

Tenzorske metode u prepoznavanju lica

Poštić, Suraja

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:774106>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-13**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Suraja Poštić

TENZORSKE METODE U
PREPOZNAVANJU LICA

Diplomski rad

Voditelj rada:
Prof. dr. sc. Zlatko Drmač

Zagreb, rujan, 2018.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Sadržaj	iii
Uvod	1
1 Tenzori	3
1.1 Definicija i osnovni pojmovi	3
1.2 Operacije s tenzorima	5
2 Dekompozicija singularnih vrijednosti (SVD)	7
2.1 Dekompozicija matrice na singularne vrijednosti	7
2.2 Aproksimacija nižeg ranga	10
3 SVD višeg reda (HOSVD)	13
3.1 HOSVD	13
3.2 Aproksimacija tenzora pomoću HOSVD-a	15
4 Prepoznavanje znamenki	17
4.1 Oblikovanje algoritma	18
4.2 Rezultati	21
5 Prepoznavanje lica	23
5.1 Oblikovanje algoritma	23
5.2 Rezultati	25
6 Uklanjanje utjecaja osvjetljenja	29
7 Implementacija algoritma u programskom jeziku R	31
7.1 Korišteni paketi i naredbe	31
7.2 HOSVD	32
7.3 Prepoznavanje znamenki	35
7.4 Prepoznavanje lica	40

7.5 Uklanjanje utjecaja osvjetljenja	48
Bibliografija	53

Uvod

Količina podataka dostupnih za obradu raste velikom brzinom. Prikladan način njihovog skladištenja ključan je za temeljitu i uspješnu analizu. Želimo li pohraniti sliku, prirodno ćemo ju zamisliti u obliku matrice. Svaki piksel nalazi se u točno određenom retku i stupcu. Njegova pozicija igra bitnu ulogu prilikom vizualizacije željenog objekta. Promotrimo sada video kao niz slika. Matrica više nije dovoljna za očuvanje svih prikupljenih informacija. Uveli smo treću dimenziju – vrijeme. Javlja se potreba za tenzorima, odnosno matricama viših dimenzija. Uz položaj na slici, pikselu moramo pridružiti i vremenski trenutak u kojem je snimljen.

Fokus ovog rada je problem klasifikacije ljudi. Za svaku osobu htjet ćemo razlikovati situacije poput kuta gledišta, izraza lica i osvjetljenja. Razumijevanje višedimenzionalne algebre potrebno je želimo li konstruirati algoritam za prepoznavanje lica. Najprije uvodimo definiciju tenzora te pripadnih pojmova i operacija. Prisjetit ćemo se matricnog SVD-a (eng. Singular value decomposition) u drugom poglavlju kako bi lakše razumjeli SVD višeg reda (HOSVD) koji potom slijedi. Rad s četiri ili više dimenzija može se isprva učiniti prilično neintuitivnim. Četvrto poglavlje posvećeno je jednostavnijem problemu prepoznavanja znamenki [6] i služi kao prilagodba na strukturu tenzora. Način na koji se dekompozicija tenzora koristi za prepoznavanje lica [6] opisuje peto poglavlje. U šestom će poglavlju HOSVD omogućiti izgladivanje svjetlosnih prijelaza na slici [4]. Na kraju se nalazi implementacija cijelog algoritma u programskom jeziku R.

Poglavlje 1

Tenzori

Postoje problemi koje obična linearna algebra neće riješiti na dovoljno efikasan način. Želimo li prijeći na multilinearu algebru potrebno je definirati novu strukturu - tenzor. Tenzor je zamišljen kao proširenje ideja skalara, vektora i matrice na više dimenzije. Možemo ga zamisliti kao N -dimenzionalan objekt čiji su elementi jednoznačno određeni pozicijom duž svake dimenzije. Za početak definiramo tenzore i osnovne pojmove koji se uz njih vežu.

1.1 Definicija i osnovni pojmovi

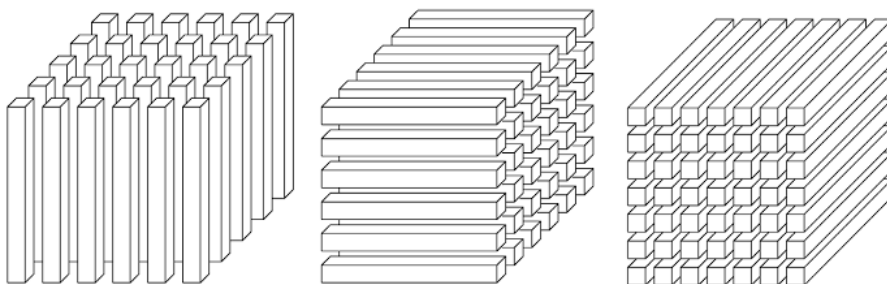
Definicija 1.1.1. *Tenzor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n \times \dots \times I_N}$ reda N sastoji se od elemenata ¹ $a_{i_1 \dots i_n \dots i_N}$ pri čemu je $1 \leq i_n \leq I_N$.

Skalar na ovaj način postaje tenzor nultog, vektor prvog, a matrica drugog reda. Raščlamba matrice na sastavne dijelove, poput redaka i stupaca, uvelike nam je olakšala razumijevanje operacija s njima. Isto ćemo učiniti i s tenzorima. Budući da oni nisu fiksne dimenzije nećemo koristiti termine kao što su *redci* i *stupci* već specificirati o kojem *modu* unutar tenzora govorimo.

Definicija 1.1.2. *Vektor moda n tenzora reda N dobit ćemo fiksiramo li sve indekse osim n -tog. Njegova duljina jednaka je I_n .*

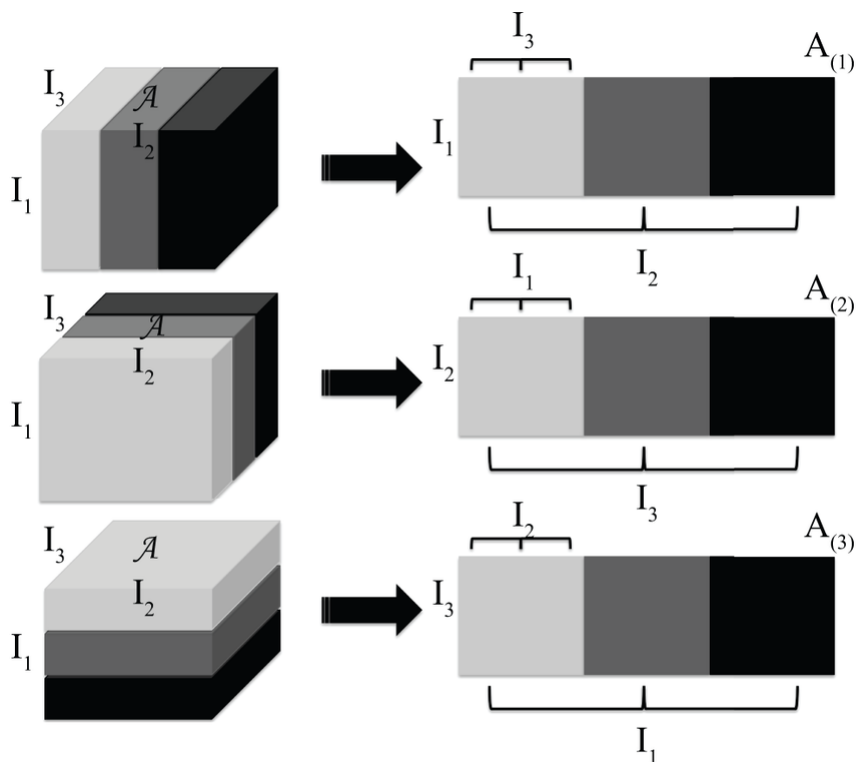
Govorimo li o matricama, vektori moda 1 su njeni stupci, a vektori moda 2 redci. Od velike važnosti bit će nam matrice koje se dobivaju pomoću upravo definiranih vektora.

¹Skalari su označeni malim tiskanim slovima (a), vektori podebljanim malim tiskanim slovima (\mathbf{a}), matrice velikim podebljanim tiskanim slovima (\mathbf{A}), a tenzori velikim pisanim slovima (\mathcal{A}).



Slika 1.1: Na slici [13] se nalaze redom vektori moda 1, 2 i 3 tenzora reda 3. Prilikom vizualizacija ćemo se jednostavnosti radi zadržati u tri dimenzije.

Definicija 1.1.3. $\mathcal{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \dots I_{n-1} I_{n+1} \dots I_N)}$ označavamo matricu čiji su stupci vektori moda n . Kažemo da je ona rezultat **izravnavanja** (eng. **unfold**) tenzora \mathcal{A} po n -toj dimenziji, odnosno n -tom modu koristimo li tenzorsku terminologiju.



Slika 1.2: Tenzor reda 3 možemo izravnati na tri načina [14].

Napomena 1.1.4. Uočimo što se dobije izravnavanjem matrice \mathbf{A} ²

$$\text{unfold}_1(\mathbf{A}) = \mathbf{A} \quad (1.1)$$

$$\text{unfold}_2(\mathbf{A}) = \mathbf{A}^T. \quad (1.2)$$

Izravnamo li matricu u modu 1 rezultat će biti početna matrica. S druge strane, izravnavanje u modu 2 ekvivalentno je transponiranju.

Pojam ranga nećemo definirati za cijeli tenzor već za svaki njegov mod posebno.

Definicija 1.1.5. Rang n tenzora $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n \times \dots \times I_N}$ jednak je rangu matrice $\mathbf{A}_{(n)}$ nastale izravnavanjem u modu n

$$R_n = \text{rank}_n(\mathcal{A}) = \text{rank}(\mathbf{A}_{(n)}) \quad \text{za } n \leq N. \quad (1.3)$$

Upravo smo definirali najosnovnije pojmove vezane uz tenzore. Uz njih ćemo mnoge višedimenzionalne račune uspješno prebaciti u prostor dvije dimenzije na koji smo navikli. To će nam olakšati razumijevanje operacija i identiteta koji slijede.

1.2 Operacije s tenzorima

Koncept tenzora nije nam od velike koristi ako nismo upoznati s operacijama koje se nad njime mogu obavljati. Ovdje ćemo objasniti samo one koje će nam biti potrebne u kasnijim poglavljima. Više o ostalim operacijama nad tenzorima može se naći u [7].

Tenzore istih dimenzija množimo međusobno po komponentama što nas dovodi do definicije skalarnog produkta.

Definicija 1.2.1. Skalarni produkt dva tenzora $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ definira se kao

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \dots \sum_{i_N} a_{i_1 \dots i_N} b_{i_1 \dots i_N}. \quad (1.4)$$

Pomoću njega lako definiramo i normu tenzora.

Definicija 1.2.2. Norma³ tenzora \mathcal{A} iznosi

$$\|\mathcal{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}. \quad (1.5)$$

Osim međusobno, tenzore možemo množiti i matricama pod uvjetom da specificiramo u kojem modu to činimo.

²Indeks označava u kojem se modu izravnavanje vrši.

³Ovako definirana norma će se u daljnjem tekstu primjenjivati i na tenzore reda 2, odnosno matrice, a ekvivalentna je Frobeniusovoj normi.

Definicija 1.2.3. *Produkt moda n tenzora $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n \times \dots \times I_N}$ i matrice $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, u oznaci $\mathcal{A} \times_n \mathbf{U}$, je tenzor $\mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times J_n \times \dots \times I_N}$ čiji su elementi*

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j_n i_{n+1} \dots i_N} = \sum_{i_n} a_{i_1 \dots i_{n-1} i_n i_{n+1} \dots i_N} u_{j_n i_n}. \quad (1.6)$$

Vidimo da ovdje, kao i kod matričnog množenja, tražimo da bude zadovoljena ulančanost dimenzija. Broj stupaca matrice \mathbf{U} mora odgovarati duljini tenzora u modu n . Ona nakon množenja matricom \mathbf{U} postaje jednaka broju njezinih redaka. Produkt moda n možemo također definirati pomoću matrica dobivenih izravnavanjem tenzora \mathcal{A} u n -tom modu.

Napomena 1.2.4. *Neka je $\mathcal{B} = \mathcal{A} \times_n \mathbf{U}$. Tada vrijedi*

$$\mathbf{B}_{(n)} = \mathbf{U} \mathbf{A}_{(n)}. \quad (1.7)$$

Sljedeća napomena će nam pomoći da uvidimo koliko je usko povezan produkt moda n s običnim množenjem matrica.

Napomena 1.2.5.

$$\mathbf{U} \mathbf{A} = \mathbf{A} \times_1 \mathbf{U} \quad (1.8)$$

$$\mathbf{A} \mathbf{V}^T = \mathbf{A} \times_2 \mathbf{V}. \quad (1.9)$$

Drugim riječima, množenje matrice \mathbf{A} matricom \mathbf{U} s lijeve strane je zapravo mod-1 produkt dvodimenzionalnog tenzora \mathcal{A} i matrice \mathbf{U} . Mod-2 produkt odgovara množenju tenzora \mathcal{A} transponiranom matricom \mathbf{V} s desne strane.

Koncept rastava tenzora na singularne vrijednosti, tzv. HOSVD (*eng. Higher Order Singular Value Decomposition*) temeljen je upravo na mod- n produktu. Međutim, u nastavku ćemo se najprije prisjetiti običnog, tj. matričnog SVD-a (*eng. Singular Value Decomposition*).

Poglavlje 2

Dekompozicija singularnih vrijednosti (SVD)

Dekompozicija matrice na singularne vrijednosti ima mnoge primjene. Od rješavanja problema najmanjih kvadrata [12] do uštede memorije ili klasifikacije slika. Nama će služiti kao temelj za dekompoziciju tenzora o kojoj će biti riječ kasnije.

2.1 Dekompozicija matrice na singularne vrijednosti

Teorem 2.1.1. *Neka su m i n prirodni brojevi takvi da je $m > n$ te $\mathbf{A} \in \mathbb{R}^{m \times n}$ proizvoljna realna matrica. Tada postoji dekompozicija $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, gdje su $\mathbf{U} \in \mathbb{R}^{m \times m}$ i $\mathbf{V} \in \mathbb{R}^{n \times n}$ ortogonalne matrice, a $\mathbf{\Sigma} = \mathbf{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ dijagonalna matrica sa svojstvom $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ [12].*

Dokaz. U slučaju $\mathbf{A} = 0$ stavimo li da su \mathbf{U} i \mathbf{V} proizvoljne ortogonalne matrice i $\mathbf{\Sigma} = 0$ tvrdnja vrijedi. Inače dokaz provodimo indukcijom po m i n .

Budući da je $m \geq n$, baza indukcije je za $n = 1$. Definiramo matrice \mathbf{U} , $\mathbf{\Sigma}$ i \mathbf{V} na način:

$$\mathbf{U} = \frac{\mathbf{A}}{\|\mathbf{A}\|_2}, \quad \mathbf{\Sigma} = \|\mathbf{A}\|_2, \quad \mathbf{V} = 1. \quad (2.1)$$

Koristeći ove definicije tvrdnja teorema vrijedi za $n = 1$ i $m \geq n$ proizvoljan.

Pretpostavimo da tvrdnja vrijedi za $\mathbf{A} \in \mathbb{R}^{(m-1) \times (n-1)}$. Odaberemo takav jedinični vektor \mathbf{v} da se na njemu postiže maksimum 2-norme za \mathbf{A}

$$\|\mathbf{A}\|_2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{Ax}\|_2 = \|\mathbf{Av}\|_2. \quad (2.2)$$

Pomoću njega definiramo vektor \mathbf{u}

$$\mathbf{u} = \frac{\mathbf{Av}}{\|\mathbf{Av}\|_2}. \quad (2.3)$$

Uzmemo proizvoljne matrice $\tilde{\mathbf{U}}$ i $\tilde{\mathbf{V}}$ takve da su sljedeće matrice ortogonalne reda m , odnosno n :

$$\mathbf{U}_0 = [\mathbf{u}, \tilde{\mathbf{U}}], \quad \mathbf{V}_0 = [\mathbf{v}, \tilde{\mathbf{V}}]. \quad (2.4)$$

Računamo

$$\mathbf{U}_0^T \mathbf{A} \mathbf{V}_0 = \begin{bmatrix} \mathbf{u}^T \\ \tilde{\mathbf{U}}^T \end{bmatrix} \mathbf{A} [\mathbf{u}, \tilde{\mathbf{V}}] = \begin{bmatrix} \mathbf{u}^T \mathbf{A} \mathbf{v} & \mathbf{u}^T \mathbf{A} \tilde{\mathbf{V}} \\ \tilde{\mathbf{U}}^T \mathbf{A} \mathbf{v} & \tilde{\mathbf{U}}^T \mathbf{A} \tilde{\mathbf{V}} \end{bmatrix}. \quad (2.5)$$

Sljedeća jednakost vrijedi zbog načina na koji smo definirali vektore \mathbf{u} i \mathbf{v} :

$$\mathbf{u}^T \mathbf{A} \mathbf{v} = \frac{\mathbf{v}^T \mathbf{A}^T \mathbf{A} \mathbf{v}}{\|\mathbf{A} \mathbf{v}\|_2} = \frac{\|\mathbf{A} \mathbf{v}\|_2^2}{\|\mathbf{A} \mathbf{v}\|_2} = \|\mathbf{A} \mathbf{v}\|_2 = \|\mathbf{A}\|_2 := \sigma. \quad (2.6)$$

Matrica \mathbf{U}_0 je ortogonalna što znači da je $\tilde{\mathbf{U}}^T \mathbf{u} = 0$, a prema tome je i

$$\tilde{\mathbf{U}}^T \mathbf{A} \mathbf{v} = \tilde{\mathbf{U}}^T \mathbf{u} \|\mathbf{A} \mathbf{v}\|_2 = 0. \quad (2.7)$$

Želimo pokazati da je $\mathbf{u}^T \mathbf{A} \tilde{\mathbf{V}} = 0$. Stavimo $\mathbf{A}_1 = \mathbf{U}_0^T \mathbf{A} \mathbf{V}_0$, $\mathbf{w}^T = \mathbf{u}^T \mathbf{A} \tilde{\mathbf{V}}$, $\mathbf{B} = \tilde{\mathbf{U}}^T \mathbf{A} \tilde{\mathbf{V}}$. Sada vrijedi

$$\mathbf{A}_1 = \mathbf{U}_0^T \mathbf{A} \mathbf{V}_0 = \begin{bmatrix} \sigma & \mathbf{w}^T \\ \mathbf{0} & \mathbf{B} \end{bmatrix}. \quad (2.8)$$

2-norma je invarijantna na ortogonalnost pa dobivamo jednakost

$$\sigma = \|\mathbf{A}\|_2 = \|\mathbf{U}_0^T \mathbf{A} \mathbf{V}_0\|_2 = \|\mathbf{A}_1\|_2. \quad (2.9)$$

Za proizvoljan vektor $\mathbf{z} \neq 0$ vrijedi

$$\|\mathbf{A}_1\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}_1 \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \geq \frac{\|\mathbf{A}_1 \mathbf{z}\|_2}{\|\mathbf{z}\|_2}. \quad (2.10)$$

Množenjem s $\|\mathbf{z}\|_2$ ostaje

$$\|\mathbf{A}_1\|_2 \|\mathbf{z}\|_2 \geq \|\mathbf{A}_1 \mathbf{z}\|_2. \quad (2.11)$$

Postavimo \mathbf{z} da bude jednak

$$\mathbf{z} = \begin{bmatrix} \sigma \\ \mathbf{w} \end{bmatrix}. \quad (2.12)$$

Tada vrijedi

$$\begin{aligned} \|\mathbf{A}_1\|_2^2 \|\mathbf{z}\|_2^2 &= \|\mathbf{A}_1\|_2^2 (\sigma^2 + \|\mathbf{w}\|_2^2) \geq \|\mathbf{A}_1 \mathbf{z}\|_2^2 = \left\| \mathbf{A}_1 \begin{bmatrix} \sigma \\ \mathbf{w} \end{bmatrix} \right\|_2^2 = \\ &= \left\| \begin{bmatrix} \sigma & \mathbf{w}^T \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \sigma \\ \mathbf{w} \end{bmatrix} \right\|_2^2 = (\sigma^2 + \mathbf{w}^T \mathbf{w})^2 + \|\mathbf{B} \mathbf{w}\|_2^2 \geq \\ &\geq (\sigma^2 + \|\mathbf{w}\|_2^2)^2, \end{aligned} \quad (2.13)$$

iz čega slijedi

$$\|\mathbf{A}_1\|_2^2(\sigma^2 + \|\mathbf{w}\|_2^2) \geq (\sigma^2 + \|\mathbf{w}\|_2^2)^2. \quad (2.14)$$

Podijelimo li s desnom stranom dobivamo

$$\sigma^2 = \|\mathbf{A}\|_2^2 = \|\mathbf{A}_1\|_2^2 \geq \sigma^2 + \|\mathbf{w}\|_2^2, \quad (2.15)$$

što vrijedi samo za $\mathbf{w} = 0$.

Pokazali smo da vrijedi

$$\mathbf{U}_0^T \mathbf{A} \mathbf{V}_0 = \begin{bmatrix} \sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}. \quad (2.16)$$

Po pretpostavci indukcije \mathbf{B} možemo zapisati kao

$$\mathbf{B} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T. \quad (2.17)$$

Uvrštavanjem u (2.16) jednažba postaje

$$\mathbf{U}_0^T \mathbf{A} \mathbf{V}_0 = \begin{bmatrix} \sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1 \end{bmatrix} \begin{bmatrix} \sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_1 \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_1 \end{bmatrix}, \quad (2.18)$$

iz čega zajedno s činjenicom da ortogonalne matrice čine multiplikativnu grupu slijedi tvrdnja. Silazan poredak singularnih vrijednosti možemo postići pomoću matrica permutacije koje su također ortogonalne. \square

Napomena 2.1.2. U slučaju da je $m < n$, SVD provodimo za matricu \mathbf{A}^T .

Definicija 2.1.3. Stupce matrice $\mathbf{U} = [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_m]$ zovemo *lijevi singularni vektori*, a stupce matrice $\mathbf{V} = [\mathbf{v}_1 \mathbf{v}_2 \dots \mathbf{v}_n]$ *desni singularni vektori*. Brojevi σ_i zovu se *singularne vrijednosti*.

Uočimo da singularne vrijednosti padaju kako se spuštamo niz dijagonalu matrice $\mathbf{\Sigma}$. Matricu \mathbf{A} možemo zapisati kao zbroj matrica ranga 1 koristeći singularne vektore

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T. \quad (2.19)$$

Analogon ovakvog rastava bit će ključan kada pređemo na tenzore.

Napomena 2.1.4. Zamislimo li matricu \mathbf{A} kao tenzor reda 2, njenu dekompoziciju na singularne vrijednosti možemo zapisati u obliku

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \mathbf{\Sigma} \times_1 \mathbf{U} \times_2 \mathbf{V}. \quad (2.20)$$

Gornja jednakost u skladu je s napomenom 1.2.5 iz predhodnog poglavlja.

2.2 Aproksimacija nižeg ranga

Ponekad ćemo, na primjer zbog nedostatka memorije, htjeti matricu aproksimirati onom nižeg ranga. Sljedeći važan teorem [12] nam govori kako to učiniti koristeći SVD i kolika greška će pritom nastati.

Teorem 2.2.1. *Neka je $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ i $r = r(\mathbf{A})$. Za $k \in 1, 2, \dots, r - 1$ definiramo matrice*

$$\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{U}(:, 1:k) \mathbf{\Sigma}(1:k, 1:k) \mathbf{V}(:, 1:k)^T. \quad (2.21)$$

Tada vrijedi

$$\min_{r(\mathbf{X}) \leq k} \|\mathbf{A} - \mathbf{X}\|_2 = \|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}. \quad (2.22)$$

Napomena 2.2.2. *Matrice \mathbf{A}_k zapravo su dobivene uzimanjem prvih k elemenata spomenutog rastava (2.19).*

Dokaz. Pokažimo najprije drugu jednakost

$$\begin{aligned} \|\mathbf{A} - \mathbf{A}_k\|_2 &= \left\| \sum_{i=k+1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T \right\|_2 = \\ &= \|\mathbf{U} \mathbf{diag}(0, \dots, 0, \sigma_{k+1}, \dots, \sigma_n) \mathbf{V}^T\|_2 = \\ &= \sigma_{k+1}. \end{aligned} \quad (2.23)$$

Neka je sada $\mathbf{B} \in \mathbb{R}^{m \times n}$ proizvoljna matrica ranga najviše k . Želimo pokazati da udaljenost između \mathbf{A} i \mathbf{B} u 2-normi nije veća od σ_{k+1} .

Nul-potprostor matrice \mathbf{B} dimenzije je veće ili jednake $n - k$. Budući da je potprostor razapet vektorima $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k+1}$ dimenzije $k + 1$ znamo da postoji netrivialan jedinični vektor \mathbf{h} koji se nalazi u njegovom presjeku s nul-potprostorom od \mathbf{B} . Za njega je $\mathbf{B}\mathbf{h} = \mathbf{0}$ i vrijedi

$$\begin{aligned} \|\mathbf{A} - \mathbf{B}\|_2 &\geq \|(\mathbf{A} - \mathbf{B})\mathbf{h}\|_2 = \|\mathbf{A}\mathbf{h}\|_2 = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{h}\|_2 = \\ &= \|\mathbf{\Sigma}\mathbf{V}^T\mathbf{h}\|_2 \geq \sigma_{k+1} \|\mathbf{V}^T\mathbf{h}\|_2 = \sigma_{k+1}. \end{aligned} \quad (2.24)$$

□

Dakle, najbolja aproksimacija matrice \mathbf{A} ranga najviše $k < r$ dobivena je od prvih k lijevih i desnih singularnih vektora te prvih k singularnih vrijednosti. Greška nastala smanjivanjem ranga jednaka je idućoj singularnoj vrijednosti σ_{k+1} .



Slika 2.1: Aproksimacija slike nižim rangom. Originalna slika (lijevo) je veličine 512×352 piksela. Sliku u sredini smo dobili postavljajući k na 35, a desno na 15 (vidi 2.2.1). Uočimo da se smanjenjem ranga značajno gubi na kvaliteti. Slika je uzeta iz Weizmann baze [1].

Poglavlje 3

SVD višeg reda (HOSVD)

Dekompozicija tenzora na singularne vrijednosti je poopćenje SVD-a iz prošlog poglavlja. Mnogo svojstava običnog SVD-a naći se svoj analogon u HOSVD-u. Međutim, postoje i poneke razlike. Prijeđimo na osnovni teorem o rastavu tenzora.

3.1 HOSVD

Teorem 3.1.1. Tenzor $\mathcal{A} \in \mathbb{R}^{i_1 \times i_2 \times \dots \times i_N}$ može se zapisati u obliku rastava [5]

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \dots \times_N \mathbf{U}^{(N)}, \quad (3.1)$$

gdje su $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times i_n}$, ortogonalne matrice. Tenzor \mathcal{S} iste je dimenzije kao i početni tenzor \mathcal{A} te ga zovemo **jezgreni tenzor**. Dodatno, on ima svojstvo međusobne ortogonalnosti: svaka dva podtenzora $^1 \mathcal{S}_{i_n=a}$ i $\mathcal{S}_{i_n=b}$ su okomita u smislu skalarnog produkta 1.2.1 ako je $a \neq b$

$$\langle \mathcal{S}_{i_n=a}, \mathcal{S}_{i_n=b} \rangle = 0 \quad \text{za } a \neq b. \quad (3.2)$$

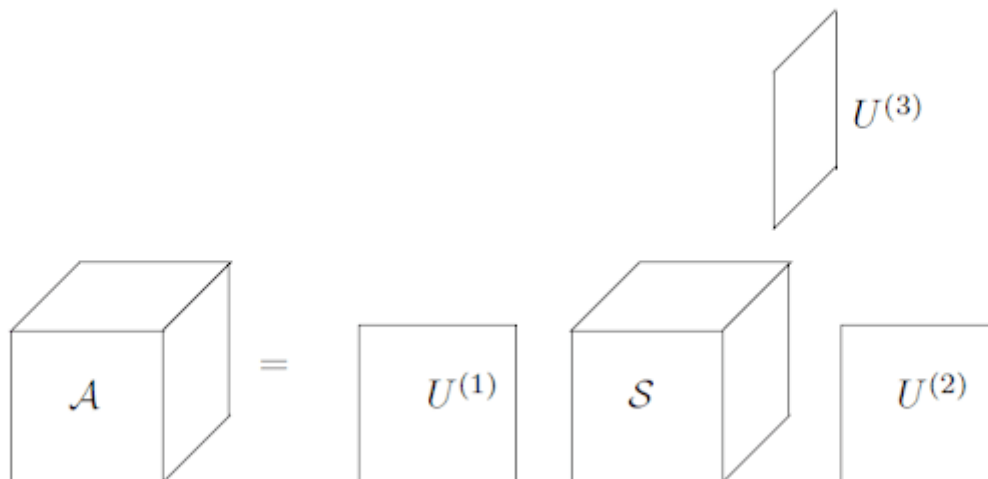
Singularne vrijednosti moda n definirane su s

$$\sigma_i^{(n)} = \|\mathcal{S}_{i_n=i}\|, \quad i = 1, \dots, I_n \quad (3.3)$$

i za njih vrijedi padajući poredak

$$\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \dots \geq \sigma_{I_n}^{(n)} \quad n = 1, \dots, N. \quad (3.4)$$

¹S $\mathcal{A}_{i_n=a}$ označavamo podtenzor koji nastaje fiksiranjem moda i_n na vrijednost a u originalnom tenzoru.



Slika 3.1: Ilustracija HOSVD-a za tenzor reda 3 [3].

Prije dokaza treba nam definicija Kroneckerovog produkta matrica.

Definicija 3.1.2. *Kroneckerov produkt matrica* $\mathbf{A} \in \mathbb{R}^{i \times j}$ i $\mathbf{B} \in \mathbb{R}^{k \times l}$, u oznaci $\mathbf{A} \otimes \mathbf{B}$, je matrica iz $\mathbb{R}^{(ik) \times (jl)}$ definirana s

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1j}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2j}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1}\mathbf{B} & a_{i2}\mathbf{B} & \dots & a_{ij}\mathbf{B} \end{bmatrix}. \quad (3.5)$$

Dokaz. Elemente rastava računamo na sljedeći način. Odredimo SVD matrica nastalih izravnavanjem tenzora \mathcal{A} u svakom modu

$$\mathbf{A}_{(i)} = \mathbf{U}^{(i)} \boldsymbol{\Sigma}^{(i)} (\mathbf{V}^{(i)})^T, \quad i = 1, 2, \dots, N. \quad (3.6)$$

Zatim stavimo

$$\mathcal{S} = \mathcal{A} \times_1 (\mathbf{U}^{(1)})^T \times_2 (\mathbf{U}^{(2)})^T \dots \times_N (\mathbf{U}^{(N)})^T. \quad (3.7)$$

Preostaje pokazati da su kriške od \mathcal{S} međusobno okomite te da singularne vrijednosti padaju u svakom modu.

Jednadžba 3.1 može se zapisati u matričnog obliku

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \mathbf{S}_{(n)} (\mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n+2)} \dots \mathbf{U}^{(N)} \times \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \dots \mathbf{U}^{(n-1)})^T. \quad (3.8)$$

Označimo s r_n najveći indeks za koji je $\sigma_{r_n}^{(n)} > 0$. Usporedbom 3.6 i (3.8), uz ortogonalnost Kroneckerovog faktora u (3.8) slijedi

$$\mathbf{S}_{(n)} = \boldsymbol{\Sigma}^{(n)} \mathbf{V}^{(n)T} (\mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n+2)} \dots \mathbf{U}^{(N)} \times \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \dots \mathbf{U}^{(n-1)}). \quad (3.9)$$

Jednadžba (3.9) nam govori da za proizvoljne ortogonalne matrice $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(n-1)}, \mathbf{U}^{(n+1)}, \dots, \mathbf{U}^{(N)}$ vrijedi

$$\langle \mathcal{S}_{i_n=a}, \mathcal{S}_{i_n=b} \rangle = 0 \quad \text{za } a \neq b, \quad (3.10)$$

$$\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \dots \geq \sigma_{I_n}^{(n)} \geq 0, \quad (3.11)$$

te za $r_n < I_n$

$$\sigma_{r_n+1}^{(n)} = \dots = \sigma_{I_n}^{(n)} = 0. \quad (3.12)$$

□

Jezgreni tenzor \mathcal{S} nije naslijedio dijagonalnu strukturu matrice $\boldsymbol{\Sigma}$ iz običnog SVD-a 2.1.1. Ipak, iz teorema 3.1.1 je jasno da je njegova masa koncentrirana u prednjem gornjem lijevom kutu (zamislimo li ga kao kvadar u tri dimenzije). Što se više odmičemo u bilo kojem smjeru elementi postaju sve manji. Računanje dekompozicije tenzora moda 3 implementirano je u funkciji 7.2.

3.2 Aproksimacija tenzora pomoću HOSVD-a

Sekcija 2.2 ističe praktičnost aproksimacije matrice nižim rangom. Isto želimo učiniti i s tenzorima. Bogata struktura tenzora omogućava nam pametan odabir dimenzije u kojoj vršimo redukciju (više u poglavljima 4 i 6). Za sada se fokusiramo na posebnu vrstu zapisa tenzora koja će biti temelj redukcije dimenzije.

Krenimo za početak s tenzorima reda 3. Preformuliramo li malo rezultat prethodnog teorema zaključit ćemo da se tenzor $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$ može napisati kao suma matrica pomnoženih sa singularnim vektorima ²

$$\mathcal{A} = \sum_{i=1}^n \mathbf{A}_i \times_3 \mathbf{u}_i^{(3)}, \quad \mathbf{A}_i = \mathcal{S}(:, :, i) \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}. \quad (3.13)$$

Vektori $\mathbf{u}_i^{(3)}$ su stupci matrice $\mathbf{U}^{(3)}$, a \mathbf{A}_i valja zamišljati kao matrice iz $\mathbb{R}^{m \times n}$, ali također kao i tenzore iz $\mathbb{R}^{m \times n \times 1}$. Nadalje, matrice \mathbf{A}_i su međusobno okomite

$$\langle \mathbf{A}_i, \mathbf{A}_j \rangle = \text{tr}[\mathbf{U}^{(2)} \mathcal{S}(:, :, i)^T (\mathbf{U}^{(1)})^T \mathbf{U}^{(1)} \mathcal{S}(:, :, j) (\mathbf{U}^{(2)})^T] = \text{tr}[\mathcal{S}(:, :, i)^T \mathcal{S}(:, :, j)] = 0. \quad (3.14)$$

Napomena 3.2.1. *Koristili smo jednakost $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$.*

²Sjetimo se zapisa matrice preko singularnih vektora (2.19).

Zaključujemo da se svaki podtenzor $\mathcal{A}_{i_3=x}$ tenzora \mathcal{A} može prikazati u ortogonalnoj bazi

$$(\mathbf{A}_i)_{i=1}^{R_3}, \quad (3.15)$$

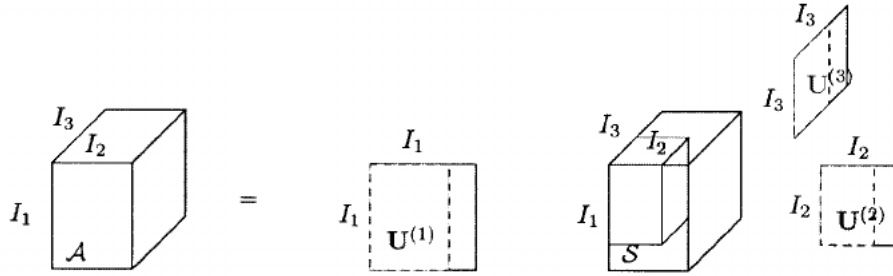
gdje je R_3 broj pozitivnih singularnih vrijednosti moda 3:

$$\mathcal{A}(:, :, j) = \sum_{i=1}^{R_3} z_i^{(j)} \mathbf{A}_i. \quad (3.16)$$

Iz jednakosti (3.13) zaključujemo da je $z_i^{(j)}$ upravo j -ta komponenta vektora $\mathbf{u}_i^{(3)}$.

Ograničimo li se na prvih $k < R_3$ baznih matrica $(\mathbf{A}_i)_{i=1}^k$ i prvih k singularnih vektora moda 3 $(\mathbf{u}_i^{(3)})_{i=1}^k$ dobit ćemo aproksimaciju \mathcal{A}' tenzora \mathcal{A} nižeg ranga $k = R'_3 < R_3$

$$\mathcal{A}' = \sum_{i=1}^k \mathbf{A}_i \times_3 \mathbf{u}_i^{(3)}. \quad (3.17)$$



Slika 3.2: Smanjenje dimenzije korištenjem HOSVD-a.

Greška koja pritom nastaje ograničena je sumom kvadrata singularnih vrijednosti koje odgovaraju odbačenim singularnim vektorima reda 3

$$\|\mathcal{A} - \mathcal{A}'\| \leq \sum_{i=k+1}^{R_3} (\sigma_i^{(3)})^2. \quad (3.18)$$

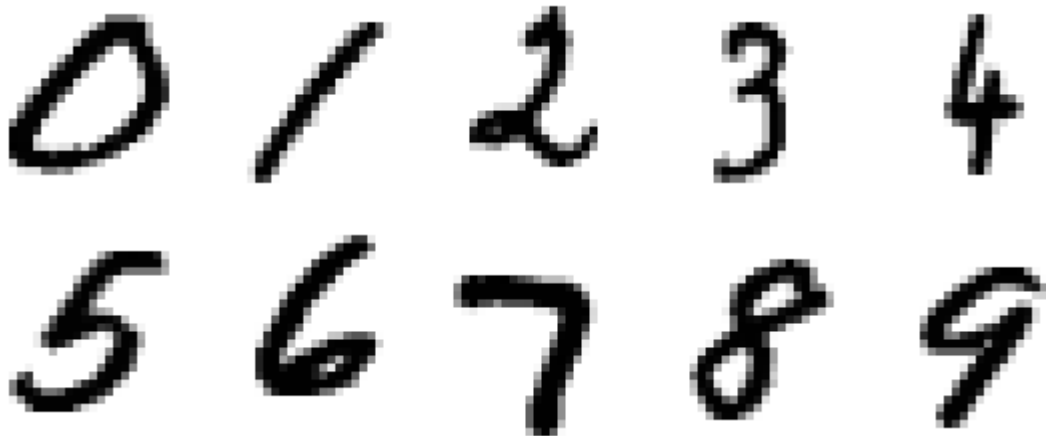
Valja napomenuti da opisani način aproksimacije tenzora nižim rangom nije optimalan kao što je to bio slučaj kod matrica 2.2.1. Više o postupku pronalaska optimalnog rješenja može se naći u [4].

Do ovog trenutka smo prikupili dovoljno znanja za uspješno rješavanje problema klasifikacije znamenki i lica. Slijede poglavlja s preciznim opisom algoritama [6] te rezultatima.

Poglavlje 4

Prepoznavanje znamenki

Prepoznavanje znamenki jednostavniji je problem od prepoznavanja lica. Spremamo samo dvije informacije: sliku kao matricu piksela i znamenku koja se na njoj nalazi. Za pohranu slike uzet ćemo prirodne dvije dimenzije: redke i stupce, a duž treće će se nalaziti različite slike iste znamenke. Matricu slike mogli smo razvući u vektor čime bi uštedjeli na jednoj dimenziji. Budući da nam znamenke služe za privikavanje na tenzore, što podrazumijeva barem tri dimenzije, to nećemo učiniti.

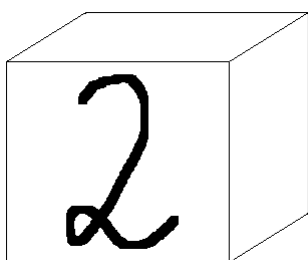


Slika 4.1: Isječak iz MNIST [10] baze rukom pisanih znamenki.

4.1 Oblikovanje algoritma

Problem 1. (*Prepoznavanje znamenki*) Dan je skup od 1000 slika rukom pisanih znamenki [10] veličine 28×28 (broj slika svake znamenke jednak je 100). Cilj je osmisliti algoritam za prepoznavanje znamenke koja se nalazi na dosad neviđenoj slici [6].

Rješenje 1. Skup slika pohranit ćemo u tenzore veličine $28 \times 28 \times 100$. Svaki tenzor sastoji se od različitih slika iste znamenke. Prve dvije dimenzije služe da razlikujemo piksele, a treća slike.



Slika 4.2: Ilustracija tenzora reda 3 u kojem su pohranjene slike znamenke 2.

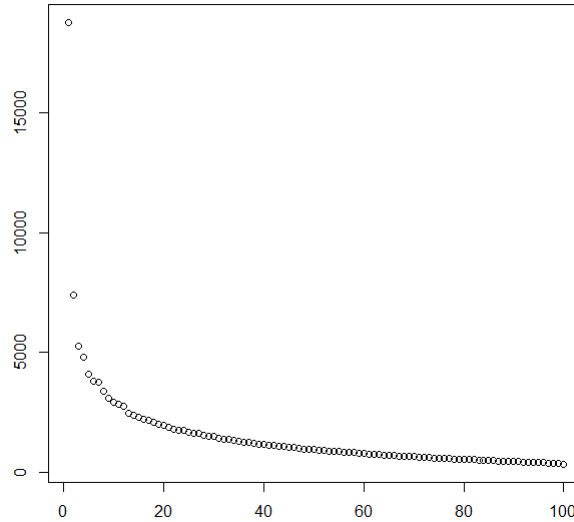
Princip je sljedeći

- za svaku znamenku provesti HOSVD i pomoću njega izračunati ortogonalnu bazu (3.15),
- zapisati novu znamenku u svakoj od gornjih baza (3.16) i odrediti grešku koja je pritom nastala
- znamenku ćemo proglasiti onom kod koje je nastala najmanja greška prilikom prikaza u bazi (3.15).

Promatramo tenzor znamenke 2. Sve singularne vrijednosti moda 3 su pozitivne (vidi 7.3), što bi značilo da valja izračunati 100 baznih matrica $(\mathbf{A}_i)_{i=1}^{100}$. To je velika količina posla zbog čega ćemo koristiti aproksimaciju (3.17)

$$\mathcal{A}' = \sum_{i=1}^k \mathbf{A}_i \times_3 \mathbf{u}_i^{(3)}, \quad \mathbf{A}_i = \mathbf{S}(:, :, i) \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}. \quad (4.1)$$

Iz grafa 1 vidimo da je veliki postotak varijacije među znamenkama dvojke objašnjen pomoću prvih 10-ak singularnih vrijednosti. Prisjetimo se da teorem 3.1.1 kaže da singularne vrijednosti padaju u svakom modu. Postotak varijacije koji smo zadržali nakon aproksimacije (3.17) približno je jednak

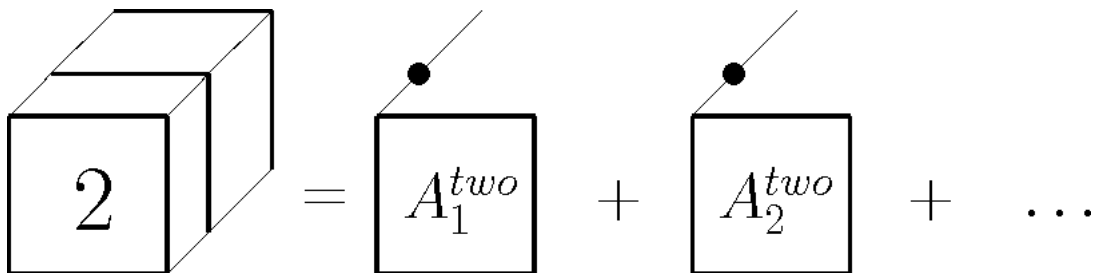


Slika 4.3: Singularne vrijednosti moda 3 tenzora znamenke 2.

$$\frac{\sum_{i=1}^k (\sigma_i^{(3)})^2}{\sum_{i=1}^{100} (\sigma_i^{(3)})^2}. \quad (4.2)$$

U slučaju $k = 10$ on iznosi oko 81%.

Označimo sa \mathbf{Z} nepoznatu znamenku koju želimo klasificirati. Potrebni su nam koeficijenti z_i 3.16 njenog zapisa u bazi $(\mathbf{A}_i)_{i=1}^k$.



Slika 4.4: Rastav (4.1) tenzora znamenke 2.

Napomena 4.1.1. U slučaju da je \mathbf{Z} slika koja se već nalazi u tenzoru na poziciji j (u

trećem modu), njezini koeficijenti su upravo j -te komponente vektora $\mathbf{u}_i^{(3)}$:

$$\mathcal{A}(:, :, j) = \sum_{i=1}^k z_i^{(j)} \mathbf{A}_i = \sum_{i=1}^k \mathbf{u}_i^{(3)}(j) \mathbf{A}_i. \quad (4.3)$$

Za nepoznatu znamenku koeficijente računamo minimizacijom greške prikaza u bazi (3.16)

$$\min_z \left\| \mathbf{Z} - \sum_j z_j \mathbf{A}_j \right\|. \quad (4.4)$$

Definiramo funkciju G kao

$$G(z) = \frac{1}{2} \left\| \mathbf{Z} - \sum_j z_j \mathbf{A}_j \right\|^2 = \frac{1}{2} \langle \mathbf{Z} - \sum_j z_j \mathbf{A}_j, \mathbf{Z} - \sum_j z_j \mathbf{A}_j \rangle. \quad (4.5)$$

Transformacija koju smo napravili je monotona bijekcija. Umjesto da tražimo rješenje od (4.4) naći ćemo rješenje za (4.5). Baza $(\mathbf{A}_i)_{i=1}^k$ je ortogonalna što znači da vrijedi

$$\langle \mathbf{A}_i, \mathbf{A}_j \rangle = 0 \quad \text{za } i \neq j, \quad (4.6)$$

pa (4.5) postaje

$$G(z) = \frac{1}{2} \langle \mathbf{Z}, \mathbf{Z} \rangle - \sum_j z_j \langle \mathbf{Z}, \mathbf{A}_j \rangle + \frac{1}{2} \sum_j z_j^2 \langle \mathbf{A}_j, \mathbf{A}_j \rangle. \quad (4.7)$$

Točku minimuma dobivamo izjednačavanjem derivacije po svakoj komponenti s nulom

$$\frac{\partial G}{\partial z_j} = -\langle \mathbf{Z}, \mathbf{A}_j \rangle + z_j \langle \mathbf{A}_j, \mathbf{A}_j \rangle = 0. \quad (4.8)$$

Konačno, izračunali smo koeficijente nepoznate znamenke:

$$z_j = \frac{\langle \mathbf{Z}, \mathbf{A}_j \rangle}{\langle \mathbf{A}_j, \mathbf{A}_j \rangle}, \quad j = 1, 2, \dots \quad (4.9)$$

Greška pritom iznosi

$$\left\| \mathbf{Z} - \sum_j z_j \mathbf{A}_j \right\|. \quad (4.10)$$

Opisani postupak koji smo proveli za znamenku 2 ponavljamo i za ostale znamenke.

4.2 Rezultati

U ovom odjeljku radimo evaluaciju predstavljenog algoritma za prepoznavanje rukom pisanih znamenki. Korišten je skup od 10000 slika nepoznatih znamenki [10], a za broj baznih matrica se uzima $k \in \{3, 5, 7, 10\}$.

k	3	5	7	10
točnost	86.05%	89.46%	90.37%	91.39%

Tablica 4.1: Točnost algoritma 1 za prepoznavanje rukom pisanih znamenki. Skup za trening sastoji se od 100 slika svake znamenke.

Već sa samo tri bazne matrice algoritam točno razvrstava znamenke u 86% slučajeva. Povećanjem k do 10 točnost je narasla na otprilike 91% gdje će se stabilizirati. Navodimo još i rezultate koji se dobiju uzimanjem po 1000 slika svake znamenke za kreiranje početnih tenzora. Povećanjem skupa za učenje, uz $k = 10$ dobiva se najveća točnost od 94.38%.

k	3	5	7	10
točnost	89.93%	91.7%	93.3%	94.38%

Tablica 4.2: Točnost algoritma za prepoznavanje rukom pisanih znamenki. Tensor pojedine znamenke sastoji se od 1000 slika.

Poglavlje 5

Prepoznavanje lica

Zamislimo da nam je umjesto znamenki dana baza slika ljudi. Razvlačenjem matrice piksela u vektor oslobodili smo jednu dimenziju. Primjerice, u nju možemo posložiti različite izraze lica koji se na slikama pojavljuju. Treba imati na umu da oni nisu bitan faktor za klasifikaciju - radi se o jednoj te istoj osobi bez obzira na njezin izraz lica. To nas dovodi do sljedećeg problema.

5.1 Oblikovanje algoritma

Problem 2. (*Prepoznavanje lica*) Posjedujemo skup slika n_p osoba gdje je svaka od njih slikana s n_e različitih izraza lica [6]. Želimo moći identificirati nepoznatu osobu kao jednu iz danog skupa.

Rješenje 2. Matricu piksela slike razvući ćemo u vektor duljine n_i . Od zadanih slika formiramo tenzor $\mathcal{A} \in \mathbb{R}^{n_i \times n_e \times n_p}$. Prvi mod tenzora \mathcal{A} razlikuje piksele na slici, drugi izraze lica, a treći osobe. HOSVD tenzora \mathcal{A} pišemo kao

$$\mathcal{A} = \mathbf{C} \times_p \mathbf{H}, \quad \mathbf{C} = \mathbf{S} \times_i \mathbf{F} \times_e \mathbf{G}. \quad (5.1)$$

Za fiksni izraz lica e stoga vrijedi jednakost

$$\mathcal{A}(:, e, :) = \mathbf{C}(:, e, :) \times_p \mathbf{H}. \quad (5.2)$$

Budući da su $\mathcal{A}(:, e, :)$ i $\mathbf{C}(:, e, :)$ zapravo matrice za njih ćemo uvesti nove oznake: \mathbf{A}_e i \mathbf{C}_e . Jednadžba (5.2) time postaje matrična jednadžba korištenjem (1.9)

$$\mathbf{A}_e = \mathbf{C}_e \mathbf{H}^T, \quad e = 1, \dots, n_e. \quad (5.3)$$

Uzmimo nadalje da $\mathbf{a}_p^{(e)}$ predstavljaju stupce matrice \mathbf{A}_e , a \mathbf{h}_p retke matrice \mathbf{H} . Valja uočiti da je matrica \mathbf{H} ista za svaki izraz lica. Pretvorimo li (5.3) u stupčanu jednadžbu dobit ćemo da vrijedi

$$\mathbf{a}_p^{(e)} = \mathbf{C}_e \mathbf{h}_p. \quad (5.4)$$

Razmislimo malo što predstavljaju pojedini elementi gornje jednakosti. Vektor $\mathbf{a}_p^{(e)}$ je rastegnuta slika osobe p s izrazom lica e . Stupci matrice \mathbf{C}_e su bazni vektori izraza e , a \mathbf{h}_p sadrži koordinate slike osobe p u toj bazi.

Neka je $\mathbf{z} \in \mathbb{R}^{n_i}$ slika nepoznate osobe s jednim od n_e poznatih izraza lica. Želimo utvrditi o kojoj osobi je riječ. Za osobu p s izrazom lica e koordinate u gore spomenutoj bazi se nalaze u vektoru \mathbf{h}_p . Provodimo postupak:

- za svaki e računamo koordinate slike \mathbf{z} u bazi stupaca od \mathbf{C}_e i tražimo kojem od vektora \mathbf{h}_p su najbliži,
- izrazu lica e pridružimo dotičnu osobu p ,
- nepoznatu osobu klasificiramo kao onu čiji pripadni izraz lica ostvaruje najmanju grešku.

Preostaje odrediti koordinate slike \mathbf{z} . Njih ćemo dobiti rješavanjem problema najmanjih kvadrata

$$\min_{\alpha_e} \|\mathbf{C}_e \alpha_e - \mathbf{z}\|_2. \quad (5.5)$$

Princip po kojem klasifikacija pomoću tenzora radi sada bi već trebao biti jasan. Prolazimo po svim uvjetima u kojima se osobe na slikama nalaze (u gornjem slučaju samo izrazi lica) i neviđenu sliku zapisujemo u njihovoj bazi. Osoba se identificira kao ona čijem je zapisu u spomenutoj bazi najbližija.

Ljudi mogu biti uslikani u raznim situacijama. Izraz lica nije jedini faktor koji će se pritom mijenjati. Tu su također osvjetljenje, kut gledišta, svjetlina, pozadina, kontrast i mnogi drugi. Mi ćemo se usredotočiti na izraze lica, osvjetljenje i kut gledišta, za koje već postoji organizirana baza slika instituta Weizmann [1].

Problem 3. (Prepoznavanje lica 2) Svaka od n_p osoba slikana je s n_e izraza lica, u n_{ill} uvjeta osvjetljenja i iz n_{vp} točaka gledišta. Ponovno želimo odrediti o kojoj osobi je riječ za dosad neviđenu sliku.

Rješenje 3. Problem postavljamo isto kao i prethodni samo što se dižemo za dvije dimenzije

$$\mathcal{A} = \mathbf{C} \times_p \mathbf{H}, \quad \mathbf{C} = \mathbf{S} \times_i \mathbf{D} \times_e \mathbf{E} \times_{ill} \mathbf{F} \times_{vp} \mathbf{G}, \quad (5.6)$$

$$\mathcal{A}(:, e, ill, vp, :) = \mathbf{C}(:, e, ill, vp, :) \times_p \mathbf{H}, \quad (5.7)$$

$$\mathbf{A}_{(e,ill,vp)} = \mathbf{C}_{(e,ill,vp)} \mathbf{H}^T, \quad e = 1, \dots, n_e, \quad ill = 1, \dots, n_{ill}, \quad vp = 1, \dots, n_{vp}, \quad (5.8)$$

$$\mathbf{a}_p^{(e,ill,vp)} = \mathbf{C}_{(e,ill,vp)} \mathbf{h}_p. \quad (5.9)$$

Ovdje je $\mathbf{a}_p^{(e,ill,vp)}$ slika osobe p s izrazom lica e , u osvjetljenju ill i iz kuta gledišta vp . Stupci matrice $\mathbf{C}_{(e,ill,vp)}$ razapinju prostor obilježja e , ill i vp , a \mathbf{h}_p su i dalje koeficijenti osobe p u navedenoj bazi. Ponavljamo već poznati postupak uz male modifikacije:

- za svaki e , ill i vp računamo koordinate slike \mathbf{z} u bazi stupaca od $\mathbf{C}_{(e,ill,vp)}$ i tražimo kojem od vektora \mathbf{h}_p su najbližije,
- izrazu lica e , osvjetljenju ill i kutu gledišta vp pridružimo dotičnu osobu p ,
- nepoznatu osobu klasificiramo kao onu čiji pripadni izraz lica, osvjetljenje i kut gledišta ostvaruju najmanju grešku.

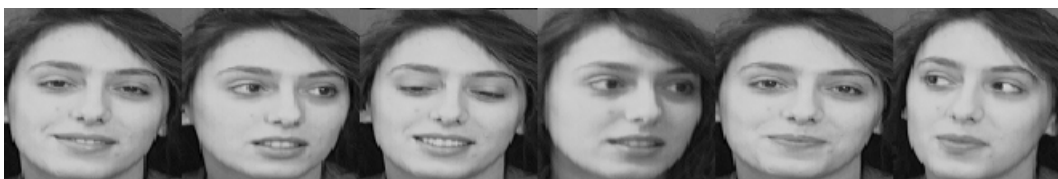
Koordinate u bazi (3.15) su ponovno rješenje problema najmanjih kvadrata

$$\min_{\alpha_{(e,ill,vp)}^s} \left\| \mathbf{C}_{(e,ill,vp)} \alpha_{(e,ill,vp)} - \mathbf{z} \right\|_2. \quad (5.10)$$

5.2 Rezultati

R baza slika

Biblioteka *rTensor* 7.1 jezika R posjeduje tenzor u koji su složene 92×112 slike 40 osoba u 10 različitih uvjeta ¹. Dio uvjeta koristimo kao skup za učenje, a dio za evaluaciju.



Slika 5.1: 6 različitih uvjeta u kojima je slikana jedna osoba iz R-ove baze slika.

Krenuvši sa samo 3 uvjeta u skupu za učenje točnost prepoznavanja iznosi solidnih 81%. Povećanjem broja uvjeta na 7 ona je narasla na približno 91%.

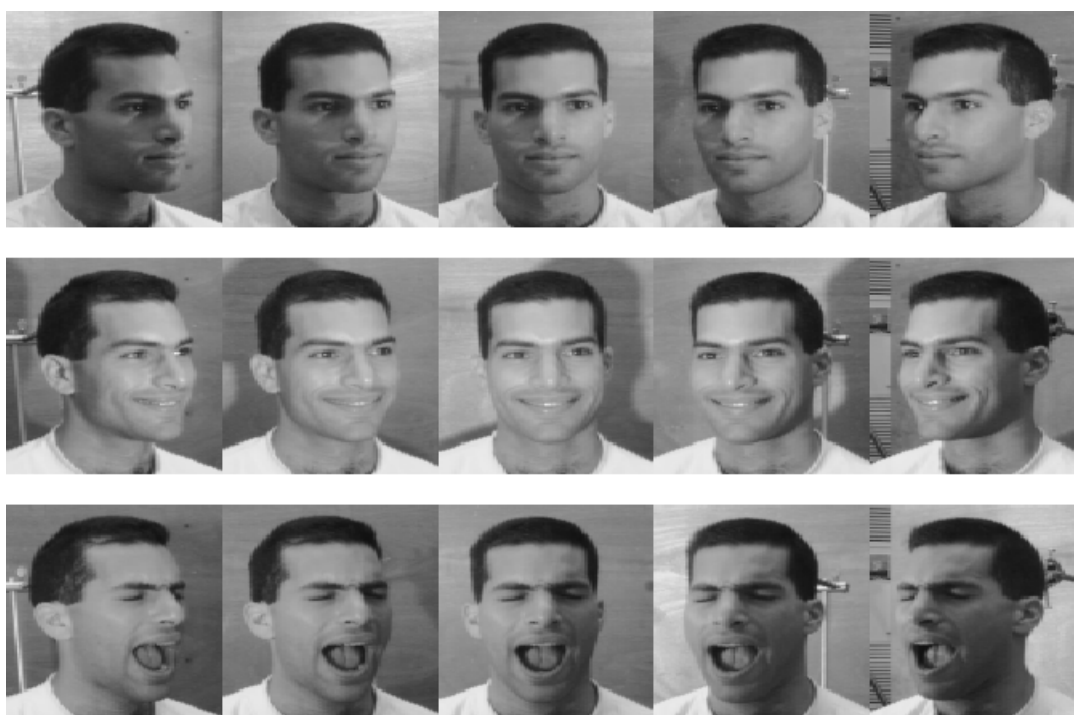
¹Uvjetima ovdje smatramo različite izraze lica, osvjetljenja i kuteve gledišta zajedno.

Prepoznavanje lica	točnost
Trening: 40 ljudi, 3 uvjeta. Test: 40 ljudi, 7 uvjeta.	81.07%
Trening: 40 ljudi, 5 uvjeta. Test: 40 ljudi, 5 uvjeta.	87%
Trening: 40 ljudi, 7 uvjeta. Test: 40 ljudi, 3 uvjeta.	90.83%

Tablica 5.1: Čak i mali skup slika pokazuje zadovoljavajuće rezultate. Osobe se prepoznaju u 81% slučajeva. Brojka raste na 91% ako se taj skup poveća.

Baza slika instituta Weizmann [1]

Korišten dio baze slika instituta Weizmann [1] sastoji se od 27 muškaraca slikanih s 3 izraza lica: normalan, blagi osmijeh i zijevanje, u 3 različita osvjetljenja: frontalno, slijeva i zdesna, te iz 5 kuteva gledišta: 0, ± 17 , ± 34 .



Slika 5.2: Primjer slika iz baze instituta Weizmann [1]. Prvi redak sadrži normalan izraz lica i osvjetljenje zdesna, drugi blagi osmijeh i frontalno osvjetljenje, a treći zijevanje i osvjetljenje slijeva. U stupcima se nalazi svih 5 kuteva gledišta.

Rezultati prepoznavanja variraju ovisno o tome uzimamo li za evaluaciju nepoznat izraz lica, osvjetljenje ili kut gledišta. Uspješnost klasifikacije prikazana je sljedećom tablicom, a kreće se između 80% i skoro 100%.

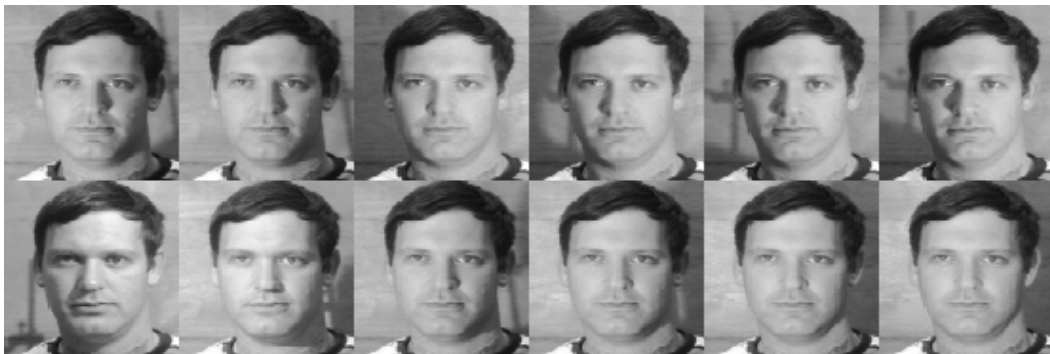
Prepoznavanje lica	točnost
Trening: 27 ljudi, 2 izraza lica, 3 osvjetljenja, 5 kuteva gledišta. Test: 27 ljudi, 1 izraz lica (normalan), 3 osvjetljenja, 5 kuteva gledišta.	99.51%
Trening: 27 ljudi, 3 izraza lica, 2 osvjetljenja, 5 kuteva gledišta. Test: 27 ljudi, 3 izraza lica, 1 osvjetljenje (slijeva), 5 kuteva gledišta.	84.2%
Trening: 27 ljudi, 3 izraza lica, 3 osvjetljenja, 4 kuta gledišta. Test: 27 ljudi, 3 izraza lica, 3 osvjetljenja, 1 kuta gledišta (-17).	81.89%

Tablica 5.2: Rezultati prepoznavanja lica. Najlakše se prepoznaje nepoznat izraz lica - u čak 99.51% slučajeva. Uspjeh pada na 81.89% kada je riječ o neviđenom kutu gledišta.

Poglavlje 6

Uklanjanje utjecaja osvjetljenja

Za kraj ćemo ukratko objasniti kako se koristeći HOSVD može ublažiti utjecaj osvjetljenja. Postupak je u praksi vrlo koristan. Naime, prilikom prepoznavanja lica nije bitno u kojem je osvjetljenju ono slikano već isključivo kojoj osobi pripada. Na neki način želimo da sve slike iste osobe budu što sličnije. Izgladimo li svjetlosne prijelaze to je upravo ono što ćemo postići.



Slika 6.1: Osoba iz Weizmann [1] baze slika u 12 različitih osvjetljenja.

Imamo slike 11 osoba u 16 različitih osvjetljenja veličine 128×88 [1]. Pohranili smo ih u tenzor ¹ $\mathcal{A} \in \mathbb{R}^{11264 \times 16 \times 11}$. Računamo HOSVD:

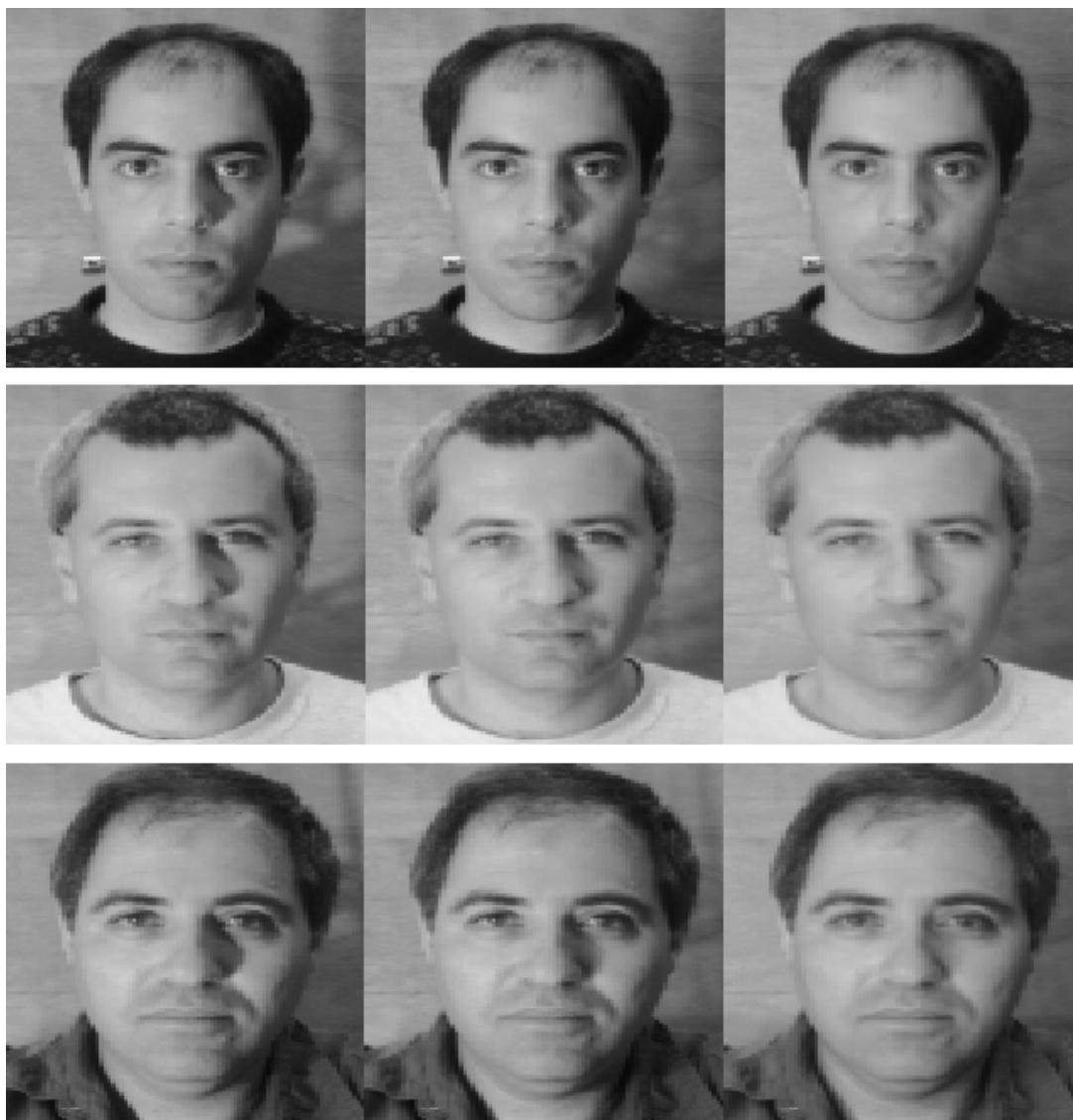
$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}. \quad (6.1)$$

Rang koji odgovara osvjetljenju smanjit ćemo na r režući matricu $\mathbf{U}^{(2)}$ i jezgri tenzor \mathcal{S}

$$\mathcal{A}' = \mathcal{S}(:, 1 : r, :) \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}(:, 1 : r) \times_3 \mathbf{U}^{(3)}. \quad (6.2)$$

¹Matrica piksela slike razvučena je u vektor duljine $128 \times 88 = 11264$.

Time smo na svim slikama tenzora \mathcal{A} izgladili svjetlosne prijelaze.



Slika 6.2: Prva slika je originalna, na drugoj je $r = 5$, a na trećoj $r = 2$. Vidi 7.5 za kod koji generira ovu sliku pomoću redukcije dimenzije. Primjetimo da se zbog pametnog načina odabira dimenzije u kojoj se redukcija vrši kvaliteta slike nije smanjila.

Poglavlje 7

Implementacija algoritma u programskom jeziku R

Ovo poglavlje posvećeno je praktičnom dijelu rada. Svi predstavljeni algoritmi za prepoznavanje znamenaka i lica te izgladivanje svjetlosnih prijelaza implementirani su u programskom jeziku R. Prvo slijedi kratak pregled gotovih funkcija koje se koriste u kodu, a zatim dolazimo do objašnjenja ručno pisanih funkcija. Svaki algoritam posjeduje svoj odjeljak. U njemu prolazimo dio po dio koda prisjećajući se svega što smo do sada već spominjali. Na kraju odjeljka nalazi se sažetak njegovih funkcija zajedno s dijelom koda koji je potrebno pozvati kako bi se izvršio cijeli algoritam.

7.1 Korišteni paketi i naredbe

R posjeduje nekoliko paketa specijaliziranih za rad s tenzorima. Obuhvaćene su osnovne operacije, norme, izravnavanje, HOSVD i dr.. U nastavku slijedi opis korištenih funkcija iz paketa *tensorA* i *rTensor* radi lakšeg snalaženja u daljnjem kodu. Detaljnije o istima može se naći u R-ovoj dokumentaciji [8][11].

tensorA

- **to.matrix.tensor(X, i)** - vraća matricu nastalu izravnavanjem tenzora \mathcal{X} u modu i .
- **to.tensor(X, dims)** - od numeričkih podataka X koji predstavljaju elemente tenzora radi tenzor dimenzija $dims$ ¹.

¹ Varijable koje predstavljaju ulaz ili izlaz neke funkcije pišemo nakošeno.

rTensor

- **ttl(X, list_mat, ms)** - množi tenzor \mathcal{X} listom matrica *list_mat* u modovima *ms*.
- **hosvd(X)** - vraća jezgri tenzor \mathcal{Z} i listu matrica \mathbf{U} iz rastava (3.1) tenzora \mathcal{X} .
- **faces_tnsr** - Skup slika 40 osoba u 10 različitih uvjeta. Slike dimenzija 92×112 složene su u $92 \times 112 \times 40 \times 10$ tenzor reda 4.

Vizualizacija i pohrana slika rađena je uz pomoć paketa *raster* koji služi za manipulaciju s prostornim podacima [9].

raster

- **raster(x)** - od podataka u varijabli *x* radi objekt tipa raster.
- **brick(x, ...)** - objekte proslijeđene funkciji slaže kao slojeve objekta tipa RasterBrick.
- **plotRGB(x)** - crta Red-Green-Blue sliku baziranu na troslojnom objektu tipa RasterBrick.

7.2 HOSVD

Baza svih algoritama opisanih u ovom radu je višedimenzionalna dekompozicija singularnih vrijednosti. Razumijevanje HOSVD-a vrlo je bitno. Iako R već posjeduje funkciju koja računa HOSVD promotrit ćemo ručno pisan analogon iste. Tako ćemo se bolje upoznati s ranije opisanim postupkom računanja elemenata dekompozicije 3.1. Jednostavnosti radi, ograničili smo se na tenzore reda 3.

Mod-*n* produkt

Koristimo vlastite funkcije za izračun mod-*n* produkta u smislu definicije 1.2.3.

```

1 # MOD-1 MNOZENJE
2
3 Model<-function(A,U){
4   # DIMENZIJE POCETNOG TENZORA I MATRICE
5   tdim<-dim(A)
6   mdim<-dim(U)
7   # DIMENZIJA TENZORA NASTALOG MNOZENJEM
8   newdim<-c(mdim[1],tdim[2:3])
9   newt<-to.tensor(c(1:prod(newdim)),

```

```

10  c(l=newdim[1],m=newdim[2],n=newdim[3]))
11  # RACUNANJE ELEMENATA NOVOG TENZORA
12  for(i1 in 1:newdim[1])
13    for(i2 in 1:newdim[2])
14      for(i3 in 1:newdim[3]){
15        el<-0
16        Atemp<-to.matrix.tensor(A[,i2,i3],1)
17        for(k in 1:mdim[2])el<-el+U[i1,k]*Atemp[k]
18        newt[i1,i2,i3]<-el
19      }
20  return(newt)
21 }
22
23 # MOD-2 MNOZENJE
24
25 Mode2<-function(A,U){
26   tdim<-dim(A)
27   mdim<-dim(U)
28   newdim<-c(tdim[1], mdim[1],tdim[3])
29   newt<-to.tensor(c(1:prod(newdim)),
30   c(l=newdim[1],m=newdim[2],n=newdim[3]))
31   for(i1 in 1:newdim[1])
32     for(i2 in 1:newdim[2])
33       for(i3 in 1:newdim[3]){
34         el<-0
35         Atemp<-to.matrix.tensor(A[i1,,i3],1)
36         for(k in 1:mdim[2])el<-el+U[i2,k]*Atemp[k]
37         newt[i1,i2,i3]<-el
38       }
39   return(newt)
40 }
41
42 # MOD-3 MNOZENJE
43
44 Mode3<-function(A,U){
45   tdim<-dim(A)
46   mdim<-dim(U)
47   newdim<-c(tdim[1:2],mdim[1])
48   newt<-to.tensor(c(1:prod(newdim)),
49   c(l=newdim[1],m=newdim[2],n=newdim[3]))
50   for(i1 in 1:newdim[1])
51     for(i2 in 1:newdim[2])
52       for(i3 in 1:newdim[3]){
53         el<-0
54         Atemp<-to.matrix.tensor(A[i1,i2,],1)
55         for(k in 1:mdim[2])el<-el+U[i3,k]*Atemp[k]
56         newt[i1,i2,i3]<-el

```

```

57     }
58     return(newt)
59 }

```

HOSVD

Funkcija $HOSVD(A)$ na temelju matričnog SVD-a računa elemente HOSVD-a tenzora \mathcal{A} (3.1). Pritom se služi funkcijama za mod- n produkt iz prethodnog odjeljka.

```

1 HOSVD<-function(A){
2
3   # IZRAVNAVANJE TENZORA
4   A1<-to.matrix.tensor(A,1)
5   A2<-to.matrix.tensor(A,2)
6   A3<-to.matrix.tensor(A,3)
7
8   # SVD MATRICA NASTALIH IZRAVNAVANJEM
9   U1<-svd(A1)$u
10  U2<-svd(A2)$u
11  U3<-svd(A3)$u
12
13 # RACUNANJE JEZGREN OG TENZORA
14 S<-Mode1(A,t(U1))
15 S<-Mode2(S,t(U2))
16 S<-Mode3(S,t(U3))
17
18 # ELEMENTI HOSVD-a
19 return(list(S,U1,U2,U3))
20 }

```

Sažetak funkcija

- **Mode1(A, U)** - vraća mod 1 produkt tenzora \mathcal{A} reda 3 i matrice \mathbf{U} .
- **Mode2(A, U)** - vraća mod 2 produkt tenzora \mathcal{A} reda 3 i matrice \mathbf{U} .
- **Mode3(A, U)** - vraća mod 3 produkt tenzora \mathcal{A} reda 3 i matrice \mathbf{U} .
- **HOSVD(A)** - vraća listu $\{S, U^{(1)}, U^{(2)}, U^{(3)}\}$ elemenata HOSVD-a tenzora \mathcal{A} reda 3.

U daljnjim kodovima koristit ćemo gotovu funkciju za HOSVD zato što radi brže.

7.3 Prepoznavanje znamenki

Poglavlje 4 objašnjava algoritam i diskutira rezultate prepoznavanja rukom pisanih znamenki. Sada ćemo vidjeti kako se spomenuti algoritam može implementirati u R-u.

Za početak skidamo MNIST bazu slika [10]. Slike su crno bijele veličine 28×28 piksela. Njih 70000 raspoređeno je u skup za treniranje (60000) i testiranje (10000). Nećemo koristiti čitavu bazu već njen manji dio. Oba skupa spremamo u obliku matrice sa 785 stupaca. Svaki redak predstavlja jednu sliku čije smo piksele razvukli u vektor duljine $28 \times 28 = 784$. Posljednji, 785. stupac, govori o kojoj je znamenki riječ. Sljedeći isječak koda je mala modifikacija koda za učitavanje i pohranjivanje MNIST baze znamenki koji se može naći na [2].

```

1 DownloadImages<-function(){
2
3   # SKIDANJE SLIKA ZNAMENKI S INTERNETA
4   download.file(
5     "http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz",
6     "train-images-idx3-ubyte.gz")
7   download.file(
8     "http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz",
9     "train-labels-idx1-ubyte.gz")
10  download.file(
11    "http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz",
12    "t10k-images-idx3-ubyte.gz")
13  download.file(
14    "http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz",
15    "t10k-labels-idx1-ubyte.gz")
16
17  # UNZIPANJE DATOTEKA
18  R.utils::gunzip("train-images-idx3-ubyte.gz")
19  R.utils::gunzip("train-labels-idx1-ubyte.gz")
20  R.utils::gunzip("t10k-images-idx3-ubyte.gz")
21  R.utils::gunzip("t10k-labels-idx1-ubyte.gz")
22
23  # UCITAVANJE DATOTEKA SA SLIKAMA
24  loadImageFile<-function(filename){
25    ret<-list()
26    f<-file(filename, 'rb')
27    readBin(f, 'integer', n=1, size=4, endian='big')
28    n<-readBin(f, 'integer', n=1, size=4, endian='big')
29    nrow<-readBin(f, 'integer', n=1, size=4, endian='big')
30    ncol<-readBin(f, 'integer', n=1, size=4, endian='big')
31    x<-readBin(f, 'integer', n=n*nrow*ncol, size=1, signed=FALSE)
32    close(f)
33    data.frame(matrix(x, ncol=ncol, byrow=TRUE))
34  }

```



```

35
36 # UCITAVANJE DATOTEKA S OZNAKAMA
37 LoadIlabelFile<-function(filename){
38   f<-file(filename, 'rb')
39   readBin(f, 'integer', n=1, size=4, endian='big')
40   n<-readBin(f, 'integer', n=1, size=4, endian='big')
41   y<-readBin(f, 'integer', n=n, size=1, signed=FALSE)
42   close(f)
43   y
44 }
45
46 # SKUPOVI ZA TRENING I TEST
47 train<-LoadImageFile("train-images-idx3-ubyte")
48 test<-LoadImageFile("t10k-images-idx3-ubyte")
49
50 # UBACIVANJE OZNAKA
51 train$y<-as.factor(LoadIlabelFile("train-labels-idx1-ubyte"))
52 test$y<-as.factor(LoadIlabelFile("t10k-labels-idx1-ubyte"))
53
54 return(list(train, test))
55 }

```

Dakle, funkcija *DownloadImages()* vraća listu koja se sastoji od skupa za treniranje i testiranje. Prije bilo kakve daljnje obrade podatke ćemo prebaciti u tenzor $28 \times 28 \times ni$, gdje je *ni* broj slika koje koristimo.

Želimo znati koliko baznih matrica iz (4.1) nam je dovoljno da objasnimo većinu varijacije unutar jedne znamenke. Za svaku znamenku crtamo njene singularne vrijednosti 1.

```

1 SingularValues<-function(){
2   par(mfrow=c(2,5))
3   for(dig in 0:9){
4     # IZDVAJANJE 100 SLIKA ZNAMENKE "DIG"
5     digits<-train[which(train[,785]==dig),-785]
6     digits<-digits[1:100,]
7     A<-new("Tensor", 3L, c(28L, 28L, 100L),
8     data=as.vector(t(digits)))
9     hoasvd<-hosvd(A)
10    S<-hoasvd$Z
11    # CRTANJE SINGULARNIH VRIJEDNOSTI
12    f<-numeric(100)
13    for(i in 1:100)f[i]<-Fnorm(matrix(vec(S[, , i]), 28, 28))
14    plot(f, main=paste0("Znamenka: ", dig), type="l",
15    ylab="Singularne Vrijednosti", xlab="")
16  }
17 }

```

Korištena je pomoćna funkcija *Fnorm(A)* koja računa Frobeniusovu normu matrice.

```

1 Fnorm<-function(A){
2   fn<-sqrt(sum(A*A))
3   return(fn)
4 }

```

Pozivom funkcije *SingularValues()* bit će nacrtan graf singularnih vrijednosti za sve znamenke redom. Naslućujemo da je dovoljno uzeti desetak baznih matrica.

Idući korak je računanje baznih matrica za pojedinu znamenku kako je opisano u (3.13).

```

1 BaseMatrices<-function(dig){
2   # IZDVAJANJE 100 SLIKA ZNAMENKE "DIG"
3   digits<-train[which(train[,785]==dig),-785]
4   A<-new("Tensor",3L,c(28L,28L,100L),data=as.vector(t(digits[1:100,])))
5
6   #HOSVD
7   hoasvd<-hosvd(A)
8   S<-hoasvd$Z
9   u1<-hoasvd$U[[1]]
10  u2<-hoasvd$U[[2]]
11  u3<-hoasvd$U[[3]]
12
13  # BAZNE MATRICE
14  A1<-ttl(S[, ,1],list(u1,u2),c(1,2))
15  A2<-ttl(S[, ,2],list(u1,u2),c(1,2))
16  A3<-ttl(S[, ,3],list(u1,u2),c(1,2))
17  A4<-ttl(S[, ,4],list(u1,u2),c(1,2))
18  A5<-ttl(S[, ,5],list(u1,u2),c(1,2))
19  A6<-ttl(S[, ,6],list(u1,u2),c(1,2))
20  A7<-ttl(S[, ,7],list(u1,u2),c(1,2))
21  A8<-ttl(S[, ,8],list(u1,u2),c(1,2))
22  A9<-ttl(S[, ,9],list(u1,u2),c(1,2))
23  A10<-ttl(S[, ,10],list(u1,u2),c(1,2))
24
25  # PREBACIVANJE U KLASU MATRICA
26  A1<-matrix(vec(A1),nrow=28,ncol=28)
27  A2<-matrix(vec(A2),nrow=28,ncol=28)
28  A3<-matrix(vec(A3),nrow=28,ncol=28)
29  A4<-matrix(vec(A4),nrow=28,ncol=28)
30  A5<-matrix(vec(A5),nrow=28,ncol=28)
31  A6<-matrix(vec(A6),nrow=28,ncol=28)
32  A7<-matrix(vec(A7),nrow=28,ncol=28)
33  A8<-matrix(vec(A8),nrow=28,ncol=28)
34  A9<-matrix(vec(A9),nrow=28,ncol=28)
35  A10<-matrix(vec(A10),nrow=28,ncol=28)
36
37  return(list(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10))
38 }

```

Listu prvih 10 baznih matrica za znamenku *znam* dobit ćemo kao rezultat funkcije *BaseMatrices(znam)*.

Dosad smo koristili samo skup za treniranje. Slijedi prepoznavanje znamenki iz skupa za testiranje. Prolazimo po cijelom skupu i svaku njegovu znamenku *Z* klasificiramo kao onu u čijoj se bazi (3.15) može najtočnije prikazati. Pomoćna funkcija *Error(Z, list)* računa grešku (4.4) prilikom prikaza slike znamenke *Z* u bazi matrica *list*.

```

1 Error<-function(Z, list){
2
3   # RASPAKIRAVANJE LISTE
4   A1<-list[[1]]
5   A2<-list[[2]]
6   A3<-list[[3]]
7   A4<-list[[4]]
8   A5<-list[[5]]
9   A6<-list[[6]]
10  A7<-list[[7]]
11  A8<-list[[8]]
12  A9<-list[[9]]
13  A10<-list[[10]]
14
15  # RACUNANJE KOEFICIJENATA ZA BAZU
16  z1<-sum(Z*A1)/sum(A1*A1)
17  z2<-sum(Z*A2)/sum(A2*A2)
18  z3<-sum(Z*A3)/sum(A3*A3)
19  z4<-sum(Z*A4)/sum(A4*A4)
20  z5<-sum(Z*A5)/sum(A5*A5)
21  z6<-sum(Z*A6)/sum(A6*A6)
22  z7<-sum(Z*A7)/sum(A7*A7)
23  z8<-sum(Z*A8)/sum(A8*A8)
24  z9<-sum(Z*A9)/sum(A9*A9)
25  z10<-sum(Z*A10)/sum(A10*A10)
26
27  # UKUPNA GRESKA
28  #error<-Z-z1*A1-z2*A2-z3*A3
29  #error<-Z-z1*A1-z2*A2-z3*A3-z4*A4-z5*A5
30  #error<-Z-z1*A1-z2*A2-z3*A3-z4*A4-z5*A5-z6*A6-z7*A7
31  error<-Z-z1*A1-z2*A2-z3*A3-z4*A4-z5*A5-z6*A6-z7*A7-z8*A8-z9*A9-z10*A10
32  error<-sqrt(sum(error*error))
33  return(error)
34 }

1 Accuracy<-function(){
2
3   # BAZNE MATRICE ZA SVAKU ZNAMENKU
4   list0<-BaseMatrices(0)
5   list1<-BaseMatrices(1)

```

```

6 list2<-BaseMatrices(2)
7 list3<-BaseMatrices(3)
8 list4<-BaseMatrices(4)
9 list5<-BaseMatrices(5)
10 list6<-BaseMatrices(6)
11 list7<-BaseMatrices(7)
12 list8<-BaseMatrices(8)
13 list9<-BaseMatrices(9)
14
15 # PROLAZAK PO TESTNOM SKUPU
16 n<-nrow(test)
17 est<-matrix(0,n,1)
18 for(i in 1:n)
19 {
20 # SLIKA "Z" IZ TESTNOG SKUPA
21 Z<-matrix(as.numeric(test[i,-785]),28,28)
22
23 # RACUNANJE GRESKE ZA SVAKU ZNAMENKU
24 error<-numeric(10)
25 error[1]<-Error(Z,list0)
26 error[2]<-Error(Z,list1)
27 error[3]<-Error(Z,list2)
28 error[4]<-Error(Z,list3)
29 error[5]<-Error(Z,list4)
30 error[6]<-Error(Z,list5)
31 error[7]<-Error(Z,list6)
32 error[8]<-Error(Z,list7)
33 error[9]<-Error(Z,list8)
34 error[10]<-Error(Z,list9)
35
36 # KLASIFIKACIJA SLIKE "Z"
37 est[i,1]<-which(error==min(error))-1
38 }
39
40 # UKUPNA TOCNOST ALGORITMA
41 res<-cbind(test,est)
42 accuracy<-length(which(res[,785]==res[,786]))/n*100
43 accuracy<-paste0("Tocnost iznosi ",accuracy,"%.")
44 print(accuracy)
45 }

```

Funkcija *Accuracy()* provodi gornji postupak i ispisuje kolika je ukupna točnost klasifikacije nakon što je obrađen cijeli testni skup. Time je završeno prepoznavanje rukom pisanih znamenki. Slijedi sažetak funkcija i kod koji je potrebno pozvati kako bi se izvršio cjelokupan algoritam.

Sažetak funkcija

- **DownloadImages()** - sa stranice [10] skida MNIST bazu rukom pisanih znamenki. Vraća listu $\{train, test\}$ skupova za treniranje i testiranje u obliku matrica gdje svaki redak predstavlja jednu sliku. U zadnjem stupcu nalazi se oznaka znamenke na slici.
- **Fnorm(A)** - računa Frobeniusovu normu matrice **A**.
- **SingularValues()** - crta grafove singularnih vrijednosti moda 3 za svaku znamenku.
- **BaseMatrices(dig)** - vraća listu prvih 10 baznih matrica (3.15) znamenke *dig*. Koristi 100 slika dotične znamenke.
- **Error(A)** - računa grešku (4.4) prilikom prikaza nepoznate znamenke *Z* u bazi matrica *list* koju vraća funkcija *BaseMatrices()*.
- **Accuracy()** - Ispisuje točnost algoritma 1 za prepoznavanje rukom pisanih znamenki.

Poziv Algoritma

```
1 r<-DownloadImages()
2 train<-r[[1]]
3 test<-r[[2]]
4 SingularValues()
5 Accuracy()
```

7.4 Prepoznavanje lica

Priprema podataka

Baza instituta Weizmann [1] sadrži slike veličine 512×352 piksela u png formatu. Prebacujemo ih u csv datoteku u istom obliku kao što smo pohranjivali znamenke: svaki redak predstavlja jednu sliku, a u zadnjem stupcu se nalazi oznaka osobe na slici. Prije spremanja slike smo smanjili na veličinu 128×88 jer su inače prevelike za efikasan izračun.

```
1 ImageCSV<-function() {
2
3   # UCITAVANJE SLIKA
4   setwd("~/FaceBase_png/ex2-3vp0-4ill0-2")
5   lf<-list.files()
6   l<-length(lf)
7
8   # DOHVACANJE IMENA SVIH OSOBA
9   names<-rep('g', l)
10  for(i in 1:l) names[i]<-strsplit(lf[i], split='-')[[1]][1]
```

```

11  names<-unique(names)
12  names<-setdiff(names, 'yaron')
13
14  # UVJETI U KOJIMA SU LJUDI SLIKANI
15  ex<-c(1,2,3)
16  vp<-c(0,1,2,3,4)
17  ill<-c(0,1,2)
18
19  # IMENA SVIH DATOTEKA
20  files<-rep('g',1)
21  index<-1
22  for(n in names){
23    for(i in vp){
24      for(j in ill){
25        for(k in ex){
26          files[index]<-paste0(n, '-vp',i, '-il',j, '-ex',k, '.png')
27          index<-index+1
28        }
29      }
30    }
31  }
32
33  # POTREBNE DIMENZIJE
34  npix<-128*88
35  nex<-3
36  nvp<-5
37  ni<-3
38  np<-27
39  l<-27*3*3*5
40  for5D<-matrix(0,1,npix)
41
42  setwd("../")
43  setwd("../")
44  setwd("~/FaceBase_png/EX1")
45
46  # POOLING
47  for(k in 1:l){
48    name<-files[k]
49    image<-raster(name, band=1)
50    a<-as.matrix(image)
51    na<-matrix(0,128,88)
52    for(i in 1:128){
53      for(j in 1:88){
54        ii<-(i-1)*4+1
55        jj<-(j-1)*4+1
56        na[i,j]<-mean(a[ii:(ii+3),jj:(jj+3)])
57      }

```

```

58     }
59     for5D[k,]<-as.vector(na)
60     if(k%%50==0)print(k)
61   }
62
63   # SPREMANJE SLIKA
64   setwd("../")
65   setwd("../")
66   write.csv(for5D,'Weizmann5D.csv')
67 }

```

Skup za treniranje i testiranje

Evaluaciju algoritama 2 3 provodimo uz pomoć skupa za treniranje i skupa za testiranje. Vidjeli smo da točnost klasifikacije ovisi o njihovom odabiru 5.2. Funkcije koje slijede rade nekoliko vrsta potrebnih skupova ovisno o tome želimo li testirati na nepoznatom izrazu lica, osvjetljenju, kutu gledišta 5.2 ili zadanom broju uvjeta 5.2.

R baza slika

Baza slika jezika R koristi dva moda tenzora za pohranjivanje piksela. Želimo li ih zamijeniti jednim pozvat ćemo funkciju *Order3Tensor* koja vraća tenzor reda 3 2.

```

1 Order3Tensor<-function() {
2   ft<-faces_tnsr
3   t<-new("Tensor",3L,c(10304L,10L,40L),data=rep(0,10304*400))
4   for(i in 1:40){
5     for(j in 1:10){
6       t[,j,i]<-vec(ft[, ,i,j])
7     }
8   }
9   return(t)
10 }

```

Dobiveni tenzor pretvaramo u skup za treniranje i testiranje uz pomoć *RfaceBase* funkcije.

```

1 RfaceBase<-function(nex,np){
2   T<-Order3Tensor()
3   train<-T[,1:nex,1:np]
4   test<-matrix(0,(10-nex)*np,dim(train)[1]+1)
5   row<-1
6   for(p in 1:np){
7     for(e in (nex+1):10){
8       z<-vec(T[,e,p])
9       test[row,]<-c(z,p)

```

```

10     row<-row+1
11   }
12 }
13 return(list(train, test))
14 }

```

Baza slika instituta Weizmann

Pomoćna funkcija *Weizmann335()* slaže bazu 5.2 u tenzor $11264 \times 3 \times 3 \times 5 \times 27 \times 3$.

```

1 Weizmann335<-function() {
2   fort<-read.csv('Weizmann5D.csv')
3   fort<-as.matrix(fort[, -1])
4   vfort<-as.vector(t(fort))
5   t<-new("Tensor", 5L, c(11264L, 3L, 3L, 5L, 27L), data=vfort)
6   return(t)
7 }

```

Zatim ju razdvajamo na dio za trening i test.

```

1 Expression<-function(ex, np) {
2   t<-Weizmann335()
3   train<-t[, -ex, , 1:np]
4   test<-matrix(0, 15*np, (dim(t)[1]+1))
5   row<-1
6   for(p in 1:np) {
7     for(i in 1:3) {
8       for(v in 1:5) {
9         z<-vec(t[, ex, i, v, p])
10        test[row, ]<-c(z, p)
11        row<-row+1
12      }
13    }
14  }
15  return(list(train, test))
16 }
17
18 Illumination<-function(il, np) {
19   t<-Weizmann335()
20   train<-t[, , -il, , 1:np]
21   test<-matrix(0, 15*np, (dim(t)[1]+1))
22   row<-1
23   for(p in 1:np) {
24     for(e in 1:3) {
25       for(v in 1:5) {
26         z<-vec(t[, e, il, v, p])
27         test[row, ]<-c(z, p)
28         row<-row+1

```



```

29     }
30   }
31 }
32 return(list(train, test))
33 }
34
35 ViewPoint<-function(vp, np){
36   t<-Weizmann335()
37   train<-t[, , -vp, 1:np]
38   test<-matrix(0, 9*np, (dim(t)[1]+1))
39   row<-1
40   for(p in 1:np){
41     for(i in 1:3){
42       for(e in 1:3){
43         z<-vec(t[, e, i, vp, p])
44         test[row, ]<-c(z, p)
45         row<-row+1
46       }
47     }
48   }
49   return(list(train, test))
50 }

```

Klasifikacija osobe

Nad skupom za trening provodimo HOSVD. Spremamo njegove elemente i računamo tenzor C iz (5.6). Zatim prelazimo na testni skup. Osobe na slikama identificiramo kao one kojima su najslabije u smislu 3. Kalkulacija indeksa uvjeta po kojima algoritam treba proći 3 prebačena je u funkciju *Indices()*. Tako klasifikacijska funkcija 7.4 prihvaća tenzore proizvoljnog reda.

```

1 Indices<-function(d){
2
3   # RED TENZORA
4   nmod<-length(d)
5
6   # PRVI I ZADNJI MOD NE SADRZE UVJETE U KOJIMA SE SLIKALO
7   loop<-d[-c(1, nmod)]
8   ind<-nmod-2
9
10  # MATRICA ZA INDEKSE
11  indices<-matrix(0, prod(loop), ind)
12  col<-1
13  while(ind>1){
14    if(col==1){x<-1}else{x<-prod(loop[1:(col-1)])}

```

```

15   indices[,col]<-rep(c(1:loop[col]),each=prod(loop[-c(1:col)]),times=x
16   )
16   ind<-ind-1
17   col<-col+1
18   }
19   indices[,col]<-rep(c(1:loop[col]),times=(nrow(indices)/loop[col]))
20   return(indices)
21 }

```

PersonClassification() koristi *Indices()* i za sada radi s tenzorima reda 3, 4, 5 ili 6. Po želji se lako može prilagoditi i ostalim. Ona minimizira maloprije spomenutu vrstu greške 3 kako bi saznala o kojoj je osobi riječ.

```

1 PersonClassification<-function(C,H,d,Z,indices){
2
3   # DEFAULTNE POSTAVKE
4   error<-100000000
5   person<-0
6   np<-d[length(d)]
7
8   # PROLAZAK PO SVIH KOMBINACIJAMA UVJETA NA SLIKAMA
9   for(ndx in 1:nrow(indices)){
10
11     # MATRICA C OVISI O REDU ORIGINALNOG TENZORA
12     if(ncol(indices)==1)
13       Cel<-matrix(vec(C[,indices[ndx,1],]),nrow=d[1],ncol=d[length(d)])
14     if(ncol(indices)==2)
15       Cel<-matrix(vec(C[,indices[ndx,1],indices[ndx,2],]),nrow=d[1],ncol=
16       np)
17     if(ncol(indices)==3)
18       Cel<-matrix(vec(C[,indices[ndx,1],indices[ndx,2],
19       indices[ndx,3],]),nrow=d[1],ncol=np)
20     if(ncol(indices)==4)
21       Cel<-matrix(vec(C[,indices[ndx,1],indices[ndx,2],
22       indices[ndx,3],indices[ndx,4],]),nrow=d[1],ncol=np)
23
24     # PROBLEM NAJMANJIH KVADRATA
25     alpha<-qr.solve(Cel,Z)
26
27     # PREPOZNAVANJE OSOBE
28     for(p in 1:np){
29       hp<-H[p,]
30       er<-alpha-hp
31       err<-sqrt(sum(er*er))
32       if(err<error){
33         error<-err
34         person<-p
35       }
36     }
37   }
38 }

```

```

35     }
36   }
37   return(person)
38 }

```

Cijeli postupak klasifikacije nalazi se unutar funkcije *TensorFaces()*.

```

1 TensorFaces<-function() {
2
3   # DIMENZIJA TENZORA
4   nmodes<-length(dim(train))
5   d<-dim(train)
6
7   # HOSVD
8   print('HOSVD u tijeku...')
9   hotsvd<-hosvd(train)
10  print('HOSVD dovršen.')
11  S<-hotsvd$Z
12  U<-hotsvd$U
13
14  Uttl<-U
15  Uttl[[nmodes]]<-NULL
16  modes<-1:(nmodes-1)
17
18  # RACUNANJE TENZORA C
19  print('Racunanje tenzora C...')
20  C<-ttl(S,Uttl,ms=modes)
21
22  # POCETAK EVALUACIJE
23  eval<-matrix(0,nrow(test),2)
24  nim<-nrow(test)
25  indices<-Indices(d)
26  print('Izvršava se klasifikacija slika iz testne skupine...')
27
28  # PROLAZAK PO TESTNOM SKUPU
29  for(im in 1:nim){
30    eval[im,1]<-test[im,(d[1]+1)]
31    image<-test[im,-(d[1]+1)]
32    eval[im,2]<-PersonClassification(C,U[[nmodes]],d,image,indices)
33    if(im%%50==0)print(paste0("Obradeno je ",im, " slika..."))
34  }
35  print('Klasifikacija dovršena.')
36  success<-round(100*length(which(eval[,1]==eval[,2]))/nim,2)
37  success<-paste0("Točnost iznosi: ",success,"%.")
38  print(success)
39  #return(eval)
40 }

```

Sažetak funkcija

- **ImageCSV()** - učitava slike iz datoteka preuzetih s [1], smanjuje ih i pohranjuje u csv datoteku u obliku matrice. Radi se o 27 osoba s 3 izraza lica, u 3 osvjetljenja i iz 5 kuteva gledišta. Svaki redak predstavlja jednu sliku, a u zadnjem stupcu nalazi se redni broj osobe na slici.
- **Order3Tensor()** - tenzor reda 4 koji vraća funkcija *faces_tnsr* (vidi 7.1) slaže u tenzor reda 3 razvlačeći sliku u vektor. Vraća dobiveni tenzor.
- **RfaceBase(nex,np)** - Od tenzora dobivenog pomoću funkcije *Order3Tensor()* radi skup za trening i test u matičnom obliku. Uzima se samo prvih *np* osoba. Skup za treniranje ima *nex* izraza lica, a skup za testiranje ostalih $10 - nex$.
- **Weizmann335()** - Uzima csv datoteku koju radi funkcija *ImageCSV()* i njene podatke slaže u tenzor reda 5. Prva dimenzija su pixeli u obliku vektora, druga izrazi lica, treća osvjetljenja, četvrta kutevi gledišta, a peta osobe. Funkcija vraća opisani tenzor.
- **Expression(ex,np)** - od tenzora dobivenog funkcijom *Weizmann335()* radi skupove za treniranje i testiranje i vraća njihovu listu: {train,test}. Varijabla *train* je tenzor, a *test* matrica. Skup za testiranje sastoji se od izraza lica *ex* u svim kombinacijama ostalih uvjeta. Za oba skupa uzima se prvih *np* osoba.
- **Illumination(ill,np)** - od tenzora dobivenog funkcijom *Weizmann335()* radi skupove za treniranje i testiranje i vraća njihovu listu: {train,test}. Varijabla *train* je tenzor, a *test* matrica. Skup za testiranje sastoji se od osvjetljenja *ill* u svim kombinacijama ostalih uvjeta. Za oba skupa uzima se prvih *np* osoba.
- **ViewPoint(vp,np)** - od tenzora dobivenog funkcijom *Weizmann335()* radi skupove za treniranje i testiranje i vraća njihovu listu: {train,test}. Varijabla *train* je tenzor, a *test* matrica. Skup za testiranje sastoji se od kuta gledišta *vp* u svim kombinacijama ostalih uvjeta. Za oba skupa uzima se prvih *np* osoba.
- **Indices()** - Prima vektor dimenzija *d* tenzora *train* iz prošlih funkcija. Vraća matricu gdje je svaki redak jedna kombinacija uvjeta na slici po kojima funkcija *PersonClassification()* treba proći.
- **PersonClassification(C,H,d,z,indices)** - prima tenzor *C*, matricu *H* (vidi 2), vektor dimenzija *d* tenzora *train*, nepoznatu sliku *Z* i rezultat prošle funkcije *indices*. Vraća redni broj osobe na slici *Z*. Tenzor *train* može biti dimenzije 3, 4, 5 ili 6.
- **TensorFaces()** - ispisuje uspješnost algoritma za prepoznavanje lica 2 i 3.

Za kraj ponovo slijedi primjer poziv algoritma za bazu instituta Weizmann [1] u kojem se testni skup sastoji od izraza lica broj 1.

Poziv Algoritma

```
1 ImageCSV()
2 ex<-1
3 np<-5
4 tt<-Expression(ex,np)
5 train<-tt[[1]]
6 test<-tt[[2]]
7 TensorFaces()
```

7.5 Uklanjanje utjecaja osvjetljenja

Fotografije spremamo na već opisan način. Ovaj put imamo 11 muškaraca u 16 osvjetljenja također iz Weizmann baze [1].

```
1 ImageCSV<-function() {
2
3   # DIREKTORIJ U KOJEM SE NALAZE SLIKE
4   setwd("~/FaceBase_png/Illu")
5   list.files()->lf
6   length(lf)->l
7
8   # POTREBNE DIMENZIJE
9   ni<-16
10  np<-l/16
11  npix<-128*88
12  fort<-matrix(0,l,npix)
13
14  # POOLING
15  for(k in 1:l){
16    name<-lf[k]
17    image<-raster(name,band=1)
18    a<-as.matrix(image)
19    na<-matrix(0,128,88)
20
21    for(i in 1:128){
22      for(j in 1:88){
23        ii<-(i-1)*4+1
24        jj<-(j-1)*4+1
25        na[i,j]<-mean(a[ii:(ii+3),jj:(jj+3)])
26      }
27    }
28    fort[k,]<-as.vector(na)
```

```

29 }
30
31 # SPREMANJE U CSV DATOTEKU
32 setwd("..")
33 setwd("..")
34 write.csv(fort, 'WeizmannIllumination.csv')
35 }

```

Posebna funkcija slaže podatke u odgovarajuć tenzor $11264 \times 16 \times 11$ i računa njegov HOSVD.

```

1 IllHOSVD<-function() {
2
3   # PREBACIVANJE SLIKA U TENZOR
4   fort<-read.csv('WeizmannIllumination.csv', header=T)
5   fort<-as.matrix(fort[, -1])
6   vfort<-as.vector(t(fort))
7   t<-new("Tensor", 3L, c(11264L, 16L, 11L), data=vfort)
8
9   # HOSVD
10  hotsvd<-hosvd(t)
11  S<-hotsvd$Z
12  U1<-hotsvd$U[[1]]
13  U2<-hotsvd$U[[2]]
14  U3<-hotsvd$U[[3]]
15
16  return(list(t, S, U1, U2, U3))
17 }

```

Izgladit ćemo svjetlosne prijelaze na slikama čitavog skupa režući jezgri tenzor S i odgovarajuću matricu $U^{(i)}$ (6.2). Skup reduciran po dimenziji osvjetljenja vraća funkcija *IlluminationReduction()*.

```

1 IlluminationReduction<-function(S, U1, U2, U3, r) {
2   # REZANJE JEZGREN OG TENZORA I MATRICE U(2)
3   tri<-ttl(S[, 1:r, ], list(U1, U2[, 1:r], U3), ms=c(1, 2, 3))
4   return(tri)
5 }

```

Na kraju imamo funkciju koja ilustrira dobiveni rezultat 6.

```

1 Illustration<-function(t, S, U1, U2, U3, person) {
2   par(mfrow=c(1, 3))
3
4   # ORIGINALNA SLIKA
5   image<-vec(t[, 1, person])
6   image<-matrix(image, 128, 88)
7   image<-raster(image)
8   plotRGB(brick(image, image, image))

```

```

9
10 # r=5
11 tri<-IlluminationReduction(S,U1,U2,U3,5)
12 image<-vec(tri[,1,person])
13 image<-matrix(image,128,88)
14 image<-raster(image)
15 plotRGB(brick(image,image,image))
16
17 # r=2
18 tri<-IlluminationReduction(S,U1,U2,U3,2)
19 image<-vec(tri[,1,person])
20 image<-matrix(image,128,88)
21 image<-raster(image)
22 plotRGB(brick(image,image,image))
23 }

```

Sažetak funkcija

- **ImageCSV()** - učitava slike iz datoteka preuzetih s [1], smanjuje ih i pohranjuje u csv datoteku u obliku matrice. Radi se o 11 osoba u 16 različitih osvjetljenja. Svaki redak predstavlja jednu sliku, a u zadnjem stupcu nalazi se redni broj osobe na slici.
- **IIIHOSVD** - vraća listu $\{t,S,U1,U2,U3\}$ tenzora iz gornje csv datoteke i elemenata njegovog HOSVD-a.
- **IlluminationReduction** - prima elemente HOSVD-a iz prošle funkcije zajedno s rangom r . Vraća tenzor kojem je rang u modu osvjetljenja reduciran na r (vidi (6.2)).
- **Illustration(t,S,U1,U2,U3,person)** - prima listu $\{t,S,U1,U2,U3\}$ koju vraća funkcija *IIIHOSVD()* i redni broj osobe. Za dotičnu osobu crta originalnu sliku zajedno sa slikama gdje je rang osvjetljenja smanjen na 5 odnosno 2.

Za kraj ponovo slijedi primjer poziv algoritma za bazu instituta Weizmann [1] u kojem se testni skup sastoji od izraza lica broj 1.

Poziv Algoritma

```

1 ImageCSV()
2 th<-IIIHOSVD()
3 t<-th[[1]]
4 S<-th[[2]]
5 U1<-th[[3]]
6 U2<-th[[4]]
7 U3<-th[[5]]

```

7.5. UKLANJANJE UTJECAJA OSVJETLJENJA

51

```
8 person<-2  
9 Illustration(t,S,U1,U2,U3, person)
```


Bibliografija

- [1] *FaceBase*, rujan 2018, <http://www.wisdom.weizmann.ac.il/~vision/FaceBase/>.
- [2] *Load MNIST*, rujan 2018, <https://gist.github.com/daviddalpiaz/ae62ae5ccd0bada4b9acd6dbc9008706>.
- [3] *Tensor in SAS*, rujan 2018, <http://www.sas-programming.com/2009/12/tensor-in-sas.html>.
- [4] M. Alex, O. Vasilescu i D. Terzopoulos, *Multilinear Subspace Analysis of Image Ensembles*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03) (2003), 5–6.
- [5] L. De Lathauwer, B. De Moor i J. Vandewalle, *Multilinear Singular Value Tensor Decompositions*, SIAM Journal on Matrix Analysis and Applications (2000).
- [6] L. Eldén, *Matrix Methods in Data Mining and Pattern Recognition*, SIAM Fundamentals of Algorithms, 2007.
- [7] T. G. Kolda i B. W. Bader, *Tensor Decompositions and Applications*, Society for Industrial and Applied Mathematics (2009), 455–500.
- [8] K. Gerald van den Boogaart, *Package 'tensorA'*, rujan 2018, <https://cran.r-project.org/web/packages/tensorA/tensorA.pdf>.
- [9] R. J. Hijmans, J. van Etten, J. Cheng, M. Mattiuzzi, M. Sumner, J. A. Greenberg, O. Perpinan Lamigueiro, A. Bevan, E. B. Racine, A. Shortridge i A. Ghosh, *Package 'raster'*, rujan 2018, <https://cran.r-project.org/web/packages/raster/raster.pdf>.
- [10] Y. LeCun, C. Cortes i C. J. C. Burges, *THE MNIST DATABASE of handwritten digits*, rujan 2018, <http://yann.lecun.com/exdb/mnist/>.

- [11] J. Li, J. Bien i M. Wells, *Package 'rTensor'*, rujan 2018, <https://cran.r-project.org/web/packages/rTensor/rTensor.pdf>.
- [12] Saša Singer, *Numerička analiza, 25. predavanje*, rujan 2018, https://web.math.pmf.unizg.hr/~singer/num_anal/NA_0910/25.pdf.
- [13] Tyler Ueltschi, *Third-Order Tensor Decompositions and Their Application in Quantum Chemistry*, rujan 2018, <http://mathbook.pugetsound.edu/tyleru/section-1.html>.
- [14] Y. Yu, J. Jin, F. Liu i S. Crozier, *Multidimensional Compressed Sensing MRI Using TensorDecomposition-Based Sparsifying Transform*, School of Information Technology and Electrical Engineering, the University of Queensland, St Lucia, Queensland, Australia (2014), 3.

Sažetak

Rad opisuje algoritam za prepoznavanje lica temeljen na tenzorima. Višedimenzionalna algebra omogućila nam je da uzmemo u obzir raznolike uvjete u kojima osoba može biti uslikana. Razlikovali smo izraze lica, osvjetljenja i kuteve gledišta. Prepoznavanje nepoznatog izraza lica obavljeno je vrlo uspješno. Suočavanjem algoritma s drugačijim osvjetljenjem ili kutem gledišta točnost klasifikacije značajno se smanjila.

Redukcijom dimenzije osvjetljenja ublažili smo izražene svjetlosne kontraste. Posjedujemo li bazu slika s velikim brojem različitih osvjetljenja ovaj postupak možemo provesti prije same klasifikacije. Time izbjegavamo nebitne razlike među slikama iste osobe. Isto vrijedi i za ostale dimenzije u kojima je aproksimacija nižim rangom logična.

Summary

This thesis describes tensor based algorithm for face recognition. Multilinear algebra enabled us to take into consideration various scene conditions usually found in images. We distinguish face expressions, illumination and view points. Recognition of unfamiliar face expression was very successful. Facing with different illumination or viewpoint resulted in a significant accuracy drop.

Reducing the dimension related to illumination we have managed to smooth noticeable light contrasts. In case we possess image base with many different illumination conditions we can conduct this procedure prior to the classification process. As a result we avoid essential differences between images of the same person. Same applies to another dimension where dimensionality reduction makes sense.

Životopis

Rođena sam 14. kolovoza 1994. godine u Zagrebu. Pohađala sam Osnovnu školu Matije Gupca nakon čega upisujem V. gimnaziju. 2012. godine sudjelovala sam na Međunarodnoj olimpijadi iz astronomije i astrofizike. Završila sam preddiplomski sveučilišni studij Matematika na Prirodoslovno-matematičkom fakultetu u Zagrebu. Svoje obrazovanje nastavljam na diplomskom studiju Financijska i poslovna matematika.