

Sveučilište u Zagrebu

Prirodoslovno-matematički fakultet

Kompresija slike pomoću tenzora

Naziv tima:

Glitch u matrici

Članovi tima:

Željka Baća, Nikola Kašnar, Ivana Kristić

Mentor:

prof. dr. sc. Zlatko Drmač

Zagreb, 3. ožujka 2025.

Sadržaj

1	Uvod	2
2	Tenzori	3
2.1	Općenito o tenzorima	3
2.2	Tenzorska dekompozicija	4
2.3	Tucker dekompozicija	6
2.4	Postojeće kompresijske metode	8
2.4.1	Kompresija iz CP dekompozicije	8
3	Eksperimentalni rezultati	11
3.1	Kompresija pomoću CP dekompozicije	11
3.1.1	$R = 10$	12
3.1.2	$R = 50$	12
3.1.3	$R = 100$	13
3.1.4	$R = 200$	14
3.1.5	$R = 500$	14
3.1.6	Analiza rezultata	15
3.2	Tucker dekompozicija	17
3.2.1	Kompresija cijele slike	18
3.2.2	Kompresiramo R kanal	20
3.2.3	Kompresiramo G kanal	22
3.2.4	Kompresiramo B kanal	24
3.2.5	Selektivno kompresiramo dio fotografije	26
4	Literatura	29

1 Uvod

U današnje vrijeme, s porastom količine podataka u različitim područjima, poput satelitskih snimaka, medicinskih slika ili video zapisa, postaje sve važnije razvijati učinkovite metode za pohranu i prijenos ovih podataka. Višedimenzionalni signali, poput multispektralnih slika i video sekvenci, predstavljaju poseban izazov zbog svoje velike veličine i složenosti. Kompresija ovih podataka ključna je kako bi se smanjili zahtjevi za pohranom i brzinom prijenosa, a istovremeno zadržala što veća kvaliteta podataka.

U ovom seminaru predstavit ćemo metodu učenja tenzorske dekompozicije za kompresiju slika. Tenzorska dekompozicija je matematički alat koji omogućuje razlaganje višedimenzionalnih podataka na manje komponente, čime se postiže smanjenje redundancije i učinkovitija kompresija. Za razliku od tradicionalnih metoda, koje koriste unaprijed definirane baze za kompresiju, naša metoda uči baze iz skupa podataka za obuku, što omogućuje bolju prilagodbu specifičnim karakteristikama podataka.

U seminaru ćemo detaljno objasniti:

1. Što su tenzori i kako se koriste za predstavljanje višedimenzionalnih podataka.
2. Kako funkcioniра **Tucker dekompozicija**, jedan od najčešće korištenih pristupa tenzorskoj dekompoziciji.
3. Kako se metodom učenja tenzorske dekompozicije postiže kompresija podataka, uključujući kvantizaciju i kodiranje.
4. Rezultate metode na sintetičkim i stvarnim podacima, poput multispektralnih satelitskih snimaka, te usporedbu s drugim metodama kompresije.

Kroz ovaj seminar, pokazat ćemo kako se tenzorska dekompozicija može primijeniti za rješavanje problema kompresije u stvarnim aplikacijama, uz istaknute prednosti u odnosu na tradicionalne pristupe.

2 Tenzori

2.1 Općenito o tenzorima

U ovom potpoglavlju ćemo ukratko proći kroz osnove tenzora s obzirom da smo u detalje išli na predavanjima. Tenzori su generalizacija vektora i matrica na višedimenzionalne prostore. Dok su vektori jednodimenzionalni nizovi, a matrice dvodimenzionalni nizovi, tenzori mogu imati tri ili više dimenzija. Formalno, tenzor reda N (ili N -dimenzionalni tenzor) definira se kao višedimenzionalni niz elemenata indeksiranih s N indeksa.

Definicija 2.1.1 Neka su I_1, I_2, \dots, I_N dimenzije tenzora. Tenzor \mathcal{X} reda N može se predstaviti kao funkcija:

$$\mathcal{X} : I_1 \times I_2 \times \cdots \times I_N \rightarrow \mathbb{R},$$

gdje je $\mathcal{X}_{i_1, i_2, \dots, i_N}$ vrijednost tenzora na poziciji (i_1, i_2, \dots, i_N) . Tenzor se može zapisati i kao višedimenzionalni niz:

$$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}.$$

Primjeri tenzora

- **Skalar:** Tenzor reda 0 (nema dimenzija). Primjer: $x = 5$.
- **Vektor:** Tenzor reda 1 (jedna dimenzija). Primjer: $\mathbf{v} = [1, 2, 3]$.
- **Matrica:** Tenzor reda 2 (dvije dimenzije). Primjer:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}.$$

- **Tenzor višeg reda:** Tenzor reda 3 ili više. Primjer: tenzor \mathcal{X} dimenzija $2 \times 2 \times 2$:

$$\mathcal{X} = \left[\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \right].$$

Svojstva tenzora

Tenzori imaju nekoliko ključnih svojstava koja ih čine korisnim za predstavljanje višedimenzionalnih podataka:

- **Multilinearnost:** Tenzori su linearni u odnosu na svaku dimenziju. To znači da se operacije na tenzorima mogu izvoditi nezavisno po svakoj dimenziji.

- **Struktura informacija:** Tenzori mogu pohraniti složene odnose između podataka, poput prostornih, vremenskih ili spektralnih ovisnosti.
- **Kompaktnost:** Tenzorska dekompozicija omogućuje smanjenje redundancije u podacima, što je korisno za kompresiju i analizu.

Još neka od svojstva spomenutih na predavanjima su:

Definicija 2.1.2 *Euklidski skalarni produkt definiramo kao:*

$$\langle X, Y \rangle_F = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}$$

Definicija 2.1.3 *Normu(Frobeniusovu) definiramo kao:*

$$\|X\|_F = \sqrt{\langle X, X \rangle_F} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2}$$

Primjena tenzora

Tenzori se koriste u različitim područjima, uključujući:

- **Obradu signala:** Za analizu višedimenzionalnih signala poput video zapisa ili multispektralnih slika.
- **Strojno učenje:** Za predstavljanje složenih podataka u modelima dubokog učenja.
- **Fiziku i inženjerstvo:** Za modeliranje fizikalnih sustava s višestrukim dimenzijama.

Tenzorska dekompozicija, poput Tucker dekompozicije, koristi se za razlaganje tenzora na manje komponente, što omogućuje učinkovitu kompresiju i analizu podataka.

2.2 Tenzorska dekompozicija

Tenzorska dekompozicija je matematička tehnika koja se koristi za razlaganje tenzora na manje komponente, čime se otkrivaju skrivene strukture i obrasci u višedimenzionalnim podacima. Slično kao što se matrica može rastaviti na proizvod manjih matrica (npr. SVD -om), tenzor se može rastaviti na kombinaciju tenzora nižeg reda i faktorskih matrica. Ova tehnika koristi se za smanjenje redundancije, kompresiju podataka i ekstrakciju značajki.

Definicija 2.2.1 Neka je $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ tenzor reda N . Tenzorska dekompozicija ovog tenzora može se zapisati kao:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \cdots \times_N \mathbf{U}^{(N)},$$

gdje je:

- $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ **tenzor jezgre** (core tensor) manjih dimenzija,
- $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ su **faktorske matrice** za svaku dimenziju n ,
- \times_n označava **mode- n produkt** (tenzorsko-matrično množenje po n -toj dimenziji).

Vrste tenzorske dekompozicije

Postoji nekoliko glavnih vrsta tenzorske dekompozicije, od kojih su najpoznatije:

- **Tucker dekompozicija:**

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \cdots \times_N \mathbf{U}^{(N)}.$$

Tucker dekompozicija je generalizacija SVD za tenzore višeg reda. Jezgrini tenzor \mathcal{G} i faktorske matrice $\mathbf{U}^{(n)}$ mogu imati proizvoljne dimenzije, što omogućuje fleksibilnost u modeliranju.

- **dekompozicija (CANDECOMP/PARAFAC):**

$$\mathcal{X} \approx \sum_{r=1}^R \lambda_r \cdot \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)},$$

gdje je \circ vanjski produkt, a $\mathbf{a}_r^{(n)}$ su vektori faktora za svaku dimenziju. CP dekompozicija je specijalni slučaj Tucker dekompozicije gdje je jezgrini tenzor dijagonalan.

- **(Tenzorski vlakić) Tensor Train (TT) dekompozicija:**

$$\mathcal{X} \approx \mathcal{G}_1 \times_1 \mathcal{G}_2 \times_2 \cdots \times_{N-1} \mathcal{G}_N,$$

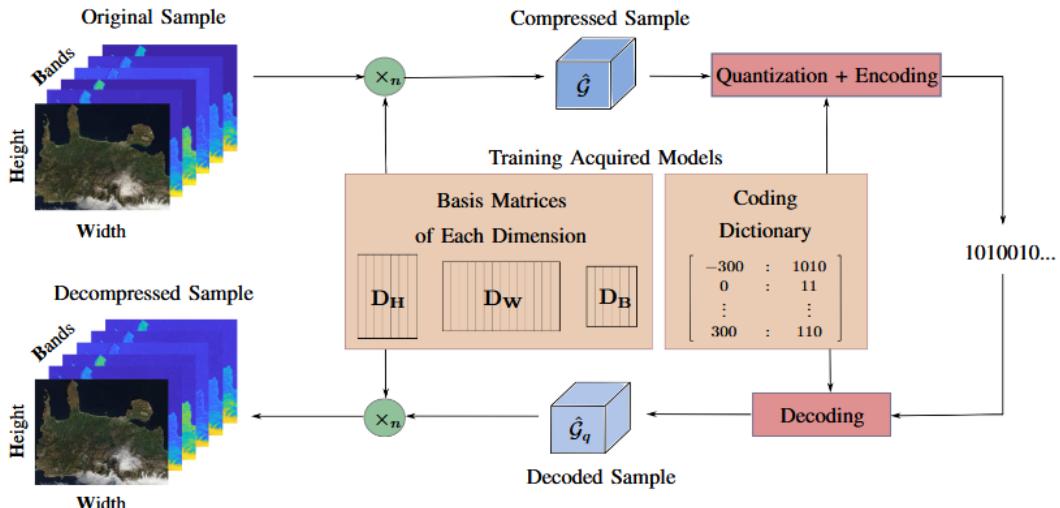
gdje su \mathcal{G}_n tenzori nižeg reda povezani u "vlak" (train). Ova dekompozicija je posebno korisna za tenzore vrlo visokog reda.

Prednosti tenzorske dekompozicije

- **Smanjenje dimenzionalnosti:** Tenzorska dekompozicija omogućuje smanjenje broja parametara potrebnih za opisivanje podataka.
- **Otkrivanje strukture:** Otkriva skrivene obrasce i odnose u podacima.
- **Fleksibilnost:** Može se primijeniti na tenzore proizvoljnog reda i dimenzija.

Tenzorska dekompozicija, posebno Tucker dekompozicija, ključna je komponenta učenja tenzorske dekompozicije za kompresiju višedimenzionalnih signala, što ćemo detaljnije objasniti u sljedećim dijelovima seminara.

Na sljedećoj slici možemo vidjeti primjer kako funkcioniра dekompozicija slike:



Slika 1: Tenzorska dekompozicija

2.3 Tucker dekompozicija

Tucker dekompozicija je jedna od najčešće korištenih metoda tenzorske dekompozicije. Ona generalizira singularnu dekompoziciju (SVD) matrice na tenzore višeg reda. Tucker dekompozicija razlaže tenzor na tenzor jezgre (core tensor) i skup faktorskih matrica, što omogućuje smanjenje dimenzionalnosti i otkrivanje skrivenih struktura u podacima.

Definicija 2.3.1 Neka je $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ tenzor reda N . Tucker dekompozicija ovog tenzora može se zapisati kao:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \dots \times_N \mathbf{U}^{(N)},$$

gdje je:

- $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$ **jezgrini tenzor** (core tensor) manjih dimenzija,
- $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ su **faktorske matrice** za svaku dimenziju n ,
- \times_n označava **mode- n produkt** (tenzorsko-matrično množenje po n -toj dimenziji).

Kratko objasnimo pojedine dijelove:

- **Tenzor jezgre (\mathcal{G})**: - Jezgrini tenzor je tenzor manjih dimenzija koji sadrži "težine" ili koeficijente za kombinaciju faktorskih matrica. - Dimenzije jezgrinog tenzora R_1, R_2, \dots, R_N određuju koliko će se informacija sačuvati u svakoj dimenziji.
- **Faktorske matrice ($\mathbf{U}^{(n)}$)**: - Faktorske matrice su ortogonalne matrice koje predstavljaju baze za svaku dimenziju tenzora. - Svaka faktorska matrica $\mathbf{U}^{(n)}$ preslikava originalnu dimenziju I_n u smanjenu dimenziju R_n .
- **Mode- n produkt (\times_n)**: - Mode- n produkt je operacija koja množi tenzor s matricom po određenoj dimenziji. - Na primjer, $\mathcal{G} \times_1 \mathbf{U}^{(1)}$ množi jezgrini tenzor \mathcal{G} s matricom $\mathbf{U}^{(1)}$ po prvoj dimenziji.

Razmotrimo jednostavan primjer Tucker dekompozicije za tenzor reda 3 (trodimenzionalni tenzor).

Primjer 2.3.1 Neka je $\mathcal{X} \in \mathbb{R}^{2 \times 2 \times 2}$ tenzor reda 3:

$$\mathcal{X} = \left[\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \right].$$

Prepostavimo da želimo izvršiti Tucker dekompoziciju s jezgrinim tenzorom dimenzija $2 \times 2 \times 2$ i faktorskim matricama $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}$ dimenzija 2×2 .

Koraci dekompozicije su sljedeći:

1. Izračun faktorskih matrica: - Faktorske matrice $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}$ mogu se izračunati koristeći metode poput SVD ili iterativnih algoritama (npr. ALS - Alternating Least Squares).

2. Izračun jezgrinog tenzora: - Jezgrini tenzor \mathcal{G} izračuna se kao:

$$\mathcal{G} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times_3 \mathbf{U}^{(3)T}.$$

3. Rekonstrukcija tenzora: - Originalni tenzor može se rekonstruirati kao:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}.$$

Prednosti Tucker dekompozicije

- **Fleksibilnost:** Tucker dekompozicija može se primijeniti na tenzore proizvoljnog reda i dimenzija.
- **Smanjenje dimenzionalnosti:** Omogućuje smanjenje broja parametara potrebnih za opisivanje podataka.
- **Otkrivanje strukture:** Otkriva skrivene obrasce i odnose u podacima.

Tucker dekompozicija ključna je komponenta učenja tenzorske dekompozicije za kompresiju višedimenzionalnih signala, što ćemo detaljnije objasniti u sljedećim dijelovima seminara. Primejr korisštenja Tucker dekompozicije možete vidjeti u poglavljju sa ekperimentalnim rezultatima.

2.4 Postojeće kompresijske metode

2.4.1 Kompresija iz CP dekompozicije

CP dekompozicija omogućava kompresiju tenzora tako što ga razlaže na konačnu sumu tenzora ranga 1, čime se smanjuje količina podataka potrebnih za pohranu. Za razliku od Tucker dekompozicije, gdje postoji jezgreni tenzor koji modulira interakcije među faktorima, CP dekompozicija koristi samo faktorske vektore.

$$\mathcal{X} \approx \sum_{r=1}^R \lambda_r \cdot \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)},$$

- **R - rang dekompozicije**, odnosno broj komponenti u aproksimaciji,
- λ_r - **skalarni koeficijenti**,
- $\mathbf{a}_r^{(n)}$ - **faktorski vektori** za svaku dimenziju n ,
- \circ - **vanjski produkt**.

Umjesto pohranjivanja cijelog tenzora \mathcal{X} , možemo pohraniti samo faktorske matrice $A^{(n)}$ i skalarne koeficijente λ_r .

Proces kompresije pomoću CP dekompozicije sastoji se od slijedećih koraka:

1. Računanje CP dekompozicije

Za dani tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, tražimo faktorske matrice $A^{(1)}, A^{(2)}, \dots, A^{(N)}$ i

vektore λ_r . Najčešće se koristi **Alternating Least Squares (ALS)** metoda, koja minimizira grešku između originalnog tenzora i njegove aproksimacije:

$$\min_{A^{(1)}, A^{(2)}, \dots, A^{(N)}} \left\| \mathcal{X} - \sum_{r=1}^R \lambda_r \cdot \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)} \right\|_F^2.$$

2. Spremanje podataka

Spremamo faktorske matrice i skalarne koeficijente dobivene CP dekompozicijom.

3. Dekompresija podataka

Kada trebamo rekonstruirati podatke, jednostavno koristimo:

$$\hat{\mathcal{X}} = \sum_{r=1}^R \lambda_r \cdot \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)}. \quad (1)$$

3 Eksperimentalni rezultati

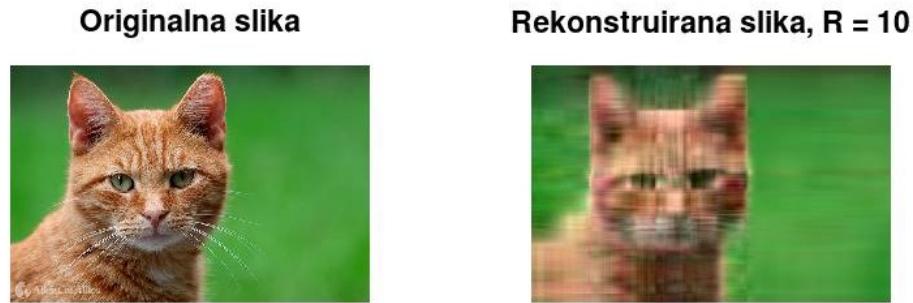
3.1 Kompresija pomoću CP dekompozicije

```
1 %% Ucitavanje slike
2 img = imread('cat.jpg');
3 img = im2double(img);
4 [I1, I2, I3] = size(img);
5
6 original_size = whos('img'); % Velicina originalne slike
7 original_size_MB = original_size.bytes / (1024 * 1024); % MB
8
9 X = tensor(img); % Pretvaranje slike u tenzor
10
11 %% CP dekompozicija
12 R = 500; % Rang kompresije (veci R = bolja rekonstrukcija)
13 [P, Uinit, output] = cp_als(X, R, 'tol', 1e-6, 'maxiters', 50, ,
14     'printitn', 1);
15
16 % Velicina faktorskih matrica nakon kompresije
17 A_size = whos('P');
18 A_bytes = numel(P.U{1}) * 8; % Svaki double zauzima 8 bajtova
19 B_bytes = numel(P.U{2}) * 8;
20 C_bytes = numel(P.U{3}) * 8;
21 lambda_bytes = numel(P.lambda) * 8;
22
23 % Ukupna velicina nakon kompresije (u MB)
24 compressed_size_MB = (A_bytes + B_bytes + C_bytes + lambda_bytes) /
25     (1024 * 1024);
26 compression_ratio = original_size_MB / compressed_size_MB;
27
28 %% Rekonstrukcija slike iz CP dekompozicije
29 X_reconstructed = full(P);
30
31 %% Prikaz originalne i rekonstruirane slike
32 figure;
33 subplot(1,2,1);
34 imshow(img);
35 title('Originalna slika');
36 subplot(1,2,2);
37 imshow(double(X_reconstructed));
38 title(['Rekonstruirana slika, R = ', num2str(R)]);
39
40 %% Izracunavanje greske kompresije
41 rmse = sqrt(mean((img(:) - double(X_reconstructed(:))).^2));
42 disp(['RMSE greska rekonstrukcije: ', num2str(rmse)]);
```

Kako bi analizirali utjecaj ranga CP dekompozicije na kvalitetu rekonstrukcije slike, izvršili smo testiranje sa različitim vrijednostima R na slijedećog slici:

3.1.1 $R = 10$

Za početak smo postavili rang dekompozicije $\mathbf{R} = 10$.



Slika 3: Rekonstruirana slika, $R = 10$

RMSE greška za ovu rekonstrukciju iznosi:

$$RMSE = 0.07883$$

Vidimo da je rekonstruirana slika primjetno zamagljena, iz čega možemo zaključiti da mali broj komponenti ne može u potpunosti sačuvati detalje slike.

3.1.2 $R = 50$

Povećanjem ranga na $\mathbf{R} = 50$, dobivamo slijedeći rezultat:



Slika 4: Rekonstruirana slika, $R = 50$

Smanjena je greška rekonstrukcije:

$$RMSE = 0.0495$$

Slika sada izgleda manje mutno, a veći broj komponenti rezultirao je i manjom greškom rekonstrukcije.

3.1.3 $R = 100$

Daljnje povećanje ranga na $R = 100$ dodatno poboljšava kvalitetu rekonstrukcije:



Slika 5: Rekonstruirana slika, $R = 100$

Izračunata RMSE greška:

$$RMSE = 0.03466$$

Vidljivo je da je slika sada još manje mutna, a detalji su bolje očuvani.

3.1.4 $R = 200$



Slika 6: Rekonstruirana slika, $R = 200$

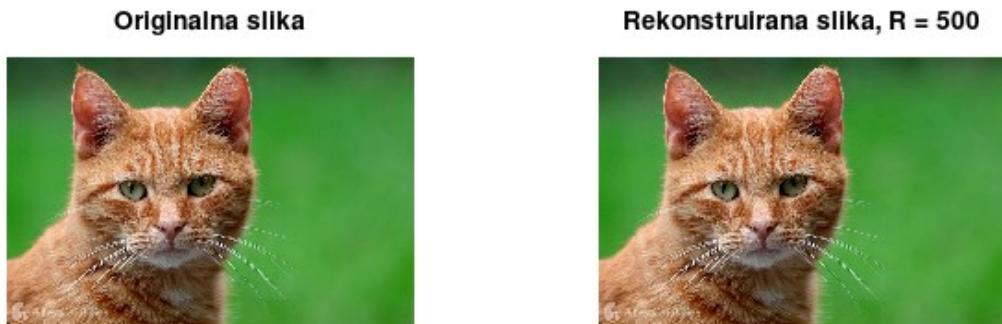
Dobivena RMSE greška je:

$$RMSE = 0.01971$$

Primijećujemo da su razlike između originalne i rekonstruirane slike sada gotovo potpuno neprimjetne.

3.1.5 $R = 500$

Na kraju smo testirali $R = 500$.



Slika 7: Rekonstruirana slika, $R = 500$

Dobivena RMSE greška je:

$$RMSE = 0.00595$$

Razlike između originalne i rekonstruirane slike su potpuno neprimjetne.

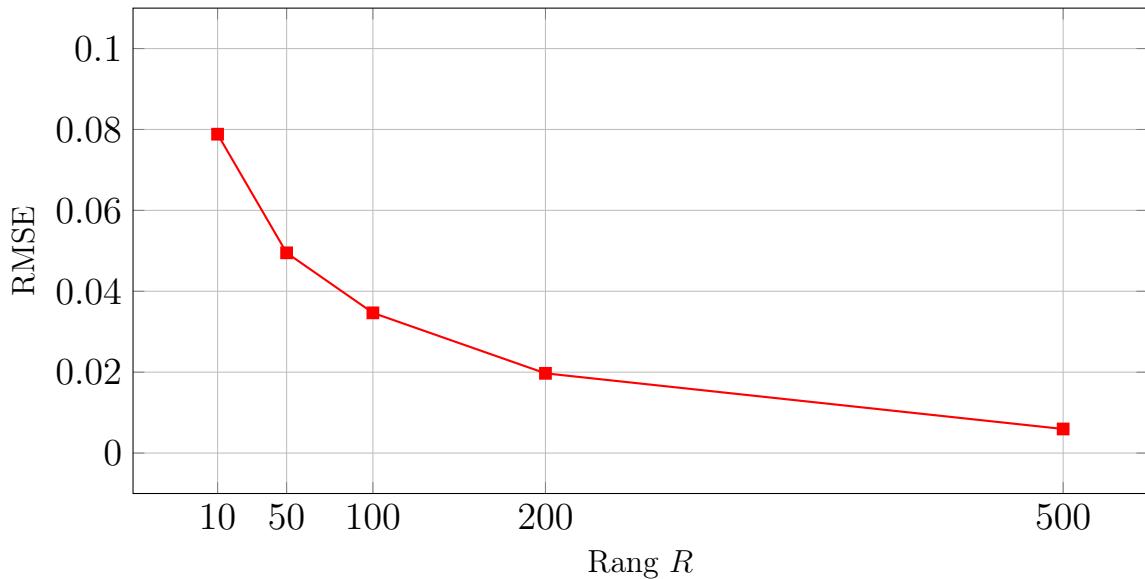
3.1.6 Analiza rezultata

Veličina početne slike je 7.4499 MB.

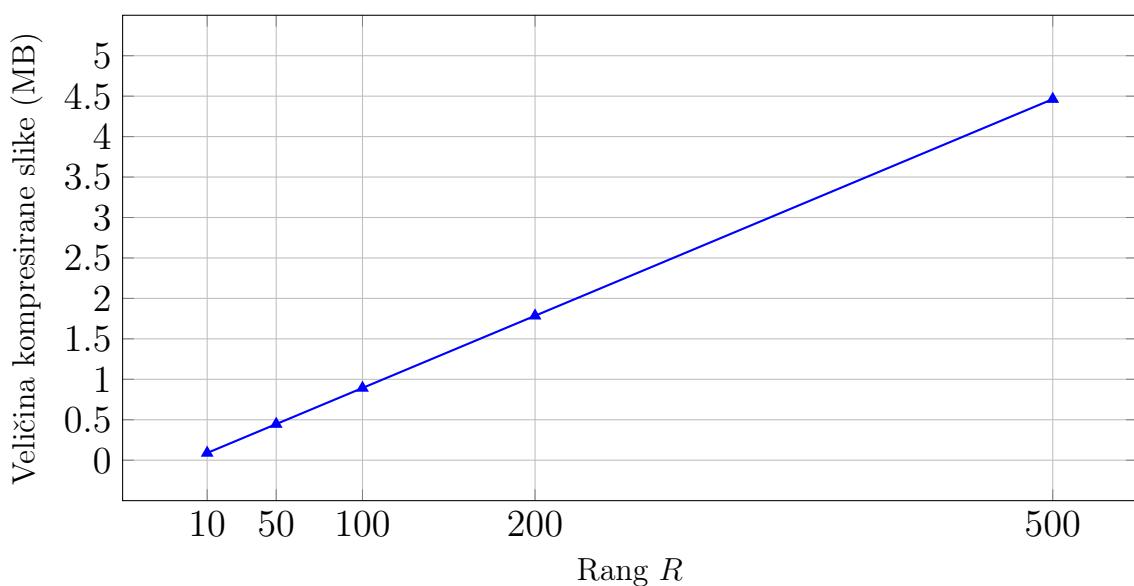
U tablici prikazujemo RMSE greške i veličinu kompresije za različite vrijednosti ranga R :

Rang R	RMSE greška	Vizualna kvaliteta	Veličina (MB)	Faktor kompresije
10	0.07883	Jako mutna slika	0.0893	83.46
50	0.04950	Blago mutna slika	0.4463	16.69
100	0.03466	Gotovo originalna slika	0.8926	8.35
200	0.01971	Razlike su neprimjetne	1.7853	4.17
500	0.00595	Razlike su neprimjetne	4.4632	1.67

Tablica 1: Rezultati kompresije sa različitim vrijednostima R



Slika 8: RMSE u ovisnosti o rangu R



Slika 9: Veličina kompresirane slike u ovisnosti o rangu R

3.2 Tucker dekompozicija

Napravili smo primjer Tucker dekompozicije koji koristi library Tensor toolbox koji se može instalirati sa izvora [5].

```
1 clear; clc;
2 % Generiramo 3D tenzor
3 I = 10; J = 10; K = 10; % Dimensions of the tensor
4 X = randn(I, J, K); % Random 3D tensor of size I x J x K
5
6 % Napravimo Tucker dekompoziciju
7 % Specificiramo rankove
8 R1 = 5;
9 R2 = 5;
10 R3 = 5;
11
12 % Napravimo Tucker dekompoziciju koristeci Tensor toolbox
13 [G, U1, U2, U3] = tucker_als(X, [R1, R2, R3]);
14
15 % Rekonstruiramo tenzor iz dekompozicije
16 X_reconstructed = ttm(G, {U1, U2, U3}, [1, 2, 3]);
17
18 % Evaluiramo gresku
19 reconstruction_error = norm(X(:) - X_reconstructed(:)) / norm(X(:));
20 fprintf('Reconstruction Error: %.4f\n', reconstruction_error);
21
22 % Vizualiziramo originalni i rekonstruirani tenzor
23 slice_original = X(:, :, 1);
24 slice_reconstructed = X_reconstructed(:, :, 1);
25
26 figure;
27 subplot(1, 2, 1);
28 imagesc(slice_original);
29 title('Original Tensor Slice');
30 colorbar;
31
32 subplot(1, 2, 2);
33 imagesc(slice_reconstructed);
34 title('Reconstructed Tensor Slice');
35 colorbar;
```

Ako koristimo kod na primjeru neke slike, kod će ispisati grešku u rekonstrukciji slike, tj. nešto u ovom obliku:

Reconstruction Error: 0.1234

Ovo znači da je greška rekonstrukcije 12.34%. To pokazuje koliko je tenzor uspješno komprimiran i rekonstruiran.

3.2.1 Kompresija cijele slike

Tucker dekompozicija se primjenjuje na cijelu sliku, a smanjenjem ranga se postiže kompresija:

```
1 clc; clear; close all;
2
3 % Ucitavanje slike
4 img = imread('cat.jpg');
5 img = im2double(img);
6 [I1, I2, I3] = size(img);
7
8 % Veličina originalne slike
9 original_size = whos('img');
10 original_size_MB = original_size.bytes / (1024 * 1024); %
    Pretvaranje u MB
11
12 % Pretvaranje slike u tenzor
13 X = tensor(img);
14
15 % CP dekompozicija
16 R = 500; % Rang kompresije (veći R = bolja rekonstrukcija)
17 [P, Uinit, output] = cp_als(X, R, 'tol', 1e-6, 'maxiters', 50, ,
    printitn', 1);
18
19 % Veličina faktorskih matrica nakon kompresije
20 A_size = whos('P');
21 A_bytes = numel(P.U{1}) * 8; % Svaki double zauzima 8 bajtova
22 B_bytes = numel(P.U{2}) * 8;
23 C_bytes = numel(P.U{3}) * 8;
24 lambda_bytes = numel(P.lambda) * 8;
25
26 % Ukupna veličina nakon kompresije (u MB)
27 compressed_size_MB = (A_bytes + B_bytes + C_bytes + lambda_bytes) /
    (1024 * 1024);
28
29 % Faktor kompresije
30 compression_ratio = original_size_MB / compressed_size_MB;
31
32 % Rekonstrukcija slike iz CP dekompozicije
33 X_reconstructed = full(P);
34
```

Kod ovog pristupa kompresija se primjenjuje na cijelu sliku, uključujući sva tri RGB kanala. Budući da se svi kanali istovremeno podvrgavaju dekompoziciji, dolazi do značajne promjene u strukturi slike.

Promjene su odmah vidljive, posebno pri nižim vrijednostima ranga.



(a) Originalna slika



(b) $R = 10$



(c) $R = 100$



(d) $R = 200$



(e) $R = 400$



(f) $R = 600$



(g) $R = 700$



(h) $R = 703$

Rank (r)	Compression Size (MB)	Compression Ratio	RMSE
703	23.6679	0.3148	0.0000
700	23.5879	0.3158	2.7659
600	20.9191	0.3561	16.1995
400	14.3372	0.5196	31.0763
200	6.2531	1.1914	40.9109
100	2.8976	2.5710	44.0696
10	0.2692	27.6776	51.9897

Tablica 2: Rezultati kompresije sva 3 kanala

3.2.2 Kompresiramo R kanal

Kompresija samo dijela slike može dovesti do većih ušteda u veličini datoteke, poboljšane performanse i optimizaciju resursa, dok se istovremeno čuva ključna vizualna kvaliteta.

```

1 % Primjena Tucker dekompozicije samo na ROI
2 rank_ROI = 100; % Smanjeni rang samo za ROI
3
4 [%core_R, factors_R] = tucker_als(R_ROI, rank_ROI, rank_ROI);
5 [%core_G, factors_G] = tucker_als(G_ROI, rank_ROI, rank_ROI);
6 [core_B, factors_B] = tucker_als(B_ROI, rank_ROI, rank_ROI);
7
8 % Rekonstrukcija ROI-a
9 %R_rec_ROI = factors_R{1} * core_R * factors_R{2}';
10 %G_rec_ROI = factors_G{1} * core_G * factors_G{2}';
11 B_rec_ROI = factors_B{1} * core_B * factors_B{2}';
12
13 % Vra anje rekonstruiranog ROI-a u originalnu sliku
14 R(y1:y2, x1:x2) = R_ROI;
15 G(y1:y2, x1:x2) = G_ROI;
16 B(y1:y2, x1:x2) = B_rec_ROI;
17
18 % Spajanje kanala
19 reconstructed_img = cat(3, R, G, B);
20

```

Slika 11: Selektivna dekompozicija



(a) Originalna slika



(b) $R = 10$



(c) $R = 100$



(d) $R = 200$



(e) $R = 300$



(f) $R = 400$



(g) $R = 600$

Rang (r)	Compression Size (MB)	Compression Ratio	RMSE
703	7.8893	0.9443	0.0000
600	6.9730	1.0684	11.5161
400	4.7791	1.5589	21.6177
300	3.3554	2.2203	26.4333
200	2.0844	3.5742	29.3990
100	0.9659	7.7130	31.8884
10	0.0897	83.0329	36.0672

Tablica 3: Rezultati kompresije za različite vrijednosti ranga (r)

3.2.3 Kompresiramo G kanal

Rang (r)	Compression Size (MB)	Compression Ratio	RMSE
703	7.8893	0.9443	0.0000
600	6.9730	1.0684	8.5338
400	4.7791	1.5589	15.3880
300	3.3554	2.2203	18.3681
200	2.0844	3.5742	20.9832
100	0.9659	7.7130	22.9628
10	0.0897	83.0329	30.0687

Tablica 4: Rezultati kompresije za različite vrijednosti ranga (r)



(a) Originalna slika



(b) $R = 10$



(c) $R = 100$



(d) $R = 200$



(e) $R = 300$



(f) $R = 400$



(g) $R = 600$



(h) $R = 700$

3.2.4 Kompresiramo B kanal

Rang (r)	Compression Size (MB)	Compression Ratio	RMSE
703	7.8893	0.9443	0.0000
600	6.9730	1.0684	7.8484
400	4.7791	1.5589	14.5699
300	3.3554	2.2203	17.3362
200	2.0844	3.5742	19.1992
100	0.9659	7.7130	20.5086
10	0.0897	83.0329	23.0602

Tablica 5: Rezultati kompresije za različite vrijednosti ranga (r)

Kada se kompresija primjenjuje selektivno na različite RGB kanale, pokazalo se da plavi kanal najbolje reagira na kompresiju. Ovo može biti zbog toga što plavi kanal često nosi manje informacija u usporedbi s crvenim i zelenim kanalima, što omogućava veću kompresiju bez značajnog gubitka kvalitete slike. Na plavom kanalu, kompresija stoga može postići bolje omjere između smanjenja veličine i očuvanja vizualne kvalitete slike, dok su crveni i zeleni kanali obično podložniji većim gubicima kvalitete pri sličnim razinama kompresije.



(a) Originalna slika



(b) $R = 10$



(c) $R = 100$



(d) $R = 200$



(e) $R = 300$



(f) $R = 400$



(g) $R = 600$



(h) $R = 700$

3.2.5 Selektivno kompresiramo dio fotografije

Ako se selektivno djeluje samo na desni, zamagljeni dio fotografije, kompresija može značajno smanjiti veličinu slike bez velikog gubitka kvalitete, jer su ti dijelovi slike manje informativni. Zamagljeni dijelovi sadrže manju količinu detalja, što znači da kompresija može ukloniti redundantne informacije dok gotovo ne utječe na percepciju slike.

Time, komprimirajući samo područje slike koje je zamagljeno, možemo drastično smanjiti veličinu slike bez da značajno narušimo kvalitetu, jer su vizualni podaci u tom području manje važni za krajnji dojam slike. Ovo je vrlo korisno za optimizaciju prostora za pohranu, osobito kada radimo s velikim slikama ili kada su neki dijelovi slike manje bitni za detalje ili prepoznavanje.

```
1 clc; clear; close all;
2
3 % Ucitavanje slike
4 img = imread('cat.jpg');
5 img = double(img);
6
7 % Ako je slika RGB, razdvoji kanale
8 if ndims(img) == 3
9     R = img(:,:,1);
10    G = img(:,:,2);
11    B = img(:,:,3);
12 else
13     R = img; % Ako je crno-bijela, koristi isti kanal
14     G = R;
15     B = R;
16 end
17
18 % Definiraj ROI (Region of Interest)
19 x1 = 450; % Po etna X koordinata
20 x2 = 700; % Krajnja X koordinata
21 y1 = 1; % Po etna Y koordinata
22 y2 = 463; % Krajnja Y koordinata
23
24 % Izdvoji samo ROI iz slike
25 R_ROI = R(y1:y2, x1:x2);
26 G_ROI = G(y1:y2, x1:x2);
27 B_ROI = B(y1:y2, x1:x2);
28
29 % Funkcija za Tucker dekompoziciju s ALS (Alternating Least Squares)
30 function [core, factors] = tucker_als(X, rank1, rank2)
31     max_iter = 50; % Maksimalni broj iteracija
32     tol = 1e-6; % Tolerancija za konvergenciju
33
34 % Inicializacija faktorskih matrica koriste i sluajne
```

```

vrijednosti
35 [U, ~] = qr(rand(size(X,1), rank1), 0);
36 [V, ~] = qr(rand(size(X,2), rank2), 0);
37
38 % Po etno jezgro
39 prev_core = U' * X * V;
40
41 for iter = 1:max_iter
42     core = U' * X * V;
43     U_new = X * V / core;
44     V_new = X' * U / core';
45
46 % Provjera konvergencije
47 if norm(core - prev_core, 'fro') < tol
48     break;
49 end
50 prev_core = core;
51 end
52
53 factors = {U, V}; % Spremanje faktorskih matrica
54 end

```

Označeni dio fotografije je onaj na koji ćemo djelovati.



(a) Originalna fotografija

Rekonstruirana slika (ROI kompresija)



(b) Selektirani dio fotografije (RGB)

Rekonstruirana slika (ROI kompresija)



(c) $R = 1$, blue kanal

Rekonstruirana slika (ROI kompresija)



(d) $R = 1$, red kanal



(a) Originalna fotografija



(b) $R = 10$, green kanal



(c) $R = 100$, red kanal



(d) $R = 500$, red kanal



(a) $R = 10$, blue kanal



(b) $R = 100$, blue kanal

r	Original ROI Size	Compressed ROI Size	Compression Ratio (ROI)	RMSE (ROI)
B Selektivno				
1	2723.7422 KB	5.5859 KB	487.6070	40.8898
10	2723.7422 KB	56.5625 KB	48.1546	18.2611
100	2723.7422 KB	635.9375 KB	4.2830	11.4765
G Selektivno				
1	2723.7422 KB	5.5859 KB	487.6070	99.2104
10	2723.7422 KB	56.5625 KB	48.1546	41.1099
R Selektivno				
1	2723.7422 KB	5.5859 KB	487.6070	46.4599
100	2723.7422 KB	635.9375 KB	4.2830	11.7639

Tablica 6: Podaci za selektivnu kompresiju u različitim kanalima

4 Literatura

Literatura

- [1] Tensor Decomposition Learning for Compression of Multidimensional Signals, by Anastasia Aidini, Grigorios Tsagkatakis and Panagiotis Tsakalides
- [2] Naš GitHub direktorij <https://github.com/NikolaKasnar/Tensor-image-compression>