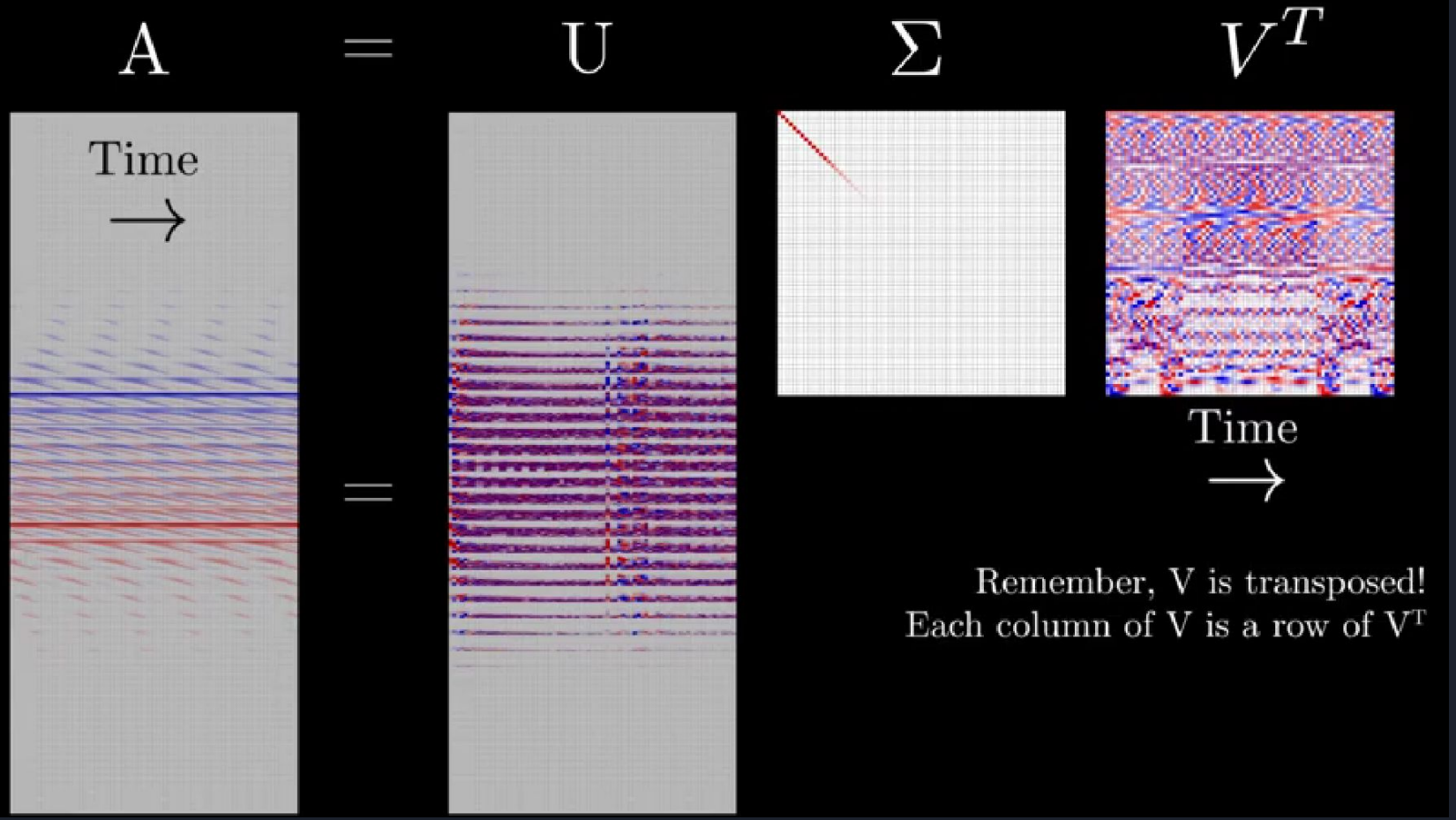
POD using Full Singular Value Decomposition

Singular Value Decomposition

It is an important tool used for data processing. It has to do with data reduction, to find the key correlations. It uses linear algebra to find the most influencing characteristic for any set of data given.

In this we decompose a matrix A into three different matrices each carrying their own significance and for the understanding of the modes.

$$M = U \cdot \Sigma \cdot V^*$$

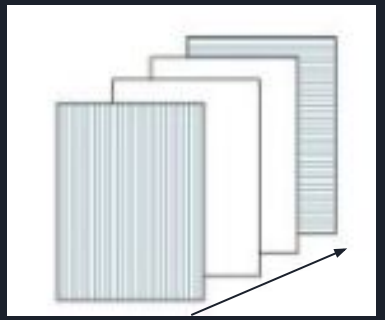


Ref: https://youtu.be/axfUYYNd-4Y

Understanding and Visualizing

$$\begin{bmatrix} | & | & & | \\ A_1 & A_2 & \cdots & A_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ U_1 & U_2 & \cdots & U_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots \\ 0 & \sigma_2 & \cdots & \\ & & \sigma_n & \end{bmatrix} \begin{bmatrix} | & | & & | \\ V_1 & V_2 & \cdots & V_m \\ | & | & & | \end{bmatrix}^T$$

$$V^T = \begin{bmatrix} - & V_1^T & - \\ - & V_2^T & - \\ & \vdots & \\ - & V_m^T & - \end{bmatrix}$$



Interpretation

For an $m \times n$ matrix $\mathbf{A}$ of rank p there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

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

$m \times m$ $\quad$ $m \times n$ $\quad$ V is $n \times n$

The columns of $\boldsymbol{U}$ are orthogonal eigenvectors of $\boldsymbol{AA^T}$.

The columns of $\boldsymbol{V}$ are orthogonal eigenvectors of $\boldsymbol{A^TA}$.

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

$$AA^T = U\Sigma V^T \; V\Sigma U^T$$

$$V^T V = I \; , \; U^T U = I$$

$$AA^T = U\Sigma^2 U^T$$

$$AA^T U = U\Sigma^2$$

$$U \longrightarrow Eigenvectors$$

$$\Sigma \longrightarrow Eigenvalues$$

Code For SVD

```
import cv2
import cv2
import os
import glob
from skimage.filters import gaussian
from skimage import img_as_ubyte
import numpy as np
```

```
images_list = []
path = r"E:\cropped3\cropped_fully_developed\*.*"
```

```
for file in glob.glob(path):
    print(file)
    img = cv2.imread(file, 0)
    images_list.append(img)

images_list = np.array(images_list)
```

Preprocessing

```
list_digit = []
list_digit_nomean = []

for i in range(67):
    list_digit_nomean.append(images_list[i, :, :].ravel().tolist())
    temp = images_list[i, :, :].ravel()
    temp_2 = temp - (np.mean(temp, axis = 0))
    list_digit.append(temp_2.tolist())

arr_digit_1 = pd.DataFrame(list_digit)
arr_digit = np.array(list_digit)
arr_digit_nomean = pd.DataFrame(list_digit_nomean)
arr_digit.shape
```

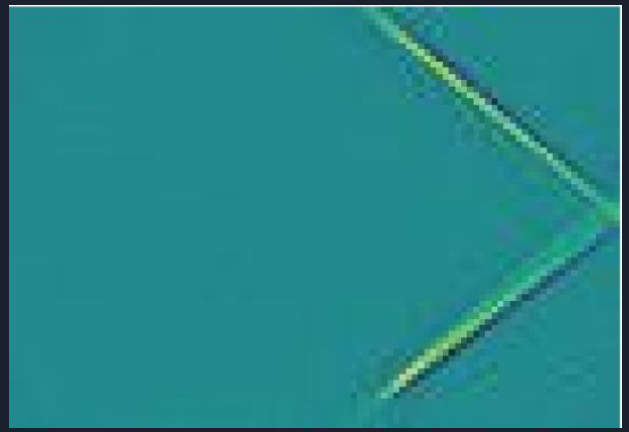
```
from scipy.linalg import svd
U, s, VT = svd(arr_digit.T)
```

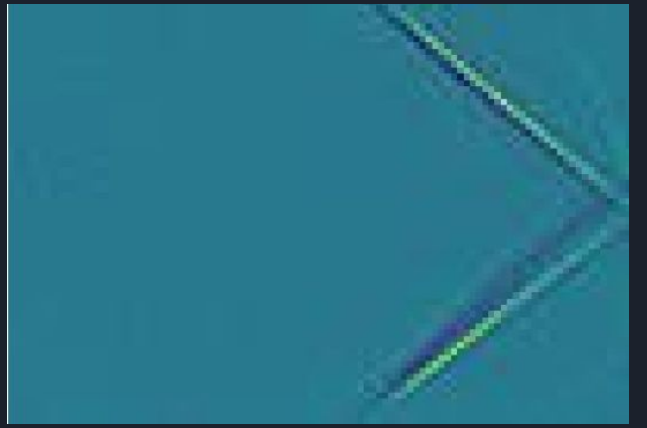
```
U.shape
```

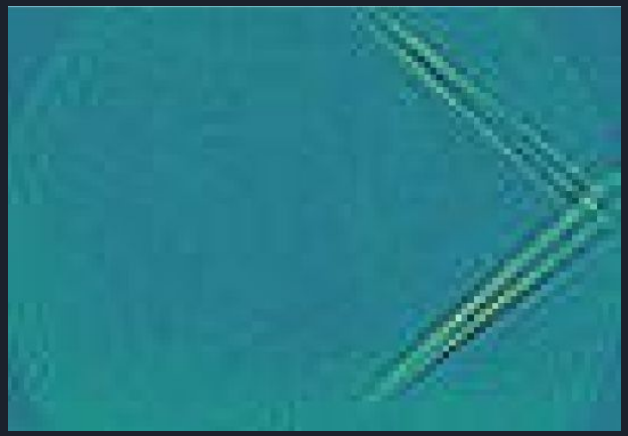
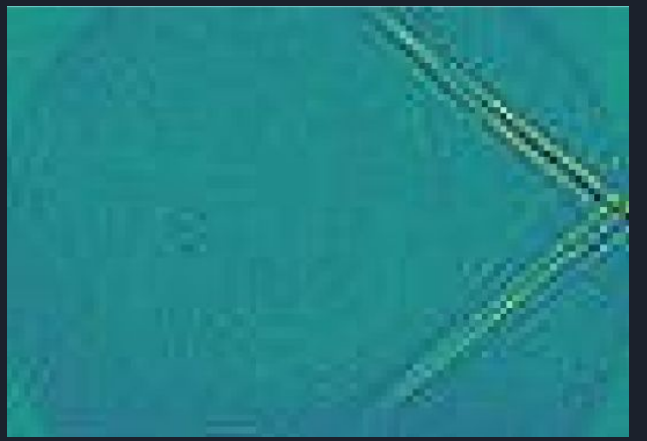
```
(6900, 6900)
```

```
import matplotlib

mode_1 = U[:, 0 ]
mode_1 = mode_1.reshape(69, 100)
visualize(mode_1)
```





```
!git clone https://github.com/xinntao/Real-ESRGAN.git
%cd Real-ESRGAN

!pip install basicsr
!pip install facexlib
!pip install gfpgan
!pip install -r requirements.txt
!python setup.py develop

!wget https://github.com/xinntao/Real-ESRGAN/releases/download/v0.1.0/RealESRGAN_x4plus.pth -P experiments/pretrained_models
```

```
import os
from google.colab import files
import shutil

upload_folder = 'upload'
result_folder = 'results'

if os.path.isdir(upload_folder):
    shutil.rmtree(upload_folder)
if os.path.isdir(result_folder):
    shutil.rmtree(result_folder)
os.mkdir(upload_folder)
os.mkdir(result_folder)

# upload images
uploaded = files.upload()
for filename in uploaded.keys():
  dst_path = os.path.join(upload_folder, filename)
  print(f'move {filename} to {dst_path}')
  shutil.move(filename, dst_path)
```