Application Of SVD

Image Compression

202018019-Nirmit Gupta
202018044-Meet Chaudhary
202018049-Bhavya Dubey
2~~020~~18050-Harman Singh

OBJECTIVE

- To **Reduce The Redundancy Of The Image** And to Store or Transmit Data in an Efficient form using SVD(Singular Value Decomposition).

What is Image Compression?

- **Image compression** is a type of data **compression** applied to digital **images**, to reduce their cost for storage or transmission.
- **Images** are represented in a rectangular array where each element corresponds to the grayscale value for that pixel.

Why do we need Compression?

Compression is needed basically to save :

- Memory/Space
- Bandwidth
- cost

Applications of SVD

- Image Decomposition
- Matrix Factorization
- Face Recognition
- Dimensionality Reduction
- Recommendation System(Amazon and Netflix)
- Matrix Completion
- Image Compression

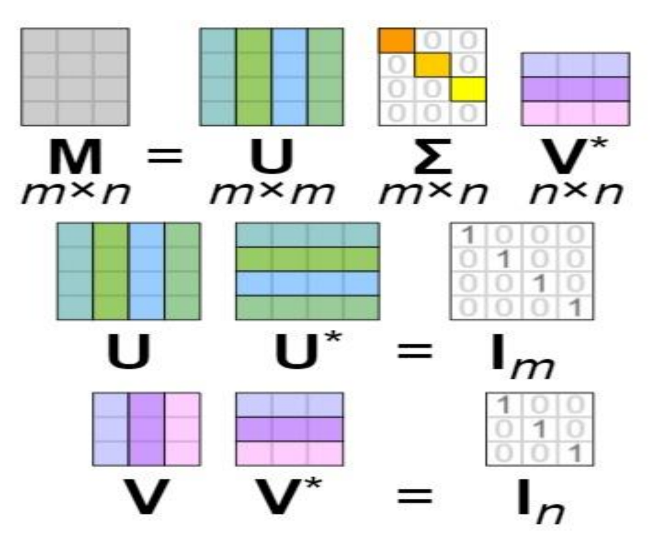
What is SVD and why we use it ?

WHAT?

- In linear algebra, the **singular value decomposition** (**SVD**) is a factorization of a real or complex matrix that generalizes the eigen decomposition of a square normal matrix to any matrix via an extension of the polar decomposition.

WHY?

- Easy to implement
- Low cost and inexpensive



$\mathbf{M}_{m \times n} = \mathbf{U}_{m \times m} \, \Sigma_{m \times n} \, \mathbf{V}^*_{n \times n}$
$\mathbf{U} \, \mathbf{U}^* = \mathbf{I}_m$
$\mathbf{V} \, \mathbf{V}^* = \mathbf{I}_n$

WORKFLOW:

1. Uploading the original input image
2. Using SVD to compress the image
   - Converting the image to the matrix $A=U\Sigma V$
   - Calculating matrix U ,Σ ,Vt and then multiplying
3. Applying k number of compressions on the uploaded image

Implementing svd image compression in python

There are two ways to implement svd:

- The first is by using normal matrices and computing each U, Vt, and **Σ** and then multiplying it to obtain the compressed matrix A. this makes the process lengthy.
- The second method is using a module of numpy which is **numpy.linalg.svd** which does all this operation of matrix on its own and gives us the result. This makes the process easy and time consuming

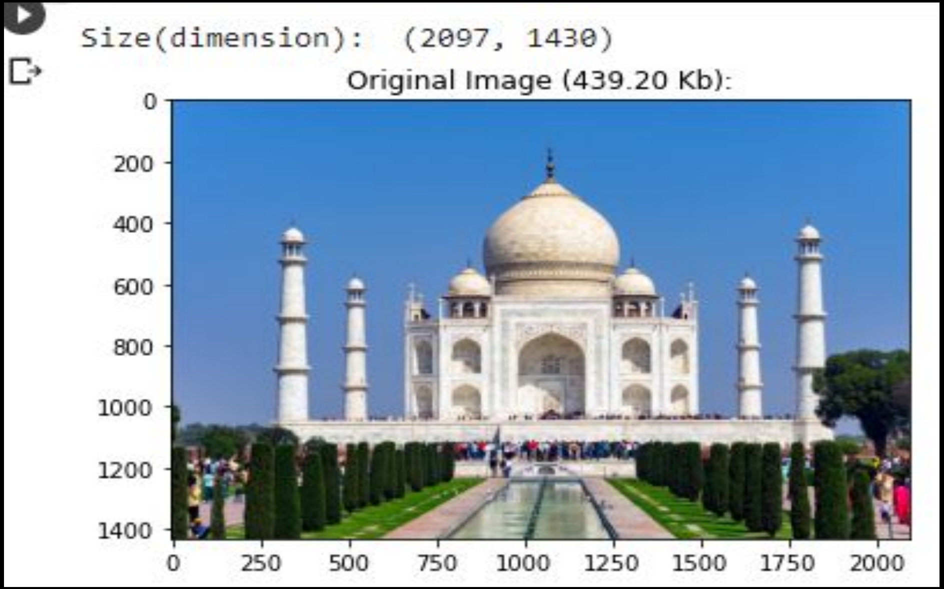
Image compression using SVD:

- SVD can compress any form of data efficiently:
- SVD takes a matrix, square or non square, and divides it into two orthogonal matrix and a diagonal matrix
- This allows to rewrite the original matrix as a sum of much simpler rank one matrices

LIBRARIES USED

- Numpy(for Svd)
- Matplotlib
- PIL(python image library)
- OS(Operating System)

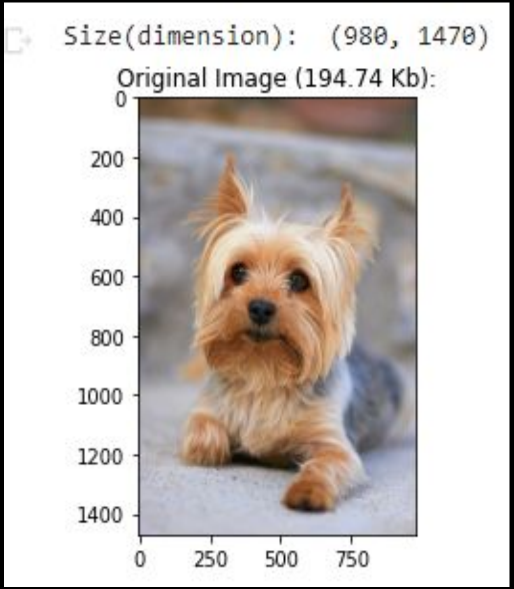
Input Image



```
Size(dimension):  (2097, 1430)
```

Original Image (439.20 Kb):

Original Image Before Compression
Dimension:(2097x1430)
Size: 439.20kb

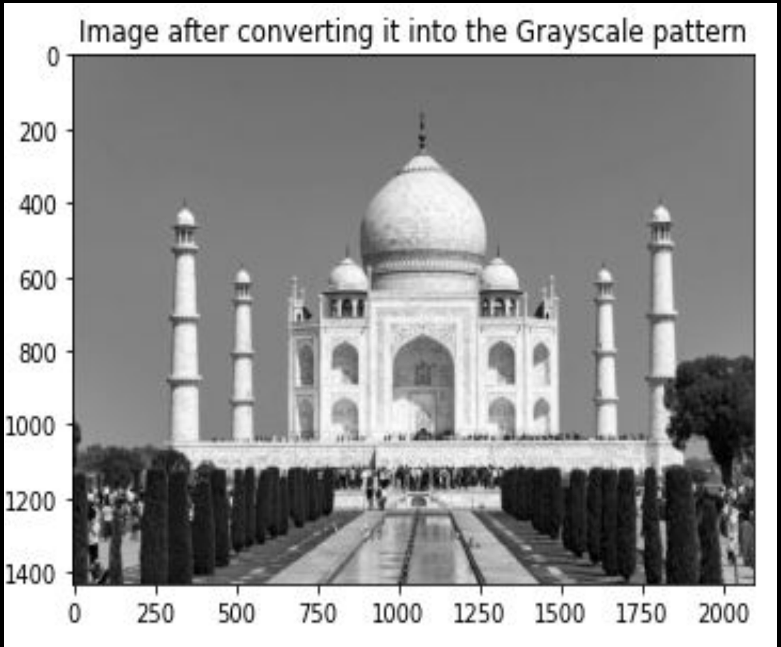


```
Size(dimension):  (980, 1470)
```

Original Image (194.74 Kb):

Original Image Before
Compression
Size:(980x1470)

Image After Converting it into Grey Scale



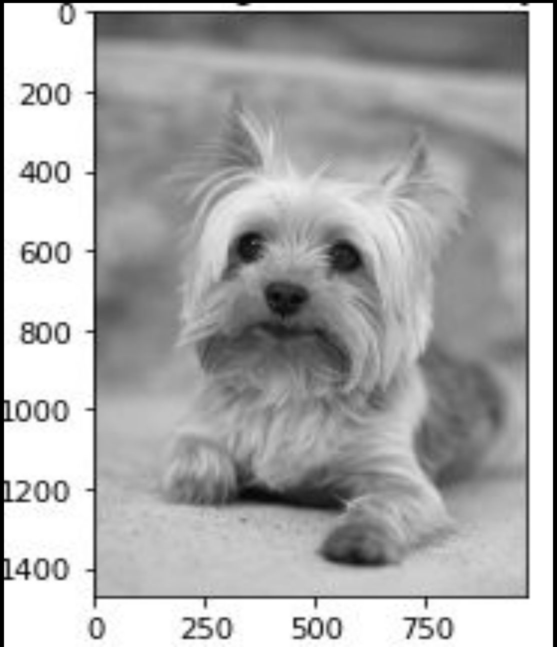
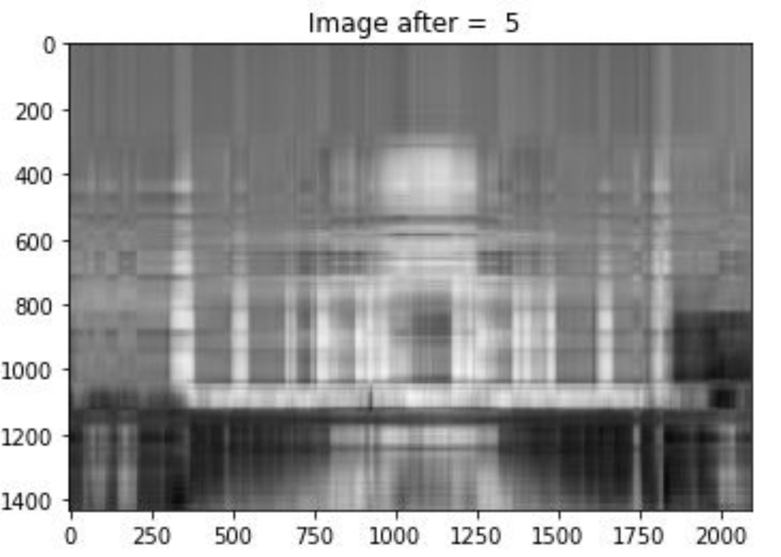


Image After Compression



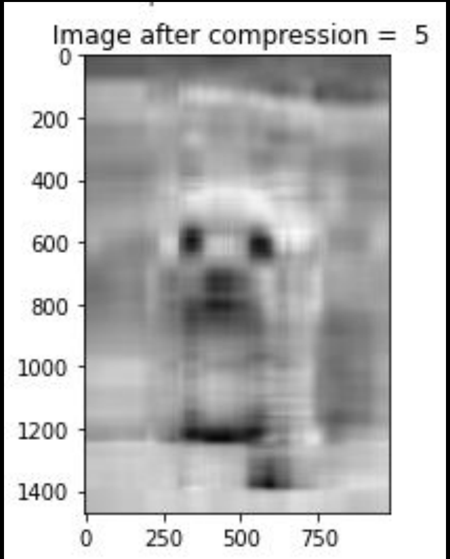
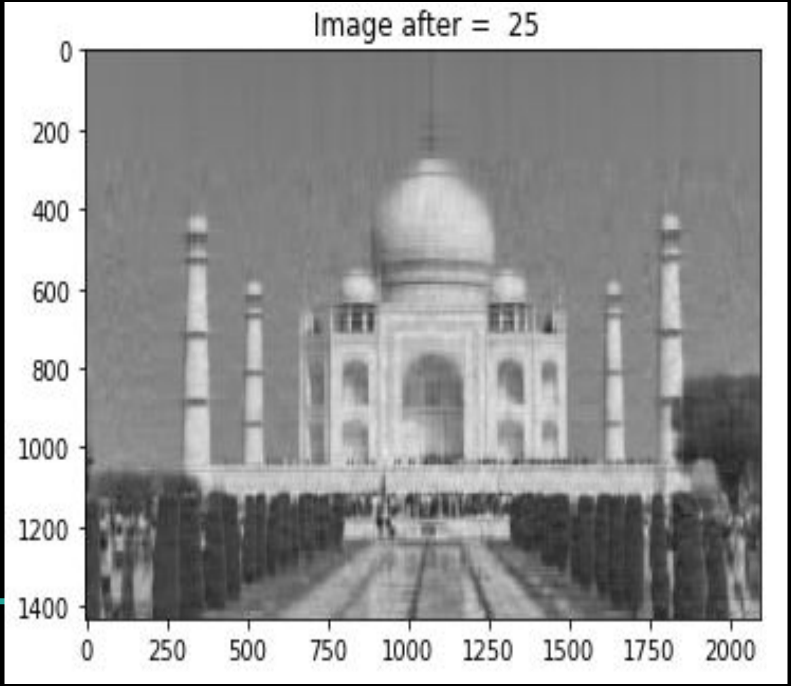
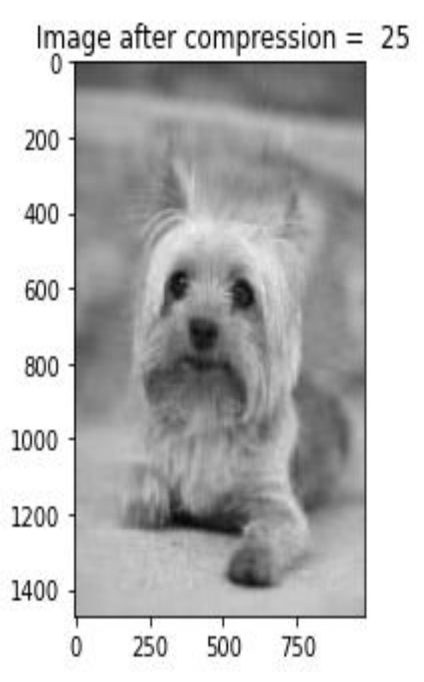


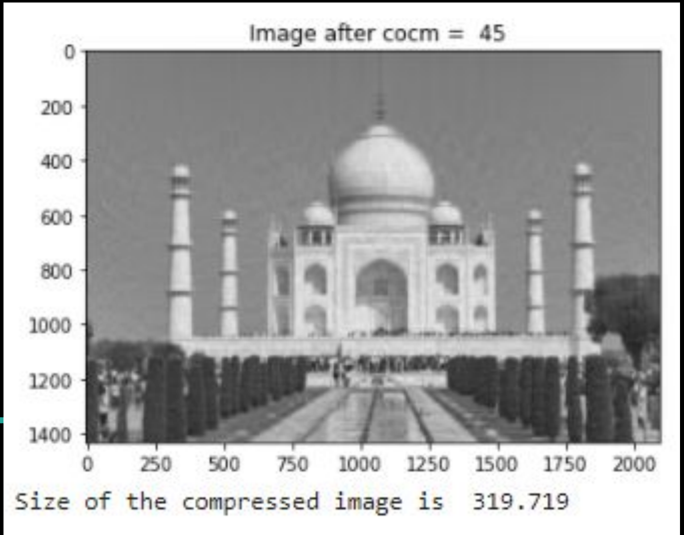
Image after 5 compressions



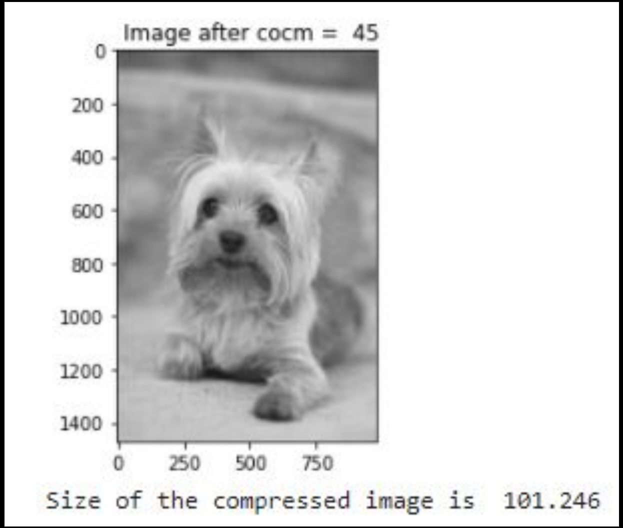


ImageAfter 25 compression

Image After 45 compression



```
Size of the compressed image is  319.719
```



```
Size of the compressed image is  101.246
```

Conclusion

- SVD Image Compression can be used to compress the image to a significant amount thus saving space , bandwidth and cost
- The compression depends on the size of image but SVD can compress the image upto more than 50% of its Original size

THANK YOU!