**math.e****Hrvatski matematički elektronički časopis**

Prepoznavanje lica pomoću tenzorske dekompozicije singularnih vrijednosti

optimizacija prepoznavanje slika tenzorska dekompozicija singularnih vrijednosti

Ivan Zirdum i Nela Bosner

Moderna tehnologija i stalan porast broja stanovnika na Zemlji razlog su velike produkcije podataka. Svaki dan se proizvede ogromna količina novih podataka koje je potrebno spremati, obraditi, a često i slati. Zbog toga je bilo potrebno osmisliti načine za efikasno spremanje podataka kako bi oni zauzimali što je manje moguće prostora, a da glavna informacija bude očuvana. Lijep primjer toga je kompresija digitalne slike. Uzmimo za primjer sliku dimenzije 3000×2000 piksela. Kako bismo takvu sliku spremili u memoriju potrebno nam je $3000 \cdot 2000 \cdot 24 = 144000000$ bitova memorije (za pamćenje jednog piksela koristimo 24 bita) što iznosi 18 megabajta.

U ovom članku bavit ćemo se matricnom (tenzorskom) dekompozicijom koja nam omogućava da iz nekog skupa podataka, prikazanog pomoću matrice (tenzora), izdvojimo "najvažniji" dio, a ostatak izostavimo. Koristeći tu tehniku možemo komprimirati slike, razviti algoritme koji: klasificiraju rukom pisane znamenke, prepoznaju ljudska lica i još mnogo toga. Zamislite sada da ste zaposleni u pošti i da je vaša zadaća sortirati pristigla pisma prema poštanskom broju kako bi ona stigla na pravu adresu. Možda biste radije bili prometni policajac na autocesti koji je zadužen za praćenje brzine automobila koje ulaze u tunel. Vaš posao je snimiti tablice automobila koji ne poštuju ograničenje brzine, a zatim u bazi podataka registracija vozila pronaći tu tablicu te na priloženu adresu vlasnika poslati kaznu za prebrzu vožnju. Biste li svaki dan isti posao radili ručno ili biste radije pokušali taj postupak automatizirati? Srećom, ljudi su se već susreli s ovim problemom i smislili mnogo načina kako taj posao ubrzati i uštediti dragocjeno vrijeme. Svi navedeni problemi mogu se riješiti upotrebom jednog moćnog alata numeričke matematike koji ćemo vam pokušati približiti u ostatku ovog članka.

1 Matricna dekompozicija singularnih vrijednosti (SVD)

Jedna od najkorisnijih dekompozicija i s teorijske strane (za dokazivanje) i s praktične strane, je dekompozicija singularnih vrijednosti (engl. singular value decomposition) ili, skraćeno, SVD. Sljedeći teorem pokazuje da za svaku matricu postoji takva dekompozicija.

Teorem 1. (SVD) Neka su m i $n, n \geq m$ prirodni brojevi te A proizvoljna $n \times m$ realna matrica. Tada postoji dekompozicija

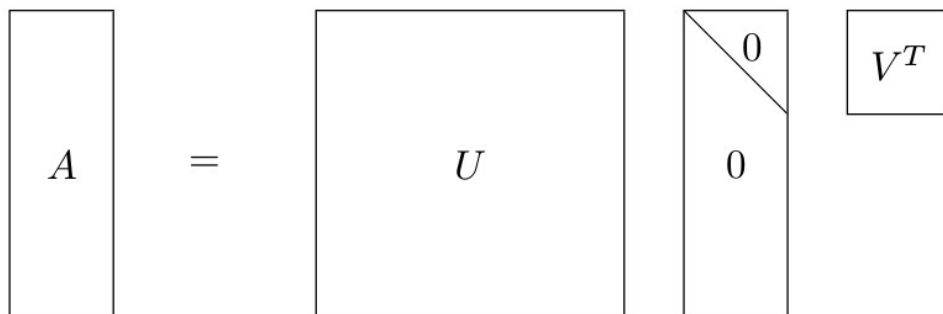
$$A = U\Sigma V^T, \quad (1)$$

gdje su $U \in \mathbb{R}^{n \times n}$ i $V \in \mathbb{R}^{m \times m}$ ortogonalne matrice, a $\Sigma \in \mathbb{R}^{n \times m}$ dijagonalna matrica,

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_m),$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m \geq 0.$$

Ilustrirajmo SVD:

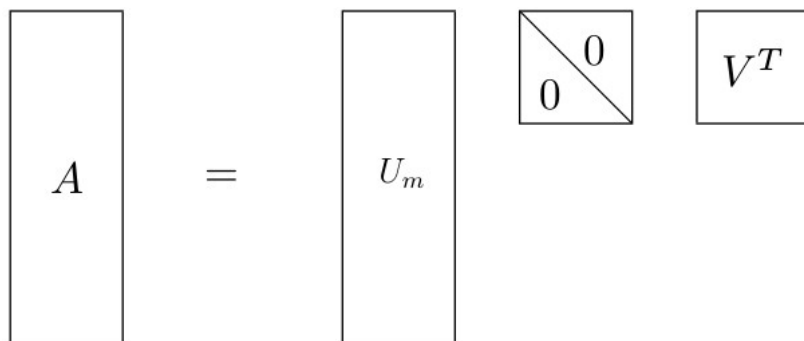


Slika 1: $A \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times m}$, $V^T \in \mathbb{R}^{m \times m}$.

Ako U zapišemo kao $U = (U_m U_m^\perp)$, gdje je $U_m \in \mathbb{R}^{n \times m}$, a Σ zapišemo kao $\begin{pmatrix} \Sigma_m \\ 0 \end{pmatrix}$ gdje je $\Sigma_m \in \mathbb{R}^{m \times m}$, tada dobijemo tanki SVD.

$$A = U_m \Sigma_m V^T,$$

Ilustrirajmo sada tanki SVD:



Slika 2: $A \in \mathbb{R}^{n \times m}$, $U_m \in \mathbb{R}^{n \times m}$, $\Sigma_m \in \mathbb{R}^{m \times m}$, $V^T \in \mathbb{R}^{m \times m}$.

Jako važnu primjenu SVD-a daje sljedeći teorem.

Teorem 2. Neka je $A = U_m \Sigma_m V^T$ dekompozicija singularnih vrijednosti (SVD) realne matrice A tipa $n \times m$, $n \geq m$. Zapišimo matrice U_m i V iz SVD-a od A u stupčanom obliku

$$U_m = (u_1, \dots, u_m), \quad V = (v_1, \dots, v_m).$$

Onda matricu A možemo pisati kao zbroj matrica ranga 1

$$A = U_m \Sigma_m V^T = \sum_{i=1}^m \sigma_i u_i v_i^T.$$

$$A = \sum_{i=1}^m \sigma_i u_i v_i^T = \left| \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \right| + \left| \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \right| + \dots$$

Matricu A_k , istog tipa kao i A , ranga $\text{rang}(A_k) \leq k < m$, koja je po 2-normi najbliža matrici A , možemo zapisati kao

$$A_k = U_m \Sigma_k V^T = \sum_{i=1}^k \sigma_i u_i v_i^T$$

pri čemu je

$$\Sigma_k = \text{diag}(\sigma_1, \dots, \sigma_k, 0, \dots, 0).$$

Pritom je

$$\|A - A_k\|_2 = \sigma_{k+1}$$

najmanja udaljenost između A i svih matrica ranga najviše k .

1.1 Primjena dekompozicije matrice singularnih vrijednosti

Jedna od primjena prethodnog teorema je kompresija slika. Neka je dana $n \times m$ matrica A s elementima $a_{i,j} \in [0, 1]$. Zamislimo da A predstavlja $n \times m$ grayscale (crno-bijelu) sliku, gdje broj na (i, j) mjestu u matrici predstavlja intenzitet svjetlosti piksela na mjestu (i, j) na slici. Ako uzmemo da vrijednost nula predstavlja crno, a jedan bijelo, zanimljivo je pitanje što će se dogoditi kad napravimo $A = U \Sigma V^T$, a onda u matrici Σ posljednjih nekoliko ne-nul elemenata stavimo na nulu te promotrimo dobivenu sliku A_k , gdje je $A_k = U \Sigma_k V^T = \sum_{i=1}^k \sigma_i u_i v_i^T$, a $\Sigma_k = \text{diag}(\sigma_1, \dots, \sigma_k, 0, \dots, 0)$.



Slika 3: Testna slika A dimenzije 376×350 u JPG formatu zauzima 129 kb.

Preciznije, neka je $A = U\Sigma V^T$, SVD matrice A . Prema teoremu 2, vrijedi da je $A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$ najbolja aproksimacija matrice A matricom ranga k , u smislu $\|A - A_k\| = \sigma_{k+1}$. Da bismo spremili podatke u_1, \dots, u_k te $\sigma_1 v_1, \dots, \sigma_k v_k$ iz kojih možemo dobiti matricu A_k potrebno je pamtit i samo $(m + n) \cdot k$ brojeva budući da su u_i i v_i vektori iz \mathbb{R}^m odnosno iz \mathbb{R}^n . S druge strane, za spremanje matrice A potrebno je pamtit i svih $m \cdot n$ brojeva. Stoga, koristimo A_k kao kompresiju slike A .



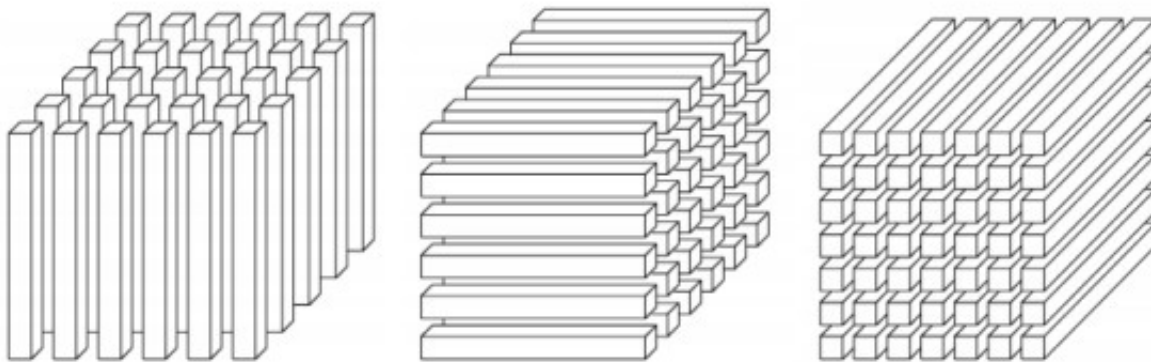
Slika 4: Kompresija A_k s a) $k = 50$, b) $k = 25$, c) $k = 15$, a zauzimaju redom 35 kb, 18 kb te 11 kb.

2 Tenzori

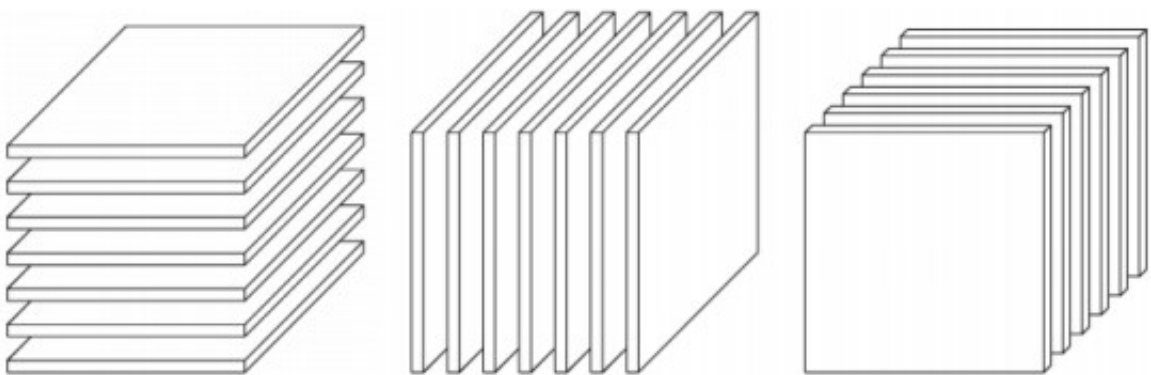
Do sada smo razmatrali linearnu algebru, gdje su glavni objekti vektori i matrice. Njih možemo smatrati kao jednodimenzionalna i dvodimenzionalna polja podataka. Tenzor je višedimenzionalni ekvivalent vektora i matrice i možemo ga predstaviti višedimenzionalnim poljem brojeva. Red tenzora jednak je broju indeksa potrebnih za indeksiranje njegovih elemenata. Primjerice, skalar možemo smatrati tenzorom reda 0, vektor tenzorom reda 1, a matricu tenzorom reda 2. Tenzori reda većeg ili jednakog 3 nazivaju se tenzorima višeg reda. Preciznije, tenzor reda N je element tenzorskog produkta N vektorskih prostora.

Red tenzora $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ je N . Elemente od \mathcal{A} označavamo s $\mathcal{A}_{i_1 i_2 \dots i_n}$, gdje su $1 \leq i_n \leq I_n$, $n = 1, 2, \dots, N$.

Svaki indeks u tenzoru nazivamo mod, a dimenzija pripadnog moda je broj različitih vrijednosti koje taj indeks može poprimiti. Unutar tenzora možemo indeksirati podtenzore ograničavanjem pojedinih indeksa. Primjerice, za tenzor $\mathcal{A} = (a_{i_1 i_2 i_3}) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ reda 3 možemo fiksiranjem indeksa u modu 1, 2 ili 3 definirati podtenzore kao $\mathcal{A}_{i_1=n} = \mathcal{A}_{n::}$, $\mathcal{A}_{i_2=n} = \mathcal{A}_{:n:}$ ili $\mathcal{A}_{i_3=n} = \mathcal{A}_{::n}$. Primjetimo da su u tom slučaju dva indeksa slobodna pa su podtenzori zapravo novi tenzori reda 2, odnosno matrice. Posebno za slučaj tenzora reda 3 te podtenzore nazivamo horizontalnim, lateralnim i frontalnim odsječcima. Također možemo definirati i vektore po modu n na način da fiksiramo sve indekse osim jednoga, npr. vektor u modu 1 se dobije kao $\mathcal{A}_{:i_2 i_3}$. Analogno za tenzor reda N možemo definirati podtenzore reda manjeg ili jednakog N . Na sljedećoj slici su prikazani primjeri podtenzora za tenzor reda 3.



Slika 5: Primjeri podtenzora. Vektorski tenzora u: modu 1, modu 2 i modu 3.



Slika 6: Primjeri podtenzora. Horizontalni, lateralni i frontalni odsječci.

MATLAB kod

```
function Av=vekt_tenz_mod(A,tmod,j,k)
% Av=vekt_tenz_mod(A,mod,i1,i2) vraća vektor tenzora A reda
% 3 u modu tmod, sa preostalim indeksima fiksiranim na j i
% k.

tdim=size(A);
ind=1:3;
ip=ind(ind=tmod);
Ap=permute(A,[tmod,ip]);
Av=reshape(Ap(:,j,k),tdim(tmod),1);

end
```

MATLAB kod

```
function Ao=ods_j_tenz_mod(A,tmod,i)
% Ao=ods_j_tenz_mod(A,tmod,j,k) vraća odsjecak tenzora A
% reda 3 u modu tmod, za i_tmod=i.

tdim=size(A);
ind=1:3;
ip=ind(ind=tmod);
Ap=permute(A,[ip,tmod]);
Ao=reshape(Ap(:, :, i),tdim(ip));

end
```

često je korisno prikazati tenzor u obliku matrice. Stoga definiramo matricu tenzora po modu n kao matricu $A_{(n)}$ koja se dobije tako da vektore iz moda n složimo kao stupce u matricu $A_{(n)}$. Poredak kojim se vektori iz moda n preslikavaju u stupce nije bitan, dok god se isti poredak koristi u svim izračunima.

Matricizacija tenzora (eng. unfold) u modu n od \mathcal{A} označava se s

$$A_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)}.$$

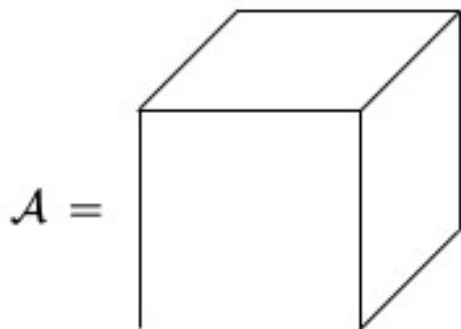
Tenzorski produkt u modu n tenzora $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ s matricom $U \in \mathbb{R}^{J_n \times I_n}$, označen s $\mathcal{A} \times_n U$, je

$$(I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)$$

tenzor čiji elementi su zadani s

$$(\mathcal{A} \times_n U)_{i_1 i_2 \dots i_{n-1} j_n i_{n+1} \dots i_N} \stackrel{def}{=} \sum_{i_n} a_{i_1 i_2 \dots i_{n-1} i_n i_{n+1} \dots i_N} u_{j_n i_n}.$$

Zbog jednostavnosti ograničavamo se na tenzore reda 3, $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Takav tenzor možemo vizualizirati na sljedeći način:



Uvedimo sada osnovne pojmove za tenzor reda 3, $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Dimenzije tenzora $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ su: I_1 , I_2 i I_3 .

2.1 Osnovni koncepti tenzora

Prvo definiramo skalarni produkt dvaju tenzora istog reda i jednakih dimenzija:

Definicija 3. *Skalarni produkt dvaju tenzora istih dimenzija $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ je suma umnožaka njihovih elemenata, tj.*

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i,j,k} a_{ijk} b_{ijk}. \quad (2)$$

Pripadna norma je

Definicija 4. *Norma tenzora $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ je kvadratni korijen sume svih kvadrata njegovih elemenata, tj.*

$$\|\mathcal{A}\| = \langle \mathcal{A}, \mathcal{A} \rangle^{1/2} = \left(\sum_{i,j,k} a_{ijk}^2 \right)^{1/2}.$$

Ako gledamo gornje definicije na matricama (dvodimenzionalnim tenzorima) vidimo da je to Frobeniusova norma.

Sljedeće što definiramo je množenje u modu 1 matrice s tenzorom.

Definicija 5. Množenje u modu 1 tenzora $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ matricom $U \in \mathbb{R}^{l_0 \times I_1}$ u oznaci $\mathcal{A} \times_1 U$, je $l_0 \times I_2 \times I_3$ tenzor dan s

$$(\mathcal{A} \times_1 U)(j, i_2, i_3) = \sum_{k=1}^{I_1} u_{j,k} a_{k,i_2,i_3}.$$

Usporedbe radi, uočimo da za množenje matrica vrijedi

$$A \times_1 U = UA, \quad (UA)(i, j) = \sum_{k=1}^l u_{i,k} a_{k,j}.$$

Znamo da je množenje matrica ekvivalentno množenju svakog stupca iz \mathcal{A} matricom U . Isto vrijedi i za množenje tenzora matricom u modu 1. Svi stupčani vektori iz tenzora \mathcal{A} se množe matricom U .

Slično množenje u modu 2 tenzora s matricom V

$$(\mathcal{A} \times_2 V)(i_1, j, i_3) = \sum_{k=1}^{I_2} v_{j,k} a_{i_1,k,i_3}$$

znači da su svi retci tenzora pomnoženi s matricom V . Opet uočavamo da je množenje u modu 2 matrice s V ekvivalentno matičnom množenju s V^T zdesna,

$$A \times_2 V = AV^T;$$

Množenje u modu 3 se dobije analogno.

Ponekad je zgodno matricizirati tenzor u matricu.

Matricizacija tenzora \mathcal{A} je definirana kroz tri moda (koristeći MATLAB-ovu notaciju):

$$\begin{aligned} \text{unfold}_1(\mathcal{A}) := A_{(1)} &:= (\mathcal{A}(:, 1, :), \mathcal{A}(:, 2, :), \dots, \mathcal{A}(:, I_2, :)) \in \mathbb{R}^{I_1 \times I_2 I_3}, \\ \text{unfold}_2(\mathcal{A}) := A_{(2)} &:= (\mathcal{A}(:, :, 1)^T, \mathcal{A}(:, :, 2)^T, \dots, \mathcal{A}(:, :, I_3)^T) \in \mathbb{R}^{I_2 \times I_1 I_3}, \\ \text{unfold}_3(\mathcal{A}) := A_{(3)} &:= (\mathcal{A}(1, :, :)^T, \mathcal{A}(2, :, :)^T, \dots, \mathcal{A}(I_1, :, :)^T) \in \mathbb{R}^{I_3 \times I_1 I_2}. \end{aligned}$$

Vrijedi i sljedeće:

- (1) {Stupčani vektori od \mathcal{A} su stupčani vektori od $A_{(1)}$ }.
- (2) {Retci od \mathcal{A} su stupčani vektori od $A_{(2)}$ }.
- (3) {Prostorni vektori od \mathcal{A} su stupčani vektori od $A_{(3)}$ }.

Unfold_1 od \mathcal{A} je ekvivalentan dijeljenju tenzora na odsječke $\mathcal{A}(:, i, :)$ (koji su matrice) i slaže ih u dugu matricu $A_{(1)}$.

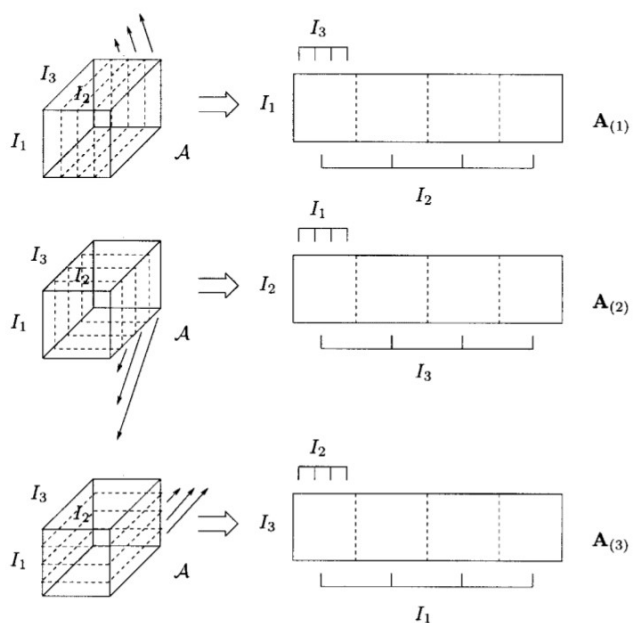
MATLAB kod

```

function Au=tenz_unfold(A,tmod)
% Au=tenz_unfold(A,tmod) vraća matricizaciju tenzora A reda
% 3 po modu tmod.

tdim=size(A);
Au=[];
io=mod(tmod,3)+1;
for i=1:tdim(io)
    if (tmod==1)
        Au=[Au,odsj_tenz_mod(A,2,i)];
    else
        Au=[Au,odsj_tenz_mod(A,io,i)'];
    end
end
end
end

```



Slika 7: Matricizacija $(I_1 \times I_2 \times I_3)$ tenzora \mathcal{A} na $(I_1 \times I_2 I_3)$ matricu $\mathbf{A}_{(1)}$, $(I_2 \times I_3 I_1)$ matricu $\mathbf{A}_{(2)}$ i $(I_3 \times I_1 I_2)$ matricu $\mathbf{A}_{(3)}$ ($I_1 = I_2 = I_3 = 4$).

Kao što nam je korisno prikazati tenzor u obliku matrice tako nam je ponekad korisno tenzorificirati matricu odnosno matricu pretvoriti u tenzor. Tenzorifikacija je inverz matricizacije odnosno operator fold je inverz operatora unfold pa vrijedi

$$\text{fold}_i(\text{unfold}_i(\mathcal{A})) = \mathcal{A}.$$

MATLAB kod

```

function A=tenz_fold(Au,tmod,tdim)
% A=tenz_fold(Au,tmod,tdim) tenzorificira matricu Au koja
% je matricizirana pomocu Au=tenz_unfold(A,tmod), natrag u
% polazni tenzor dimenzija tdim(1) x tdim(2) x tdim(3).

A=[];
[dim1,dim2]=size(Au);
if (dim1==tdim(tmod) && dim2==prod(tdim(tdim=tmod)))
    fprintf(1,'Krive dimenzije!\n');
    return;
end

io=mod(tmod,3)+1; ib=mod(tmod-2,3)+1; bdim=tdim; bdim(io)=1;
for i=1:tdim(io)
    switch(tmod)
        case 1
            A(:,i,:)=reshape(Au(:,(i-1)*tdim(ib)+1:
                                i*tdim(ib)),bdim);
        case 2
            A(:, :, i)=reshape(Au(:,(i-1)*tdim(ib)+1:
                                i*tdim(ib))',bdim);
        case 3
            A(i, :, :)=reshape(Au(:,(i-1)*tdim(ib)+1:
                                i*tdim(ib))',bdim);
    end
end
end

```

Korištenjem unfold-fold operacija, možemo formulirati matrično množenje ekvivalentno u modu i tenzorskom množenju:

$$\mathcal{A} \times_i U = \text{fold}_i(U \text{ unfold}_i(\mathcal{A})) = \text{fold}_i(UA_{(i)}). \quad (3)$$

MATLAB kod

```

function B = tenz_mult(A, U, tmod)
% B = tenz_mult(A, U, mod) vraca tenzorski produkt u modu
% tmod tenzora A reda 3 sa matricom U.

B=[];
Au=tenz_unfold(A,tmod);
udim=size(U);
audim=size(Au);

if (udim(2)=audim(1))
    fprintf(1,'Krive dimenzije!\n');
    return;
end

tdim = size(A);
switch(tmod)
    case 1
        B=tenz_fold(U*Au,tmod,[udim(1),tdim(2:3)]);
    case 2
        B=tenz_fold(U*Au,tmod,[tdim(1),udim(1),tdim(3)]);
    case 3
        B=tenz_fold(U*Au,tmod,[tdim(1:2),udim(1)]);
end

end

```

2.2 Tenzorski SVD (HOSVD)

Matrični SVD može se generalizirati na više načina kako bi dobili tenzorski. Mi promatramo jedan koji se često naziva "Higher order SVD" (HOSVD).

Vidjeli smo da matrice možemo promatrati kao tenzore reda 2. Koristeći tenzorsku notaciju, jednadžbu (1) iz Teorema 1 (SVD) sada možemo zapisati kao

$$A = U\Sigma V^T = \Sigma \times_1 U \times_2 V = S \times_1 U^{(1)} \times_2 U^{(2)}$$

isto vrijedi i za tenzore višeg reda. O tome nam više govori sljedeći teorem.

Teorem 6. (HOSVD) Tenzor $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$ može se zapisati kao

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}, \quad (4)$$

gdje su $U^{(1)} \in \mathbb{R}^{l \times l}$, $U^{(2)} \in \mathbb{R}^{m \times m}$ i $U^{(3)} \in \mathbb{R}^{n \times n}$ ortogonalne matrice. \mathcal{S} je tenzor istih dimenzija kao \mathcal{A} . Svaka dva odsječka tenzora \mathcal{S} su ortogonalna u smislu skalarnog produkta:

$$\langle \mathcal{S}(i, :, :), \mathcal{S}(j, :, :) \rangle = \langle \mathcal{S}(:, i, :), \mathcal{S}(:, j, :) \rangle = \langle \mathcal{S}(:, :, i), \mathcal{S}(:, :, j) \rangle = 0$$

za $i \neq j$. Singularne vrijednosti u modu 1 su definirane s

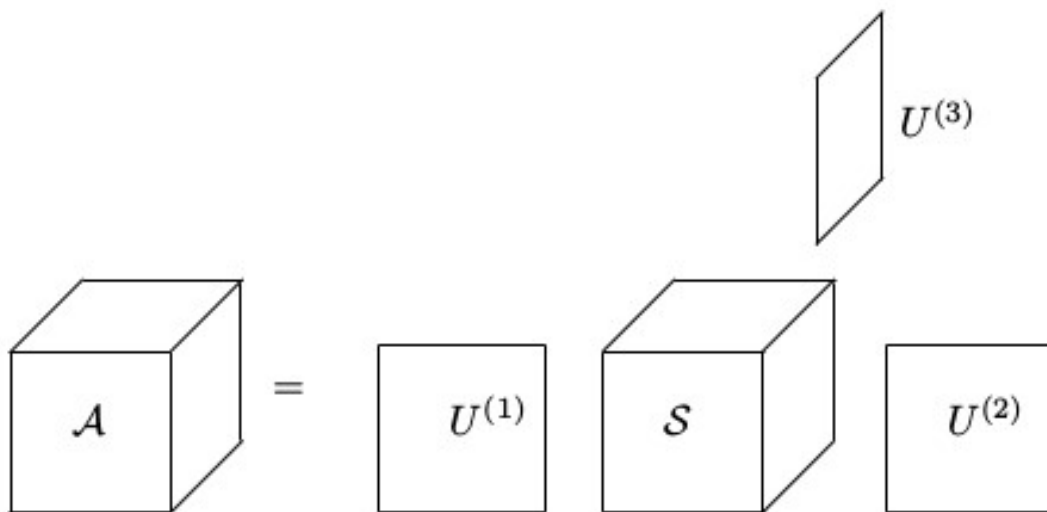
$$\sigma_j^{(1)} = \|\mathcal{S}(j, :, :)\|_F, \quad j = 1, \dots, l$$

i one su sortirane silazno

$$\sigma_1^{(1)} \geq \sigma_2^{(1)} \geq \dots \geq \sigma_l^{(1)}. \quad (5)$$

Singularne vrijednosti u ostalim modovima i njihovo sortiranje je analogno.

Ortogonalni tenzor \mathcal{S} nazivamo jezgri tenzor tenzora \mathcal{A} . HOSVD je vizualiziran na Slici 8.



Slika 8: Vizualizacija HOSVD-a.

Koristeći tenzorsku dekompoziciju, svaki tenzor reda N možemo prikazati kao produkt

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 \dots \times_N U^{(N)} \quad (6)$$

gdje nam je \mathcal{S} jezgri tenzor, a U_n , $n = 1, 2, \dots, N$ ortogonalne matrice.

Iz dokaza prethodnog teorema može se zapravo vidjeti kako se računa HOSVD određenog tenzora \mathcal{A} :

- (1) Matrica $U^{(i)}$ u modu i može se direktno naći kao lijeva singularna matrica od matricizacije tenzora \mathcal{A} u modu i :

$$A_{(i)} = U^{(i)} \Sigma^{(i)} V^{(i)T}.$$

- (2) Nakon toga, jezgreni tenzor \mathcal{S} može se izračunati prebacivanjem matrica singularnih vektora na lijevu stranu u jednadžbi (6):

$$\mathcal{S} = \mathcal{A} \times_1 U^{(1)T} \times_2 U^{(2)T} \times_3 \dots \times_N U^{(N)T}.$$

Zbog toga se računanje HOSVD-a tenzora reda N svodi na računanje N različitih matricnih SVD-a matrica formata $(I_n \times I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N)$.

Ponekad se događa da je dimenzija od jednog moda veća od produkta dimenzija drugih modova. Pretpostavimo, npr. da je $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$ gdje je $l > mn$. Može se pokazati da tada za tenzor \mathcal{S} vrijedi

$$\mathcal{S}(i, :, :) = 0, \quad i > mn, \quad (7)$$

i možemo odbiti dio s nulama tenzora \mathcal{S} te (4) napisati kao tanki HOSVD

$$\mathcal{A} = \hat{\mathcal{S}} \times_1 \hat{U}^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}, \quad (8)$$

gdje je $\hat{\mathcal{S}} \in \mathbb{R}^{mn \times m \times n}$ i $\hat{U}^{(1)} \in \mathbb{R}^{l \times mn}$.

MATLAB kod

```

function [S,U1,U2,U3]=tenz_hosvd(A)
% [S,U1,U2,U3]=tenz_hosvd(A) racuna HOSVD tenzora A reda 3.

[U1,S1,V1]=svd(tenz_unfold(A,1));
[U2,S2,V2]=svd(tenz_unfold(A,2));
[U3,S3,V3]=svd(tenz_unfold(A,3));
S=tenz_mult(tenz_mult(tenz_mult(A,U1',1),U2',2),U3',3);
sdim=size(S);

%Slucaj tankog HOSVD-a
psd12=sdim(1)*sdim(2);
psd13=sdim(1)*sdim(3);
psd23=sdim(2)*sdim(3);

if (sdim(1)>psd23)
S=S(1:psd23,1:sdim(2),1:sdim(3));
U1=U1(1:sdim(1),1:psd23);
return;
end

if (sdim(2)>psd13)
S=S(1:sdim(1),1:psd13,1:sdim(3));
U2=U2(1:sdim(2),1:psd13);
return;
end

if (sdim(3)>psd12)
S=S(1:sdim(1),1:sdim(2),1:psd12);
U3=U3(1:sdim(3),1:psd12);
return;
end

```

2.3 Aproximacija tenzora HOSVD-om

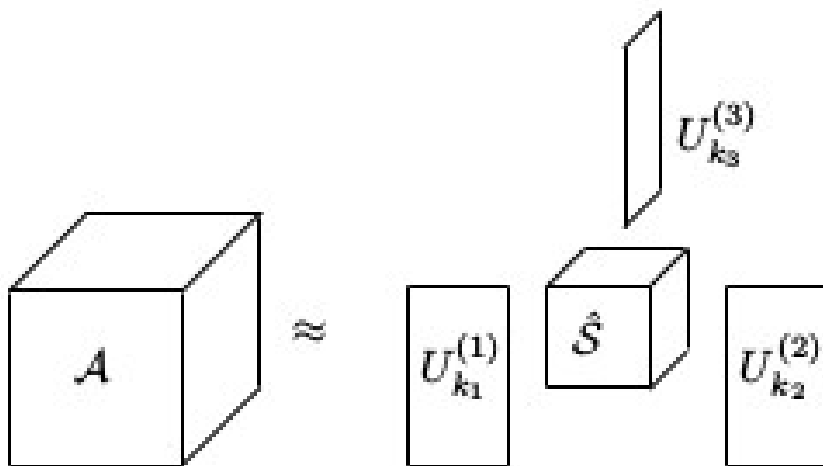
I HOSVD se može koristiti za kompresiju tenzora. Budući da je masa jezgrenog tenzora \mathcal{S} koncentrirana za male vrijednosti od sva tri indeksa, moguće je "stisnuti" podatke u sva tri moda HOSVD-om. Pretpostavimo sada da smo podatke stisnuli u manji tenzor gdje ćemo

imati k_i stupaca u modu i . Neka je $U_{k_i}^{(i)} = U^{(i)}(:, 1 : k_i)$ i

$\hat{\mathcal{S}} = \mathcal{S}(1 : k_1, 1 : k_2, 1 : k_3)$. Zatim razmotrimo aproksimaciju

$$\mathcal{A} \approx \hat{\mathcal{A}} = \hat{\mathcal{S}} \times_1 U_{k_1}^{(1)} \times_2 U_{k_2}^{(2)} \times_3 U_{k_3}^{(3)}.$$

Ilustrirajmo to na sljedećoj slici:



Na ovaj način smo uz minimalne gubitke uštedjeli memoriju. Tenzor $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$ ima ukupno $l \cdot m \cdot n$ elemenata, dok za njegovu aproksimaciju $\hat{\mathcal{A}}$ moramo pamtitati ukupno $k_1 \cdot k_2 \cdot k_3 + l \cdot k_1 + m \cdot k_2 + n \cdot k_3$ elemenata.

Ako je tenzor $\mathcal{A} \in \mathbb{R}^{1000 \times 1000 \times 1000}$ onda on ima $1000 \cdot 1000 \cdot 1000 = 10^9$ elemenata, dok njegova aproksimacija $\hat{\mathcal{A}}$ za $k_1 = k_2 = k_3 = 100$ ima 10^6 elemenata u $\hat{\mathcal{S}}$, $1000 \cdot 100 = 10^5$ elemenata u $U_{k_1}^{(1)}$, $U_{k_2}^{(2)}$ i $U_{k_3}^{(3)}$, ukupno $1.3 \cdot 10^6$ elemenata. što znači da umjesto da koristimo 1 000 000 000 elemenata mi koristimo samo 1 300 000 elemenata, a to je ušteda od 997 000 000 elemenata, odnosno umjesto da zauzmemo 3GB memorije zauzimamo samo 3.9MB.

3 Prepoznavanje lica korištenjem HOSVD-a

Ljudska bića su vrlo vješta pri prepoznavanju lica čak i kada izraz lica, osvjetljenje, kut gledanja itd. variraju. Razviti automatske postupke za prepoznavanje lica koji su robusni s obzirom na različite uvjete, izazovan je istraživački problem koji je istražen pomoću nekoliko različitih pristupa. Analiza glavnih komponenti (eng. principal component analysis, PCA), temeljena na SVD-u, popularna je tehnika koja često ide pod imenom "eigenfaces". Međutim, ova metoda je najbolja kada su sve fotografije nastale pod sličnim uvjetima, i ne radi dobro kad se mijenjaju neki faktori okoline.

Nedavno su istraživane metode za multilinearu analizu skupa fotografija. Konkretno, problem prepoznavanja lica razmatran je korištenjem tenzorskog modela, TensorFaces pristupa. I to tako da modovi tenzora predstavljaju drukčiji način gledanja, npr. osvjetljenje ili izraz lica, postalo je moguće poboljšati preciznost algoritma prepoznavanja u usporedbi s PCA metodom.

U ovom ćemo odsječku opisati metodu tenzorskog prepoznavanja lica, koja se odnosi na TensorFaces. Budući da se radi o fotografijama, koje se često pohranjuju kao matrice $m \times n$, s m i n reda 100-500, izračuni za svako lice koje treba identificirati su prilično teški. Razmotrit ćemo kako se tenzorski SVD (HOSVD) također može koristiti za smanjenje dimenzije u svrhu smanjenja broja operacija.

3.1 Tenzorski prikaz fotografija

Pretpostavimo da imamo kolekciju fotografija od n_p osoba, gdje je svaka fotografija $m_{i_1} \times m_{i_2}$ matrica s $m_{i_1} m_{i_2} = n_i$ elemenata. Pretpostavljamo da su stupci fotografija složeni tako da svaka fotografija predstavlja vektor u \mathbb{R}^{n_i} . Nadalje pretpostavljamo da je svaka osoba fotografirana u n_e različitih izraza lica. često je $n_i \geq 5000$, i obično je n_i znatno veći od n_e i n_p . Takva kolekcija fotografija je spremljena u tenzor:

$$\mathcal{A} \in \mathbb{R}^{n_i \times n_e \times n_p} \quad (9)$$

Pozivamo se na različite modove kao što su mod fotografije (eng. the image mode), mod izraza lica (eng. the expression mod) i mod osobe (eng. the person mode), zbog toga koristimo indekse i, e, p u (9).

Ako, primjerice, također imamo fotografije svake osobe s različitim osvjetljenjem, kutem gledanja, itd., onda bi mogli predstavljati kolekciju fotografija tenzorom višeg reda. Radi jednostavnosti ovdje promatramo samo tenzore reda 3. Generalizacija tenzorima višeg reda je jasna.

Primjer 7. Prethodno smo obradili fotografije od 10 osoba iz baze podataka Yale Face koje smo postavili na 112×78 piksela i pohranili u vektor duljine 8736. Pet fotografija je prikazano na sljedećoj slici.



Slika 9: Osoba 1 s pet različitih izraza lica (iz Yale Face Database).

Svaka osoba je fotografirana u ukupno 11 različitih izraza lica.

Redoslijed modova je naravno proizvoljan; u svrhu ilustriranja pretpostavljat ćemo redoslijed iz (9). Međutim kako bi naglasili proizvoljnost, koristit ćemo oznaku \times_e za množenje tenzora matricom u modu izraza lica (the expression mode), i slično za druge modove. Sada pretpostavljamo da je $n_i \gg n_e n_p$ i pišemo tanki HOSVD (Pogledati Teorem 6 i (8)),

$$\mathcal{A} = \mathcal{S} \times_i F \times_e G \times_p H,$$

gdje je $\mathcal{S} \in \mathbb{R}^{n_e n_p \times n_e \times n_p}$ jezgra tenzora \mathcal{A} , $F \in \mathbb{R}^{n_i \times n_e n_p}$ ima ortogonalne stupce, i $G \in \mathbb{R}^{n_e \times n_e}$ i $H \in \mathbb{R}^{n_p \times n_p}$ su ortogonalne matrice.

3.2 Prepoznavanje lica

Sada ćemo razmatrati problem klasifikacije kako slijedi:

Za danu fotografiju nepoznate osobe, koju predstavlja vektor u \mathbb{R}^{n_i} odredit ćemo koju od n_p osoba predstavlja ili odlučiti da nepoznata osoba nije u bazi podataka.

Za klasifikaciju pišemo HOSVD (4) u sljedećem obliku:

$$\mathcal{A} = \mathcal{C} \times_p H, \quad \mathcal{C} = \mathcal{S} \times_i F \times_e G. \quad (10)$$

Za određeni izraz lica e imamo

$$\mathcal{A}(:, e, :) = \mathcal{C}(:, e, :) \times_p H. \quad (11)$$

Očito tenzore $\mathcal{A}(:, e, :)$ i $\mathcal{C}(:, e, :)$ možemo promatrati kao matrice, koje označavamo s A_e i C_e . Stoga, za sve izraze imamo linearne relacije

$$A_e = C_e H^T, \quad e = 1, 2, \dots, n_e \quad (12)$$

Imajmo na umu da se ista (ortogonalna) matrica H pojavljuje u svih n_e relacija. S $H^T = (h_1 \cdots h_{n_p})$, stupac p iz (12) može biti zapisan kao

$$a_p^{(e)} = C_e h_p \quad (13)$$

Možemo interpretirati (12) i (13) na sljedeći način:

Stupac p od matrice A_e sadrži sliku osobe p u izrazu e . Stupci matrice C_e su bazni vektori za izraz e , i red p matrice H , tj. h_p , sadrži koordinate fotografija osobe p u toj bazi. Nadalje isti h_p sadrži koordinate fotografija osobe p u svim izraznim bazama.

Zatim pretpostavimo da je $z \in \mathbb{R}^{n_i}$ slika nepoznate osobe u nepoznatom izrazu (iz n_e) i da ju želimo klasificirati. Taj z ćemo zvati testna slika. Očito, ako je slika osobe p u izrazu e , onda su koordinate vektora z u toj bazi jednake koordinatama vektora h_p . Dakle, možemo klasificirati z računanjem njegovih koordinata u svim bazama ekspresije i provjeravajući, za svaki izraz, podudaraju li se koordinate

od z (ili se gotovo podudaraju) s elementima bilo kojeg reda matrice H .

Koordinate od z u izraznoj bazi e mogu se naći rješavanjem problema najmanjeg kvadrata

$$\min_{\alpha_e} \|C_e \alpha_e - z\|_2 \quad (14)$$

Klasifikacijski algoritam

```

for  $e = 1, 2, \dots, n_e$ 
    Riješi  $\min_{\alpha_e} \|C_e \alpha_e - z\|_2$ 
    for  $p = 1, 2, \dots, n_p$ 
        if  $\|\alpha_e - h_p\|_2 < \text{tol}$ , onda klasificiraj kao osobu  $p$  i
    stani.
    end
end

```

Količina posla u ovom algoritmu je velika: za svaku test sliku z moramo riješiti n_e problema najmanjih kvadrata (14) s $C_e \in \mathbb{R}^{n_i \times n_p}$.

4 Primjeri dobiveni MATLAB rutinama

Program smo testirali na 10 osoba fotografiranih u 10 različitih ekspresija(izraza) lica. Fotografije su spremljene u matricu tako da svaki stupac predstavlja jednu fotografiju i to na način da prvih 10 stupaca predstavlja prvu osobu u svih 10 ekspresija, drugih 10 stupaca predstavlja drugu osobu itd. Svaka fotografija je dimenzije 64×64 što znači da je matrica dimenzije 4096×100 . Pogledajmo sada fotografije tih osoba u prvom izrazu lica:





Slika 10: 10 osoba u prvom izrazu lica.

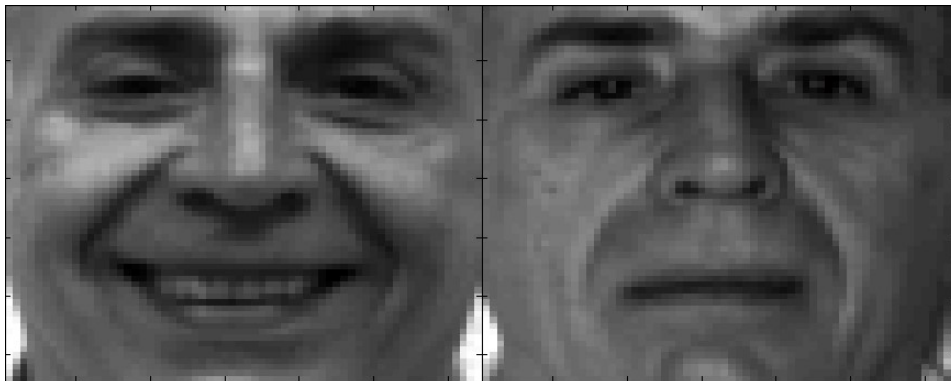
Svaka od tih osoba je fotografirana u 10 različitih izraza lica, pa pogledajmo na prvoj osobi koji su to sve izrazi: \nopagebreak[4]



4.1 Klasifikacija

Za potrebe prepoznavanja lica uzeli smo prvih 5 osoba u prve 3 ekspresije lica (15 fotografija) i stavili ih u bazu, te na njima "trenirali" program. Nakon toga smo izbacili te fotografije i na preostalim fotografijama testirali program. Preostalo je po 7 fotografija za svaku osobu koja je u bazi, a po 10 od osoba koje nisu u bazi (ukupno 35 fotografija osoba koje su u bazi i 50 fotografija osoba koje nisu u bazi). Naš cilj je da danu testnu fotografiju program klasificira kao jednu od tih 5 osoba ili da vrati: "Tražena osoba nije u bazi!". Odnosno, ukoliko je testna fotografija jedna od tih 7 fotografija da program prepozna tu osobu kao odgovarajuću osobu iz baze, a ako je jedna od 10 fotografija osoba koje nisu u bazi da vrati: "Tražena osoba nije u bazi!". Tolerancija u ovom radu iznosi 0.475, a do nje dolazimo eksperimentima.

Uzmimo sada za testnu fotografiju proizvoljnu fotografiju iz skupa od 35 fotografija osoba koje su u bazi i na njoj testirajmo program. Pogledajmo rezultate na sljedećim slikama: (Lijeva fotografija predstavlja osobu koju želimo klasificirati, a desna klasificiranu osobu.)





Vidimo da je program dobro prepoznao lica. Za sljedeće fotografije je program vratio: "Tražena osoba nije u bazi!".





što također vidimo da je dobro jer niti jedna od tih osoba nije u prvih 5 osoba na kojima smo trenirali naš program.

Napomenimo na kraju da program nije savršen i da se može dogoditi da krivo klasificira osobu. Primjer krive klasifikacije: Uzeli smo novu 11. osobu i pokušali je klasificirati.



Klasifikacija očito nije dobra jer to nisu iste osobe!

Da bismo to spriječili možemo smanjiti toleranciju, ali u tom slučaju se može dogoditi da osoba koja je u bazi ne bude prepoznata, odnosno da program vrati: "Tražena osoba nije u bazi!".

Pogledajmo i takav primjer: (Tolerancija u ovom primjeru iznosi 0.35).



Za gornju fotografiju je program vratio: "Tražena osoba nije u bazi!", a trebao ju je klasificirati kao:



Unatoč tome smatramo da je "manja" pogreška ako program ne prepozna osobu koju bi trebao, nego da osoba koja nije u bazi bude prepoznata kao neka osoba koja je u bazi. Iz tih razloga ipak uzimamo manju toleranciju do koje dolazimo eksperimentima.

Bibliografija

- [1] L. Elden, Matrix methods in data mining and pattern recognition, Philadelphia, SIAM., 2007.

- [2] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21:1253-1278, 2000.
- [3] J. Demmel. *Applied Numerical Linear Algebra*. SIAM . MIT , 1996.
- [4] T. G. Kolda, *Multilinear Operators for Higher Order Decompositions*, Tech. Report SAND2006-2081, 2006.
- [5] Mladen Rogina, Sanja Singer, Saša Singer. *Numerička analiza*. PMF - Matematički odsjek, Zagreb, 2003.
- [6] Z. Drmač. *Bilješke s predavanja iz kolegija Uvod u složeno pretraživanje podataka*, PMF - Matematički odsjek, Zagreb, 2016.
- [7] A. Novak, D. Pavlović. *Dekompozicija matrice na singularne vrijednosti i primjene*. math.e, Zagreb, 2013.

