

7η Εργαστηριακή Άσκηση - Ιδιάζουσα Παραγοντοποίηση (SVD)

Για κάθε πίνακα $A \in \mathbb{R}^{m \times n}$ ($m \geq n$), υπάρχουν ορθογώνιοι πίνακες $U \in \mathbb{R}^{m \times m}$ και $V \in \mathbb{R}^{n \times n}$ τέτοιοι ώστε

$$A = USV^T = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \end{pmatrix} \begin{pmatrix} \sigma_1 & & & 0 & \cdots & 0 \\ & \ddots & & & & \\ & & \sigma_n & & & \\ 0 & & & 0 & & \\ \vdots & & & & \ddots & \\ 0 & & & & & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_n^T \end{pmatrix}$$

Η παραπάνω ισότητα ονομάζεται **ιδιάζουσα παραγοντοποίηση (SVD - Singular Value Decomposition)** του πίνακα A. Ειδικότερα:

- Οι στήλες $\mathbf{u}_i, i = 1, \dots, m$ του U είναι ορθοκανονικά ιδιοδιανύσματα του $AA^T \in \mathbb{R}^{m \times m}$.
- Οι γραμμές $\mathbf{v}_i^T, i = 1, \dots, n$ του V^T είναι ορθοκανονικά ιδιοδιανύσματα του $A^T A \in \mathbb{R}^{n \times n}$.
- Τα διαγώνια στοιχεία του $S \in \mathbb{R}^{m \times n}$ είναι οι ιδιάζουσες τιμές $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.
- Αν $r = \text{rank}(A) \leq n$, τότε ισχύει η ισότητα:

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T, \quad (1)$$

Μία δεύτερη έκδοση της SVD είναι γνωστή ως ``SVD οικονομικού μεγέθους'', η οποία εξοικονομεί μνήμη στον υπολογιστή. Στην περίπτωση αυτή υπολογίζονται μόνο οι πρώτες n στήλες του U και οι πρώτες n γραμμές του S, ενώ ο V παραμένει πάντοτε διάστασης n σε αμφότερες τις περιπτώσεις, δηλαδή

$$A = \hat{U} \hat{S} V^T = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{pmatrix} \begin{pmatrix} \sigma_1 & \cdots & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_n^T \end{pmatrix}$$

Το Matlab εκτελεί την πλήρη ιδιάζουσα παραγοντοποίηση ενός πίνακα A με την εντολή

`[U,S,V]=svd(A)` και την οικονομική έκδοσή της με την εντολή `[U,S,V]=svd(A,0)`.

Το άθροισμα (1) μπορούμε να το υπολογίσουμε με την εντολή:

$$U(:,1:r)*S(1:r,1:r)*V(:,1:r)'$$

Table of Contents

Ερώτημα 1.....	2
Ερώτημα 2 - Εφαρμογή στη συμπίεση εικόνων.....	4
Μονοχρωματικές εικόνες (αποχρώσεις του γκρι) – gray scale images.....	5

Ερώτημα 1

Ορίζουμε τον 10x10 πίνακα Hilbert και υλοποιούμε την SVD.

```
A = hilb(10)
```

```
A = 10x10
    1.0000    0.5000    0.3333    0.2500    0.2000    0.1667    0.1429    0.1250 ...
    0.5000    0.3333    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111
    0.3333    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000
    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909
    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833
    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769
    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714
    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667
    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667    0.0625
    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667    0.0625    0.0588
```

```
[U,S,V] = svd(A)
```

```
U = 10x10
   -0.6995    0.6376    0.3034   -0.1055    0.0293   -0.0067    0.0012   -0.0002 ...
   -0.4260   -0.0704   -0.5996    0.5763   -0.3229    0.1275   -0.0377    0.0084
   -0.3170   -0.2336   -0.3786   -0.2019    0.5831   -0.4974    0.2552   -0.0889
   -0.2555   -0.2821   -0.1443   -0.4120    0.1696    0.4110   -0.5665    0.3569
   -0.2153   -0.2937    0.0275   -0.3513   -0.2295    0.3633    0.2251   -0.5706
   -0.1866   -0.2910    0.1469   -0.1954   -0.3748   -0.0487    0.4263    0.1341
   -0.1649   -0.2825    0.2291   -0.0194   -0.3094   -0.3432    0.0294    0.4469
   -0.1480   -0.2717    0.2856    0.1472   -0.1121   -0.3529   -0.3719   -0.0615
   -0.1343   -0.2603    0.3243    0.2942    0.1542   -0.0758   -0.3265   -0.4947
   -0.1230   -0.2490    0.3504    0.4195    0.4482    0.4301    0.3665    0.2699

S = 10x10
    1.7519         0         0         0         0         0         0         0 ...
         0    0.3429         0         0         0         0         0         0
         0         0    0.0357         0         0         0         0         0
         0         0         0    0.0025         0         0         0         0
         0         0         0         0    0.0001         0         0         0
         0         0         0         0         0    0.0000         0         0
         0         0         0         0         0         0    0.0000         0
         0         0         0         0         0         0         0    0.0000
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0

V = 10x10
   -0.6995    0.6376    0.3034   -0.1055    0.0293   -0.0067    0.0012   -0.0002 ...
   -0.4260   -0.0704   -0.5996    0.5763   -0.3229    0.1275   -0.0377    0.0084
   -0.3170   -0.2336   -0.3786   -0.2019    0.5831   -0.4974    0.2552   -0.0889
   -0.2555   -0.2821   -0.1443   -0.4120    0.1696    0.4110   -0.5665    0.3569
   -0.2153   -0.2937    0.0275   -0.3513   -0.2295    0.3633    0.2251   -0.5706
   -0.1866   -0.2910    0.1469   -0.1954   -0.3748   -0.0487    0.4263    0.1341
   -0.1649   -0.2825    0.2291   -0.0194   -0.3094   -0.3432    0.0294    0.4469
   -0.1480   -0.2717    0.2856    0.1472   -0.1121   -0.3529   -0.3719   -0.0615
   -0.1343   -0.2603    0.3243    0.2942    0.1542   -0.0758   -0.3265   -0.4947
   -0.1230   -0.2490    0.3504    0.4195    0.4482    0.4301    0.3665    0.2699
```

Υπολογίζουμε τον βαθμού 1 πίνακα $A_1 = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ και τη νόρμα Frobenius $\|A - A_1\|_F$

```
A1=U(:,1)*S(1,1)*V(:,1)'
```

```
A1 = 10x10
    0.8573    0.5221    0.3885    0.3131    0.2638    0.2287    0.2021    0.1814 ...
    0.5221    0.3179    0.2366    0.1907    0.1607    0.1392    0.1231    0.1104
    0.3885    0.2366    0.1760    0.1419    0.1195    0.1036    0.0916    0.0822
    0.3131    0.1907    0.1419    0.1144    0.0964    0.0835    0.0738    0.0662
    0.2638    0.1607    0.1195    0.0964    0.0812    0.0704    0.0622    0.0558
    0.2287    0.1392    0.1036    0.0835    0.0704    0.0610    0.0539    0.0484
    0.2021    0.1231    0.0916    0.0738    0.0622    0.0539    0.0477    0.0428
    0.1814    0.1104    0.0822    0.0662    0.0558    0.0484    0.0428    0.0384
    0.1646    0.1002    0.0746    0.0601    0.0507    0.0439    0.0388    0.0348
    0.1508    0.0918    0.0683    0.0551    0.0464    0.0402    0.0355    0.0319
```

```
error1 = norm(A-A1, 'fro')
```

```
error1 = 0.3448
```

Υπολογίζουμε τον βαθμού 2 πίνακα $A_2 = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T$ και τη νόρμα Frobenius $\|A - A_2\|_F$

```
A2 = U(:,1:2)*S(1:2,1:2)*V(:,1:2)'
```

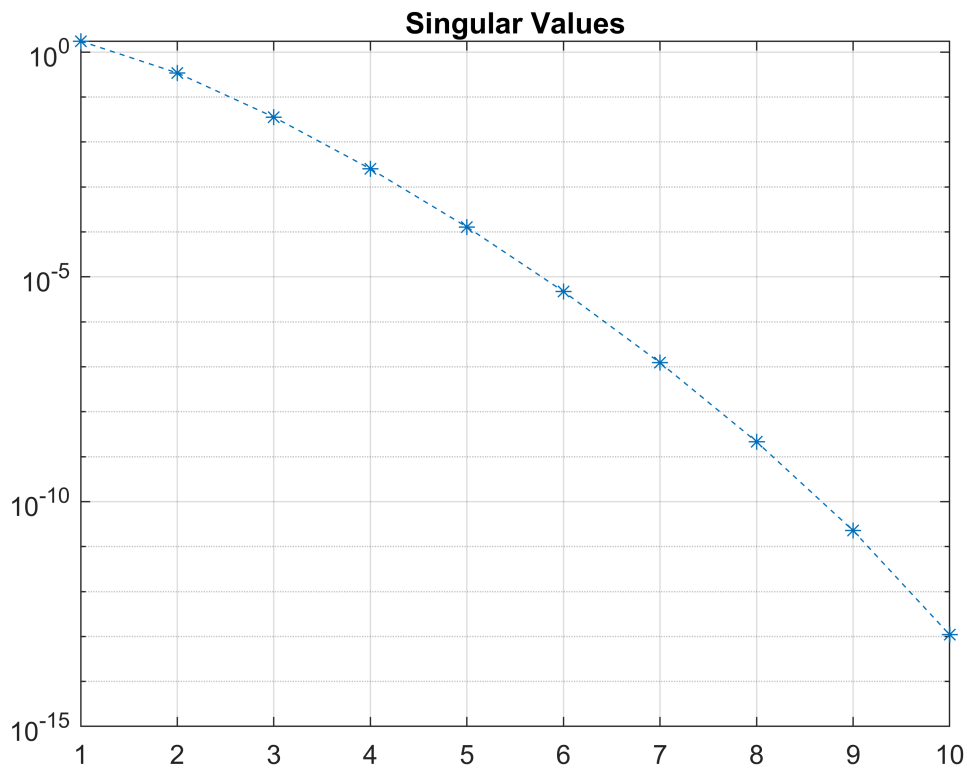
```
A2 = 10x10
    0.9967    0.5067    0.3374    0.2515    0.1996    0.1650    0.1404    0.1219 ...
    0.5067    0.3196    0.2422    0.1975    0.1678    0.1463    0.1299    0.1170
    0.3374    0.2422    0.1947    0.1645    0.1431    0.1269    0.1142    0.1039
    0.2515    0.1975    0.1645    0.1417    0.1248    0.1117    0.1012    0.0925
    0.1996    0.1678    0.1431    0.1248    0.1108    0.0997    0.0907    0.0832
    0.1650    0.1463    0.1269    0.1117    0.0997    0.0900    0.0821    0.0755
    0.1404    0.1299    0.1142    0.1012    0.0907    0.0821    0.0750    0.0691
    0.1219    0.1170    0.1039    0.0925    0.0832    0.0755    0.0691    0.0637
    0.1077    0.1065    0.0954    0.0853    0.0769    0.0699    0.0640    0.0591
    0.0963    0.0978    0.0883    0.0792    0.0715    0.0651    0.0597    0.0551
```

```
error2 = norm(A-A2, 'fro')
```

```
error2 = 0.0358
```

Γραφική παράσταση των ιδιζουσών τιμών του πίνακα A σε ημι-λογαριθμική κλίμακα.

```
semilogy(diag(S), '*--')
grid on, title('Singular Values')
```



Ερώτημα 2 - Εφαρμογή στη συμπίεση εικόνων

Μια σημαντική εφαρμογή της SVD βρίσκεται στη συμπίεση εικόνων. Μια εικόνα μπορεί να αναπαρασταθεί από έναν πίνακα δεδομένων με πληροφορίες για την απόχρωση της. Για παράδειγμα, οι εικόνες με χρωματισμό σε κλίμακα του γκρι αποτελούνται από μια ορθογώνια διάταξη pixels, το χρώμα καθενός από τα οποία συμβολίζεται με έναν ακέραιο αριθμό μεταξύ 0 (μαύρο) και 255 (άσπρο), δίνοντας

συνολικά $2^8 = 256$ διαφορετικές αποχρώσεις του γκρι για κάθε ένα pixel. Οι έγχρωμες εικόνες (μοντέλο RGB) αντιπροσωπεύονται από τρεις τέτοιους πίνακες, έναν για κάθε μια χρωματική συνιστώσα κόκκινου,

πράσινου και μπλε, δίνοντας έτσι συνολικά $(2^8)^3 = 2^{24} = 16.777.216$ διαφορετικές αποχρώσεις για καθένα pixel.

Για παράδειγμα, ο συνδυασμός (0, 0, 0) αντιστοιχεί στο μαύρο, ενώ ο συνδυασμός (255, 255, 255) αντιστοιχεί στο λευκό. Το κόκκινο καθορίζεται με την τριάδα αριθμών (255, 0, 0), το πράσινο με (0, 255, 0) και το μπλε με (0, 0, 255). Οι ενδιάμεσες αποχρώσεις δημιουργούνται με διαφορετικούς συνδυασμούς των τριών αριθμών.



Μια απίστευτη εικόνα του νεφελώματος Tarantula που βρίσκεται περίπου 160.000 έτη φωτός μακριά μας και προέρχεται από το τηλεσκόπιο Hubble της NASA. Η αραχνοειδής εμφάνιση του νεφελώματος είναι υπεύθυνη για το δημοφιλές όνομά του και αποτελεί ίσως το εντυπωσιακότερο του Μεγάλου Μαγγελανικού Νέφους, ενός δορυφορικού νανογαλαξία που είναι ο τρίτος πλησιέστερος σε εμάς γαλαξίας (14.000 έτη φωτός μακριά).

Το Matlab παρέχει διάφορες built-in συναρτήσεις επεξεργασίας εικόνας:

- `imread`: διαβάζει τις τιμές των pixels της εικόνας και επιστρέφει ένα πίνακα με όλες αυτές τις τιμές
- `imshow`: προβολή της εικόνας στην οθόνη
- `rgb2gray`: αλλάζει μια έγχρωμη εικόνα σε ασπρόμαυρη
- `im2double`: μετατρέπει την εικόνα σε αριθμό κινητής υποδιαστολής διπλής ακρίβειας στο διάστημα (0,1) για την εκτέλεση αριθμητικών υπολογισμών.
- `whos`: βλέπουμε τι ακριβώς περιέχει το αρχείο εικόνας

Μονοχρωματικές εικόνες (αποχρώσεις του γκρι) – gray scale images

- Δισδιάστατοι πίνακες [MxN] - Οι τιμές των pixels είναι ακέραιοι στο πεδίο [0,255].

```
% Read image file
RGB=imread('TarantulaNebula.jpg');
whos RGB
```

Name	Size	Bytes	Class	Attributes
RGB	567x630x3	1071630	uint8	

```
% Plot true color image
figure('Position',[50 50 800 800])
```

```

subplot(3,2,1), imshow(RGB)
title ('Original Color Image')

% Convert color image to gray scale
grayimage = rgb2gray(RGB);
whos grayimage

```

Name	Size	Bytes	Class	Attributes
grayimage	567x630	357210	uint8	

```

% Convert image to double
BN=im2double(grayimage);

% Plot true gray image
subplot(3,2,2), imshow(BN)
title ('Original Gray Image')

% Perform svd
[U,S,V]=svd(BN);

% Plot rank 5, 10, 20, 50 approximations
k = 5;
BN5 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
subplot(3,2,3), imshow(BN5), title ('Rank 5')
k = 10;
BN10 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
subplot(3,2,4), imshow(BN10), title ('Rank 10')
k = 20;
BN20 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
subplot(3,2,5), imshow(BN20), title ('Rank 20')
k = 50;
BN50 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
subplot(3,2,6), imshow(BN50), title ('Rank 50')

```

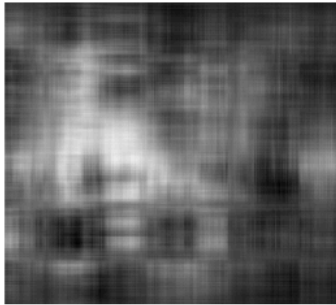
Original Color Image



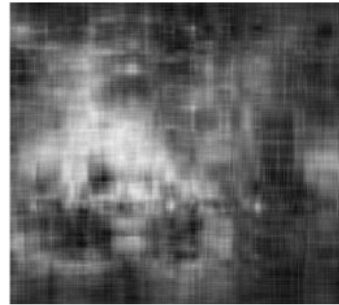
Original Gray Image



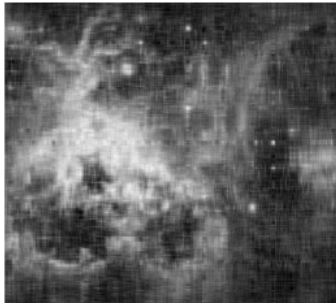
Rank 5



Rank 10



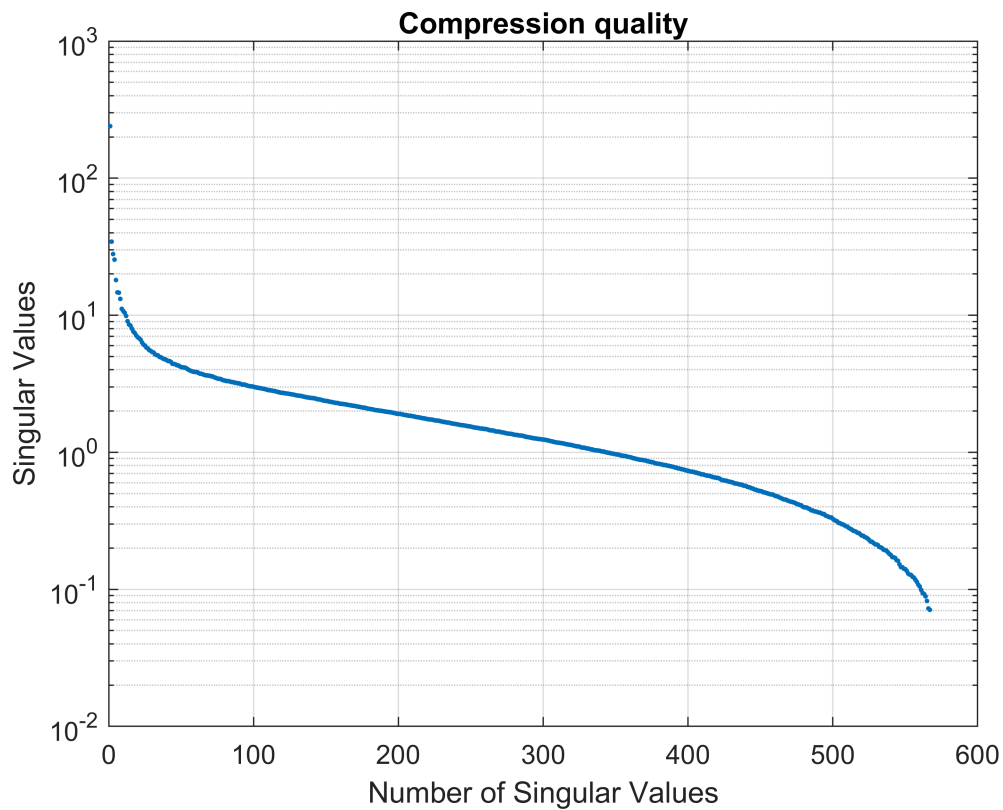
Rank 20



Rank 50



```
% Plot singular values
figure, semilogy(diag(S),'.')
grid on
title('Compression quality');
xlabel('Number of Singular Values');
ylabel('Singular Values');
```

Έγχρωμες εικόνες – RGB images

- Τρισδιάστατοι πίνακες $[M \times N \times 3]$. Οι τιμές των pixels είναι συνήθως ακέραιοι στο πεδίο $[0 \ 255]$. Κάθε pixel ορίζεται από τρεις τέτοιους αριθμούς (R,G,B).

```
% Read image file
A = imread('peppers.tiff');
% A = imread('fl2.jpg');
whos A
```

Name	Size	Bytes	Class	Attributes
A	512x512x3	786432	uint8	

```
% Convert A to double
A = im2double(A);

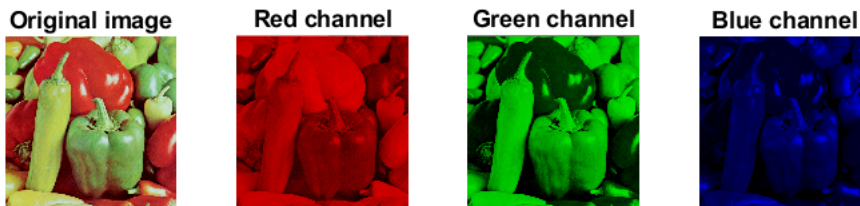
% Extract Red channel
R = A(:, :, 1);
% Extract Green channel
G = A(:, :, 2);
% Extract Blue channel
B = A(:, :, 3);
```



```

% Plot color versions of the individual color channels.
O = zeros(size(A,1),size(A,2));
subplot(1,4,1), imshow(cat(3, R, G, B)), title('Original image')
subplot(1,4,2), imshow(cat(3, R, O, O)), title('Red channel')
subplot(1,4,3), imshow(cat(3, O, G, O)), title('Green channel')
subplot(1,4,4), imshow(cat(3, O, O, B)), title('Blue channel')

```



```

% Perform svd
[UR,SR,VR] = svd(R);
[UG,SG,VG] = svd(G);
[UB,SB,VB] = svd(B);

% Plot true image and rank 3, 5, 10 approximations
figure('Position',[50 50 800 800])
subplot(2,2,1)
imshow(A); title ('Original Image')
subplot(2,2,2)
k = 3;
Mk(:, :, 1) = UR(:,1:k)*SR(1:k, 1:k)* VR(:,1:k)';
Mk(:, :, 2) = UG(:,1:k)*SG(1:k, 1:k)* VG(:,1:k)';
Mk(:, :, 3) = UB(:,1:k)*SB(1:k, 1:k)* VB(:,1:k)';
imshow(Mk); title('Rank 3')
subplot(2,2,3)
k = 5;
Mk(:, :, 1) = UR(:,1:k)*SR(1:k, 1:k)* VR(:,1:k)';
Mk(:, :, 2) = UG(:,1:k)*SG(1:k, 1:k)* VG(:,1:k)';

```

```

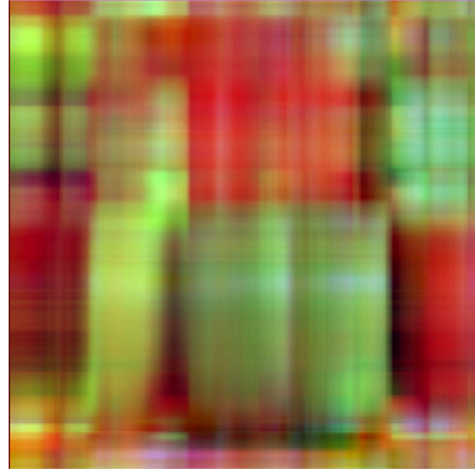
Mk(:, :, 3) = UB(:,1:k)*SB(1:k, 1:k)* VB(:,1:k)';
imshow(Mk); title('Rank 5')
subplot(2,2,4)
k = 10;
Mk(:, :, 1) = UR(:,1:k)*SR(1:k, 1:k)* VR(:,1:k)';
Mk(:, :, 2) = UG(:,1:k)*SG(1:k, 1:k)* VG(:,1:k)';
Mk(:, :, 3) = UB(:,1:k)*SB(1:k, 1:k)* VB(:,1:k)';
imshow(Mk); title('Rank 10')

```

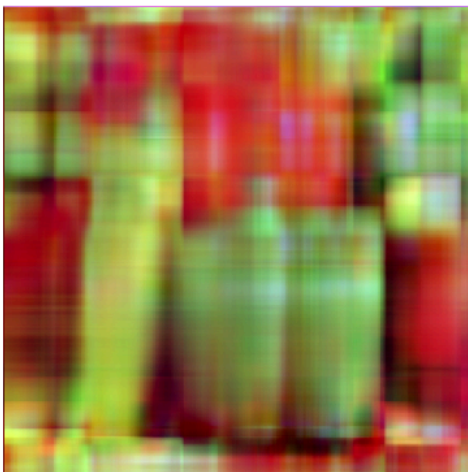
Original Image



Rank 3



Rank 5



Rank 10

