Utilizarea algoritmului DVS pentru compresia imaginilor color

Panțu Mihnea-Andrei
Toma Alexandru-Ionel
322AC

Introducere

- Am dezvoltat aceasta aplicație pentru nevoia de a reduce memoria ocupata de poze.
- Am construit aplicația astfel încât sa funcționeze pe o multitudine de extensii de nume de fișier.
- Se folosește de conceptul DVS (descompunerea unei valori singulare) pentru a reconstrui imaginea.

Ce face aplicația?

- Prima dată in aplicație introducem numele pozei.
- Apoi va apărea un slider pe care avem valorile singulare.
- Îl putem modifica astfel maxim stânga avem o singura valoarea singulară iar la maxim dreapta avem toate valorile singulare.
- Mai jos va apărea imaginea reconstruită conform cu numărul de valori singulare specificate.

Testare

- Am testat aplicația si funcționează corespunzător pe următoarele extensii de nume de fișier uzuale:
- .jpg
- .jfif
- .gif
- .png

Noțiuni teoretice

- Din punct de vedere conceptual, DVS reduce într-o maniera eficientă sisteme cu date masive la probleme mai mici, eliminând surplusul inutil de informație si reținând datele esențiale.
- Avantajul oferit de DVS constă în flexibilitatea sa, deoarece poate acționa pe orice matrice.

Cum folosim DVS?

- Acum vom explora cum să aplicăm descompunerea valorii singulare a unei matrice la problema compresiei imaginii. DVS descompune o matrice dreptunghiulară A în trei părți.
- U — matricea vectorilor singulari la stânga în coloane.
- Σ — matrice diagonală cu valori singulare.
- V — matricea vectorilor singulari la dreapta în coloane.

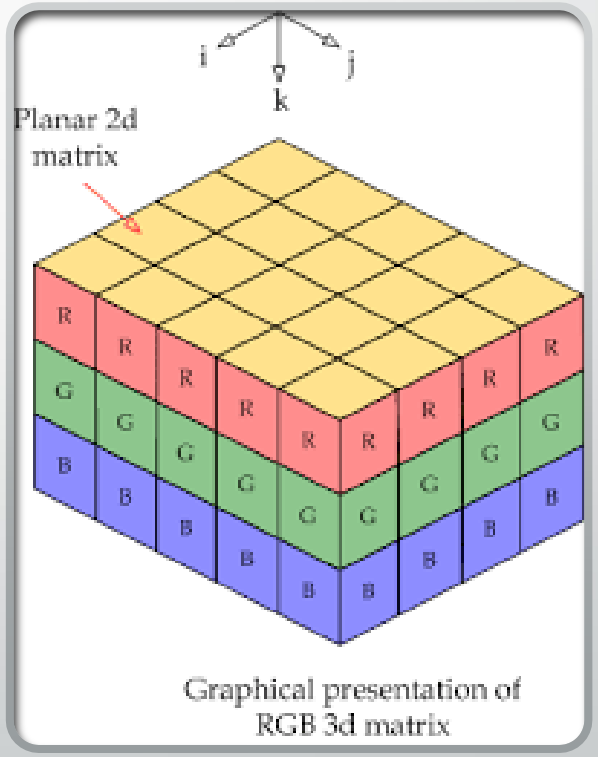
$$A = U\Sigma V^T$$

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T$$

- Aici σ1,σ2,σ3... sunt valori singulare. u1,u2,u3... și v1,v2,v3... sunt vectori singulari la stânga și respectiv la dreapta.
- O matrice de rang r va avea r termeni de felul acesta.

Descrierea aplicației

- Prin intermediul unei interfețe, utilizatorul va introduce de la tastatură numele imaginii sale în formatul: 'nume_imagine.extensie'
- Ulterior, imagini sale îi va fi atribuită o matrice 3D, matrice în care vor fi stocate numerele canalelor de imagine prezente în imaginea inițială.
- Modelul de culoare RGB (red-green-blue) este un model aditiv de culoare în care cele 3 culori sunt amestecate pentru a produce o gamă largă de alte culori.
- Principalul scop al paletei de culori RGB este de a reprezenta imagini pe sistemele electronice, precum televizoare, calculatoare, telefoane, tablete,etc.



Descrierea aplicației

- Într-un fisier obișnuit de imagine RGB de 24 de biți, imaginile sunt în esență realizate din 3 fișiere mai mici, fiecare a cate 8 biți.
- Aceste fișiere mai mici reprezintă canalele imaginii și au o gamă de 256 de valori, ce variază de la 0 la 255.
- Fiecare canal reprezintă cantitatea de lumină pe care fiecare culoare izolată trebuie să o combine pentru a crea imaginea digitală.
- Aceste combinații de culori unice, de 256 de culori, combină informațiile de culoare pentru a crea milioane de nuanțe noi de culori.

```
In [31]: np.amax(pic)
Out[31]: 255

In [32]: np.amin(pic)
Out[32]: 0
```

Descrierea aplicației

- Revenind la matricea în care vom stoca numerele canalelor de culoare, apelând comanda **np.shape(numele imaginii)** putem vedea câți pixeli ocupă imaginea respectivă.
- Înmulțind primele două valori ale funcției obținem numărul de pixeli ai imaginii.
- Matricea va avea formatul afișat, iar pe linii se regăsește informația asociată fiecărui pixel în valori ce reprezintă componentele canalelor RGB (roșu-verde-albastru).

```
In [20]: np.shape(pic)
Out[20]: (667, 1000, 3)
```

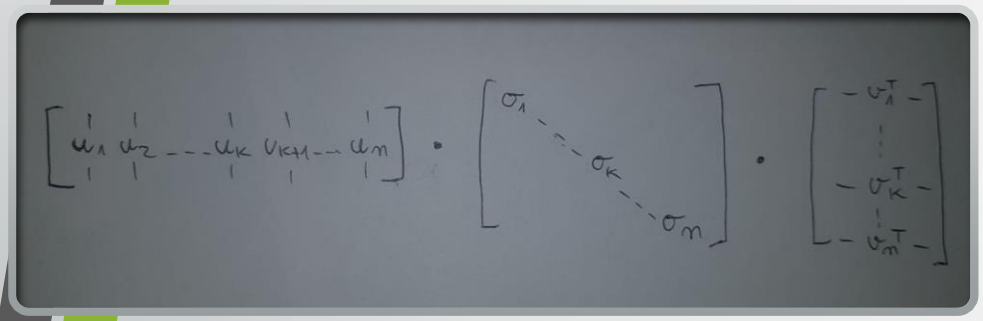
```
In [26]: pic
Out[26]: array([[[226, 204, 190],
                 [226, 204, 190],
                 [226, 204, 190],
                 ...,
                 [135, 115,  52],
                 [134, 114,  51],
                 [133, 113,  52]],

                [[222, 200, 186],
                 [222, 200, 186],
                 [222, 200, 186],
                 ...,
                 [136, 116,  53]
```

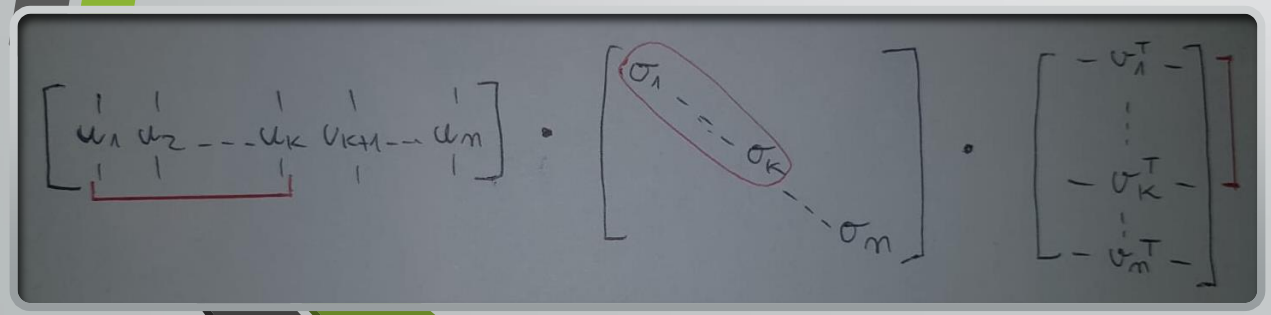
Descrierea aplicației

- Ulterior am stocat valorile componentelor modelului RGB în 3 matrice, câte una pentru fiecare culoare.
- Vom aplica asupra fiecărei din cele 3 matrice algoritmul DVS și vom stoca rezultatul funcției **np.linalg.svd(nume_imagine)** în alte trei matrice U,S si V, U fiind matricea ortogonală, coloanele ei numindu-se vectori singulari la stânga, S este un vector de valori singulare ce va fi transformat în matrice diagonală ,V matrice ale cărei coloane se numesc și vectori singulari la dreapta.
- Am creat apoi o funcție de compresie care primește ca parametru numărul de valori singulare pe care vrem să le folosim pentru a reconstrui imaginea.

Descrierea aplicației



- Vom reconstrui treptat imaginea. Vom obține pe rând trei noi matrice, câte una pentru fiecare din cele 3 culori ale modelului RGB.
- Notând cu k numărul de valori singulare pe care vrem să le utilizăm, vom înmulți matricea ortgonală U din care vom folosi primele k coloane, cu matricea diagonală S, din care vom folosi primele k valori de pe diagonala principală și înmulțim mai departe cu transpusa matricei V din care vom folosi primele k linii. (*accesând liniile matricei V este echivalent cu accesarea coloanelor matricei V transpus).
- Rezultatul va fi stocat pe rand în fiecare din cele trei matrice corespunzătoare tripletului RGB.



Descrierea aplicației

- Mai departe, vom reasambla cele 3 matrice bidimensionale într-o nouă matrice 3D folosind funcția **np.stack**.
- În noua matrice tridimensională vom avea din nou pe fiecare linie noile valori ale modelului RGB în ordinea componentelor rosu, verde, albastru, asociate fiecărui pixel din care va fi compusă noua imagine compresată.
- Am extras ulterior numărul de linii și coloane ale noii matrice pentru a determina rata compresiei.

```
r,c = np.shape(pic_red) #Returnam dimensiunile m

compress_ratio = (100 * k * (r + c + 1))/(r * c)
```

Descrierea aplicației

- În final, am introdus un slider interactiv cu ajutorul căruia putem selecta câte valori singulare vrem sa utilizăm in reconstruirea imaginii.
- Slider-ul merge de la 1 pana la numarul maxim de valori singulare ale matricei asociate imaginii.
- Cu ajutorul funcției **plt.imshow(nume_matrice_finala)** am afișat noua imagine rezultată din aplicarea compresiei DVS.