جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلـــــوم
قســـــــم الانــظـــمـــة الـــطـبـيـة الــذكـــــيــة

المحاضرة الثامنة

Applications of the Singular Value Decomposition

المادة : DSP
المرحلة : الثالثة
اسم الاستاذ: م.م. ريام ثائر احمد

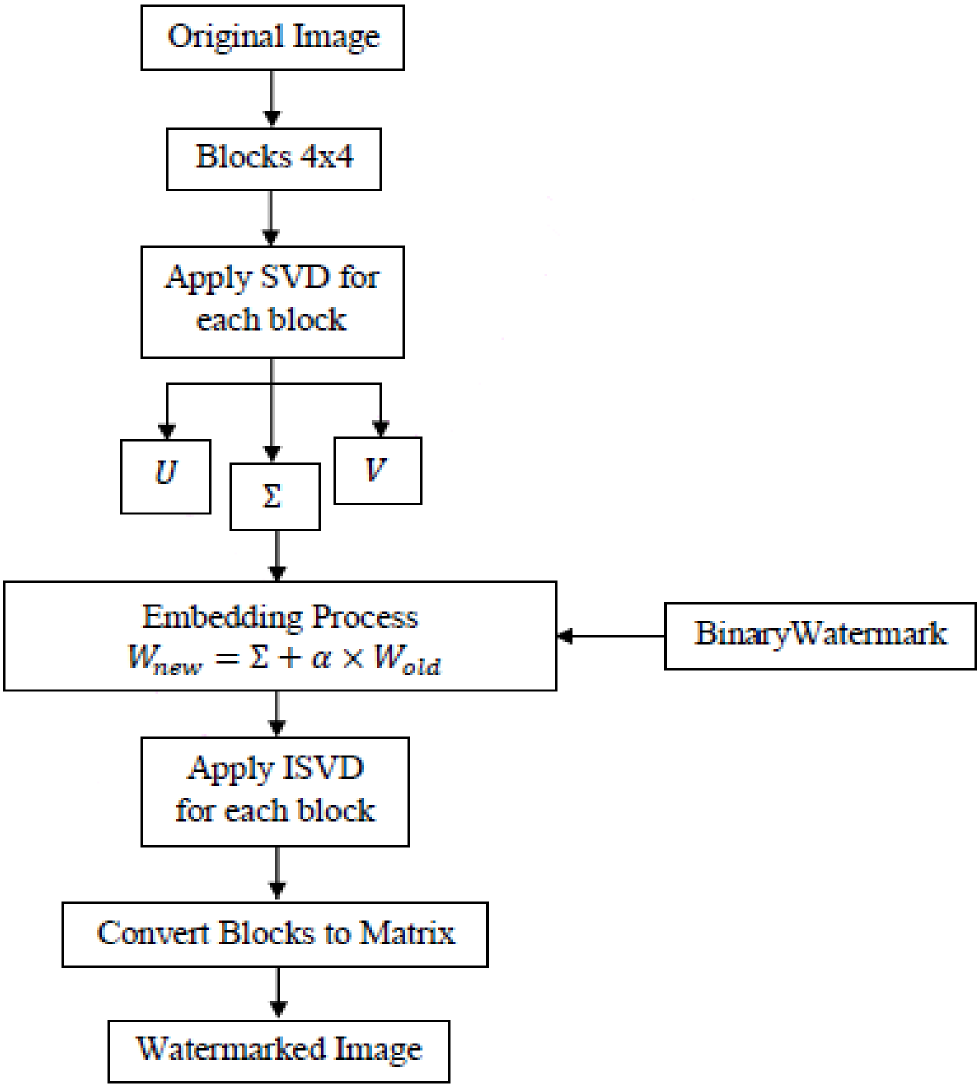
1. Some Applications of the Singular Value Decomposition (SVD) in DIP:

a- Security: Image watermarking
b- Image Compression
c- Image Denoising
d- Image Forensic
e- Solving of Image Similarity Puzzle.
f- In data mining
g- Background Subtraction.
h- Clustering and Classification.
i- Structure and Motion.
j- Restoration, Denoising, and Deblurring.
k- Image Segmentation.
l- Medical Image Reconstruction.
m- Image Encryption.
n- Face Recognition by Low-Rank Matrix approximation of the SVD.
o- Restoration (restoring blurry and noisy images) by Inverse and Pseudo-inverse

1- Security: Image watermarking

Most of the developed SVD based watermarking techniques utilizes the stability of singular values (SVs) that specify the luminance (energy) of the image layer. That is why slight variations of singular values could not influence remarkably on the cover image quality. Developed SVD based techniques either used the largest SVs or the lowest SVs to embed the watermark components either additively or by using quantization.

Example:
As an application of the SVD method we give the following algorithm of Image watermarking:



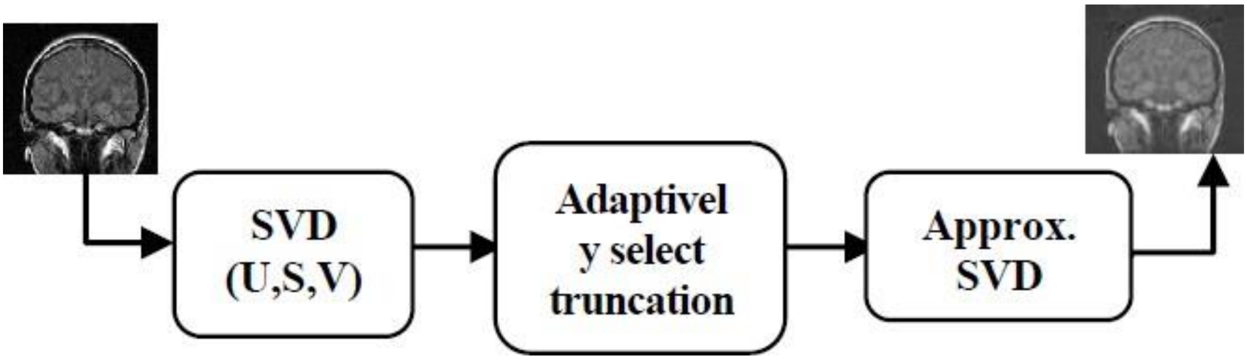
2- Image Compression

SVD with the maximum energy packing property is usually used in compression. When an image is SVD transformed, it is not compressed, but the data take a form in which the first singular value has a great amount of the image information. With this, we can use only a few singular values to represent the image with little differences from the original.

As mentioned above, SVD decomposes a matrix into orthogonal components with which optimal sub rank approximations may be obtained.

When compressing the image, the sum is not performed to the very last SVs, the SVs with small enough values are dropped.

Using the Truncated form of SVD, the transformation with rank r may offer significant savings in storage over storing the whole matrix with acceptable



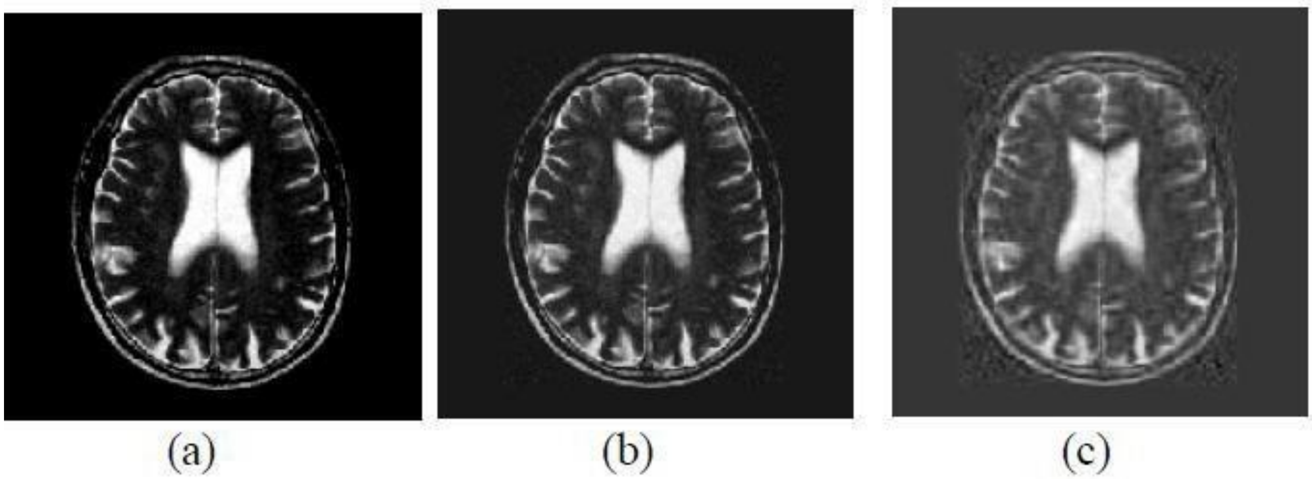
The block diagram of the SVD based Compression

The compression ratio can be calculated as follows:

$$R = \frac{n_k + k + m_k}{nm} * 100$$

where R is the compression percentage, k is the chosen rank for truncation; m and n are the number of rows and columns in the image respectively.

The following shows compressed images with different chosen ranks for truncation that result in different compression ratios.



SVD Based Compression (a) Original (b) Compression 47% (truncation to k=60)

(c) Compression 16% (truncation to k=20)

The following table illustrates the different truncation levels k used for compressing the image shown in the above figure and the resultant compression ratio for each truncation level.
Peak Signal to Noise Ratio (PSNR) is also illustrated corresponding to the different compression ratios to offer objective quality measures.

Number of truncated levels "k"	Compression "R"	PSNR
90	70.4498	37.7018
80	62.6221	36.0502
60	46.9666	32.7251
40	31.311	32.7251
20	15.6555	24.2296
10	7.8278	21.3255

Compression vs. PSN

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed or reconstructed image.

Example 1:

Multiplication of matrices also has applications in digital image processing. Consider the singular value decomposition (SVD) that consists of writing a matrix $A_{m\times n}$ as the product of three matrices:

$$A_{m\times n} = U_{m\times m}\, S_{m\times n}\, V_{n\times n}^T$$

where U and V are orthogonal matrices (that is, U^TU and $V^T V$ are $m\times m$ and $n\times n$ identity matrices, respectively) and S is a matrix whose elements $\sigma_{i,j}$ are equal to zero for $i \neq j$ and $\sigma_{1,1} \geq \sigma_{2,2} \geq \cdots \geq \sigma_{k,k} \geq 0$, with $k=min\{m,n\}$. Here is an example of the SVD decomposition:

$$A = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} = USV^T = \begin{pmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{6}}{3} & 0 & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{3} \end{pmatrix}^T$$

It can be shown that every matrix has an SVD decomposition. Moreover, algorithms exist that allow us to calculate such decompositions using a computer. The key point of our example is to observe that if $u_1, u_2, \ldots, u_m$ are the columns of the matrix U and $v_1, v_2, \ldots, v_n$ are the columns of the matrix V, then

$$A = USV^T = \sigma_{1,1}u_1v_1^T + \sigma_{2,2}u_2v_2^T + \cdots + \sigma_{k,k}u_kv_k^T$$

Why is that? Suppose that A, a grayscale image of size 1000×1000, must be transmitted from a satellite to a laboratory on Earth. In principle, the satellite would have to send 1 million numbers (one for each pixel). As typically only the first elements $\sigma_{i,i}$ of the matrix S of the SVD decomposition for A are significant (the others are "small"), it is enough, then, that the satellite sends, say, the 20 first columns of U and V, and the 20 first numbers $\sigma_{i,i}$ (totaling only 20 * 1000 + 20 * 1000 + 20 = 40020 numbers that must be sent). Upon receiving these data, the laboratory on Earth calculates the matrix $\sigma_{1,1}u_1v_1^T + \sigma_{2,2}u_2v_2^T + \cdots + \sigma_{20,20}u_{20}v_{20}^T$ that will give an approximation of the original image. Let's see an example: the picture below has 720 × 524 = 377280 pixels.

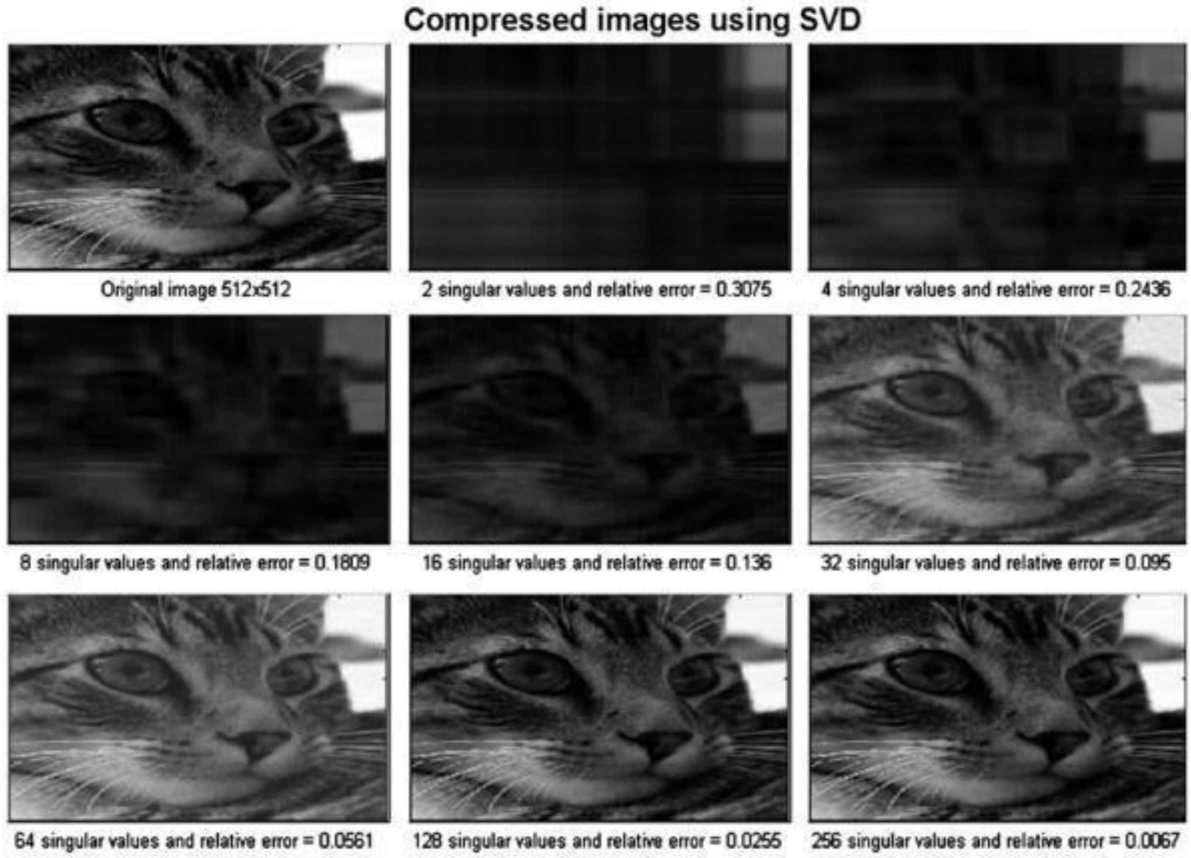
From the SVD decomposition of the corresponding matrix of this image, we can calculate the matrices $s_{1,1}u_1v_1^T + s_{2,2}u_2v_2^T + \cdots + s_{r,r}u_rv_r^T$ for r = 1, 5, 10 and 20. These matrices generate approximations to the original image, as illustrated in the following figures. Notice that the original image corresponds to the case r = 524. It is quite impressive, is it not?

$r = 1$ $r = 5$ $r = 10$ r=20

Example 2:

Students can be introduced to the singular value decomposition (SVD) method to compress data so that the original image could be reconstructed with much fewer data. An example of image compression using SVD is shown below. The relative error in the Frobenius norm of the original image is given at different ranks.

Note: Using Matlab, $[U, S, V] = \text{SVD}(A)$ performs the singular value decomposition.



Compressed images and relative errors using SVD at different ranks

Example 3:

The following is a 600×600 image of a grassy field.

This image has a rank of 600, which means it will have 600 singular values. We can apply a low-rank approximation and take a reduced number of singular values to form a rank-k approximation of the original matrix.

For 10 singular values, we get the following image:

This image is a little too fuzzy since only taking the 10 largest singular values compressed it too much. After a few tries, the number of singular values that gave a decent image was determined to be around 50. Here is the image with 50 singular values:

This image is not as high of quality as the original, but for using 50 singular values instead of 600, the quality is pretty good. We also compressed the image in the process of removing singular values. If we take the SVD of the original image, we get the following matrix sizes:

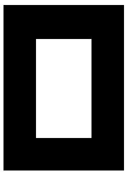
U=600×600=360000 elements
S=600×600=360000 elements
V=600×600=360000 elements
This means the total number of matrix entries needed to reconstruct the image using the SVD is 1080000 which did not do any compression on the image (it increases the number of elements needed). If we apply the low-rank approximation we get the following matrix sizes:
U=600×50=30000 elements
S=50×50=2500 elements
V=50×600=30000 elements
Using the low-rank approximation by taking only the 50 largest singular values reduces the total number of matrix elements needed to 62500. The original 600×600 image matrix needs 360000 entries to represent grass, but with the low-rank approximation, we can now represent the grass using just 62500 elements. We just reduced the total number of elements needed by 82.6%.

3. Image Denoising

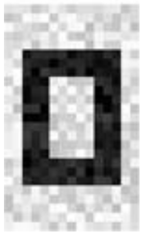
SVD has the ability to manipulate the image in the base of two distinctive data and noise subspaces which is usually used in noise filtering and also could be utilized in watermarking. Since the generic **noise signal filtering** model assumes the noise can be separated from the data, SVD locates the noise component in a subspace orthogonal to the data signal subspace. Therefore, SVD is used to approximate the matrix decomposing the data into an optimal estimate of the signal and the noise components. Image noise manifests itself as an increased "spatial activity" in the spatial domain that guides to increasing the smaller singular values in the SVD domain.

Example: Noise Reduction

An image of a black and white rectangle:



The image is treated as a matrix where 1 is white and 0 is black. Using the SVD leads to singular values of 14.72, 5.13, and 3.314. There are three singular values because the rank of the matrix is 3. Now let's add some uniform random noise to the image:



The rank of the new image matrix is now 15. Using the SVD to find the singular values of the new image matrix gives the first three singular values as 13.17, 3.85, and 2.52. The rest of the singular values are less than 0.6. If we set all the singular values less than 0.6 to zero, we get the following image:

By setting all the singular values to zero that are less than 0.6, we reduced the noise in the image. The noise is still visible, but it is much less.

4. Image Forensic

For the current digital age, digital forensic research becomes imperative. Counterfeiting and falsifying digital data or digital evidence with the goal of making illegal profits or bypassing laws is the main objective for the attackers. The forensic research focuses on many tracks; steganography, watermarking, authentication, labeling, captioning, etc. Many applications were developed to satisfy consumer requirements such as labeling, fingerprinting, authentication, copy control for DVD, hardware/ software watermarking, executables watermarks, signaling (signal information for automatic counting) for propose of broadcast monitoring count.
The proposed forensic tool is based on efficient additively embedding the optimal watermark data subspace into the host less significant subspace (noise subspace). This forensic tool can be utilized in all the forensic applications with some kind of adaptation in the embedding region based on the required robustness. Although many SVD based embedding techniques for many forensic purposes are carried out additively in singular values, they considered scaled addition without considering the wide range of singular values.

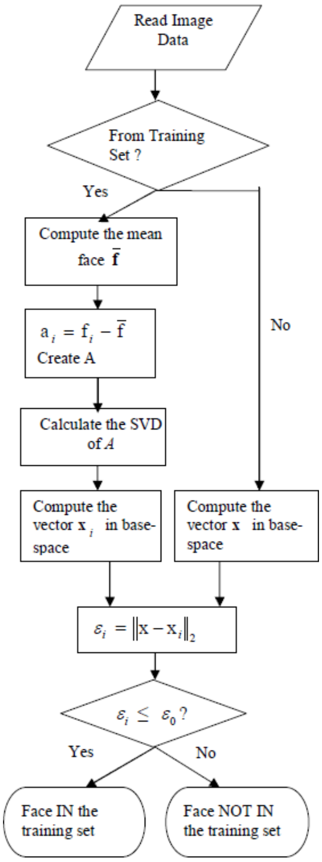
Example 1: Face Recognition

This example demonstrates how to use the SVD approach for image processing in the area of Face Recognition (FR).

To perform face recognition with SVD, we treated the set of known faces as vectors in a subspace, called “face space”, spanned by a small group of "basefaces". The projection of a new image onto the baseface is then compared to the set of known faces to identify the face.

The test is under the training set with image Size: M = 92 x 112 = 10,304, the number of known individuals: N = 20, Different Conditions: All frontal and slight tilt of the head, different facial expressions.

Essentially, a face image is of M (say 10,000) dimension. But the rank r of matrix A is less than or equals N. For most applications, a smaller number of basefaces than r are sufficient for identification. In this way, the amount of computation is greatly reduced. The following figures show the base face image, the average of training set image, and the training set image we used for this experiment.



A Computed Mean Flow chart of Face Recognition with SVD Training Set Images

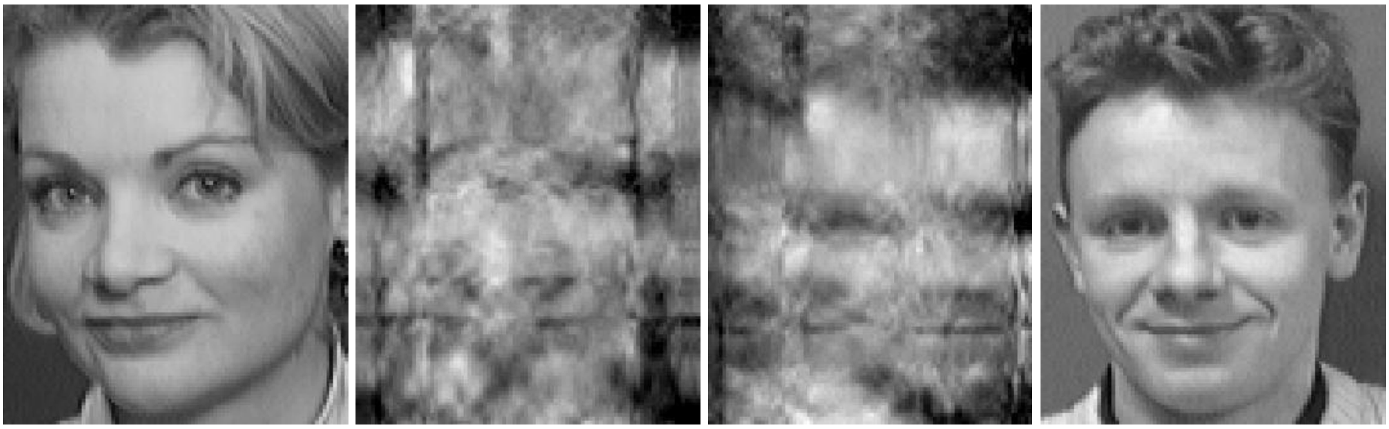
Face Image of Training Set Images

On the other hand, it is possible to be widely believed that the singular values are more important since its uniqueness; but it is naturally thinking that SVs are the most important attribute of image matrix used for face recognition. However, with experiments on exchange the SVs of two images, the result is very interesting and it shows that the singular vectors (left and right) are more important for reconstruction of the original image. The first experiment was designed to use two person's face images, on which we performed SVD decomposition. For example, the face images of Janet ($A1$) and Andy ($A2$) was decomposed into U, S, V so that:

$$A_1 = U_1 \times S_1 \times V_1^T$$

$$A_2 = U_2 \times S_2 \times V_2^T$$

Then we did the combination of the singular values and singular vectors, the result shows in the following figures:

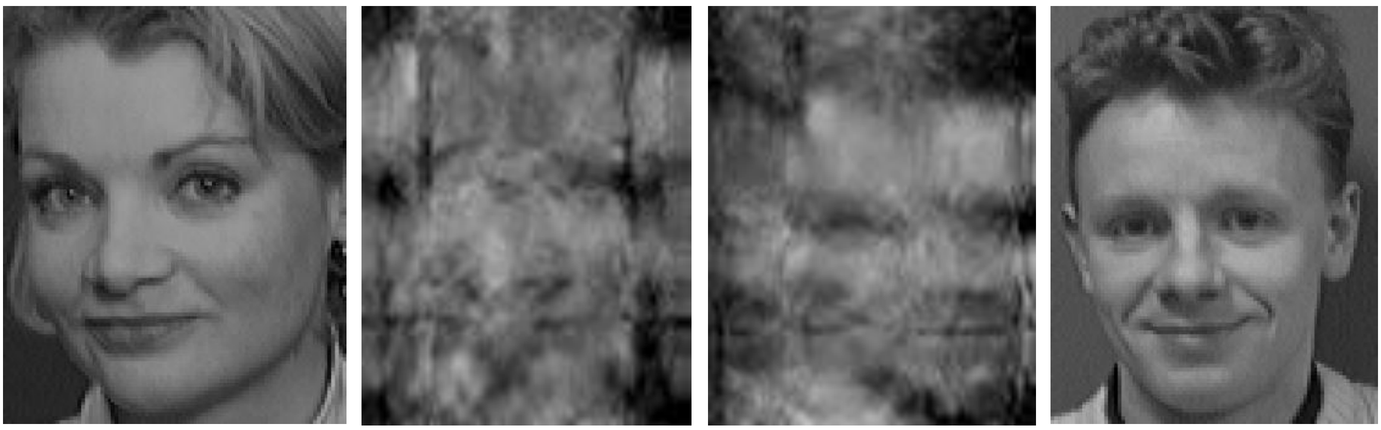


a.) Combination of $U_1 \times S_1 \times V_1^T$

b.) Combination of $U_1 \times S_1 \times V_2^T$

c.) Combination of $U_2 \times S_1 \times V_1^T$

d.) Combination of $U_2 \times S_1 \times V_2^T$



e.) Combination of $U_1 \times S_2 \times V_1^T$

f.) Combination of $U_1 \times S_2 \times V_2^T$

g.) Combination of $U_2 \times S_2 \times V_1^T$

h.) Combination of $U_2 \times S_2 \times V_2^T$

Result of Exchanged Singular Value with Singular Vectors

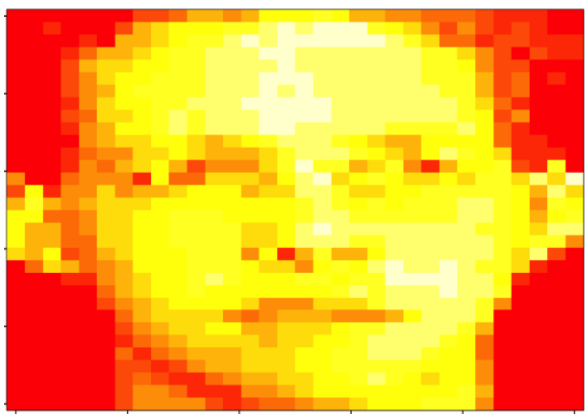
In the above images:

1- Image a) shows the combination of $U_1 \times S_1 \times V_1$, which is an original image of Janet.
2- When we combined Janet's SVs with Andy's singular vector, it shows Andy's face image (d). The image has a different brightness with Andy's original image.
3- Image e) shows Janet's face, but it is a combination of Andy's SVs and Janet's singular vectors.
4- When we combined two pairs of singular vectors U and V, which are from two images respectively, the outcome images look like a "ghost". The result shows in b), c), f) and g).

We also tested the two images, one is a face image and the other is not a face image (e.g. flower). The experimentation showed the same result as two face images.
From the result, we see that, though the singular values are unique in SVD decomposition, the singular vectors are more important for image recognition. This fact indicates that deep research and further investigation of characteristics of SVD in image processing are necessary.

Example 2: Face data

In this example, we use some data that make up an image of a face and show how the SVD can be used to produce varying approximations to this "dataset". Here is the original data.

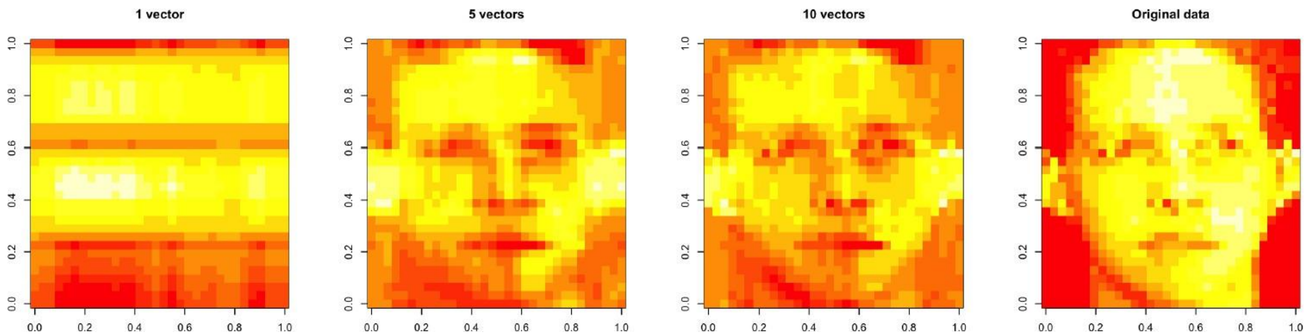


If we take the SVD and plot the squared and normalized singular values, we can see that the data can be explained by just a few singular vectors, maybe 4 or 5.

Now we can start constructing approximations to the data using the left and right singular vectors. Here we create one using just the first left and right singular vectors.

We can also create ones using 5 and 10 singular vectors, which presumably would be better approximations.

Now we can plot each one of these approximations along with the original data.



Here, the approximation using 1 singular vector is pretty poor, but using 5 gets us pretty close to the truth. Using 10 vectors doesn't seem to add much to the features, maybe just a few highlights. So 5 singular vectors is a reasonable approximation in this c