

PREPOZNAVANJE LICA korištenjem t - SVD metode

Bratulić Petar
Stanišić Matea

Ožujak 2020.

Sadržaj

1	Sažetak	1
2	Uvod	2
3	Notacija	3
4	Opis PCA i TensorFaces	4
4.1	Tradicionalni PCA - Eigenfaces	4
4.2	TensorFaces	5
5	Tensor framework	6
5.1	Tenzor - tenzor množenje	6
5.2	T-SVD	9
6	Novi pristup baziran na T-SVD-u	10
7	Alternativni pristup pomoću tenzorske QR faktorizacije	16
8	Testiranje	18
8.1	Podaci	18
8.2	Rezultati	18
8.2.1	TSVD	18
8.2.2	TQR	19
8.2.3	HOSVD	20
8.2.4	Zaključak	21
8.3	Fotografije prije i poslije primjena TSVD-a	22

1 Sažetak

Tenzor je višedimenzionalni vektori. Tenzori prvog reda se mogu promatrati kao vektori, a drugog reda kao matrice. Tenzori višeg reda, sa svojstvom da mogu čuvati više informacija, danas se sve češće javljaju u obradi slika i signala, *data miningu*, biomedicinskom inženjerstvu i slično. Uz pomoć rezultata radova Kilmera i Martina, faktORIZACIJA bazirane na matricama mogu se proširiti do tenzora trećeg reda pomoću novih množenja tenzora i koncepata. Metoda opisana u radu ima prednost nad popularnim algoritmima za prepoznavanje lica TensorFaces, baziranim na dekompoziciji tenzora zvanoj *higher-order* SVD (HOSVD), jer nije potrebno rješavanje najmanjih kvadrata da bi dobili njene koeficijente.

U ovom radu, opisano je kako iskoristiti dekompoziciju tenzora na problem prepoznavanja lica. Na kraju, prikazati ćemo uspješnost takvih algoritama.

2 Uvod

Tenzor je višedimenzionalno polje. Tenzor prvog reda je vektor, tenzor drugog reda je matrica, a tenzor reda tri ili više se naziva tenzor višeg reda. U mnogo slučajeva, primjerenije je pohraniti podatke u višedimenzionalnim poljima višeg reda, nego u ekvivalentnoj matričnoj formi. Promotrimo primjer pohrane slika različitih osoba uzetih pod različitim kutevima, osvjetljenjem i izrazima lica. Korištenjem tenzora višeg reda omogućuje pohranu takvih slika u obliku tenzora petog reda gdje modovi predstavljaju osobe, osvjetljenje, izraze lica, kuteve gledanja i piksele. Općenito, tenzori višeg reda pojavljuju se u primjeni mnogih znanstvenih disciplina poput psihometrije ili obradi signala i slika. Rezultat česte primjene su razni algoritmi i dekompozicije koji su razvijeni u tu svrhu poput Higher-Order SVD-a (HOSVD), TUCKER-a ili CANDECOMP/PARAFAC-a (CP).

Jednom kada se tenzori višeg reda koriste kao pohrana za višedimenzionalne podatke, primjena na podacima će diktirati koju vrste manipulacije i preuređivanja na njima je potrebno napraviti. U mnogim primjenama, najbitniji korak je višestruka kompresija podataka koja omogućuje da komprimirana reprezentacija tenzora očuva svojstva početnog tenzora. Zbog toga postoje različite vrste dekompozicije tenzora.

Za razliku od matričnog slučaja, gdje se kompresija podataka postiže tzv. *rank-revealing* faktorizacijama poput SVD-a, za tenzore reda većeg od 2, čak i pojam ranga je komplicirano definirati. CP i Tucker dekompozicija su dvije najčešće korištene dekompozicije tenzora.

Neka je \mathcal{A} tenzor dimenzija $I \times J \times K$ i **CP dekompozicija** napisana kao suma vanjskih produkata vektora

$$\mathcal{A} = \sum_{l=1}^R u^{(l)} \circ v^{(l)} \circ w^{(l)} \implies (\mathcal{A})_{i,j,k} = \sum_{l=1}^R u_i^{(l)} v_j^{(l)} w_k^{(l)},$$

gdje je $u^{(l)} \in \mathbb{R}^I, v^{(l)} \in \mathbb{R}^J, w^{(l)} \in \mathbb{R}^K$, za $l = 1, 2, \dots, R$.

Tucker dekompozicija je oblika

$$\mathcal{A} = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} = \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} \sigma_{ijk} (u^{(i)} \circ v^{(j)} \circ w^{(k)})$$

gdje je $R_1 \leq I, R_2 \leq J, R_3 \leq K$ i za $i, j, k, u^{(i)} \in \mathbb{R}^I, v^{(j)} \in \mathbb{R}^J, w^{(k)} \in \mathbb{R}^K, \mathcal{G} = \sigma_{ijk} \in \mathbb{R}^{R_1 \times R_2 \times R_3}, \mathbf{U} = [u^1, u^2, \dots, u^{R_1}], \mathbf{V} = [v^1, v^2, \dots, v^{R_2}], \mathbf{W} = [w^1, w^2, \dots, w^{R_3}]$.

Obje dekompozicije se mogu promatrati kao generalizacija matričnih dekompozicija singularnih vrijednosti i analize glavnih komponenata. No, za razliku od matričnog SVD-a čiji rang je broj singularnih vrijednosti različitih od nula, minimalna vrijednost od R u CP metodi se ne može uvijek izračunati.

U primjenama često se fiksira R i traži optimalna Frobeniusova norma između \mathcal{A} i sume vanjskih produkata ranga jedan, sa i bez dodatnih ograničenja na komponente vektora.

Prethodno opisani pristup može biti zadovoljavajući u nekim primjenama. U ovom je radu dan potpuno drukčiji pristup temeljen na *frameworku* objašnjenim u nastavku. Također, definirat ćemo *energy-revealing* faktORIZACIJE tenzora trećeg reda koje primijenjujemo na problem prepoznavanja lica.

Opisani postupak nije prvi takav pristup u rješavanju problema prepoznavanja lica. Prvi radovi koji su koristili tenzore te pripadne faktORIZACIJE pojavio se početkom 2000-tih godina i rezultat tih radova je algoritam TensorFaces koji koristi HOSVD.

3 Notacija

U ovom odjeljku opisuju se osnovni koncepti tenzora i uvodi se notacija koja se koristi u radu.

Skalare označavamo malim tiskanim slovom, npr. a . Vektori su označeni podebljanim tiskanim slovom, npr. \mathbf{a} . Matrice su označene podebljanim velikim tiskanim slovom, npr. \mathbf{A} . Tenzori višeg reda označeni su Eulerovim *script* slovima, kao npr. \mathcal{A} . Također, koristi se **Matlab** konvencija u nekim notacijama kao npr. $A(:, i)$ označava sve redove te i -ti stupac matrice A .

Red tenzora je dimenzionalnost vektora kojeg predstavlja koji se još nazivaju i modovi. Tenzor \mathcal{A} reda p je p -dimenzionalni vektor. Možemo ga zapisati kao

$$\mathcal{A} = (a_{i_1 i_2 \dots i_p}) \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_p}.$$

U ovom radu interes proučavanja su dekompozicije tenzora reda tri pa se fokusiramo na takve tenzore.

Nit je 1-dimenzionalni dio tenzora definiran fiksiranjem svih indeksa osim jednog. Tenzori trećeg reda imaju niti stupaca, redova i cijevi. Odsječak tenzora trećeg reda je dvodimenzionalan dio tenzora definiran fiksiranjem svih indeksa osim dva. $A(:, :, k)$, $A(:, k, :)$, $A(k, :, :)$ redom označavaju frontalni, lateralni i horizontalni odsječak. $A(:, :, k)$ se češće označava kao $A^{(k)}$.

Definiramo još sljedeće: Neka je $A \in \mathbb{R}^{l \times m \times n}$, tada je

$$\text{bcirc}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n)} & \mathbf{A}^{(n-1)} & \dots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \mathbf{A}^{(n)} & \dots & \mathbf{A}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(n)} & \mathbf{A}^{(n-1)} & \mathbf{A}^{(n-2)} & \dots & \mathbf{A}^{(1)} \end{bmatrix}$$

je blok cirkularna matrica veličine $ln \times mn$.

Naredba `unfold(A)` uzima tenzor veličine $l \times m \times n$ i vraća blok matricu veličine $ln \times m$:

$$\text{unfold}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \vdots \\ \mathbf{A}^{(n)} \end{bmatrix}$$

Operacija `fold` je inverzna operaciji `unfold`

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

Frobeniusova norma tenzora $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$ je

$$\|\mathcal{A}\|_F = \sqrt{\sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n a_{ijk}^2},$$

gdje je a_{ijk} i -ti, j -ti, k -ti element od \mathcal{A} .

4 Opis PCA i TensorFaces

U nastavku je kratki opis metoda PCA i TensorFaces s kojima metode predložene u radu imaju sličnosti.

4.1 Tradicionalni PCA - Eigenfaces

PCA je jedna od klasičnih tehnika u prepoznavanju slika i kompresiji podataka. Koristi se kako bi transformirala originalni, velik broj varijabli koje su moguće korelirane u manji broj nekoreliranih varijabli znane još kao glavne komponente. Prvih nekoliko glavnih komponenti objašnjava većinu varijance u podacima, dok ostale imaju mali doprinos. Dakle, možemo podatke prikazati ekonomičnije koristeći prvih nekoliko glavnih komponenti.

U nastavku je algoritam te kratka objašnjenja nakon njega.

Neka je $\{\mathbf{X}_1, \dots, \mathbf{X}_m\}$ kolekcija $l \times n$ slika. Matricu \mathbf{A} definiramo tako da je j -ti stupac od \mathbf{A} $\text{vec}(\mathbf{X}_j)$. Sa M označimo srednju sliku ulaznih slika. Srednjom devijacijskom formom nazivamo oblik $\mathbf{A}(:, j) \leftarrow \mathbf{A}(:, j) - \mathbf{M}$.

Kovarijacijska matrica od \mathbf{A} je dana s $\mathbf{C} = \frac{1}{m-1} \mathbf{A} \mathbf{A}^T$. Očito je ona simetrična pozitivno semidefinitna te se može ortogonalno dijagonalizirati u obliku $\mathbf{C} = \mathbf{U} \mathbf{D} \mathbf{U}^T$ gdje je \mathbf{D} dijagonalna matrica koja sadrži svojstvene vrijednosti od \mathbf{C} , dok su stupci od \mathbf{U} svojstveni vektori od \mathbf{C} .

Glavna ideja PCA-eigenfaces pristupa je da je $\mathbf{A}(:, j) \approx \mathbf{U}_r \mathbf{U}_r^T \mathbf{A}(:, j)$ gdje je $\mathbf{U}_r := \mathbf{U}(:, 1 : r)$ za neki $r \ll m$ jer svojstvene vrijednosti nakon r -te su

Algorithm 1 Tradicionalna matrična PCA metoda

Input: Slike za treniranje: \mathbf{I}_i , $i = 1, 2, \dots, N$; Test slika \mathbf{J} , indeks kraćenja k

Output: Uspredba test slike sa slikom iz skupa za treniranje

for $i = 1$ **to** N **do**
 $\mathbf{L}(:, i) \leftarrow$ vektorizirani \mathbf{I}_i

end for

$\mathbf{M} \leftarrow$ srednja slika

$\mathbf{A} \leftarrow$ srednja devijacijska forma od \mathbf{L}

$\mathbf{U} \leftarrow$ lijevi singularni vektori od \mathbf{A}

$\mathbf{G} \leftarrow \mathbf{U}(:, 1 : k)^T * \mathbf{A}$

$\mathbf{T} \leftarrow (\mathbf{J} - \mathbf{M})$

$\mathbf{t} \leftarrow$ vektorizirani forma od \mathbf{T}

$\mathbf{c} \leftarrow \mathbf{U}(:, 1 : k)^T * \mathbf{t}$

for $j = 1$ **to** N **do**

 Izračunaj $\|\mathbf{c} - \mathbf{G}(:, j)\|_F$

end for

 Slika za treniranje čiji su koeficijenti najbliži test slici je rezultat.

vrlo male relativno s m , označavajući da je većina varijance u podacima dobro obuhvaćena u smjerovima koji odgovaraju najvećim r svojstvenim vrijednostima.

4.2 TensorFaces

TensorFaces je bio prvi algoritam koji je koristio tenzore u rješavanju problema prepoznavanja lica. Podaci su reprezentirani kao tenzor s razičitim modovima za različite faktore. Kao i u PCA, slike su vektorizirane. Ostali modovi se koriste kako bi se reprezentirali ostala svojstva slike. Aproksimacija tenzora se postiže multidimenzionalnom metodom *higher-order* SVD (HOSVD).

Kao ulaz se koristi tenzor reda pet koji označava ljude (P), točke gledanja (V), osvjetljenje (I), izraze lica (E) i piksele. Tada se po formuli opisanoj u algoritmu 2 računa \mathcal{Z} gdje matrice \mathbf{U}_i predstavljaju lijeve matrice u SVD-u matrica \mathbf{L}_i tenzora \mathcal{L} po i -tom modu. \mathcal{B} definira $V \times I \times E$ različitih baza za svaku kombinaciju točki gledanja, osvjetljenja i izraza lica i pripadni podtenzor $\mathcal{B}_{v,i,e}$ veličine $P \times 1 \times 1 \times 1 \times Pix$ za svaki v, i, e . Koristimo spljoštene matrice tih podtenzora kako bi ih projicirali na skup zadan s koeficijentima vektora $c_{v,i,e}$. Tada izračunamo $c_{v,i,e} - \mathbf{U}_p(p, :)^T$ i za nepoznatu sliku odlučimo da pripada osobi p za koju je taj izraz najmanji.

Algorithm 2 TensorFaces metoda

Input: Slike za treniranje: $\mathbf{I}_{p,v,t,e}$, $p = 1, \dots, P$, $v = 1, \dots, V$, $i = 1, \dots, I$, $e = 1, \dots, E$; Test slika \mathbf{J}

Output: Koeficijenti testne slike s nekim tenzorskim bazama i usporedbama

for $p = 1 \dots P$, $v = 1 \dots V$, $i = 1 \dots I$, $e = 1 \dots E$ **do**

$\mathcal{L}(p, v, i, e, :) \leftarrow$ vektorizirani $\mathbf{I}_{p,v,i,e}$

end for

Napravi HOSVD da se dobije jezgri tenzor \mathcal{Z} i ortogonalne faktorske matrice \mathbf{U}_i :

$\mathcal{L} \leftarrow \mathcal{Z} \times_1 \mathbf{U}_p \times_2 \mathbf{U}_v \times_3 \mathbf{U}_i \times_4 \mathbf{U}_e \times_5 \mathbf{U}_{p,i,e}$

$\mathcal{B} \leftarrow \mathcal{Z} \times_2 \mathbf{U}_v \times_3 \mathbf{U}_i \times_4 \mathbf{U}_e \times_5 \mathbf{U}_{p,i,e}$

$\mathcal{T}(1, 1, 1, 1, :) \leftarrow$ vektorizirana slika za testiranje

for $v = 1, \dots, V$, $i = 1, \dots, I$, $e = 1, \dots, E$ **do**

$c_{v,i,e} \leftarrow (\mathcal{B}_{v,i,e}^\dagger)^T \mathcal{T}(1, 1, 1, 1, :)$

for $p = 1, \dots, P$ **do**

Izračunaj $\|c_{v,i,e} - \mathbf{U}_p(p, :)^T\|_F$

end for

end for

Slika za treniranje čiji su koeficijenti najbliži test slici je rezultat.

5 Tensor framework

5.1 Tenzor - tenzor množenje

Definicija 1. Neka je $\mathcal{A} \in \mathbb{R}^{l \times p \times n}$ i $\mathcal{B} \in \mathbb{R}^{p \times m \times n}$, tada definiramo produkt tenzora $\mathcal{A} * \mathcal{B}$ kao $l \times m \times n$ tenzor

$$\mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})) \quad (1)$$

Primjer: Pretpostavimo da su $\mathcal{A} \in \mathbb{R}^{l \times p \times 3}$ i $\mathcal{B} \in \mathbb{R}^{p \times m \times 3}$. Onda je

$$\mathcal{A} * \mathcal{B} = \text{fold} \left(\begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(3)} & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \mathbf{A}^{(3)} \\ \mathbf{A}^{(3)} & \mathbf{A}^{(2)} & \mathbf{A}^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \\ \mathbf{B}^{(3)} \end{bmatrix} \right) \in \mathbb{R}^{l \times m \times 3}.$$

S obzirom na činjenicu da se cirkularne matrice mogu dijagonalizirati normalnom diskretnom matricom Fourierove transformacije (DFT matrica), možemo dijagonalizirati cirkularne blok matrice po blokovima. Pretpostavimo da je $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$ i \mathbf{F}_n je $n \times n$ DFT matrica. Dobivamo $\text{bcirc}(\mathcal{A})$ koja je $n \times n$ blok matrica s $l \times m$ blokova. Neka \mathbf{P}_1 i \mathbf{P}_2 označavaju matrice permutacijskih koraka tako da

$$\mathbf{P}_1 \cdot \text{bcirc}(\mathcal{A}) \cdot \mathbf{P}_2 = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} & \cdots & \mathbf{N}_{1m} \\ \mathbf{N}_{21} & \mathbf{N}_{22} & \cdots & \mathbf{N}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{N}_{l1} & \mathbf{N}_{l2} & \cdots & \mathbf{N}_{lm} \end{bmatrix}$$

je $l \times m$ blok matrica s blokovima N_{ij} veličine $n \times n$ gdje se prvi stupac od N_{ij} sastoji od svih i, j podataka od $\mathbf{A}^{(k)}$, $k = 1, 2, \dots, n$, i

$$\mathbf{P}_2^T \cdot (\mathbf{F}_n^* \otimes \mathbf{I}_m) \cdot \mathbf{P}_2 = \mathbf{I}_m \otimes \mathbf{F}_n^*.$$

Primijetimo da

$$\mathbf{P}_1 \cdot (\mathbf{F}_n \otimes \mathbf{I}_l) \cdot \mathbf{P}_1^T \mathbf{P}_1 \cdot \text{bcirc}(\mathcal{A}) \cdot \mathbf{P}_2 \mathbf{P}_2^T \cdot (\mathbf{F}_n^* \otimes \mathbf{I}_m) \cdot \mathbf{P}_2$$

je jednako

$$(\mathbf{I}_l \otimes \mathbf{F}_n) \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} & \cdots & \mathbf{N}_{1m} \\ \mathbf{N}_{21} & \mathbf{N}_{22} & \cdots & \mathbf{N}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{N}_{l1} & \mathbf{N}_{l2} & \cdots & \mathbf{N}_{lm} \end{bmatrix} (\mathbf{I}_m \otimes \mathbf{F}_n^*) = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \cdots & \mathbf{D}_{1m} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \cdots & \mathbf{D}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{l1} & \mathbf{D}_{l2} & \cdots & \mathbf{D}_{lm} \end{bmatrix}$$

gdje je svaki D_{ij} $n \times n$ dijagonalna matrica. Množeći izraz sprijeda s \mathbf{P}_1^T i otraga s \mathbf{P}_2^T

$$\begin{aligned} (\mathbf{F}_n \otimes \mathbf{I}_l) \text{bcirc}(\mathcal{A}) (\mathbf{F}_n^* \otimes \mathbf{I}_m) &= \mathbb{P}_1^T \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \cdots & \mathbf{D}_{1m} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \cdots & \mathbf{D}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{l1} & \mathbf{D}_{l2} & \cdots & \mathbf{D}_{lm} \end{bmatrix} \mathbb{P}_2^T \\ &= \begin{bmatrix} \hat{\mathbf{A}}^{(1)} & & & \\ & \hat{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathbf{A}}^{(n)} \end{bmatrix}. \end{aligned}$$

Primijenjujući brzu Fourierovu transformaciju duž trećeg moda \mathcal{A} daje nam alternativni način računanju $\hat{\mathbf{A}}^{(i)}$. Neka je $\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$, pa je za svaki i , $\hat{\mathbf{A}}^{(i)} = \hat{\mathcal{A}}(:, :, i)$. Kad tenzori nisu rijetki, možemo izračunati $\mathcal{A} * \mathcal{B}$ računajući

FFT po svakoj niti moda tri od \mathcal{A} i \mathcal{B} da bismo dobili $\hat{\mathcal{A}}$ i $\hat{\mathcal{B}}$, pomnožimo svaki par strana od $\hat{\mathcal{A}}$ i $\hat{\mathcal{B}}$ da bismo dobili frontalne odsječke tensora $\hat{\mathcal{C}}$, onda izračunamo inverzni FFT po niti moda tri od $\hat{\mathcal{C}}$ da bismo dobili željeni rezultat. S obzirom da u najviše primjena vrijedi $\log(n) \leq m, l$ ovo računanje će trebati $O(lpmn + (lp + pm)n \log(n)) \approx O(lpn^2m)$ što je brže od $O(lpn^2m)$ po definiciji 1.

Algorithm 3 Tensor - tensor množenje, koristeći računanje u Fourierovoj domeni

Input: $\mathcal{A} \in \mathbb{R}^{l \times p \times n}, \mathcal{B} \in \mathbb{R}^{p \times m \times n}$
Output: $\mathcal{C} := \mathcal{A} * \mathcal{B}$
 $\hat{\mathcal{A}} \leftarrow \text{fft}(\mathcal{A}, [], 3); \hat{\mathcal{B}} \leftarrow \text{fft}(\mathcal{B}, [], 3)$
for $i = 1$ **to** n **do**
 $\hat{\mathcal{C}}(:, :, i) \leftarrow \hat{\mathcal{A}}(:, :, i) \hat{\mathcal{B}}(:, :, i)$
end for
 $\mathcal{C} \leftarrow \text{ifft}(\hat{\mathcal{C}}, [], 3)$

Može se primijetiti da zbog simetrije, petlja ne treba prolaziti po svim frontalnim odsječcima, već do $\frac{n}{2} + 1$ i onda izračunamo ostalih $n - (\frac{n}{2} + 1)$ odsječaka kao $\hat{\mathcal{C}}(:, :, \frac{n}{2} + j) = \text{conj}(\hat{\mathcal{C}}(:, :, \frac{n}{2} - j - 2))$ za $j = 2, \dots, n - \frac{n}{2}$.

Sljedeće leme i definicije navodimo jer su potrebne u algoritmima tenzorskih faktORIZACIJA.

Lema 2. $\mathcal{A} * (\mathcal{B} * \mathcal{C}) = (\mathcal{A} * \mathcal{B}) * \mathcal{C}$

Definicija 3. Ako je \mathcal{A} tenzor dimenzija $l \times m \times n$, onda je \mathcal{A}^T tenzor dimenzija $m \times l \times n$ dobiven transponiranjem svakog frontalnog odsječaka te mijenjanjem poretka transponiranih frontalnih odsječaka od 2 do n .

Primjer: Neka je $\mathcal{A} \in \mathbb{R}^{l \times m \times 4}$ i njegovi frontalni odsječci su dani $l \times m$ matricama $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}, \mathbf{A}^{(4)}$ pa je

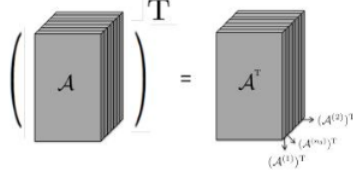
$$\mathcal{A}^T = \text{fold} \left(\left[\begin{array}{c} (\mathbf{A}^{(1)})^T \\ (\mathbf{A}^{(4)})^T \\ (\mathbf{A}^{(3)})^T \\ (\mathbf{A}^{(2)})^T \end{array} \right] \right).$$

Definicija 4. Kažemo da je tenzor f-dijagonalan ako je svaki frontalni odsječak dijagonalna matrica.

Definicija 5. Tenzor dimenzija $l \times l \times n$ $\mathcal{I}_{l,n}$ je identiteta čiji je prvi frontalni odsječak $l \times l$ matrica identiteta, a svi ostali frontalni odsječci su nulmatrice.

Definicija 6. Tenzor realnih vrijednosti \mathcal{Q} dimenzija $l \times l \times n$ je ortogonalan ako

$$\mathcal{Q}^T * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^T = \mathcal{I}. \quad (2)$$



Slika 1: Transponiranje

To znači da je skup svih lateralnih odsječaka ortogonalan, tj. vrijedi

$$\mathcal{Q}(:, i, :)^T * \mathcal{Q}(:, j, :) = \begin{cases} \mathbf{e}_1 & i = j \\ \mathbf{0} & i \neq j \end{cases} \quad (3)$$

gdje je \mathbf{e}_1 cijevni tenzor sa frontalnim odsječkom jedinica, a ostalim nulama.

5.2 T-SVD

U dekompozicija tenzora koristimo množenje tenzor-tenzor. Takvu dekompoziciju nazivamo T-SVD.

Teorem 7. *Neka je \mathcal{A} tenzor realnih vrijednosti dimenzija $l \times m \times n$. Tada se \mathcal{A} može faktorizirati u obliku*

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$$

pri čemu su \mathcal{U} ortogonalni tenzor dimenzija $l \times l \times n$, \mathcal{V} ortogonalni tenzor dimenzija $m \times m \times n$ te \mathcal{S} f-dijagonalni tenzor dimenzija $l \times m \times n$. ■

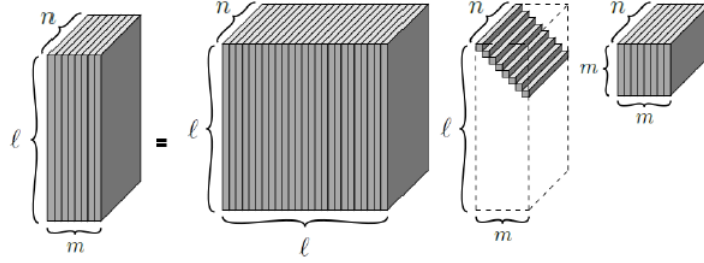
Dokaz. Prvo, transformirajmo $\text{bcirc}(\mathcal{A})$ u Fourierovu domenu:

$$(\mathbf{F}_n \otimes \mathbf{I}_l) \text{bcirc}(\mathcal{A}) (\mathbf{F}_n^* \otimes \mathbf{I}_m) = \begin{bmatrix} \hat{\mathbf{A}}^{(1)} & & & \\ & \hat{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathbf{A}}^{(n)} \end{bmatrix}$$

Izračunajmo matrični SVD za svaki $\hat{\mathcal{A}}^{(i)}$ kao $\hat{\mathcal{A}}^{(i)} = \hat{\mathcal{U}}^{(i)} \hat{\mathcal{S}}^{(i)} (\hat{\mathcal{V}}^{(i)})^T$ pa imamo

$$\begin{bmatrix} \hat{\mathbf{A}}^{(1)} & & \\ & \ddots & \\ & & \hat{\mathbf{A}}^{(n)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{U}}^{(1)} & & \\ & \ddots & \\ & & \hat{\mathbf{U}}^{(n)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{S}}^{(1)} & & \\ & \ddots & \\ & & \hat{\mathbf{S}}^{(n)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}^{(1)} & & \\ & \ddots & \\ & & \hat{\mathbf{V}}^{(n)} \end{bmatrix}^T \quad (4)$$

Sada nakon što izaberemo odgovarajuće dimenzije za \mathbf{I} , primijenimo $\mathbf{F}_n^* \otimes \mathbf{I}$ lijeva i $\mathbf{F}_n \otimes \mathbf{I}$ s desna svake blok dijagonalne matrice da dobijemo tri cirkularne matrice čiji su prvi blok stupci $\text{unfold}(\mathcal{U})$, $\text{unfold}(\mathcal{S})$, $\text{unfold}(\mathcal{V}^T)$. Primijenjujući fold dobivamo $\mathcal{U}, \mathcal{S}, \mathcal{V}^T$.



Slika 2: T-SVD

Ortogonalnost matrica, tj. $\mathcal{U}^T * \mathcal{U} = \mathcal{U} * \mathcal{U}^T = \mathcal{I}$ i $\mathcal{V}^T * \mathcal{V} = \mathcal{V} * \mathcal{V}^T = \mathcal{I}$ vrijedi iz njihove definicije te definicije operatora $*$. \square

Jedno lijepo svojstvo dekompozicije T-SVD je da daje način za pronalazak optimalne aproksimacije tenzora kao sume $k < \min(n_1, n_2)$ matričnih vanjskih produkata na analogan način matričnom SVD-u.

Teorem 8. Neka je T-SVD od $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$ dana s $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ i za svaki $k < \min(l, m)$ definiramo

$$\mathcal{A}_k = \sum_{i=1}^k \mathcal{U}(:, i, :) * \mathcal{S}(i, i, :) * \mathcal{V}(:, i, :)^T \quad (5)$$

Tada je $\mathcal{A}_k = \arg \min \|\mathcal{A} - \tilde{\mathcal{A}}\|_F$ gdje je $M = \{\mathcal{C} = \mathcal{X} * \mathcal{Y} \mid \mathcal{X} \in \mathbb{R}^{l \times k \times n}, \mathcal{Y} \in \mathbb{R}^{k \times m \times n}\}$. \blacksquare

6 Novi pristup baziran na T-SVD-u

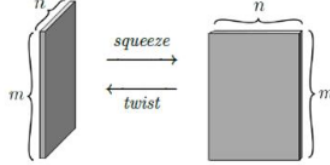
Predložena metoda razlikuje se od PCA za matrice na način da se podaci čuvaju u tenzorima trećeg reda te se koriste tenzor-teznor faktorizacije bazirane na operatoru $*$. Također, razlikuje se od TensorFaces algoritma jer slike nisu pikselizirane, nema dodatnih dimenzija koje predstavljaju osvjetljenje, poze i slično te koristi T-SVD umjesto HOSVD-a. U ovom odjeljku opisan je pristup temeljen na T-SVD, dok je u sljedećem alternativna metoda koja koristi QR faktorizaciju tenzora.

U postupku će nam koristiti sljedeća notacija: **squeeze** operacija koja uzima tenzor \mathcal{X} iz $\mathbb{R}^{l \times 1 \times n}$ i daje matricu

$$\mathbf{X} := \text{squeeze}(\mathcal{A}) \Rightarrow \mathbf{X}(i, j) := \mathcal{X}(i, 1, j)$$

dok je **twist** inverzna operacija operaciji **squeeze**

$$\text{twist}(\text{squeeze}(\mathcal{X})) = \mathcal{X}.$$



Slika 3: Prikaz operacija **squeeze** i **twist**

Neka \mathbf{M} predstavlja slika koja je prosjek svih slika. Računamo \mathcal{A} kao $\mathcal{A}(:, j, :) = \text{twist}(\mathbf{X}_j - \mathbf{M}), j = 1, \dots, m$.

Analogno s kovarijacijskom matricom u tradicionalnom PCA-u, $\mathcal{K} := \frac{1}{m-1} \mathcal{A} * \mathcal{A}^T$ je simetrična pozitivno semidefinitna i ortogonalno dijagonalizabilna, tj. $\mathcal{K} = \mathcal{U} * \mathcal{D} * \mathcal{U}^T$ gdje je \mathcal{D} f-dijagonalni tenzor koji sadrži svojstvene n-torke od \mathcal{K} na dijagonali. „Stupci” $\mathcal{U}(:, j, :)$ se mogu promatrati kao svojstvene matrice od \mathcal{K} . Kovarijacijski tenzor nam omogućuje da dobijemo maksimalne smjerove varijance po stupcima zato jer svaki odsječak $\mathcal{K}(:, :, j)$ mjeri kovarijancu svih j -tih stupaca slika.

Zbog sličnih razloga kao i u matricnoj metodi PCA, ne računamo svojstvenu dekompoziciju od \mathcal{K} , nego koristimo tenzor verziju SVD-a $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$. Slijedi iz $\mathcal{K} = \frac{1}{m-1} \mathcal{A} * \mathcal{A}^T = \frac{1}{m-1} \mathcal{U} * \mathcal{S} * \mathcal{S}^T * \mathcal{U}^T$ da nenula svojstvene n-torke od \mathcal{K} su kvadrati (po t-produktu) odgovarajućih singularnih n-torka od \mathcal{A} te \mathcal{U} sadrži glavne komponente s obzirom na \mathcal{K} .

S obzirom da je \mathcal{U} ortogonalan, $\mathcal{U}(:, 1 : k, :)$ je ortogonalan skup lateralnih odsječaka $\mathcal{U}(:, j, :) \in \mathbb{R}^{l \times 1 \times n}$ i $\mathcal{U}(:, 1 : k, :) * \mathcal{U}(:, 1 : k, :)^T$ definira ortogonalni projektor koji projektira centriranu sliku u prostor manjih dimenzija. Ako koristimo nekoliko lijevih singularnih lateralnih odsječaka od \mathcal{U} kao novu bazu, tada

$$\mathcal{A}(:, j, :) \approx \mathcal{U}(:, 1 : k, :) * \mathcal{U}(:, 1 : k, :)^T * \mathcal{A}(:, j, :) = \sum_{t=1}^k \mathcal{U}(:, t, :) * \mathcal{C}(t, j, :) \quad (6)$$

gdje je $\mathcal{C}(:, j, :) = \mathcal{U}(:, 1 : k, :)^T * \mathcal{A}(:, j, :)$ s $\mathcal{C}(t, j, :) = \mathcal{U}(:, t, :)^T * \mathcal{A}(:, j, :)$, $t = 1, 2, \dots, k$. Zbog toga se svaka centrirana slika $\mathcal{A}(:, j, :)$ može promatrati kao tenzor-linearna kombinacija ortogonalnih baznih elemenata $\mathcal{U}(:, t, :)$ za $t = 1, 2, \dots, k$ s koeficijentima $\mathcal{C}(t, j, :)$.

Problem prepoznavanja reduciramo na uspoređivanje koeficijenata između slika za testiranje i treniranje. Primijetite da moramo spremati $k, l \times n$ baznih elemenata i $k \times (\text{broj slika za treniranje}) \times n$ tenzor \mathcal{C} koeficijenata za slike za treniranje.

Algorithm 4 T-SVD metoda 1

Input: Slike za treniranje: \mathbf{I}_i , $i = 1, 2, \dots, N$; Test slika \mathbf{J} , indeks kraćenja k
Output: Odgovarajuća slika za test sliku iz reprezentacije skupa za treniranje

```

for  $i = 1$  to  $N$  do
     $\mathcal{L}(:, i, :) \leftarrow \mathbf{I}_i$ 
end for
 $\mathcal{M} \leftarrow$  srednja slika
 $\mathcal{A} \leftarrow$  srednja devijacijska forma od  $\mathcal{L}$ 
 $\mathcal{U} \leftarrow$  lijevi singularni vektori tenzora  $\mathcal{A}$ 
 $\mathcal{C} \leftarrow \mathcal{U}(:, 1 : k, :)^T * \mathcal{A}$ 
 $\mathcal{T}(:, 1, :) \leftarrow \text{twist}(\mathbf{J} - \mathcal{M})$ 
 $\mathcal{B} \leftarrow \mathcal{U}(:, 1 : k, :)^T * \mathcal{T}$ 
for  $j = 1$  to  $N$  do
    Izračunaj  $\|\mathcal{B} - \mathcal{C}(:, j, :)\|_F$ 
end for
Slika za treniranje čiji su koeficijenti najbliži test slici je rezultat.

```

Lema 9. *Kada su slike jednodimenzionalni signali ($n = 1$), SVD metoda za tenzore se reducira do tradicionalne PCA metode.*

Dokaz. Tvrdnja slijedi iz činjenice da za $n = 1$, T-SVD se reducira na matrični SVD. \square

U analizi iskazanog pristupa, korisno je napisati 6 u matričnoj notaciji. Neka je $\mathbf{Y}_j := \text{squeeze}(\mathbf{A}(:, j, :))$. Primijetimo da iz algoritma 2, \mathbf{Y}_j je samo j -ta slika za treniranje. Neka je $\text{circ}(v)$ cirkularna matrica čiji je prvi stupac v . Iz 6, definicije $*$ i malo manipulacije, može se pokazati da je

$$\mathbf{Y}_j \approx \sum_{t=1}^k \mathbf{W}_t \text{circ}(\mathbf{c}_{t,j}) \quad (7)$$

gdje je $\mathbf{W}_t := \text{squeeze}(\mathcal{U}(:, t, :))$ je $l \times n$ matrica i $\mathcal{C}_{i,j}^T := \text{squeeze}(\mathbf{C}(t, j, :)^T)$ je vektor redak duljine n . Dobivamo sljedeći rezultat.

Teorem 10. *Neka \mathbf{F} predstavlja $n \times n$ DFT matricu. Neka su $\widehat{\mathbf{Y}}_j := \mathbf{Y}_j \mathbf{F}$, $\widehat{\mathbf{W}}_j := \mathbf{W}_j \mathbf{F}$ i $\widehat{\mathbf{c}}_{t,j}$ Fourierovi koeficijenti vektora stupca $\mathbf{c}_{t,j}$ duljine n ,*

$$\widehat{\mathbf{Y}}_j \approx \left[\sum_{t=1}^k \widehat{\mathbf{c}}_{t,j}^{(1)} \widehat{\mathbf{U}}^{(1)}(:, t), \sum_{t=1}^k \widehat{\mathbf{c}}_{t,j}^{(2)} \widehat{\mathbf{U}}^{(2)}(:, t), \dots, \sum_{t=1}^k \widehat{\mathbf{c}}_{t,j}^{(n)} \widehat{\mathbf{U}}^{(n)}(:, t) \right] \quad (8)$$

gdje $\widehat{\mathbf{c}}_{t,j}^{(i)}$ predstavlja i -tu komponentu tog vektora i $\widehat{\mathbf{W}}_t(:, i) = \widehat{\mathbf{U}}^{(i)}(:, t)$, gdje je $\widehat{\mathbf{U}}^{(i)}$ unitarna matrica iz dokaza teorema T-SVD-a. Slijedi da je algoritam 2 ekvivalentan simultanoj PCA izvršenoj na svakom stupcu transformirane slike u Fourireovom prostoru, koristeći iste indekse kraćenja k na svakom stupcu. \blacksquare

Dokaz. Zbog $\mathbf{Y}_j \approx \sum_{t=1}^k \mathbf{W}_t \text{circ}(\mathbf{c}_{t,j})$, imamo

$$\mathbf{Y}_j \mathbf{F} \approx \sum_{t=1}^k \mathbf{W}_t \text{circ}(\mathbf{c}_{t,j}) \mathbf{F} = \sum_{t=1}^k \mathbf{W}_t \mathbf{F} \mathbf{F}^H \text{circ}(\mathbf{c}_{t,j}) = \sum_{t=1}^k \widehat{\mathbf{W}}_t \text{diag}(\widehat{\mathbf{c}}_{t,j})$$

tj.

$$\widehat{\mathbf{Y}}_j \approx \sum_{t=1}^k \left[\widehat{\mathbf{c}}_{t,j}^{(1)} \widehat{\mathbf{W}}_t(:, 1), \widehat{\mathbf{c}}_{t,j}^{(2)} \widehat{\mathbf{W}}_t(:, 2), \dots, \widehat{\mathbf{c}}_{t,j}^{(n)} \widehat{\mathbf{W}}_t(:, n) \right] \quad (9)$$

$$= \left[\sum_{t=1}^k \widehat{\mathbf{c}}_{t,j}^{(1)} \widehat{\mathbf{U}}^{(1)}(:, t), \sum_{t=1}^k \widehat{\mathbf{c}}_{t,j}^{(2)} \widehat{\mathbf{U}}^{(2)}(:, t), \dots, \sum_{t=1}^k \widehat{\mathbf{c}}_{t,j}^{(n)} \widehat{\mathbf{U}}^{(n)}(:, t) \right] \quad (10)$$

Ostatak tvrdnji slijedi direktno. \square

Glavna posljedica ovog teorema je ta da ilustrira potencijal za daljnju kompresiju informaciju u bazi podataka, s tim da radimo na Fourierovom prostoru. Da bismo to vidjeli, prvo primijetimo da nema gubitka u procesu prepoznavanja ako računamo u Fourierovoj domeni (zbog jednostavnosti pretpostavimo da je n paran). Također, usporedba proširenih koeficijenata testne slike \mathbf{J} s *training* koeficijenatima tenzora se može napraviti u Fourierovoj domeni. Primijetimo da se produkt $\mathcal{Z} := \mathcal{U}(:, 1 : k, :)^T * \text{twist}(\mathbf{J} - \mathbf{M})$ može izračunati koristeći algoritam 3, gdje ionako akumuliramo objekt $\widehat{\mathcal{Z}}$ dimenzija $k \times 1 \times n$ u Fourierovoj domeni. Kao i kod algoritma 3 zbog simetrije, možemo izračunati samo pola matrično-vektorskih produkata (samo prvih $\frac{n}{2} + 1$ frontalnih odsječaka treba biti spremljeno). Ako želimo usporediti $\widehat{\mathcal{Z}}$ s lateralnim odsječcima od \mathcal{C} , možemo ekvivalentno usporediti $\widehat{\mathcal{Z}}(:, 1, 1 : \frac{n}{2} + 1)$ s $\widehat{\mathcal{C}}(:, j, 1 : \frac{n}{2} + 1)$ za $j = 1, \dots$, broj slika za treniranje.

Spremanje prvih $\frac{n}{2} + 1$ frontalnih odrezaka od $\widehat{\mathcal{C}}$ treba istu količinu prostora kao i za spremanje \mathbf{U} i \mathbf{C} jer moramo spremati brojeve dvostruke preciznosti. No, jer teorem kaže da radimo nezavisni PCA na svakom od stupaca transformiranih slika u Fourierovom prostoru, više smisla ima varirati k po stupcima. Specijalno, izmijenimo 9 s

$$\widehat{\mathbf{Y}}_j \approx \left[\sum_{t=1}^{k_1} \widehat{\mathbf{c}}_{t,j}^{(1)} \widehat{\mathbf{U}}^{(1)}(:, t), \sum_{t=1}^{k_2} \widehat{\mathbf{c}}_{t,j}^{(2)} \widehat{\mathbf{U}}^{(2)}(:, t), \dots, \sum_{t=1}^{k_n} \widehat{\mathbf{c}}_{t,j}^{(n)} \widehat{\mathbf{U}}^{(n)}(:, t) \right], \quad (11)$$

gdje da bi sačuvali konjugatnu simetriju, indeks kraćenja za stupce j , $2 \leq j \leq \frac{n}{2}$ je isti kao i za stupce $n - j + 2$.

Zbog konjugatne simetrije i potrebe za izračunavanjem samo prve polovice koeficijenata slike, ne trebamo eksplicitno računati i čuvati posljednjih $n - \frac{n}{2} - 1$ stupaca.

Iz prethodno navedenog navodimo što je potrebno spremati. Za $l \times n$ sliku iz skupa za treniranje i $\frac{n}{2} + 1$ indeksa kraćenja $k_1, \dots, k_{\frac{n}{2}+1}$, pri čemu je

$k_1 \geq k_i$, trebamo spremiti

$$K = k_1 + 2k_2 + \dots + 2k_{\frac{n}{2}} + k_{\frac{n}{2}+1} \quad (12)$$

realnih brojeva dvostruke preciznosti koji predstavljaju koeficijente koji nisu skraćeni u Fourierovom prostoru. Za svaki $\hat{\mathbf{U}}^{(i)}, i = 1, \dots, \frac{n}{2} + 1$, trebamo spremiti k_i stupaca. To zahtjeva spremanje lK realnih brojeva dvostruke preciznosti. Znači za $m =$ broj slika za treniranje, moramo spremiti ukupno $lK + mK$ brojeva dvostruke preciznosti.

Preostaje razmotriti kako identificirati parametre kraćenja $k_i, i = 1, \dots, \frac{n}{2} + 1$. Podsjetimo se da je $\hat{\mathbf{A}}$ dimenzija $l \times m \times n$. Iz teorema 7 imamo da $\hat{\mathbf{U}}^{(i)}$ sadrži lijeve singularne vrijednosti i -tog frontalnog odsječka od $\hat{\mathbf{A}}$ i $\hat{\mathbf{S}}^{(i)}$ sadrži odgovarajuće singularne vrijednosti $\hat{\mathbf{s}}_j^{(i)}, j = 1, \dots, \min\{l, m\}$ za i -ti frontalni odsječak. Dalje, primijetimo da

$$\|\hat{\mathbf{A}}(:, :, 1 : \frac{n}{2} + 1)\|_F^2 = \sum_{i=1}^{\frac{n}{2}+1} \|\hat{\mathbf{A}}(:, :, i)\|_F^2 = \sum_{i=1}^{\frac{n}{2}+1} \sum_{j=1}^{\min(l, m)} (\hat{\mathbf{s}}_j^{(i)})^2. \quad (13)$$

Također, da bi optimalno odredili indekse kraćenja, promotrimo mjeru relativne energije

$$\frac{\sum_{i=1}^{\frac{n}{2}+1} \sum_{j=1}^{k_i} (\hat{\mathbf{s}}_j^{(i)})^2}{\|\hat{\mathbf{A}}(:, :, 1 : \frac{n}{2} + 1)\|_F^2}. \quad (14)$$

Jasno je da trebamo sadržati najveće svojstvene vrijednosti da održimo relativnu energiju blizu 1. Nadalje, odlučimo željenu relativnu energijsku vrijednost (npr. 0.9), posložimo $\hat{\mathbf{s}}_j^{(i)}$ u vektor \mathbf{q} od najvećeg do najmanjeg, pronađemo indeks t koji je najmanji indeks za koji je $\sum_{i=1}^{\frac{n}{2}+1} \sum_{j=1}^{k_i} \{(\hat{\mathbf{s}}_j^{(i)})^2 : (\hat{\mathbf{s}}_j^{(i)})^2 > \mathbf{q}(t)^2\} > \|\hat{\mathbf{A}}(:, :, 1 : \frac{n}{2} + 1)\|_F^2 \times .9$

Podsjetimo se da $\hat{\mathbf{s}}_1^{(i)} \geq \hat{\mathbf{s}}_2^{(i)} \geq \dots \hat{\mathbf{s}}_{\min\{l, m\}}^{(i)}$. Dakle, k_i se definiraju tako da je $\hat{\mathbf{s}}_{k_i}^{(i)} > \mathbf{q}(t)$, ali $\hat{\mathbf{s}}_{k_i+1}^{(i)} \leq \mathbf{q}(t)$. Metoda bazirana na ovoj strategiji dana je u algoritmu 5.

Kao završna napomena, iako se u algoritmu 5 zalažemo za rad u Fourierovoj domeni, možemo napraviti usporedbu s onim što se događa u prostornoj domeni. Definiramo f-dijagonalni tenzor \mathcal{D} tako da Fourierovi koeficijenti duž svakog dijagonalne niti moda tri zadovoljavaju, za svaki $i = 1, \dots, \frac{n}{2} + 1$

$$\hat{\mathbf{d}}_j^{(i)} = \begin{cases} 1 & \text{ako je } \mathbf{s}_j^{(i)} \text{ sačuvan} \\ 0 & \text{inače} \end{cases}$$

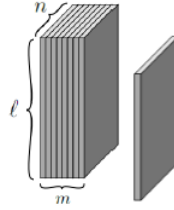
i postavimo $\hat{\mathbf{d}}_j^{(n-i+2)} = \hat{\mathbf{d}}_j^{(i)}$ da bi sačuvali konjugatnu simetriju. Tada, u prostornoj domeni, komprimirani skup za testiranje je reprezentiran koristeći ortogonalnu projekciju od \mathcal{A} na određeni prostor, tj. prostornu domenu, reprezentacija može biti napisana kao $\mathcal{U}(:, 1 : k, :) * \mathcal{D} * \mathcal{U}(:, 1 : k, :)^T * \mathcal{A}$ gdje je tenzor u zgradama ortogonalni projektor.

Algorithm 5 T-SVD metoda 2

Input: Slike za treniranje: \mathbf{I}_i , $i = 1, 2, \dots, N$; Test slika \mathbf{J} , indeks kraćenja k
Output: Usporedba test slike s reprezentacijom skupa za treniranje u Fourierovoj domeni
for $i = 1$ **to** N **do**
 $\mathcal{L}(:, :, i) \leftarrow \mathbf{I}_i$
end for
 $\mathcal{M} \leftarrow$ srednja slika
 $\mathcal{A} \leftarrow$ srednja devijacijska forma od \mathcal{L}
Frontalni odsjeci SVD-a od $\hat{\mathbf{A}}^{(i)} \leftarrow \hat{\mathcal{A}}(:, :, i)$, $i = 1, \dots, \frac{n}{2} + 1$
Drop strategija; sadrži jedino $\hat{\mathbf{U}}^{(i)}(:, 1 : k_i)$
 $\mathcal{T}(:, 1, :) \leftarrow \text{twist}(\mathbf{J} - \mathcal{M})$
for $i = 1 : \frac{n}{2} + 1$, $t = 1 : k_t$, $j = 1 : N$ **do**
 $\hat{\mathbf{c}}_{i,j}^{(i)} \leftarrow (\hat{\mathbf{U}}^{(i)}(:, t))^H \hat{\mathbf{A}}^{(i)}(:, j)$
end for
for $i = 1 : \frac{n}{2} + 1$, $t = 1 : k_t$ **do**
 $\hat{\mathbf{b}}_t^{(i)} \leftarrow (\hat{\mathbf{U}}^{(i)}(:, t))^H \hat{\mathbf{T}}^{(i)}(:, 1)$
end for
for $j = 1 : N$ **do**
 Izračunaj $\|\hat{\mathbf{b}}_t^{(i)} - \hat{\mathbf{c}}_{t,j}^{(i)}\|_F$
end for
Slika za treniranje čiji su koeficijenti najbliži test slici je rezultat.

7 Alternativni pristup pomoću tenzorske QR faktORIZACIJE

U ovom odjeljku predložena je alternativa T-SVD faktORIZACIJI, T-PQR faktORIZACIJA. Razlog predlaganja ovog algoritma je potencijalna prednost u slučaju ažuriranje baze lica. Mana je da faktORIZACIJA neće biti optimalna. Na slici je prikazano dodavanje ili brisanje lateralnog odsječka \mathcal{L} u bazu podataka \mathcal{A} veličine $l \times m \times n$.



Slika 4: Ažuriranje baze podataka

Prije početka, definirajmo permutacijski tenzor. Permutacijski tenzor $\mathcal{P} \in \mathbb{R}^{m \times m \times n}$ je tenzor čiji su elementi nule i jedinice i vrijedi $\mathcal{P}^T * \mathcal{P} = \mathcal{P} * \mathcal{P}^T = \mathcal{I}$.

Teorem 11. *Neka je \mathcal{A} $l \times m \times n$ tenzor realnih vrijednosti. Tada se \mathcal{A} može faktORIZIRATI u obliku*

$$\mathcal{A} * \mathcal{P} = \mathcal{Q} * \mathcal{R} \quad (15)$$

gdje je \mathcal{Q} $l \times l \times n$ ortogonalni tenzor, \mathcal{R} $l \times m \times n$ f-gornje trokutasti tenzor i \mathcal{P} permutacijski tenzor. ■

Dokaz. Dokaz kada je $\mathcal{P} = \mathcal{I}$, uključen je u [3]. Za općeniti slučaj, kao u T-SVD dokazu, prebacimo se u Fourierovu domenu kao u 5.2. Tada možemo dobiti QR-faktORIZACIJU pivotiranu po stupcima $\hat{\mathbf{A}}^{(1)} \mathbf{P}^{(1)} = \hat{\mathbf{Q}}^{(1)} \hat{\mathbf{R}}^{(1)}$. Prvo primijenimo istu permutaciju na svaki drugi frontalni odsječak od $\hat{\mathbf{A}}$, pa izračunamo QR faktORIZACIJU za svaki permutirani frontalni odsječak

$$\hat{\mathbf{A}}^{(i)} \mathbf{P}^{(1)} = \hat{\mathbf{Q}}^{(i)} \hat{\mathbf{R}}^{(i)}.$$

Neka su \mathbf{P} , $\hat{\mathbf{Q}}$ i $\hat{\mathbf{R}}$ blok dijagonalne matrice s $\mathbf{P}^{(1)}$, $\hat{\mathbf{Q}}^{(i)}$, $\hat{\mathbf{R}}^{(i)}$ na glavnim dijagonalama i primijenimo $\mathbf{F}^* \otimes \mathbf{I}$ slijeva i $\mathbf{F} \otimes \mathbf{I}$ sdesna svake od blok dijagonalne matrice da bismo dobili blok cirkularne matrice. Tada dobivamo \mathcal{P} , \mathcal{Q} , \mathcal{R} primijenjujući fold na prvi blok stupac blok cirkularnih matrica. Primijetimo da, s obzirom da je \mathbf{P} konstantna na blok dijagonali, \mathcal{P} mora biti permutacijski tenzor. Nadalje, \mathcal{P} mora imati $\mathbf{P}^{(1)}$ na prvom frontalnom odsječku i nule drugdje, pa je $\mathcal{A} * \mathcal{P}$ ekvivalentna permutiranju lateralnih odsječaka. □

Potencijalna korist permutacijskog tenzora je permutirati lateralne odsječke tako da je što više energije sadržanu u prvih nekoliko lateralnih odsječaka ortogonalnog tenzora \mathcal{Q} . To je slično korištenju permutacija stupaca s QR-om za matrice. Analogno s algoritmom 4 dolazimo do algoritma 6.

Algorithm 6 Tensor QR metoda

Input: Slike za treniranje: \mathbf{I}_i , $i = 1, 2, \dots, N$; Test slika \mathbf{J} , indeks kraćenja k

Output: Koeficijenti testnih slika s bazom smanjene dimenzije

```

for  $i = 1$  to  $N$  do
     $\mathcal{L}(:, :, i) \leftarrow \mathbf{I}_i$ 
end for
 $\mathcal{M} \leftarrow$  srednja slika
 $\mathcal{A} \leftarrow$  srednja devijacijska forma od  $\mathcal{L}$ 
 $[\mathcal{Q}, \mathcal{R}] \leftarrow$  tenzor (pivotirana) QR dekompozicija od  $\mathcal{A}$ 
 $\mathcal{C} \leftarrow \mathcal{Q}(:, 1:k, :)^T * \mathcal{A}$ 
 $\mathcal{T}(:, 1, :) \leftarrow \text{twist}(\mathbf{J} - \mathcal{M})$ 
 $\mathcal{B} \leftarrow \mathcal{Q}(:, 1:k, :)^T * \mathcal{T}$ 
for  $j = 1$  to  $N$  do
    Izračunaj  $\|\mathcal{B} - \mathcal{C}(:, j, :)\|_F$ 
end for
Slika za treniranje čiji su koeficijenti najbliži test slici je rezultat.

```

U slučaju ažuriranja, u kojem se povećava lateralna dimenzija za 1 tako da $l \times m \times n$ baza postaje $l \times (m + 1) \times n$ baza. Dodajući sliku znači da imamo još jedan lateralni odsječak u originalnom tenzor. S obzirom da je faktORIZACIJA (i ažuriranja na faktORIZACIJU) napravljena u Fourierovoj domeni, prvo moramo uzeti dodatni FFT po niti trećeg moda novog odsječka slike da ga preselimo u Fourierovu domenu. Jednom kad je u Fourierovoj domeni, novi se stupac dodaje na svaku od $\hat{\mathbf{R}}^{(i)}$, i standardni matični QR s pivotiranjem je primijenjen na ažurirane $\hat{\mathbf{P}}^{(1)}$; $\hat{\mathbf{Q}}^{(1)}$; $\hat{\mathbf{R}}^{(1)}$. Ažurirana permutacija se primijeni na $\hat{\mathbf{R}}^{(i)}$ i ažuriranja $\hat{\mathbf{Q}}^{(i)}$ i $\hat{\mathbf{R}}^{(i)}$ su isto izračunata. Drugim riječima, korist od matičnog QR s pivotiranjem na matricu $\hat{\mathbf{A}}^{(i)}$ u Fourierovoj domeni inaprotiv SVD-a na svakoj od tih matrica je da je QR faktORIZACIJA s pivotiranjem puno jeftinija od SVD-a.

8 Testiranje

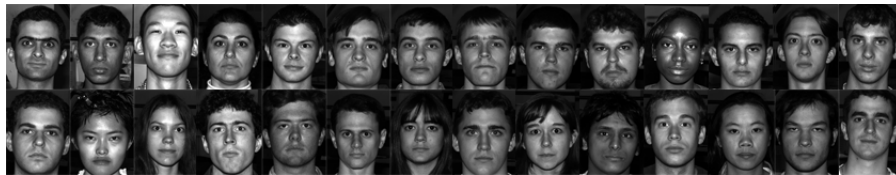
Prepoznavanje lica prethodno opisanim algoritmima (4, 5, 6) testirali smo na podacima iz [7] u Matlab okruženju. Točnost algoritma mjerili smo računajući prosječni broj dobro prepoznatih lica, odnosno formulom

$$\frac{\text{broj dobro prepoznatih lica}}{\text{broj testnih fotografija}}. \quad (16)$$

Kao što smo napomenuli u 4, za nepoznato lice na testnoj slici tvrdimo da je to osoba čiji koeficijent slike je najbliži koeficijentu trenirane slike, u metrici *Frobeniusove* norme.

8.1 Podaci

Slike preuzete iz [7] korištene su po uzoru na [2]. Preuzeta baza podataka sadrži 16128 slika 28 osoba uslikanih u 9 različitih pozicija i u 64 različita osvjetljenja. Primjer fotografija iz baze možemo vidjeti na Slici 5.



Slika 5: Primjer fotografija iz preuzete baze podataka [7]

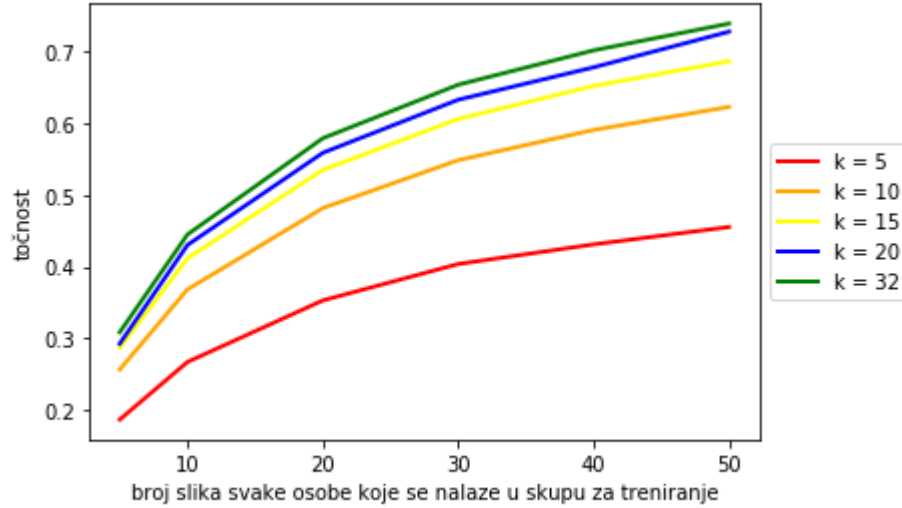
Algoritmi za prepoznavanje lica sastoje se od faze *treniranja* i *testiranja*. Skupovi koji sadrže indekse fotografija koje se nalaze u skupu slika za treniranje, odnosno za testiranje također su preuzeti sa [7]. Za svaki $p = 5, 10, 20, 30, 40, 50$, koji predstavlja broj različitih slika svake osobe koje se nalaze u skupu za treniranje, na nasumičan način generirano je 50 mogućnosti skupa za treniranje, odnosno skupa za testiranje algoritma. Za svaki p pokrenuli smo algoritme na svih 50 mogućnosti i usporedili rezultate.

8.2 Rezultati

8.2.1 TSVD

U ovom ćemo poglavlju prikazati rezultate prethodno opisanih algoritama 4 i 5 koji koriste novi pristup tenzorske dekompozicije opisan u poglavlju 5 i 6.

Uspješnost algoritma 4 u prepoznavanju nepoznatih lica s različitim brojem slika iste osobe u skupu za treniranje, mogu se vidjeti na slici 6.



Slika 6: Rezultati pokretanja algoritma koji koristi TSVD metodu dekompozicije tenzora s fiksnim indeksom suženja k na podacima iz [7].

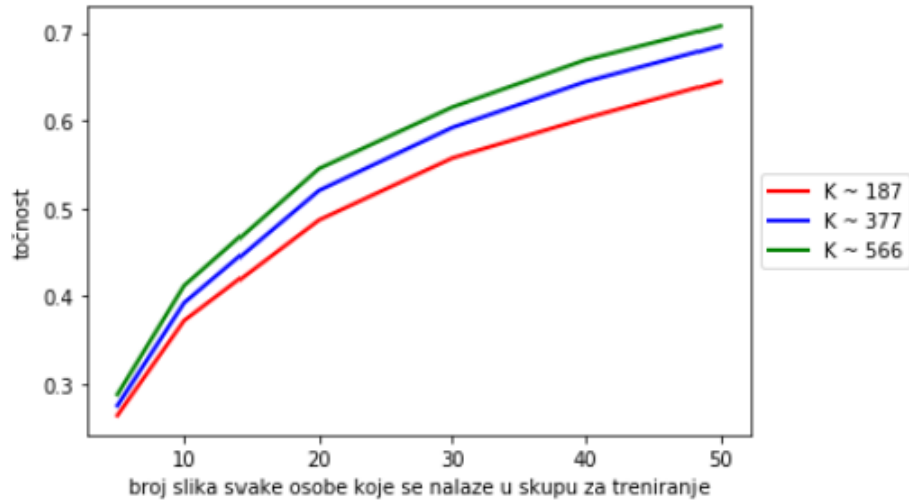
Kako je i očekivano, što skup za treniranje sadrži manje istih slika neke osobe, to je njegova mogućnost u prepoznavanju nove osobe smanjena. Također, iz grafa se jasno vidi da su rezultati točniji ako je k veći, odnosno ako manje *odrežemo* ortogonalni tenzor \mathcal{U} koji projektira slike u prostor manjih dimenzija.

Prednost ovoga algoritma nasprem ostalima je, u slučaju kada su visina i duljina slike približno ista, njegova točnost ne opada drastično u slučaju transponiranja slika koje se nalaze u lateralnim odsječcima. Razlog tome je matricni način spremanja slika u tenzor - timev korelacija između piksela ostaje nepromjenjena.

Algoritam 5 smo također implementirali i testirali na podacima iz [7]. Nažalost, nismo uspjeli dobiti zadovoljavajuće rezultate za $K \approx 100$ u usporedbi sa onima iz [2]. Točnost prepoznavanja lica tu je iznosila tek nekih $\approx 25\%$ što je izrazito loše. Povećavanjem K , odnosno povećavanjem *thresholda* dobivali smo, naravno, i bolje rezultate, usporedive točnosti sa algoritmom 4 kada bi za fiksni k uzeli upravo K/n . Primjerice, u brojkama, u slučaju kada je $K = 32 * 32$, odnosno fiksni $k = 32$, algoritam 4 za $p = 20$ dao je prosječnu točnost od $\approx 50\%$, dok bi algoritam 5 dobro prepoznao $\approx 65\%$ lica. Na slici 7 grafički su prikazani rezultati za $K \approx 187$, $K \approx 377$ i $K \approx 566$.

8.2.2 TQR

Sljedećih nekoliko rečenica posvećeno je algoritmu 6 koji je svojim rezultatima testiranja prepoznavanja lica na podacima iz [7] dao nešto lošije podatke u



Slika 7: Rezultati pokretanja algoritma koji koristi TSVD metodu dekompozicije tenzora s različitim indeksima kraćenja k_i na podacima iz [7].

usporedbi sa algoritmom 4. Glavni razlog takvog ponašanja je taj da je ova metoda dekompozicije tenzora utemeljena na QR dekompoziciji, a ne na SVD dekompoziciji matrice koja daje daleko bolje rezultate u svrhi klasifikacije ([5]).

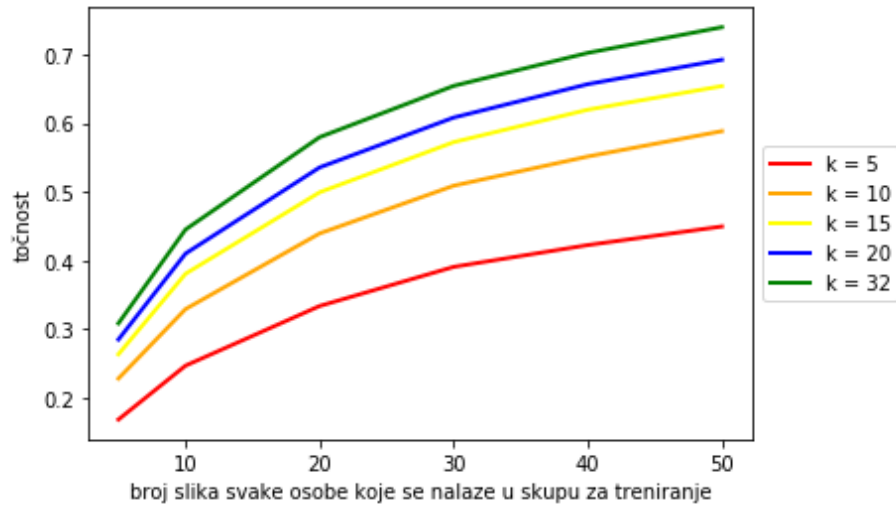
Uspješnost ovog algoritma u varijaciji sa fiksiranim indeksom k za smanjenje prostora i brojem slika iste osobe u skupu za treniranje prikazana je na slici 8.

8.2.3 HOSVD

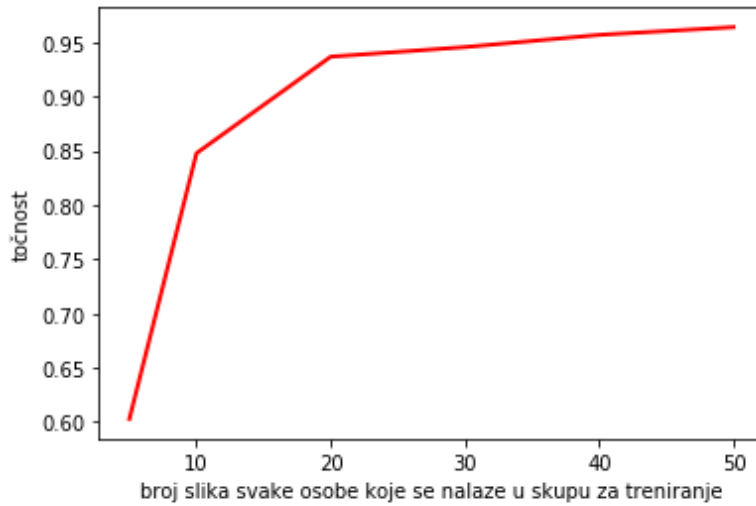
S ciljem usporedbe uspješnost naših algoritama koji su koristili TSVD faktorizaciju tenzora, napisali smo i program za prepoznavanje lica koji koristi HOSVD faktorizaciju po uzoru na [6].

Ukratko, algoritam na testnom skupu fotografija grupira one na kojima se nalazi ista osoba, te *tenzor svake osobe* pretvori `unfold` metodom u tri matrice koje faktorizira koristeći SVD metodu. Svakako, to oduzima mnogo prostora jer su nastale matrice velikih dimenzija. Uz dodatnu činjenicu da se ovo treba izvršiti za svaku osobu, ovo je glavni nedostatak ove metode. Nakon faktorizacije svih *tenzora osoba*, za fotografiju s nepoznatim licem potrebno je izračunati kojoj od osoba je *najbliža* koristeći dobivene ortogonalne matrice \mathcal{U} .

Ova metoda, iako prvenstveno nije zamišljena za prepoznavanje lica, dobro se pokazala u eksperimentima. Uspješnost ovog algoritma možemo vidjeti na slici 9.



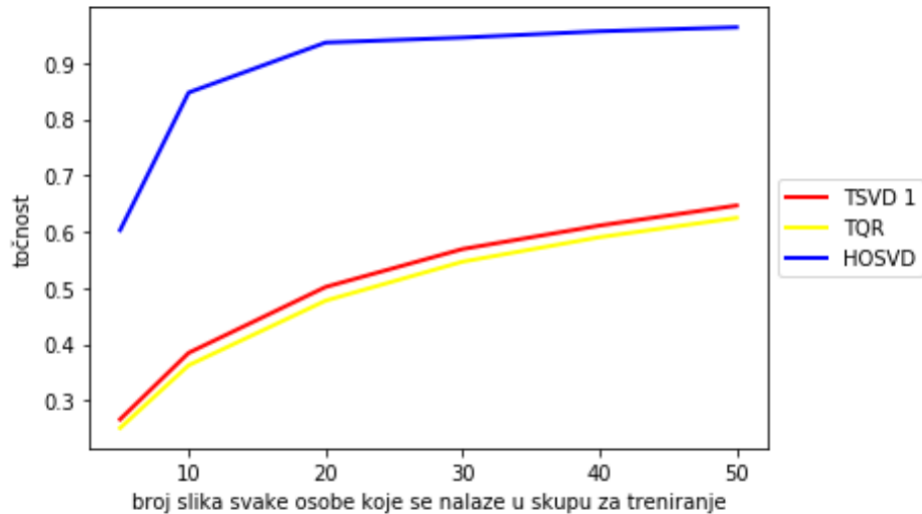
Slika 8: Rezultati pokretanja algoritma koji koristi TQR metodu dekompozicije tenzora na podacima iz [7].



Slika 9: Rezultati pokretanja algoritma koji koristi HOSVD metodu dekompozicije tenzora na podacima iz [7].

8.2.4 Zaključak

Većinom sve metode tenzorskih dekompozicija temeljene su na upotrebi SVD faktorizacije matrice. Metode variraju u načinu odabira matrica - npr. lateralni



Slika 10: Usporedba svih rezultata pokretanja prethodno navedenih algoritma na podacima iz [7].

odsječci ili **unfold** tenzora u nekom modu, koje će se faktorizirati. Algoritmi za prepoznavanja lica na fotografijama opisani u ovom seminaru svi su utemeljeni na tenzorskim operacijama i dekompoziciji opisanim u 5 i 6 i nastali su kao generalizacija SVD, HOSVD, PCA i drugih metoda.

Takvi algoritmi koriste se za aproksimacije s malim rangom, rekonstrukciju i prepoznavanje slika pod nadzorom ([4]). Brojna testiranja i eksperimenti pokazali su da su takvi algoritmi dovoljno dobri u usporedbi s klasičnim matričnim algoritmima.

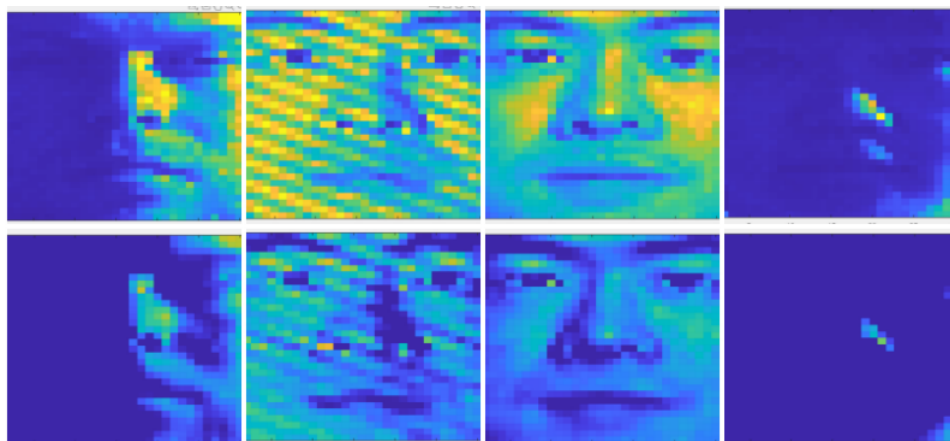
Na slici 10 možemo vidjeti prosječnu uspješnost svih promatranih algoritama u našem radu.

8.3 Fotografije prije i poslije primjena TSVD-a

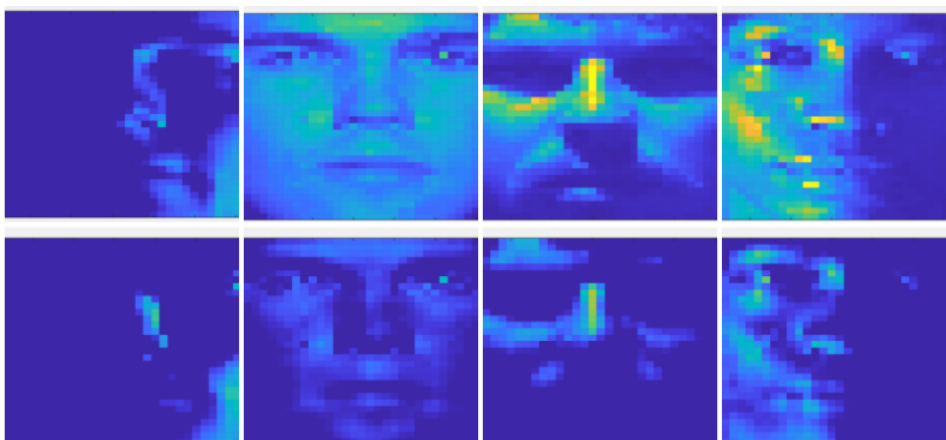
Na sljedećih nekoliko fotografija možemo vidjeti kako TSVD metoda utječe na njihovu kvalitetu.

Na slici 11 mogu se vidjeti neki primjeri nepoznatih lica koje je algoritam 4 dobro prepoznao, a na slici 12 možemo vidjeti neke primjere kada je algoritam krivo prepoznao osobu.

U prvom retku svake slike nalaze se originalne fotografije, dok su u donjem retku slike nastale nakon primjene TSVD metode s fiksnim indeksom suženja k .



Slika 11: Primjer dobro prepoznatih lica. Gore su originalne slike, ispod su slike s primjenom t-SVD metode.



Slika 12: Primjer loše prepoznatih lica. Gore su originalne slike, ispod su slike s primjenom t-SVD metode.

Dodatno, algoritam za faktORIZACIJU tenzora metodom TSVD, iskoristili smo i na video sadržaju. Jedna snimka zaslona koja prikazuje dio originalnog video sadržaja (lijevo) i dio video sadržaja nakon primjene TSVD-a (desno) nalazi se na slici [13](#).



Slika 13: Video prije i nakon primjene TSVD-a na njegov tenzor podataka.

Literatura

- [1] A.S. Georgiades, P.N. Belhumeur i D.J. Kriegman. „From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose”. *IEEE Trans. Pattern Anal. Mach. Intelligence* 23.6 (2001), str. 643–660.
- [2] Ning Hao i dr. „Facial Recognition Using Tensor-Tensor Decompositions”. *SIAM Journal on Imaging Sciences [electronic only]* 6 (veljača 2013). DOI: [10.1137/110842570](https://doi.org/10.1137/110842570).
- [3] M.E. Kilmer, C.D. Martin i L. Perrone. „A third-order generalization of the matrix svd as a product of third-order tensors”. *Tufts Computer Science Technical Report* (2008).
- [4] Liang Liao i Stephen Maybank. „Generalized Visual Information Analysis Via Tensorial Algebra”. *Journal of Mathematical Imaging and Vision* (veljača 2020). DOI: [10.1007/s10851-020-00946-9](https://doi.org/10.1007/s10851-020-00946-9).
- [5] Neil Muller, Lourenço Magaia i Ben Herbst. „Singular Value Decomposition, Eigenfaces, and 3D Reconstructions”. *Society for Industrial and Applied Mathematics* 46 (rujan 2004), str. 518–545.
- [6] Berkant Savas i Lars Eldén. „Handwritten digit classification using higher order singular value decomposition”. *Pattern Recognit.* 40 (2007), str. 993–1003.
- [7] *The Extended Yale Face Database B*. <http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html>.. Dohvaćeno: 30.3.2020.