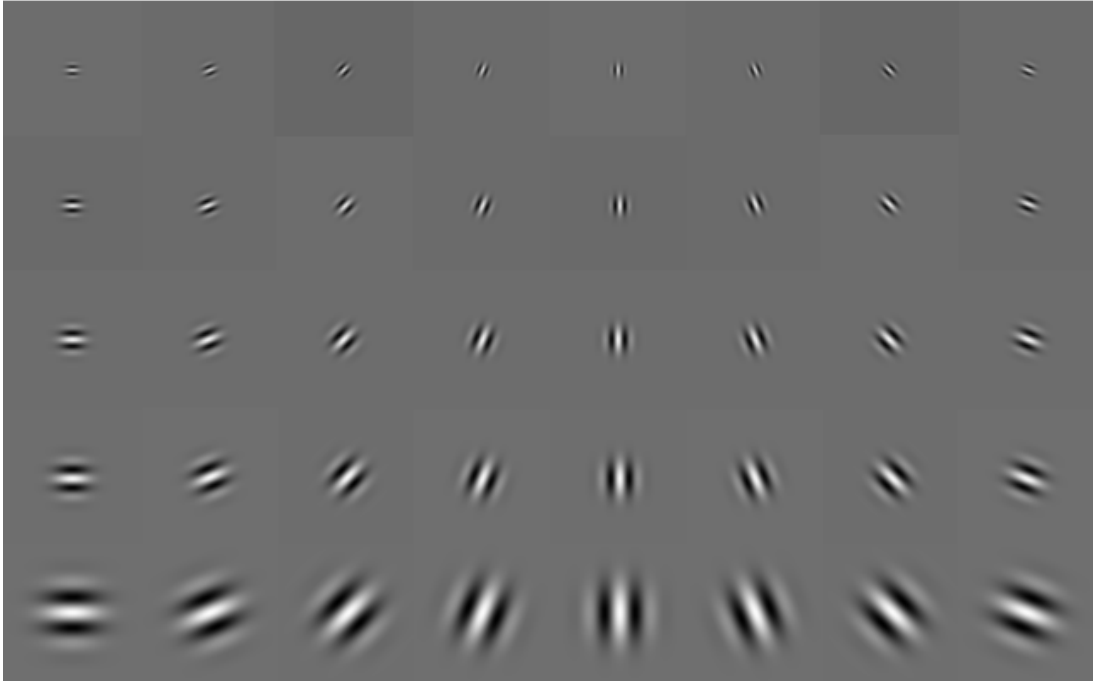
Vrijednosti parametara  $\sigma$  (skaliranje) se generiraju pomoću:

$$\sigma_i = A * 2^{\frac{(-i+3/2)}{2}}, i \in \{0, 1, 2, \dots, 4\}, A = \text{const.} \quad (22)$$

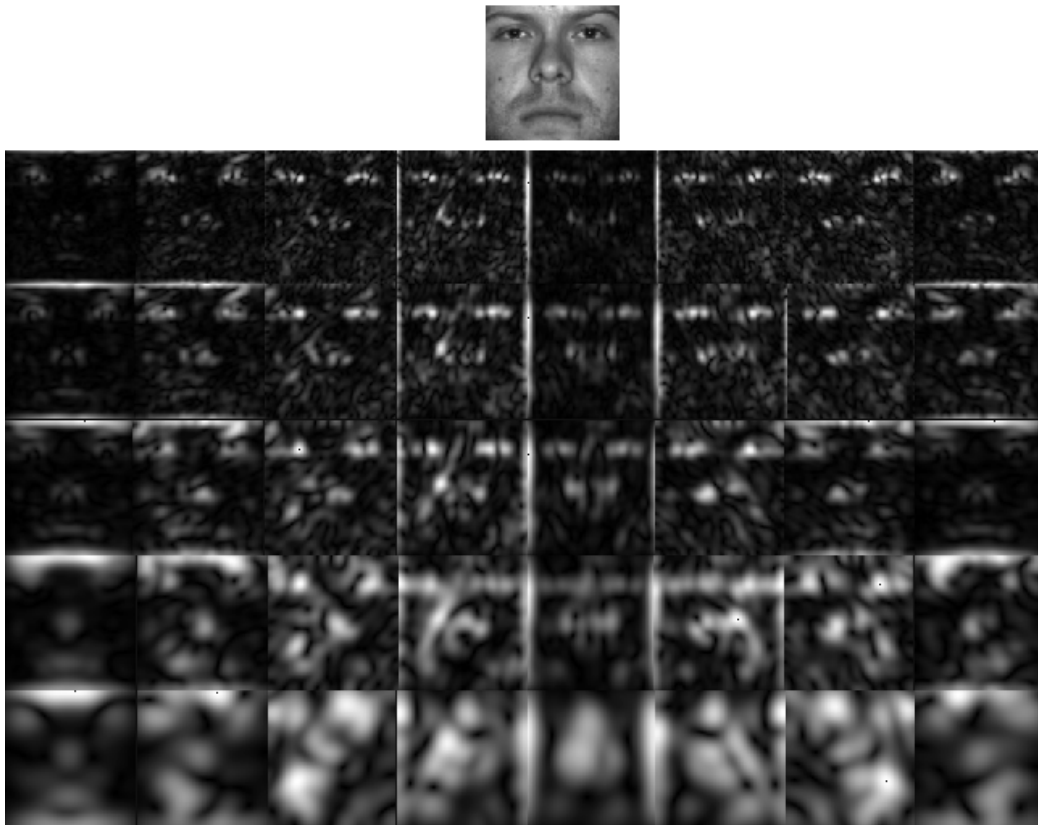
Vrijednosti parametra  $\Theta$  (kuta rotacije) se dobivaju pomoću:

$$\theta_j = j \frac{\pi}{8}, j \in \{0, 1, 2, \dots, 7\} \quad (23)$$

U nastavku je prikazan rezultat konvolucije 40 Gaborovih filtara, koji su generirani pomoću gornjih izraza, sa slikom lica. Mogu se primijetiti naglašene značajke lica (oči, nos, okolina usana, različite nepravilnosti lica).



Slika 6. Gaborov set filtara (realni dio)



Slika 7. Primjer primjene filtara nad slikom lica (slike amplituda)

## 4. Principal component analysis (PCA)

Prilikom konstruiranja sustava za prepoznavanje lica vektor značajki koji sadrži pojedine slikovne elemente može biti veoma velik. Ako imamo sliku lica dimenzija od samo  $64 \times 64$  slikovnih elemenata sa 8 slikovnih elemenata imamo vektor značajki koji sadrži 32,768 značajki odnosno točku u 32,768-dimenzionalnom prostoru značajki.

Ideja kojom se vodi PCA je ta da se može odrediti potprostor značajki manjih dimenzija koji će dovoljno točno opisivati slike lica, uz najmanji mogući gubitak informacija, te omogućiti izgradnju jednostavnijeg klasifikatora.

Ovaj postupak se temelji na Karhunen-Loeveovoj transformaciji. Ona se vodi idejom da ako je moguće izvršiti kompresiju slike lica pojedine osobe na način da se svako lice prikaže kao niz koeficijenata u smanjenom potprostoru značajki koji prilično dobro opisuju pojedino lice da se isti koeficijenti mogu koristiti za uspješnu klasifikaciju. Na ovaj način se osoba koja je trenutno predložena sustavu može prepoznati određivanjem koeficijenata izračunatih za tu osobu i već otprije poznatih koeficijenata pohranjenih u sustav, a koji karakteriziraju sustavu već poznate osobe.

### 4.1. Karhunen-Loeve transformacija

Ako svaku sliku lica promatramo kao uzorak u prostoru značajki razapet ortonormiranim jediničnim vektorima  $\vec{e}_j$  tada uzorak  $\vec{x}$  možemo zapisati kao:

$$\vec{x} = \sum_{j=1}^r c_j \vec{e}_j \quad (24)$$

gdje su  $c_j$  koeficijenti reda. Mi želimo odrediti  $n < r$  vektora  $\vec{e}_j$  za koje vrijedi da minimiziraju nastalu pogrešku aproksimacije slike lica prikazane samo pomoću njih, odnosno minimiziraju izraz:

$$\bar{\epsilon}^2 = E \left\{ \sum_{j=1}^r c_j \vec{e}_j - \sum_{j=1}^n c_j \vec{e}_j \right\} \quad (25)$$

gdje je  $E$  matematičko očekivanje. Ako se uzme ortonormalnost vektora odnosno ako znamo da vrijedi

$$\vec{e}_j^T \vec{e}_i = \begin{cases} 0 & \text{ako je } i \neq j \\ 1 & \text{ako je } i = j \end{cases} \quad (26)$$

srednju kvadratnu pogrešku danu s 25 možemo zapisati kao

$$\bar{\varepsilon}^2 = E \left\{ \sum_{j=n+1}^r c_j^2 \right\} \quad (27)$$

te ako primijetimo da vrijedi uz primjenu ortonormalnosti jediničnih vektora  $\vec{e}_j$  sljedeće

$$\begin{aligned} \vec{x} &= \sum_{j=1}^n c_j \vec{e}_j \\ \frac{\vec{x}}{\vec{e}_j} &= \frac{\sum_{j=1}^n c_j \vec{e}_j}{\vec{e}_j} \\ \vec{e}_j^T \vec{x} &= \vec{e}_j^T \vec{e}_j c_j \\ c_j &= \vec{e}_j^T \vec{x} \end{aligned} \quad (28)$$

uvrštavanjem 28 u 27 srednje kvadratna pogreška iznosi 29.

$$E \left\{ \sum_{j=n+1}^r \vec{e}_j^T \vec{x} \vec{x}^T \vec{e}_j \right\} \quad (29)$$

Kako su vektori  $\vec{e}_j$  deterministički možemo zamijeniti operaciju zbrajanja i matematičkog očekivanja te 29 možemo zapisati u obliku prikazanom s 30.

$$\bar{\varepsilon}^2 = \sum_{j=n+1}^r \vec{e}_j^T \left[ E \{ \vec{x} \vec{x}^T \} \right] \vec{e}_j \quad (30)$$

Ako pretpostavimo da je srednja vrijednost uzoraka jednaka 0 (ako nije od uzorka sa može oduzeti srednja vrijednost tako da je kriterij zadovoljen) vrijedi da  $E \{ \vec{x} \vec{x}^T \}$  odgovara korelacijskoj matrici uzoraka R.

Korelacijska matrica n slučajnih varijabli  $x_1, x_2, \dots, x_n$  je  $n \times n$  matrica čiji i, j element odgovara

$$\text{corr}(X_i, Y_j) = \frac{\text{cov}(X_i, Y_j)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (31)$$

gdje je E matematičko očekivanje, cov kovarijanca između dviju slučajnih varijabli X i Y. Korelacijska matrica je simetrična jer je korelacija između slučajnih varijabli  $X_i$  i  $Y_i$  jednaka korelaciji između varijabli  $Y_i$  i  $X_i$ . U slučaju da se radi o standardiziranim slučajnim varijablama  $X_i / \sigma(X_i)$  za

svaki  $i = 1, \dots, n$ .

Primjenjujući gornju pretpostavku 30 možemo zapisati u obliku prikazanom s 32.

$$\bar{\varepsilon}^2 = \sum_{j=n+1}^r \vec{e}_j^T \mathbf{R} \vec{e}_j \quad (32)$$

Skup ortonormiranih vektora  $\vec{e}_j$  odnosno koordinatni sustav potprostora značajki koji minimizira srednju kvadratnu pogrešku možemo dobiti ako odredimo one vektore  $\vec{e}_j$  koji zadovoljavaju jednadžbu prikazanu s 33 odnosno jednadžbu prikazanu s 34.

$$(\mathbf{R} - \lambda_j \mathbf{I}) \vec{e}_j = \vec{0} \quad (33)$$

$$\mathbf{R} \vec{e}_j = \lambda_j \vec{e}_j \quad (34)$$

Ako u izraz 32 za srednju kvadratnu pogrešku uvrstimo 34 te primijenimo svojstvo ortonormalnosti vektora  $\vec{e}_j$  iz 26 dobivamo da ako uzorak koji je početno bio  $n$ -dimenzionalan aproksimiramo sa  $r < n$  značajki činimo srednju kvadratnu pogrešku koja iznosi

$$\bar{\varepsilon}^2 = \sum_{j=n+1}^r \lambda_j \quad (35)$$

Vektori  $\vec{e}_j$  iz 33 su zapravo svojstveni  $n$ -dimenzionalni vektori kojima odgovara svojstvena vrijednost  $\lambda_j$  korelacijske matrice  $\mathbf{R} = E\{\vec{x} \vec{x}^T\}$ .

Iz danog razmatranja možemo zaključiti da ćemo ostvariti najmanju srednju kvadratnu pogrešku danu s 35 ako za prikaz uzorka  $\vec{x}$  odaberemo one svojstvene vektore korelacijske matrice koji imaju najveće svojstvene vrijednosti  $\lambda_j$  jer pogrešku zapravo čine vrijednosti onih svojstvenih vrijednosti čiji svojstveni vektori nisu korišteni za prikaz uzorka u reduciranom  $r$ -dimenzionalnom prostoru značajki.

Transformacijska matrica  $A$  koja transformira početni  $n$ -dimenzionalni vektor značajki u reducirani  $r$ -dimenzionalni vektor značajki  $i$  pritom minimizira srednju kvadratnu pogrešku činimo od prvih  $n$  vektora koordinatnih osi  $\vec{e}_j$   $j = 1, \dots, n$ :

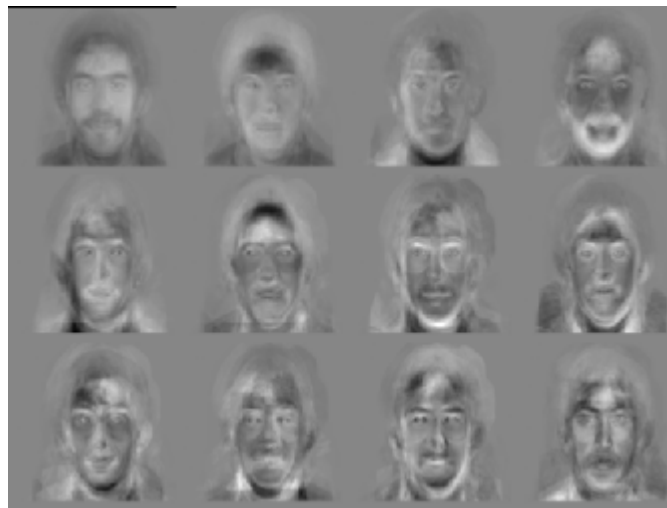
$$A = \begin{bmatrix} \vec{e}_1^T \\ \vec{e}_2^T \\ \vdots \\ \vec{e}_n^T \end{bmatrix} \quad (36)$$

gdje su  $\vec{e}_j^T$   $j = 1, \dots, n$  uređeni po padajućem redoslijedu njima odgovarajućih svojstvenih vrijednosti korelacijske matrice  $R$  tako da vrijedi

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n \geq \dots \geq \lambda_r \quad (37)$$

Reducirani vektor  $y$  dobijemo množeći izvorni uzorak  $x$  sa matricom  $A$ :  $\vec{y} = A \vec{x}$ .

Na području raspoznavanja lica vektori koji predstavljaju koordinatne osi reduciranog prostora značajki, promatrani u originalnom prostoru značajki, također predstavljaju lica. Razlika je u tome što to nisu lica stvarnih osoba koje su bile predstavljene sustavu već takozvana svojstvena lica (*engl. Eigenfaces*). Prikaz jednog takvog lica je vidljiv na slici 8. Ta svojstvena lica se određuju na temelju uzoraka za učenje, a imaju svojstvo da se svako lice koje je predloženo sustavu može prikazati kao njihova linearna kombinacija što je vidljivo na slici 10.



Slika 8 . Svojstvena lica



Slika 9 . Lice kao linearna kombinacija svojstvenih lica



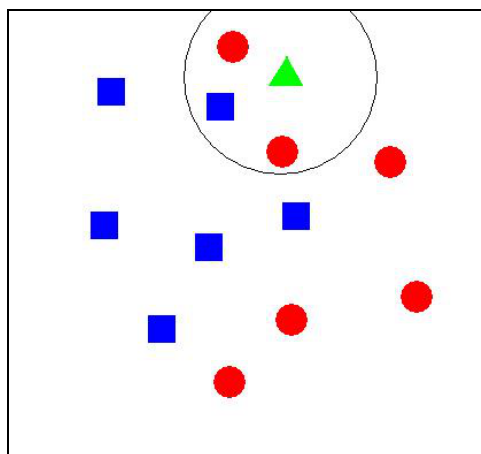
## 5. KNN algoritam

Algoritam  $k$  najbližih susjeda se temelji se na klasifikaciji uzoraka na temelju njima najbližih uzoraka iz skupa za učenje u  $n$ -dimenzionalnom vektorskom prostoru značajki. Ideja ovog algoritma je vrlo jednostavna i može se opisati u sljedećih nekoliko koraka:

- 1) Potrebno je zapamtiti pozicije u prostoru značajki za sve uzorke iz skupa za učenje kojima je poznata klasifikacija
- 2) Kad se dovede novi uzorak na ulaz klasifikatora pronade se  $k$  najbližih uzorka (iz skupa za učenje) tom uzorku nepoznate klasifikacije, pri čemu se udaljenost uzoraka računa kao euklidska udaljenost u vektorskom prostoru
- 3) Klasifikacija uzorka se obavlja sistemom glasanja, tako da se uzorak klasificira u razred koji se najčešće pojavljuje među susjedima tog uzorka (uzima se neparni broj susjeda  $k$  da se smanji vjerojatnost da broj glasova za 2 razreda bude jednak)

Klasifikator temeljen na ovom algoritmu, bez obzira na svoju jednostavnost, vrlo je robustan i daje konzistentne rezultate. Dokazano je da pogreška u klasifikaciji KNN algoritmom, ako broj uzoraka teži u beskonačnost, nije veća od dvostruke minimalne pogreške koju je moguće postići s obzirom na razdiobu kojoj pripadaju uzorci koje klasificiramo.

Na slici 10 vidimo primjer rada algoritma  $k$  najbližih susjeda u 2-dimenzionalnom prostoru značajki kojem su uzorci jedne klase označeni plavim kvadratima, a uzorci druge klase crvenim krugovima, a novi uzorak zelenim trokutom. Ako uzmemo  $k=3$  onda su tri najbliža susjedna tri označena uzorka, te sistemom glasanja klasificiramo novi uzorak u klasu predstavljenu crvenim krugovima.



Slika 10 . Primjer klasifikacije pomoću KNN algoritma

## 6. Struktura i implementacija sustava za raspoznavanje lica

Sustav je građen po MVC (model-view-controller) načelu, što znači da je sučelje sustava prema korisniku maksimalno odvojeno od dijela sustava koji zaista vrši obradu podataka i samu klasifikaciju. Sučelje je povezano s modelom preko kontrolera – razreda koji korisničkom sučelju omogućuje poziv metoda modela, poput učenja klasifikatora ili same klasifikacije. Prilikom izrade sustava korištena je programska biblioteka *Accord.NET*<sup>1</sup>. Sam model se sastoji od više razreda, koji su opisani u nastavku.

### 6.1. Gaborovi filtri

Zbog potrebe korištenja sloga Gaborovih filtara za filtriranje ulazne slike i različitih mogućnosti ekstrakcije značajki implementacija je podijeljena u više razreda. Osnovni razred je *Gabor*. Slog Gaborovih filtara implementira se pomoću razreda *FilterBank*, dok se ekstrakcija značajki vrši pomoću razreda *AmplitudeFeatureExtractor* i *PhaseFeatureExtractor* koje dijele isto sučelje. Dijagram razreda koji implementiraju Gaborove filtre je prikazan na slici 11.

#### 6.1.1. Razred *Gabor*

Razred predstavlja implementaciju Gaborovih filtara. Inicijalizacijski parametri su veličina jezgre, opcija za uklanjanje srednje vrijednosti i parametri koji su sami dio formule za generiranje Gaborovih filtara, a to su standardna devijacija, kut zakreta filtra, njegova frekvencija. Razred sadrži metode za konvoluciju slike i dobivanje značajki predstavljenih kao slike. Značajke su dobivanje amplitudnog, faznog, realnog i imaginarnog dijela konvoluirane slike sa filtrom. Razred se koristi samo interno.

#### 6.1.2. Razred *FilterBank*

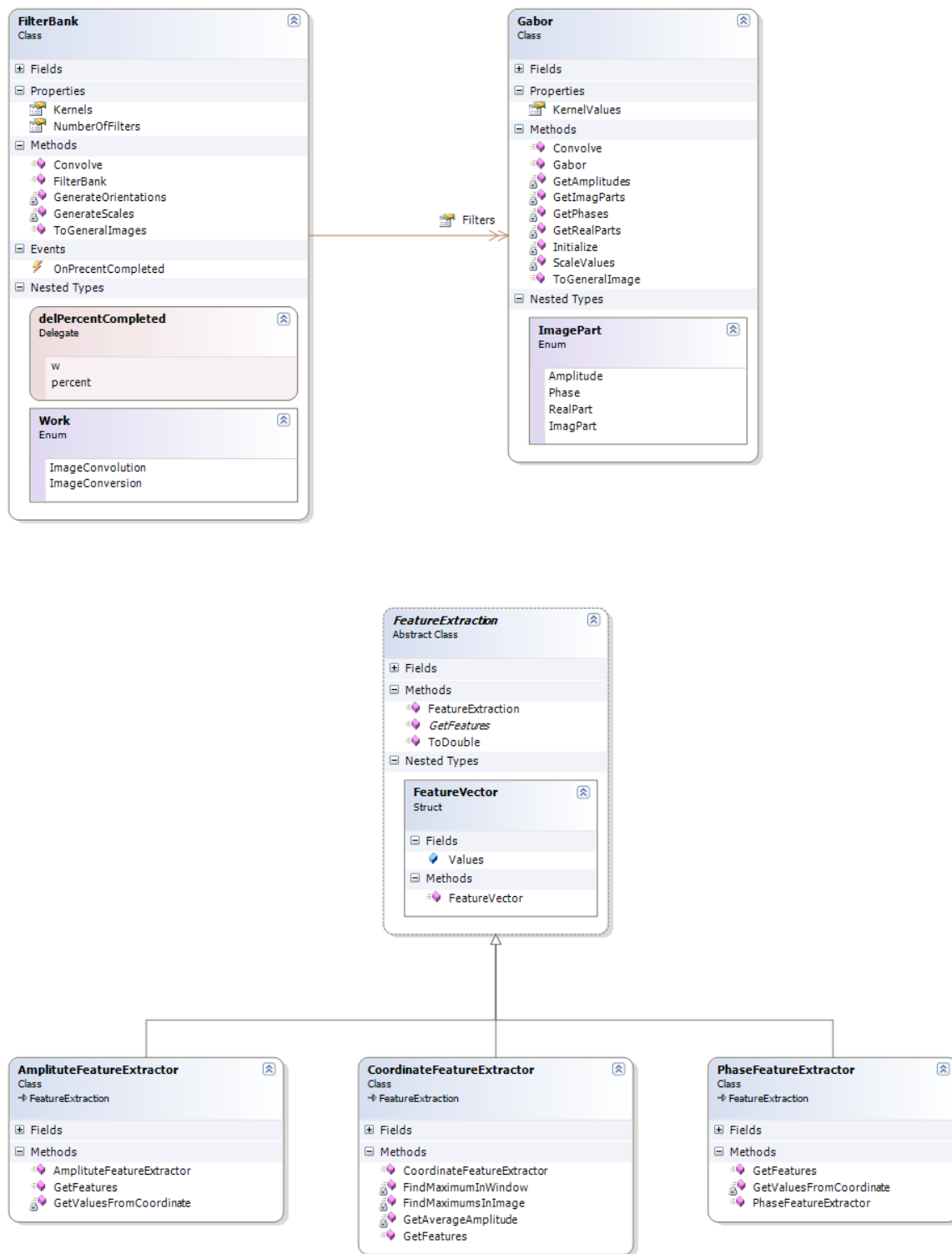
Razred izravno koristi razred *Gabor*. Stvara određeni broj Gaborovih filtara ovisno o broju zadanih orijentacija i standardnih devijacija. Sadrži metode za konvoluciju slike i dobivanje stvorenih Gaborovih filtara. Njezine metode iako su označene kao javne ne koriste se izravno.

#### 6.1.3. Ekstraktori značajki

Svaka slika konvoluirana sa Gaborovim filterom ima kompleksne vrijednosti. Kod kompleksnih vrijednosti možemo odabrati nekoliko značajki. Implementirani su ekstraktori za amplitudne i fazne vrijednosti. Kao inicijalizacijski parametar prima se razred *FilterBank* koji sadrži zbor Gaborovih filtara, usto potrebna je i slika iz koje želimo ekstrahirati značajke. Svaki razred ekstraktora

<sup>1</sup> <http://accord-net.origo.ethz.ch/>

implementira svoju metodu za dobivanje vektora značajki koji je predstavljen nizom vrijednosti piksela konvoluiranih slika na nekoj koordinati. U projektu se koristi samo amplitudni ekstraktor značajki.

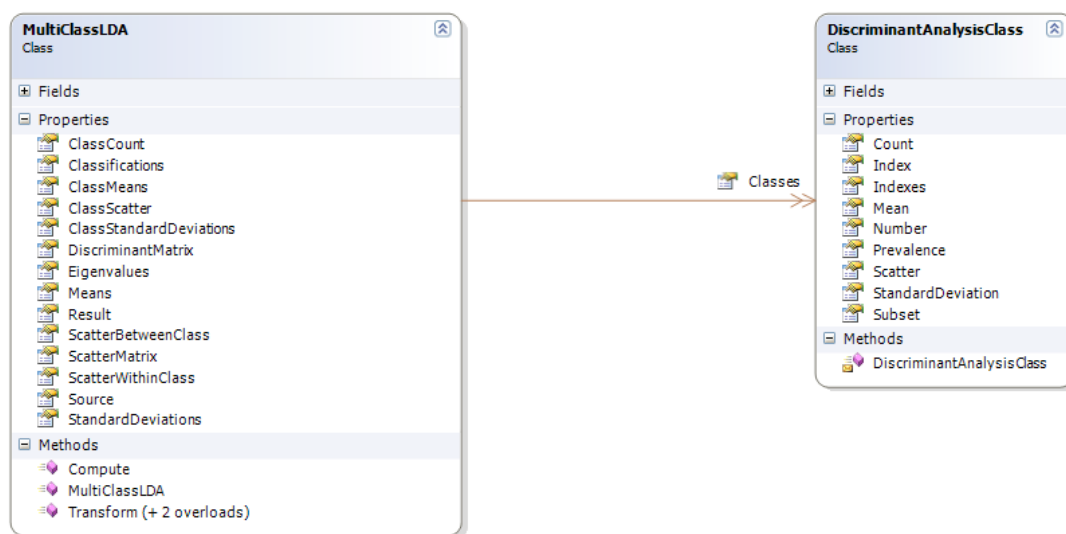


Slika 11: Dijagram razreda za filtriranje Gaborovim filtrima

## 6.2. Višestruka diskriminantna analiza

Razredi *MultiClassLda* i *DiscriminantAnalysisClass* sadrže metode za izvršavanje višestruke diskriminantne analize nad danim podacima te redukciju dimenzionalnosti ulaznih vektora. Dijagram navedenih razreda je prikazan na slici 12.

Arhitektura razreda *MultiClassLda* je vrlo jednostavna te prilagođena potrebi rješavanja zadatka. Konstruktor razreda za parametre prima matricu ulaznih vektora koja je organizirana na način da jedan redak predstavlja jedan vektor značajki. Osim matrice ulaznih vektora potrebno je zadati i vektor pripadnosti koji nam govori kojoj klasi pripada zadani redak matrice jer smo rekli da svaki redak predstavlja jedan uzorak za učenje. Unutar konstruktora se ovi podaci pohranjuju za daljnju uporabu. Sama provedba diskriminantne analize je implementirana u javnoj metodi *Compute*. Implementacijski detalji se ne razlikuju previše od teorijskog oblika samog problema. Nakon što se provede diskriminantna analiza moguće je korištenjem javne metode *Transform* dobiti novi vektor projiciran u prostor manje dimenzionalnosti. Ova metoda je po svojoj konstrukciji vrlo jednostavna te kao parametar prima ulaznu matricu koja predstavlja podatke koje želimo preslikati u prostor manjih dimenzija. Drugi parametar ove metode određuje dimenzionalnost prostora u kojeg projiciramo podatke. Ukoliko se ovaj parametar ne navede uzimaju se svi dostupni vektori. Osim ove glavne klase postoje još dvije pomoćne klase *DiscriminantAnalysisClass* te *DiscriminantAnalysisClassCollection*. Te dvije klase jednostavno služe kao spremišta podataka prilikom same provedbe linearne diskriminantne analize.



Slika 12: Dijagram razreda za višestruku diskriminantnu analizu

### 6.3. Klasifikator

Za potrebe klasifikacije koriste se razredi koji nasljeđuju apstraktni razred *Classifier*. Kao što je ranije spomenuto, naš sustav koristi kNN klasifikator, a isti je implementiran u razredu *KNNClassifier*. Dijagram razreda za klasifikaciju je prikazan na slici 13.

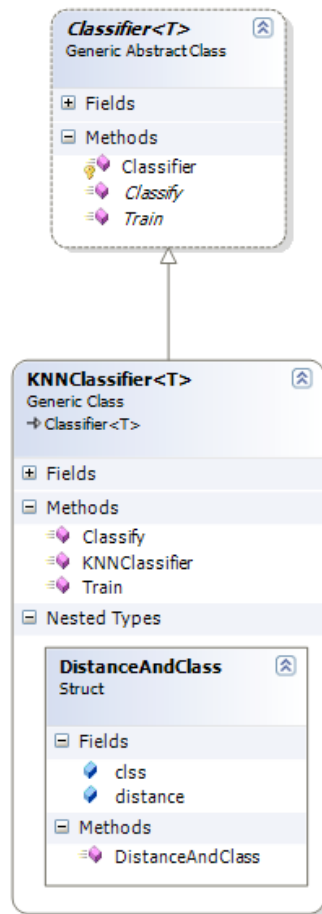
Razred *Classifier*, kao i razredi koji ga nasljeđuju su generički razredi da bi se omogućilo proizvoljno biranje vrste podataka koja će se koristiti kao oznaka pripadnosti pojedinog uzorka nekom razredu.

Učenje klasifikatora je vrlo jednostavno i sastoji se samo od pamćenja svih uzoraka za učenje, što se izvršava metodom *Train* koja ulazni skup uzoraka pohranjuje u listu *patternList*, dok se oznaka razreda kojem pripada pojedini uzorak pohranjuje u listu *classList*.

Novi uzorci se klasificiraju metodom *Classify*, u kojoj se pronađe  $k$  najbližih susjeda iz skupa za učenje uzorku kojeg se želi klasificirati. Nakon toga se odredi najčešća klasa u tom skupu od  $k$  susjeda, te se ista vraća kao rezultat klasifikacije. Kako su za samu klasifikaciju bitne jedino udaljenosti uzoraka i pripadnost uzoraka nekom od razreda uvedena je dodatna struktura *DistanceAndClass* u kojoj se za pojedini uzorak pamti razred i udaljenost od uzorka kojeg se želi klasificirati. Kad se sortira skup takvih struktura prema udaljenosti uzoraka jednostavno se iz njega iščita informacija o najčešćoj klasi u prvih  $k$  među njima.

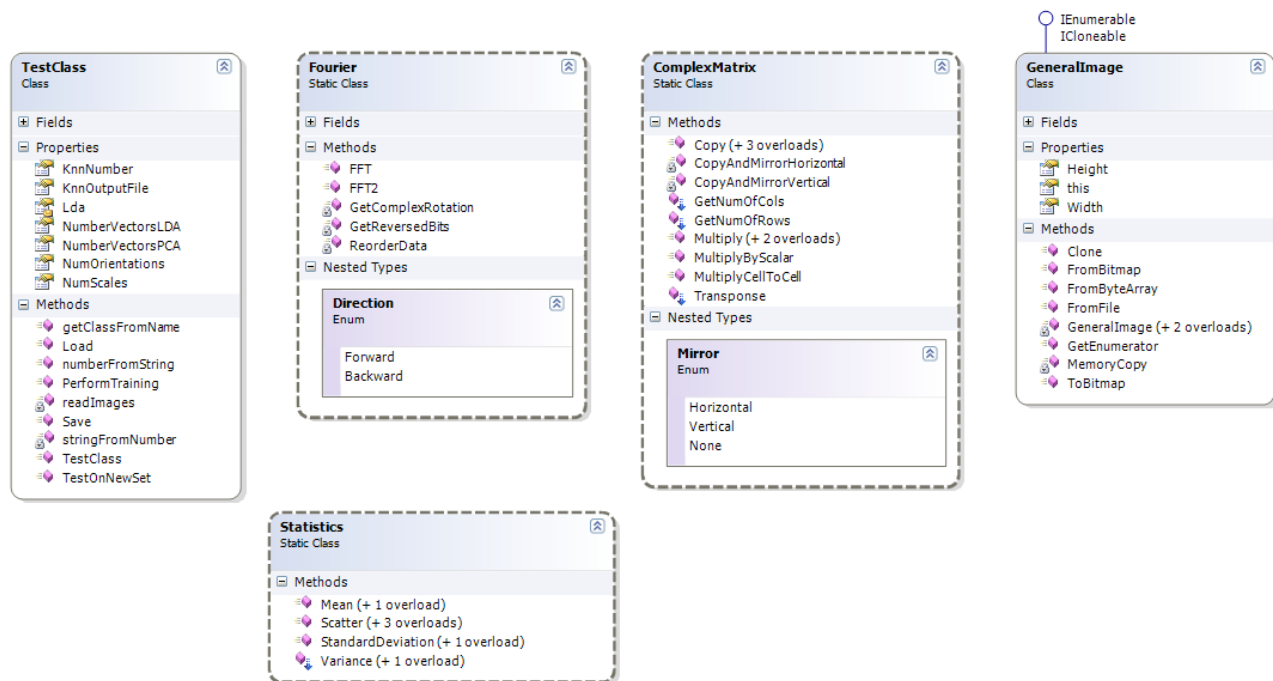
### 6.4. Ostali razredi

Osim navedenih, potrebne su još neki razredi, poput razreda *GeneralImage*, koji predstavlja sliku koju obrađujemo. Statički razred *Statistics* sadrži metode za izračunavanje raznih statističkih parametara, poput srednje vrijednosti, varijance itd. Statički razred *Fourier* sadrži metode za izračunavanje diskretne Fourierove transformacije, u jednoj ili dvije dimenzije. Statički razred *ComplexMatrix* sadrži metode za rad s kompleksnim matricama. To uključuje metodu za množenje matrica, transponiranje itd. Navedeni razredi su prikazani na dijagramu na slici 14.



Slika 13: Dijagram razreda za klasifikaciju

Razred *TestClass* predstavlja kontroler, odnosno razred koji upravlja modelom, a čije metode su predviđene da se pozivaju iz dijela sustava koji implementira korisničko sučelje. Razred sadrži metode *Save* i *Load* koje služe za spremanje odnosno učitavanje razreda iz datoteke. Na taj način je moguće spremiti odnosno učitati cijeli klasifikator, budući da su nakon učenja svi parametri potrebni za klasifikaciju novih primjera zapisani ili u razredu *TestClass* ili u razredu *KNNClassifier*, a razred *TestClass* sadrži privatnu varijablu razreda *KNNClassifier* i oba se razreda mogu serijalizirati. Metoda *PerformTraining* nad danim primjerima provede učenje klasifikatora, dok metoda *TestOnNewSet* provede klasifikaciju danih primjera te izračuna broj točno klasificiranih primjera i broj pogrešno klasificiranih primjera te za svaku pogrešnu klasifikaciju obavijesti korisnika.



Slika 14: Dijagram ostalih razreda

## 6.5. Postupci učenja i klasifikacije

Postupak učenja je implementiran u metodi *PerformTraining*. Metoda prvo učitava slike iz direktorija s primjerima za učenje. Zatim se te slike filtriraju slogom Gaborovih filtara čime se generiraju vektori značajki. Dobivenim vektorima značajki se PCA postupkom smanji dimenzionalnost. Za provođenje PCA postupka koristi se implementacija istog u programskoj biblioteci *Accord.NET*. Tako dobivenim vektorima se dodatno smanjuje dimenzionalnost pomoću Fisherove linearne diskriminantne analize. Konačno dobiveni vektori se koriste za treniranje kNN klasifikatora.

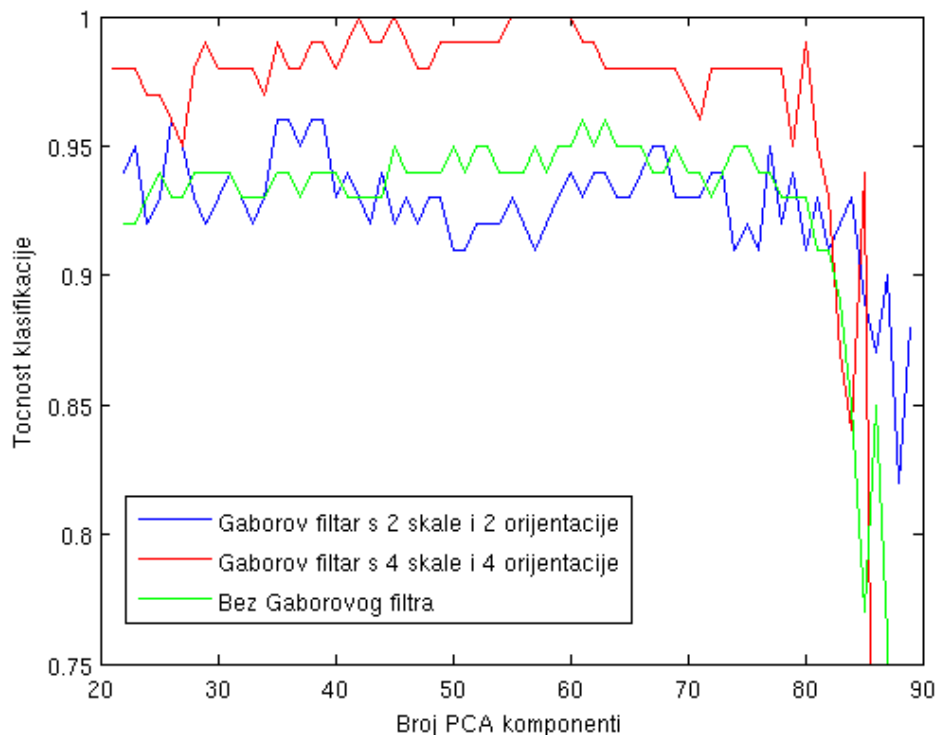
Postupak klasifikacije je gotovo pa identičan postupku učenja. Razlika je jedino u tome što se konačno dobiveni vektori ne koriste za učenje kNN klasifikatora već se klasifikator koristi za klasifikaciju tih vektora u konačne razrede.

## 7. Rezultati rada sustava

Kako je ostvaren sustav koji u sebi kombinira Gaborove filtre, PCA, LDA i k-NN postoji mnogo parametara koji se mogu po potrebi mijenjati te na taj način možemo utjecati na točnost klasifikacije. Samo testiranje je provedeno na bazi slika dostupnoj na stranici predmeta. Skup je podijeljen u dva dijela, skup za učenje te skup za testiranje. Cijela baza je podijeljena jednoliko pa su prema tome skup za učenje i skup za testiranje bili jednako brojni. Provedeno je više različitih mjerenja, a rezultati su dani u daljnjem tekstu.

Na grafu prikazanom na slici 15 vidimo usporedbu rada našeg sustava u ovisnosti o tome koristimo li četiri Gaborova filtra, šesnaest filtara ili ukoliko uopće ne koristimo Gaborove filtre. Mjerenja su provedena na skupu od 100 primjera za učenje i 100 primjera za testiranje zbog velike prostorne i vremenske složenosti Gaborovog filtriranja. Rezultati su očekivani te pokazuju ukoliko koristimo dovoljan broj Gaborovih filtara možemo postići točnost veću od 99%. Ako se ne koristi Gaborovi filtri, točnost klasifikacije je još uvijek relativno visoka, ali je otprilike 5% niža od slučaja kad se koriste Gaborovi filtri. Možemo primijetiti da se najveća točnost postiže za srednje velik broj PCA komponenti što je i očekivano ponašanje prema većini dostupnih znanstvenih radova.





*Slika 15: Usporedba točnosti klasifikacije u ovisnosu o broju korištenih Gaborovih filtara*

Osim navedenog eksperimenta testiranje smo proveli i na cijelom raspoloživom skupu u kojem smo imali 1180 primjera za učenje te 1180 primjera za testiranje. Ukoliko smo koristili samo PCA bez Gaborovih filtara, dobili smo točnost od 88.32% što je i očekivani rezultat jer pokazuje kako linearne metode klasifikacije imaju "problema" s linearno neodvojivim razredima. Cilj uvođenja Gaborovih filtara je bio jasan, željeli smo provesti nelinearnu transformaciju ulaznih podataka te smo time željeli povećati točnosti. Rezultati to i potvrđuju pa smo postigli točnost od 94.67 %. Moguće je dobiti i bolji rezultat od ovog, ali je vremenska i prostorna cijena tog postupka previsoka kako bismo opravdali manja poboljšanje točnosti klasifikacije.

## 8. Zaključak

U ovom projektu je implementiran sustav za raspoznavanje lica. Sustav koristi slog Gaborovih filtara za generiranje značajki slike. Budući da je dimenzionalnost vektora tako dobivenih značajki vrlo velika, potrebno je smanjiti dimenzionalnost vektora kako bi postupak klasifikacije bio moguć. Dimenzionalnost se smanjuje u dva koraka. U prvom koraku se koristi PCA postupak za smanjenje dimenzionalnosti na način da se minimizira rekonstrukcijska pogreška izvornih značajki. U drugom koraku se koristi višestruka diskriminantna analiza kako bi se povećala odvojivost među razredima. Konačno dobiveni vektor značajki se klasificira pomoću kNN klasifikatora.

Rezultati rada sustava su zadovoljavajući te pokazuju kako se jednostavne linearne metode klasifikacije mogu koristiti za uspješno raspoznavanje ljudskih lica. Osim toga pokazali smo kako nelinearne transformacije povećavaju točnost klasifikacije, ali pod cijenu velike vremenske i prostorne složenosti.

## 9. Literatura

- [1] Petkov, Grigorescu; *2-D Gabor function - interactive visualization*,  
<http://www.cs.rug.nl/~imaging/simplecell.html>
- [2] *Web enabled image processing operators*, <http://matlabserver.cs.rug.nl/>
- [3] *Gabor Wavelets*, <http://www.ansatt.hig.no/erikh/papers/scia99/node6.html>
- [4] *Gabor Filter* - <http://www.neuroinformatik.ruhr-uni-bochum.de/VDM/research/computerVision/imageProcessing/wavelets/gabor/gaborFilter.html>
- [5] Kepenekci B.; *Face Recognition Using Gabor Wavelet Transform*, 2001.,  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.6259&rep=rep1&type=pdf>
- [6] *Web aplikacija* - <http://matlabserver.cs.rug.nl/cgi-bin/matweb.exe>
- [7] *Gabor Wavelet transformation*, <http://www.neuroinformatik.ruhr-uni-bochum.de/VDM/research/computerVision/imageProcessing/wavelets/gabor/contents.html>
- [8] Welling, Max; *Fisher Linear Discriminant Analysis.*, Department of Computer Science, University of Toronto, 2001.
- [9] Gutierrez-Osuna, Ricardo; *Introduction to Pattern Analysis.*, Texas A&M University, 2003.
- [10] M. Turk, A. Pentland, *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience, Vol. 3, No. 1, 1991, pp. 71-86
- [11] M.A. Turk, A.P. Pentland, *Face Recognition Using Eigenfaces*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3-6 June 1991, Maui, Hawaii, USA, pp. 586-591
- [12] A. Pentland, B. Moghaddam, T. Starner, *View-Based and Modular Eigenspaces for Face Recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 21-23 June 1994, Seattle, Washington, USA, pp. 84-91
- [13] K. Etemad, R. Chellappa, *Discriminant Analysis for Recognition of Human Face Images*, Journal of the Optical Society of America A, Vol. 14, No. 8, August 1997, pp. 1724-1733
- [14] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, *Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection*, Proc. of the 4th European Conference on Computer Vision,

ECCV'96, 15-18 April 1996, Cambridge, UK, pp. 45-58

[15] Wikipedia, *k-nearest neighbor algorithm*, <http://en.wikipedia.org/wiki/KNN>

[16] S. Ribarić, *Predavanja iz kolegija Raspoznavanje uzoraka*, FER, 2010.