• Instructor: He Wang Email: he.wang@northeastern.edu

In this section, we will learn **singular value decomposition**(SVD) of a matrix. It is very useful in image compress and principal component analysis(PCA) in statistics.

Recall the spectral decomposition for symmetric matrices:

Theorem. [Spectral Decomposition for Symmetric Matrices]

A is an $m \times m$ **symmetric** matrix if an only if $A = VDV^{-1}$ such that D is diagonal and V is an orthogonal matrix.
Let $\lambda_1, \dots, \lambda_m$ be the diagonal entries of D, and let $\vec{v}_1, \dots, \vec{v}_m$ be the column vectors of V. Then $A = VDV^T$ can be written as

$$A = \lambda_1\Big(\vec{v}_1 \cdot (\vec{v}_1)^T\Big) + \cdots + \lambda_n\Big(\vec{v}_n \cdot (\vec{v}_m)^T\Big)$$

We want to find a similar decomposition for **any** $n \times m$ matrix M. For example, $M = \begin{bmatrix} 1 & 4 \\ 2 & 2 \\ 2 & -4 \end{bmatrix}$

First, we know that $A := M^T M$ is an $m \times m$ symmetric matrix. We already know that the eigenvalues of A are non-negative.

Definition. Singular Values

Let $\lambda_1, \lambda_2, \dots, \lambda_m$ be the m real eigenvalues of the symmetric matrix $M^T M$, and ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$. The **singular values** of an $n \times m$ matrix M are

$$\sigma_i := \sqrt{\lambda_i}$$

Theorem.

Let M be an 2×2 invertible matrix. The image of M of the unit circle is an ellipse. The lengths of the semimajor and the semiminor axes of the ellipse are the singular values of M.

By the spectral theorem for symmetric matrix $A = M^T M$, let $A = VDV^{-1} = VDV^T$. Let $\vec{v}_1, \dots, \vec{v}_m$ be the (orthonormal) column vectors of V. (By H.W. 5.4.17, $\text{rank}(A) = \text{rank}(M)$)

Theorem.

If $\text{rank}(M) = r$, then $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and $\sigma_{r+1} = \cdots = \sigma_m = 0$.

Theorem.

1. $M\vec{v}_i \cdot M\vec{v}_j = 0$ for $i \neq j$.
2. $||M\vec{v}_i|| = \sigma_i$ for all $i = 1, 2, \dots, m$.
3. In particular, $M\vec{v}_i = 0$ for $i = r+1, \cdots, m$.

$$M\vec{v}_i \cdot M\vec{v}_j = (M\vec{v}_i)^T M\vec{v}_j = \vec{v}_i^T M^T M \vec{v}_j$$
$$= \vec{v}_i^T A \vec{v}_j$$
$$= \lambda_j \vec{v}_i^T \vec{v}_j = \lambda_j (\vec{v}_i \cdot \vec{v}_j)$$

Define the unit vectors for $i = 1, 2, \dots, r$.

$$\vec{u}_i := \frac{M\vec{v}_i}{||M\vec{v}_i||} = \frac{1}{\sigma_i} M\vec{v}_i$$

Then we have $Mv_i = \sigma_i \vec{u}_i$ for $i = 1, 2, \dots, r$. Together with $M\vec{v}_i = 0$ for $i = r+1, \cdots, m$, we can write this as a matrix multiplication, we have

$$MV = U\Sigma$$

Here, the $n \times n$ matrix U is an orthogonal matrix with above $\vec{u}_i$ for $i = 1, 2, \dots, r$ extended $u_{i+1}, \cdots, u_n$. The $n \times m$ matrix Σ has $\Sigma_{ii} = \sigma_i$ for $i = 1, 2, \dots, r$ and zeros otherwise.

Theorem. Singular Value Decomposition(SVD)

And $n \times m$ matrix M can be decomposed as

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

or as

$$M = \sigma_1 \vec{u}_1 \vec{v}_1^T + \cdots + \sigma_r \vec{u}_r \vec{v}_r^T$$

Example 1. Find an SVD decomposition for the matrix

$$M = \begin{bmatrix} 1 & 4 \\ 2 & 2 \\ 2 & -4 \end{bmatrix}$$

1. $A = M^TM = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 2 & -4 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & 2 \\ 2 & -4 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 36 \end{bmatrix}$

$\lambda_1 = 36$ and $\lambda_2 = 9$ are eigenvalues of $A = M^TM$

The singular values of M are $\sigma_1 = \sqrt{\lambda_1} = 6$ $\sigma_2 = \sqrt{\lambda_2} = 3$

2. For $\lambda_1 = 36$ $A - \lambda_1 I = \begin{bmatrix} -27 & 0 \\ 0 & 0 \end{bmatrix}$ $x_1 = 0$

E_{λ_1} has a basis $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

For $\lambda_2 = 9$, $A - \lambda_2 I = \begin{bmatrix} 0 & 0 \\ 0 & 27 \end{bmatrix}$ $x_2 = 0$. E_{λ_2} has a basis $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\vec{v}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\vec{v}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

3. $M\vec{v}_1 = \begin{bmatrix} 4 \\ 2 \\ -4 \end{bmatrix}$ $M\vec{v}_2 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$

$\vec{u}_1 = \frac{A\vec{v}_1}{\|A\vec{v}_1\|} = \frac{1}{6}\begin{bmatrix} 4 \\ 2 \\ -4 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 \\ 1 \\ -2 \end{bmatrix}$ $\vec{u}_2 = \frac{1}{3}\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$

We need to find unit $\vec{u}_3$ such that $\vec{u}_1 \cdot \vec{u}_3 = 0$ and $\vec{u}_2 \cdot \vec{u}_3 = 0$

$\vec{u}_3 = \frac{1}{3}\begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix}$

Solve $\begin{bmatrix} 2 & 1 & -2 \\ 1 & 2 & 2 \end{bmatrix}\vec{x} = \vec{0}$.

Then normalize.

4. So $\underset{3\times 2}{M} = \underset{3\times 3}{U}\,\underset{3\times 2}{\Sigma}\,\underset{2\times 2}{V^T}$

Here $U = \frac{1}{3}\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ -2 & 2 & 1 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 6 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix}$ $V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Example 2. Find an SVD decomposition for the matrix

$$M = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

(1) $A = M^TM = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

The eigenvalues of M^TM are $\lambda_1 = 3$, $\lambda_2 = 1$, $\lambda_3 = 0$

The singular values of M are $\sigma_1 = \sqrt{3}$, $\sigma_2 = 1$, $\sigma_3 = 0$.

(2). For $\lambda_1 = 3$ $A - \lambda_1 I = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -2 \end{bmatrix} \to \cdots \to \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{bmatrix}$ $x_1 = x_3$, $x_2 = 2x_3$

E_{λ_1} has a basis $\left\{ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \right\}$. E_{λ_2} has a basis $\left\{ \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \right\}$ E_{λ_3} has a basis $\left\{ \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \right\}$

So $\vec{v}_1 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ $\vec{v}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ $\vec{v}_3 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$

(3) $M\vec{v}_1 = \frac{1}{\sqrt{6}} \begin{bmatrix} 3 \\ 3 \end{bmatrix}$ $M\vec{v}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ $M\vec{v}_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\vec{u}_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$ $\vec{u}_2 = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$

(4) $M = U \Sigma V^T$

Here $U = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$ $\Sigma = \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ $V = \begin{bmatrix} 1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ 2/\sqrt{6} & 0 & -1/\sqrt{3} \\ 1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \end{bmatrix}$

Example 3.

$$M = \begin{bmatrix} 1 & -1 \\ 1 & 2 \\ -1 & 1 \end{bmatrix}$$

(1). Calculate M^TM and MM^T.

(2). Find all eigenvalues and an eigenbasis of M^TM.

(3). Find all eigenvalues and an eigenbasis of MM^T.

(4). Find an SVD decomposition for the matrix M. That $M = U\Sigma V^T$

(1) $M^TM = \begin{bmatrix} 3 & 0 \\ 0 & 6 \end{bmatrix}$

$MM^T = \begin{bmatrix} 2 & -1 & -2 \\ -1 & 5 & 1 \\ -2 & 1 & 2 \end{bmatrix}$

$U = \begin{bmatrix} -1/\sqrt{6} & 1/\sqrt{3} & 1/\sqrt{2} \\ 2/\sqrt{6} & 1/\sqrt{3} & 0 \\ 1/\sqrt{6} & -1/\sqrt{3} & 1/\sqrt{2} \end{bmatrix}$ $\Sigma = \begin{bmatrix} \sqrt{6} & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix}$

$V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

(2) Eigenvalues for M^TM are $\lambda_1 = 6$ $\lambda_2 = 3$

Eigenvalues for MM^T are $\lambda_1 = 6$ $\lambda_2 = 3$ $\lambda_3 = 0$

(3) Eigenbasis for MM^T are $\begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$ $\begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

Eigenbasis for M^TM are $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

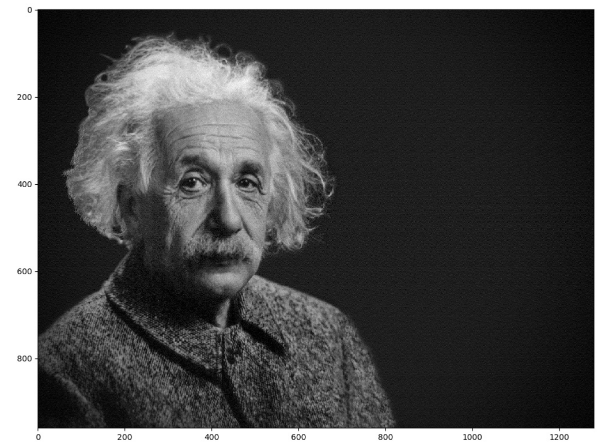
In this example, we use a trick to find U and V.

$M = U\Sigma V^T$. So $M^T = (U\Sigma V^T)^T = V\Sigma^T U^T$

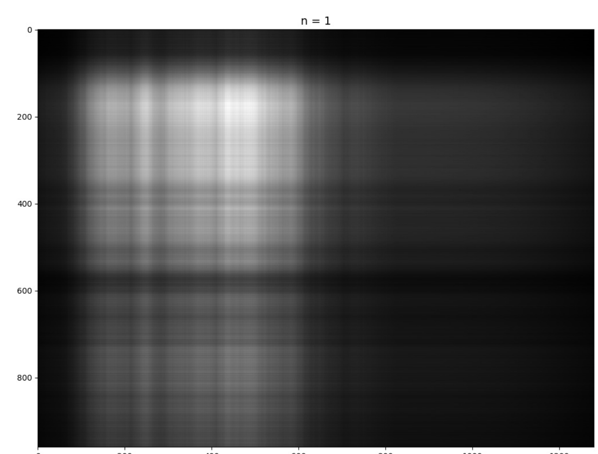
In the computation of SVD of $M = U\Sigma V^T$ we use M^TM. We can obtain V from it by the eigenbasis.

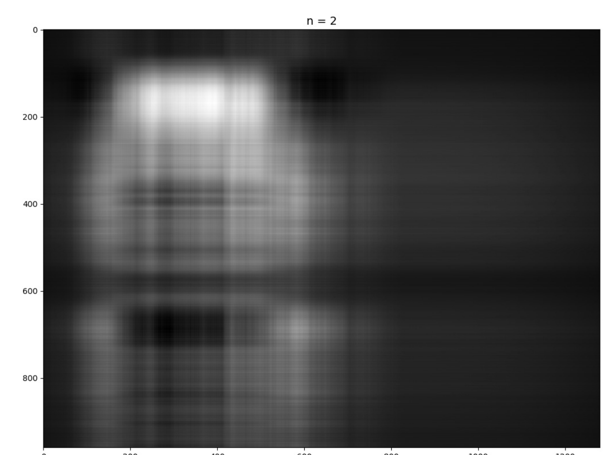
In the computation of SVD of $M^T = V\Sigma^T U^T$ we use MM^T. We can obtain U from it by the eigenbasis.

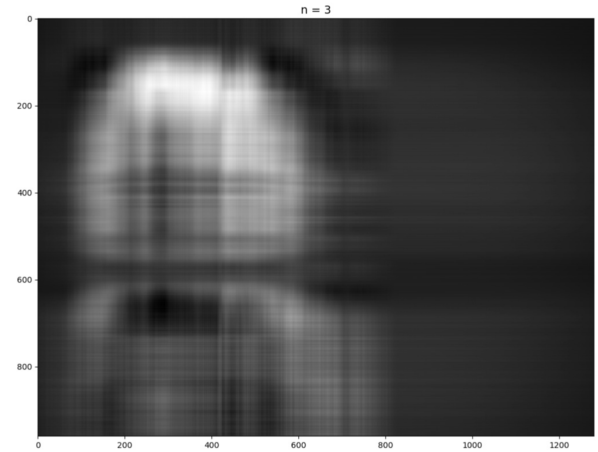
An digital image of size $m \times n$ is recorded as a matrix. SVD provides a method to reduce the size of the image by keeping a good quality. For example,

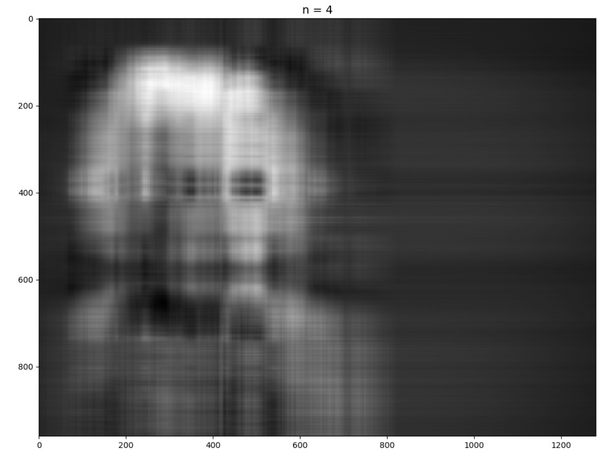


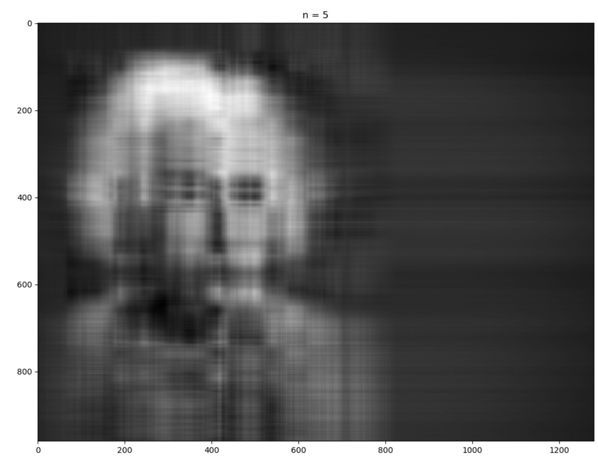
the original image is an 960×1280 matrix A. We can use Matlab or Python to find the SVD decomposition of the matrix and then reconstruct the approximation of the image. (Codes of Mathlab and Python in the end)

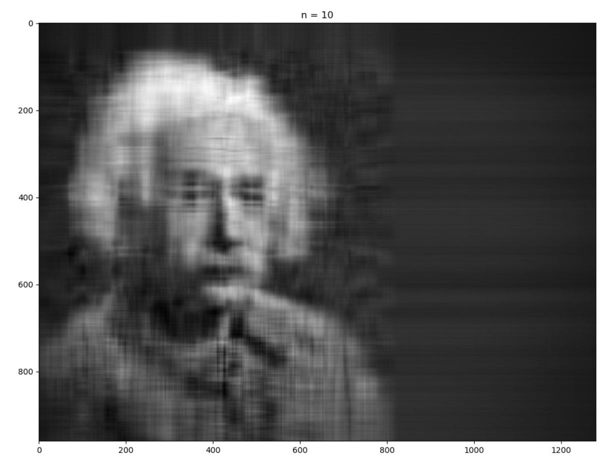


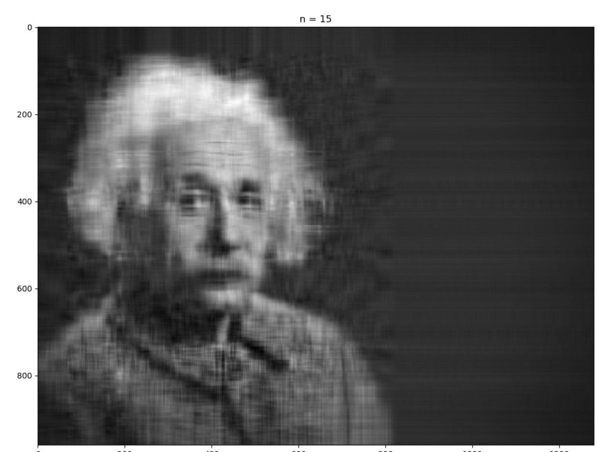


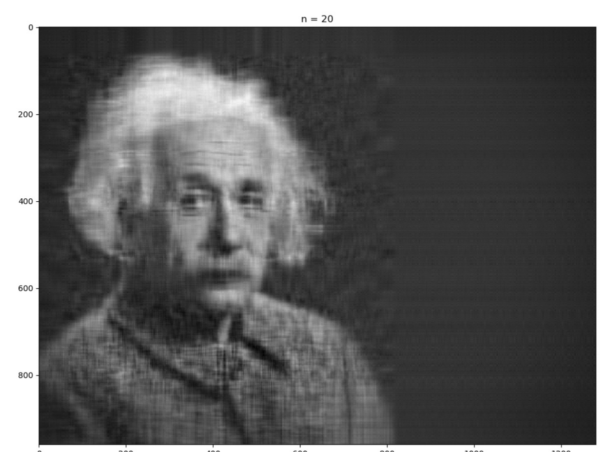

n = 3
0
200
400
600
800
0
200
400
600
800
1000
1200

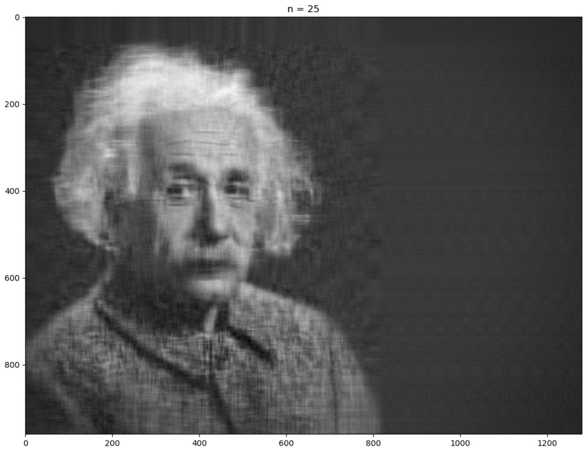

n = 4
0
200
400
600
800
0
200
400
600
800
1000
1200

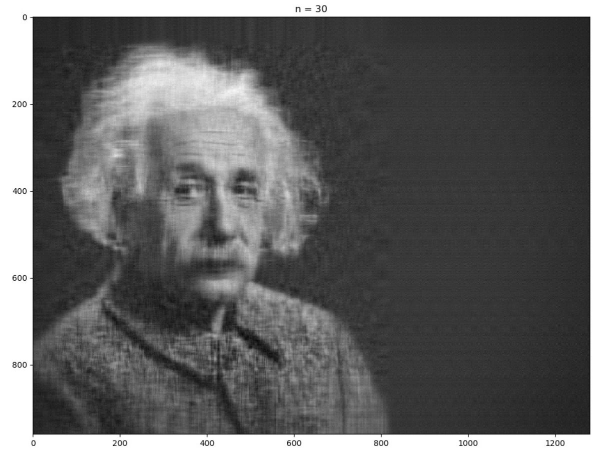

n = 5
0
200
400
600
800
0
200
400
600
800
1000
1200

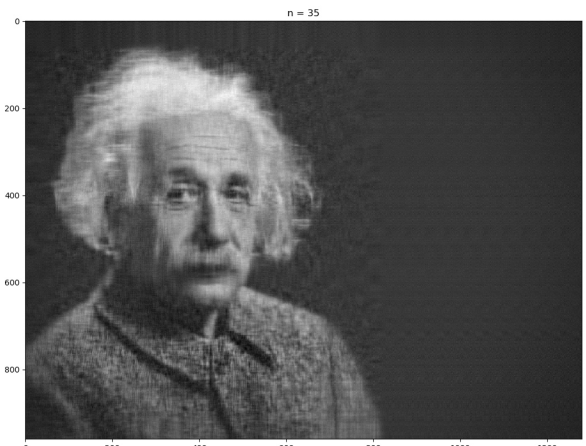

n = 10
0
200
400
600
800
0
200
400
600
800
1000
1200

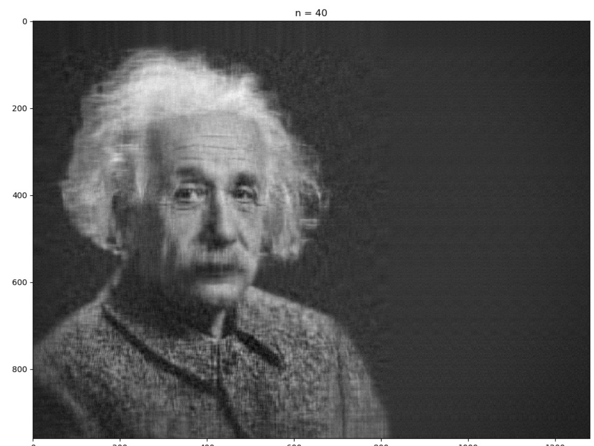

n = 15
0
200
400
600
800
0
200
400
600
800
1000
1200

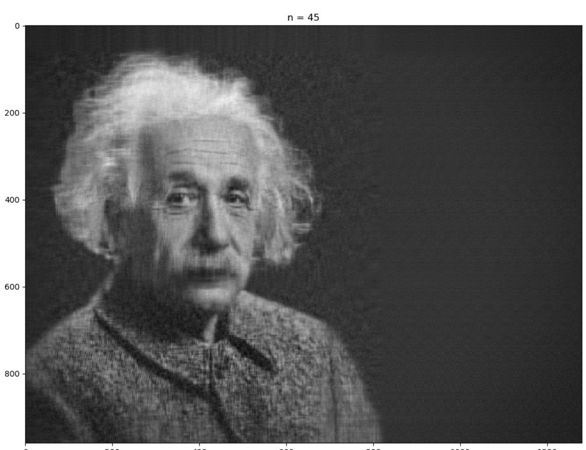

n = 20
0
200
400
600
800
0
200
400
600
800
1000
1200

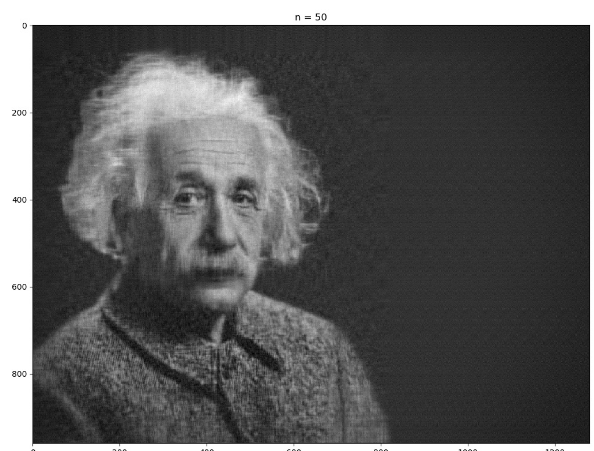

n = 25
0
200
400
600
800
0 200 400 600 800 1000 1200


n = 30
0
200
400
600
800
0 200 400 600 800 1000 1200

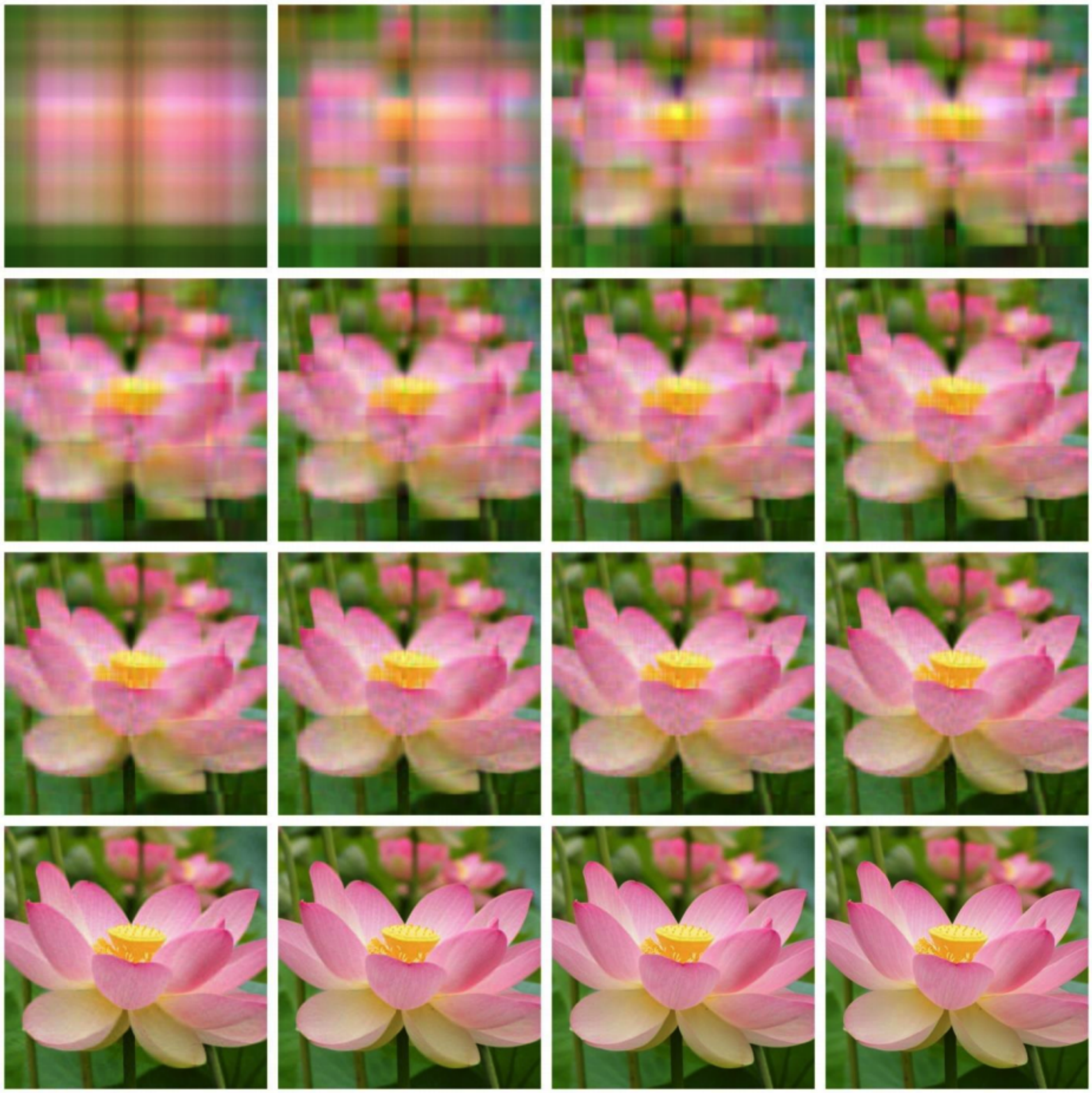

n = 35
0
200
400
600
800
0 200 400 600 800 1000 1200


n = 40
0
200
400
600
800
0 200 400 600 800 1000 1200


n = 45
0
200
400
600
800
0 200 400 600 800 1000 1200


n = 50
0
200
400
600
800
0 200 400 600 800 1000 1200

For a color picture of size 2000×2000, with $n = 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 50, 75, 100$ and the last picture is the original picture.



Code for the Albert Einstein picture compression:

1. Matlab code:

```matlab

%% import image
a=imread('einstein.jpg');
[m,n,d]=size(a);
kmax=25;
da=double(a);
U=zeros(m,m);S=zeros(m,n);V=zeros(n,n);e=zeros(kmax,d);cr=zeros(kmax,1);rmse=zeros(kmax,d);
%% SVD decomposition
for i=1:d
    [U(:,:,i),S(:,:,i),V(:,:,i)]=svd(da(:,:,i));
end
%% Reconstruction the image using the first k-th eignvalues.
for l=1:kmax
    k=2*l;
    ca=zeros(m,n,d);
    cr(k)=m*n/(k*(m+n+1));
    for i=1:d
        cai=zeros(m,n,d);
        [ca(:,:,i),cai(:,:,i)]=deal(U(:,1:k,i)*S(1:k,1:k,i)*V(:,1:k,i)');
        e(k,i)=S(k+1,k+1,i)/S(1,1,i);
        rmse(k,i)=sqrt(sum(sum(((da(:,:,i)-ca(:,:,i)).^2)))/(m*n));
    end
    imwrite(uint8(ca), sprintf('%dk.jpg', k));
end

%%
```

2. Python code:

```python
# import packages
get_ipython().run_line_magic('matplotlib', 'inline')
import matplotlib.pyplot as plt
import numpy as np
import time
from numpy import linalg as LA
from PIL import Image

# import image
img = Image.open('einstein.jpg')
imggray = img.convert('LA')
plt.figure(figsize=(12.8,9.6),dpi=100)
plt.imshow(imggray);


#convert the image data into a numpy matrix
imgmat = np.array(list(imggray.getdata(band=0)), float)
imgmat.shape = (imggray.size[1], imggray.size[0])
imgmat = np.matrix(imgmat)
plt.figure(figsize=(12.8,9.6),dpi=100)
```

```
plt.imshow(imgmat, cmap='gray');
plt.savefig('/Users/hewang/Dropbox/Sec8.3/AE.jpg', bbox_inchesstr='tight')

#SVD
U, sigma, V = np.linalg.svd(imgmat)

#Computing an approximation of the image using the first i-th eigenvalues
for i in range(1, 5):
    reconstimg = np.matrix(U[:, :i]) * np.diag(sigma[:i]) * np.matrix(V[:i, :])
    plt.figure(figsize=(12.8,9.6),dpi=100)
    plt.imshow(reconstimg, cmap='gray')
    title = "n = %s" % i
    plt.title(title, fontsize=14)
    plt.savefig('/Users/hewang/Dropbox/Sec8.3/AE'+str(i)+'.jpg', bbox_inchesstr='tight')
    plt.show()

for i in range(5, 51, 5):
    reconstimg = np.matrix(U[:, :i]) * np.diag(sigma[:i]) * np.matrix(V[:i, :])
    plt.figure(figsize=(12.8,9.6),dpi=100)
    plt.imshow(reconstimg, cmap='gray')
    title = "n = %s" % i
    plt.title(title)
    plt.savefig('/Users/hewang/Dropbox/Sec8.3/AE'+str(i)+'.jpg', bbox_inchesstr='tight')
    plt.show()
```