SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COURSE NAME : 19EC513 – IMAGE PROCESSING AND COMPUTER VISION

III YEAR / V SEMESTER

Unit I- DIGITAL IMAGE FUNDAMENTALS AND TRANSFORMS

Topic : SVD transform

Singular Value Decomposition

- Handy mathematical technique that has application to many problems

- Given any $m \times n$ matrix **A**, algorithm to find matrices **U**, **V**, and **W** such that

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T$$

U is $m \times n$ and orthonormal

W is $n \times n$ and diagonal

V is $n \times n$ and orthonormal

$$\begin{bmatrix} \\ \mathbf{A} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \mathbf{U} \\ \\ \end{bmatrix} \begin{pmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n \end{pmatrix} \begin{bmatrix} \\ \mathbf{V} \\ \\ \end{bmatrix}^T$$

- Treat as black box: code widely available In Matlab:

```
[U,W,V]=svd(A,0)
```

- The w_i are called the singular values of **A**
- If **A** is singular, some of the w_i will be 0
- In general *rank*(**A**) = number of nonzero w_i
- SVD is mostly unique (up to permutation of singular values, or if some w_i are equal)

- Application #1: inverses
- $\mathbf{A}^{-1} = (\mathbf{V}\mathbf{T})^{-1}\,\mathbf{W}^{-1}\,\mathbf{U}^{-1} = \mathbf{V}\,\mathbf{W}^{-1}\,\mathbf{U}\mathbf{T}$
  - Using fact that inverse = transpose for orthogonal matrices
  - Since **W** is diagonal, $\mathbf{W}^{-1}$ also diagonal with reciprocals of entries of **W**

Singular Value Decomposition aka SVD is one of many matrix decomposition Technique that decomposes a matrix into 3 sub-matrices namely U, S, V where U is the left eigenvector, S is a diagonal matrix of singular values and V is called the right eigenvector. We can reconstruct SVD of an image by using **linalg.svd()** method of NumPy module.

Syntax:

linalg.svd(matrix, full_matrices=True, compute_uv=True, hermitian=False)

Parameters:

1.matrix : A real or complex matrix of size > 2.

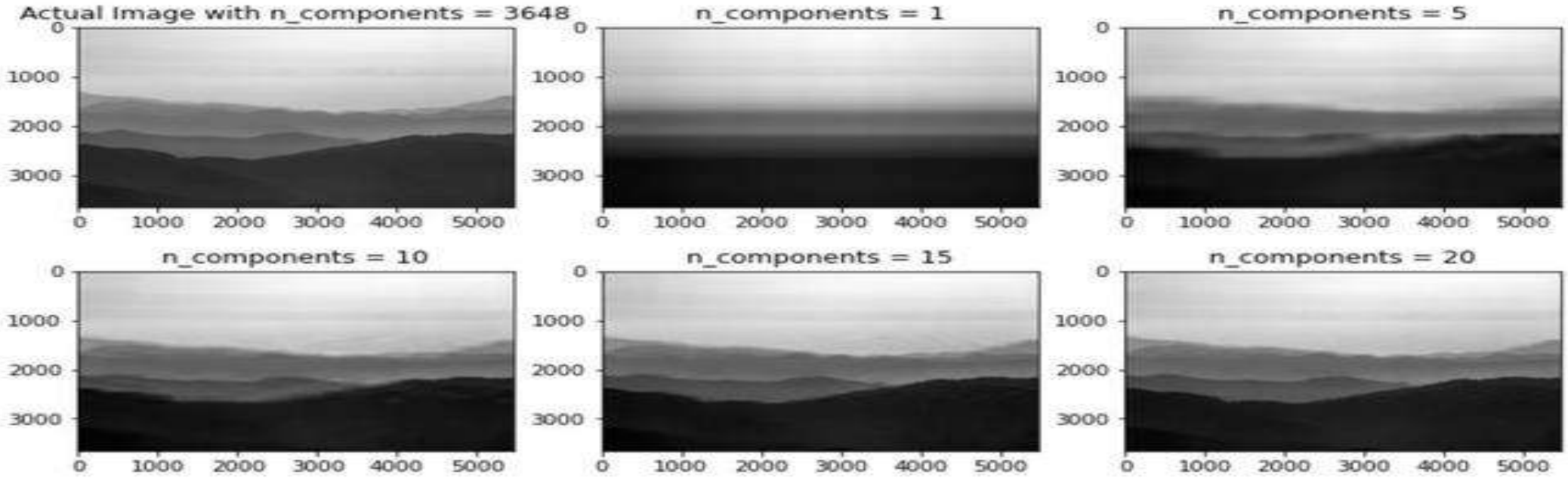
2.full_matrices: If True the size of u and v matrices are m x n , if False then the shape of u and v matrices are m x k , where k is non-zero values only.

3.compute_uv: Takes in boolean value to compute u and v matrices along with s matrix.

4.hermitian: By default matrix is assumed to be Hermitian if it contains real-values, this is used internally for efficiently computing the singular values.

Image Used:

Output:



Any Query????

Thank you……