MATH: 4840 Mathematics of Machine Learning

Dimension Reduction Using Singular Value Decomposition (SVD)

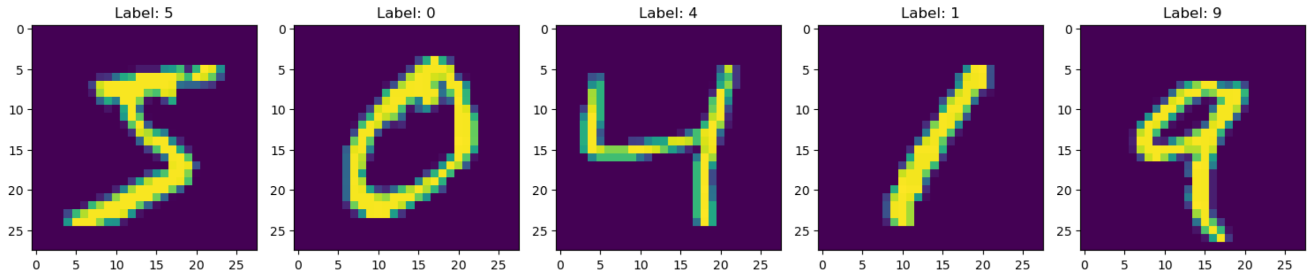
MNIST Dataset

Roy Huang, Vincent Cai, Mingwei Xi

December 11, 2024

Introduction

- Datasets often exist in high-dimensional spaces, where each dimension corresponds to a distinct feature of the data
  - MNIST
  - 784 pixels in each image



- However, the data points often lie closely on a manifold
  - A surface of lower dimensionality within a higher-dimensional space
  - Meaningful information can be captured in fewer dimensions

Dimension Reduction

- Applications
  - Medical Imaging (Data Volume Reduction)
  - Financial Fraud Detection (Abnormal Activity)
  - Autonomous Vehicle (Image Recognition)
  - E-Commerce Recommendation (Costumer Segmentation)

- Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD)

- Any matrix A can be unconditionally decomposed into three very special matrices:

$$\boxed{A = U\,\Sigma\,V^T}$$

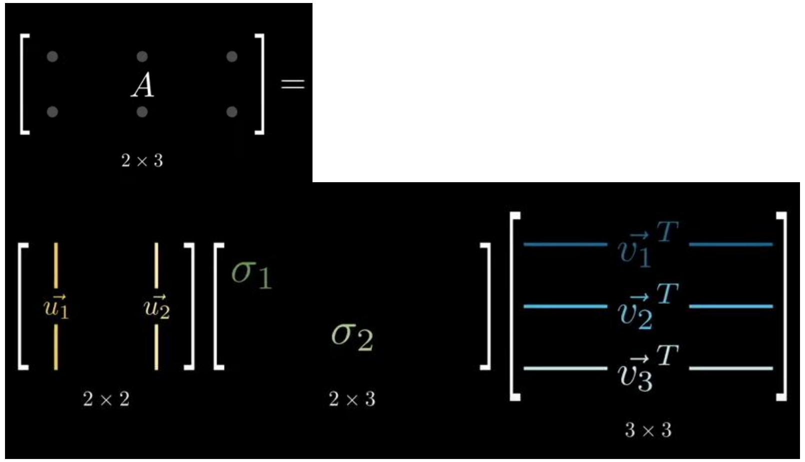
$A \in \mathbb{R}^{m \times n}$

AA^T
symmetric
$\downarrow$
$u_1, u_2, u_3 \ldots$
left singular vectors

A^TA
symmetric
$\downarrow$
$v_1, v_2, v_3 \ldots$
right singular vectors

$u_1, u_2, u_3 \ldots$ $\qquad$ $v_1, v_2, v_3 \ldots$

After sorting
$\lambda_1, \lambda_2, \lambda_3 \ldots 0, 0, 0$
$\downarrow$
$\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3} \ldots$
singular values
$(\sigma_1, \sigma_2, \sigma_3, \ldots)$



U $\qquad$ Σ $\qquad$ V^T

Singular Value Decomposition (SVD)

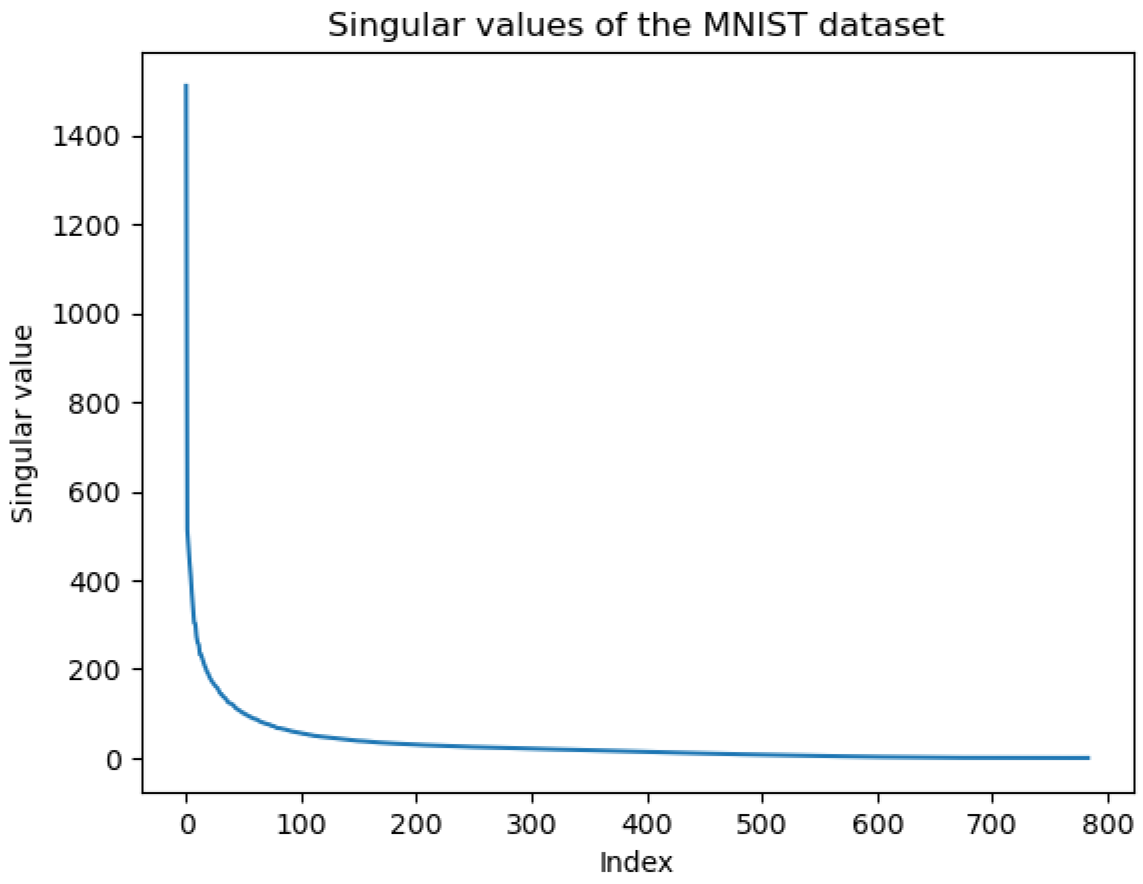
- Flatten each image (28*28)->(784)
- Create data matrix (60,000, 784)

```
# perform SVD
u, s, vh = np.linalg.svd(data_matrix, full_matrices=False)
```

```
(60000, 784) (784,) (784, 784)
```

U $\qquad\qquad$ Σ $\qquad\qquad$ V^T

Singular Value Decomposition (SVD)



- Proportion of variance explained by the first 10 singular values:

$$Total\ var = \sum \sigma^2$$

$$Sum\ of\ top\ 10\ var = \left(\sum_{i=1}^{10} {\sigma_i}^2\right)$$

$$Var\ eplained = \frac{Sum\ of\ top\ 10\ var}{Total\ var} = 0.6916$$

Image Reconstruction

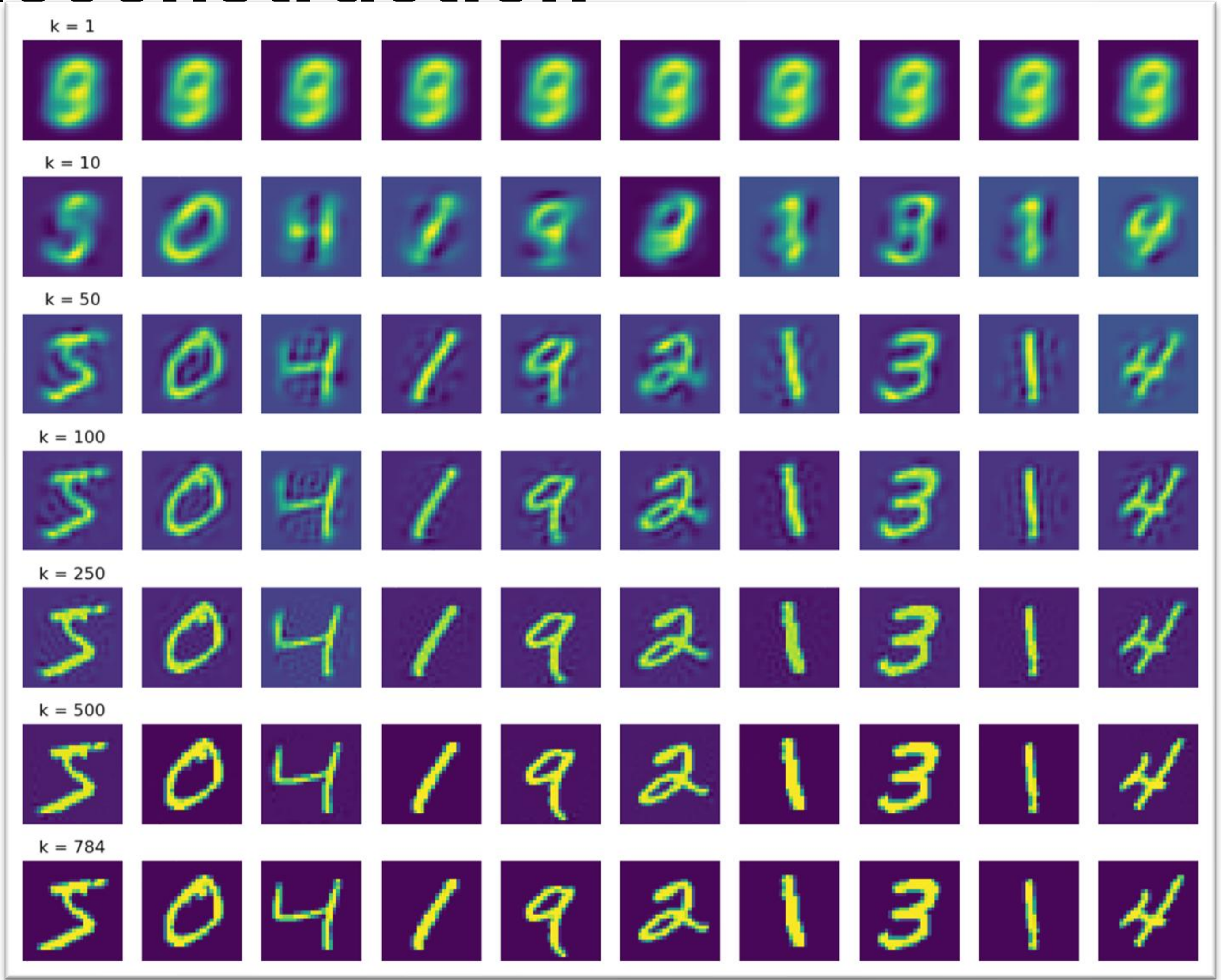
- Σ is sorted in descending order
- Retaining only the top k singular values $\sigma_1, \dots, \sigma_k$ to preserve most of the data's variance:

$$\boxed{A_k = U_k \Sigma_k V_k^T}$$

- If we choose k=50:
  - U_k: (60,000 * 50)
  - Σ_k: (50 * 50)
  - V_k^T: (50 * 784)
  - A_k: (60,000 * 784)

Image Reconstruction

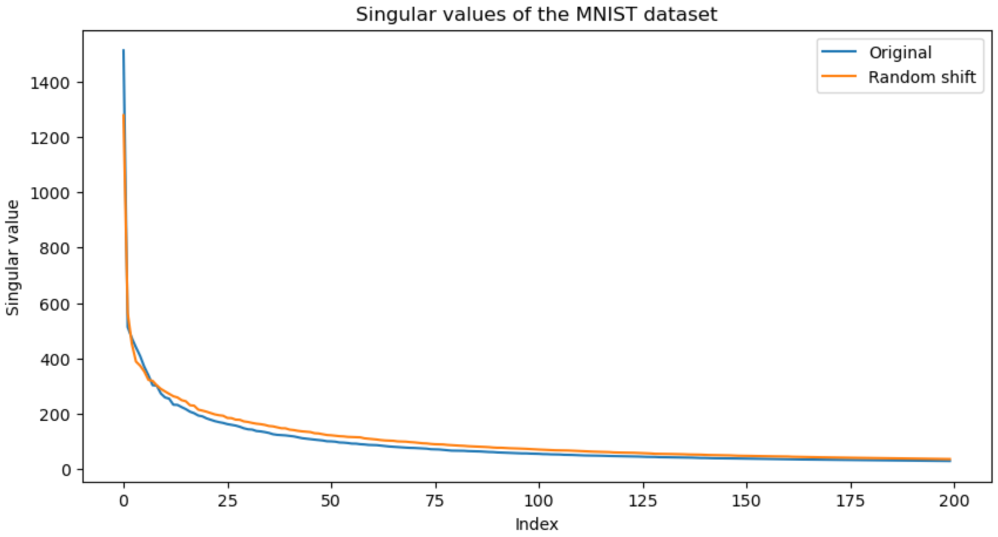
Visualize 10 images(rows) from A_k



Transformation

- Apply transformation on dataset
- Perform SVD again
- Observe changes
  - Explained variance
  - Reconstruction results

Centering, Random Shifting



Explained variance by first 10 components (original/centered):

0.6916

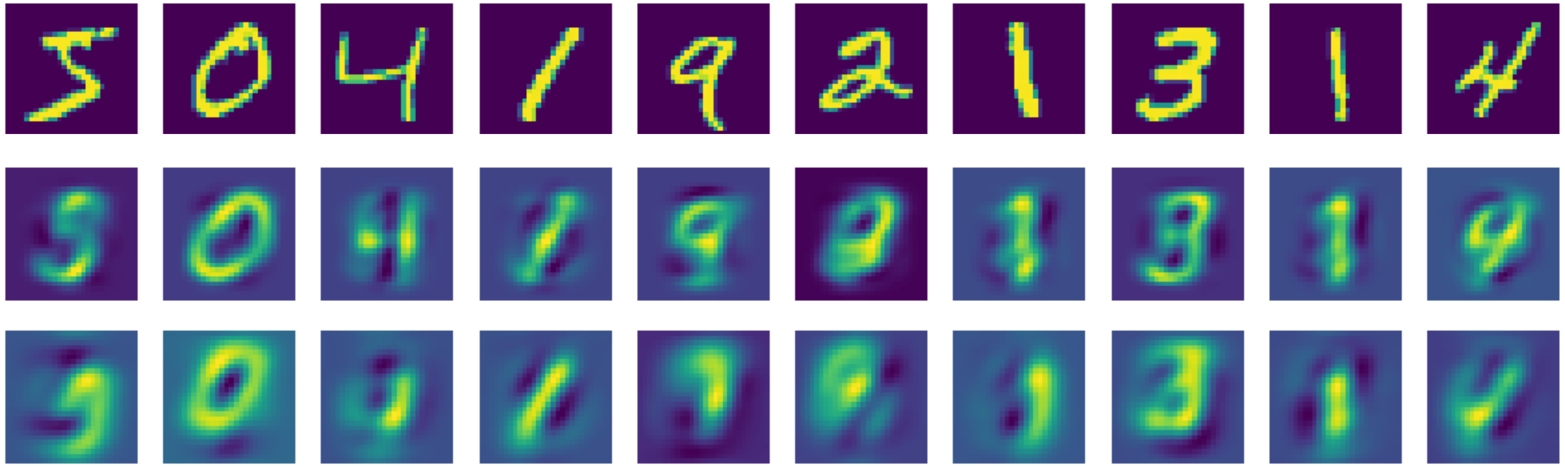
Explained variance by first 10 components (random shift):

0.5607

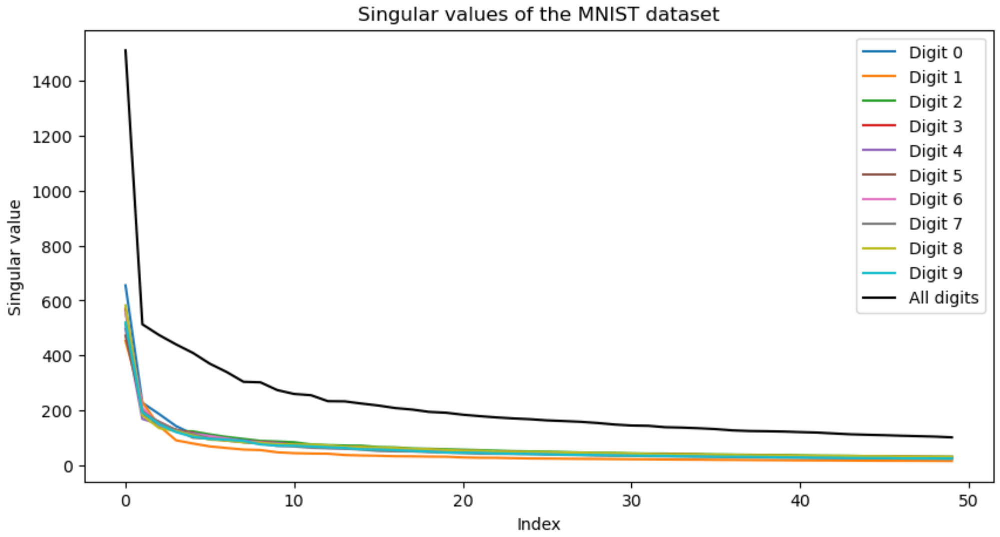
Original

Reconstruction (Centered)

Reconstruction (Random Shifted)



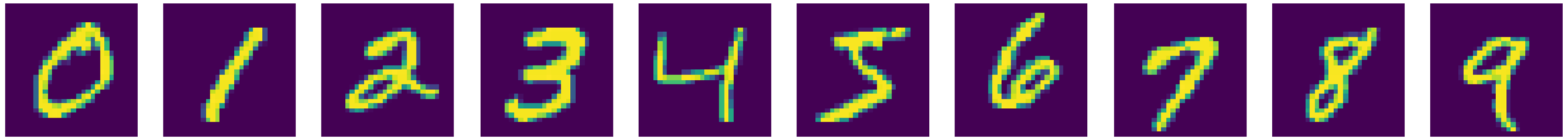
Each Digit



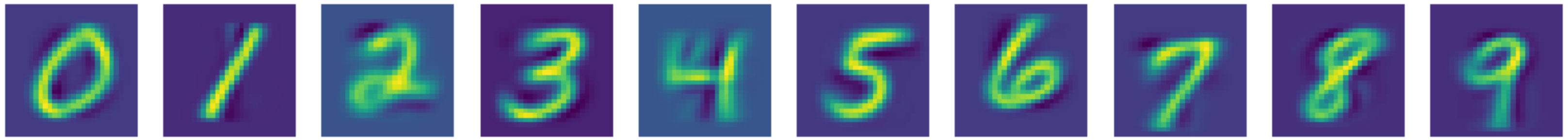

```
digit 0: 0.8419
digit 1: 0.8942
digit 2: 0.7545
digit 3: 0.7789
digit 4: 0.7768
digit 5: 0.7497
digit 6: 0.8098
digit 7: 0.8095
digit 8: 0.7689
digit 9: 0.8034
```

Explained variance by first 10 components (original): 0.6916

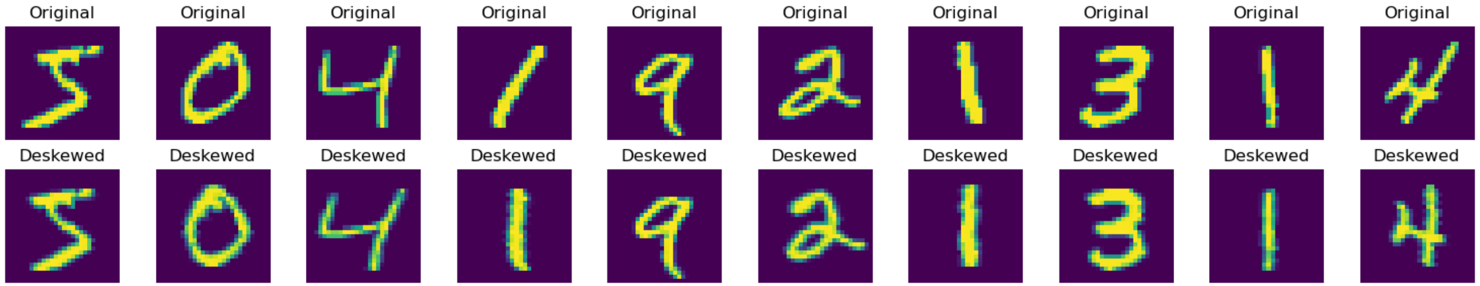
Original images



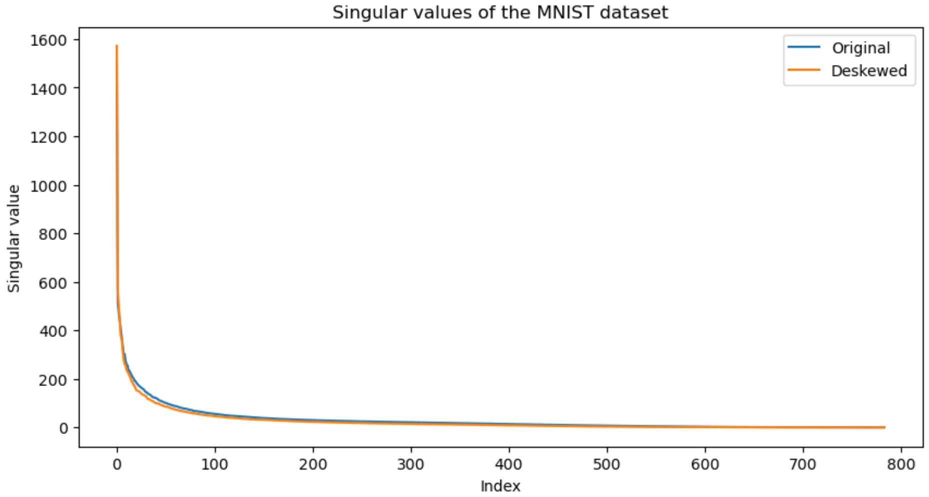
Reconstructed images (k = 10)



Deskewing



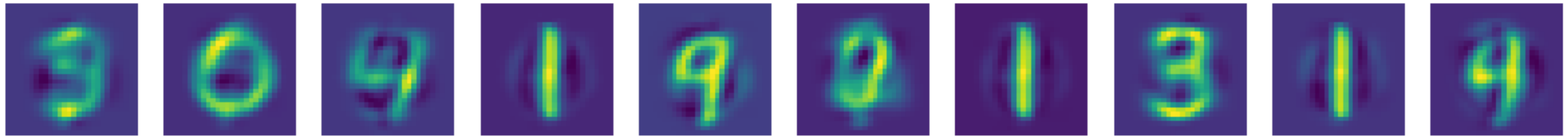

```
m = cv2.moments(image)
skew = m['mu11'] / m['mu02']
M = np.float32([[1, skew, -0.5 * 28 * skew], [0, 1, 0]])
image_deskewed = cv2.warpAffine(image, M, (28, 28))
```



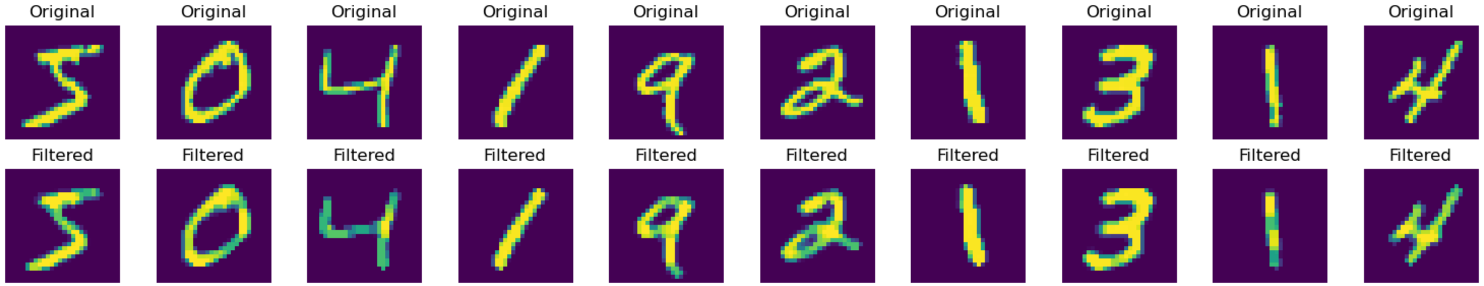
Explained variance by first 10 components (original): 0.6916

Explained variance by first 10 components (deskewed): 0.7642

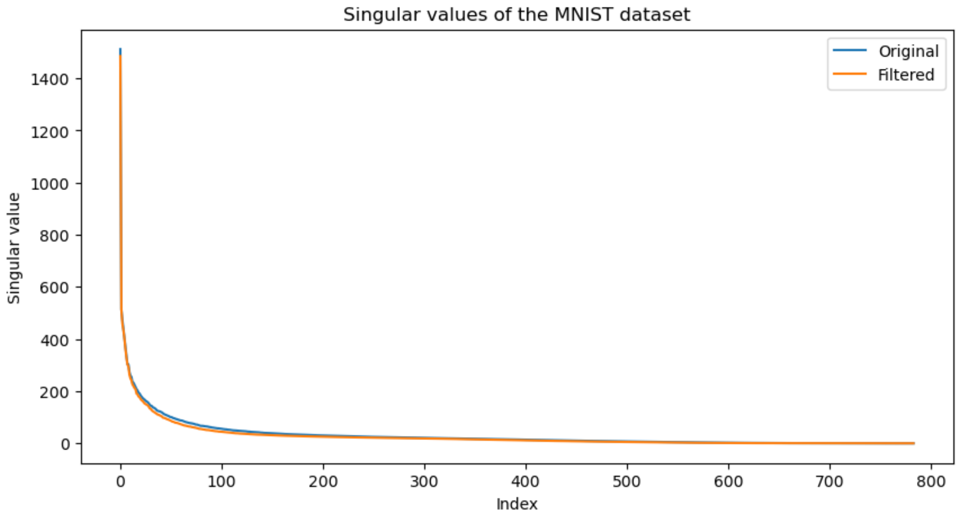
Reconstructed images using deskewing



Median Filtering



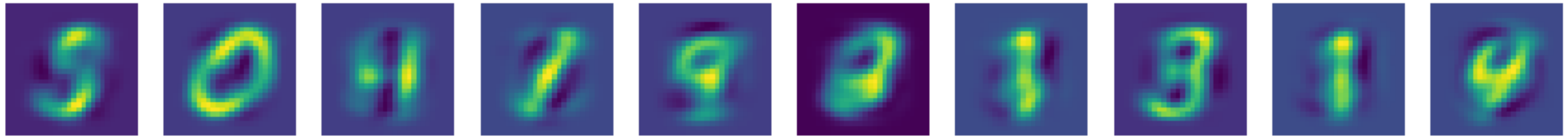

```
filtered_image = cv2.medianBlur(image, kernel_size)
```



Explained variance by first 10 components (original): 0.6916

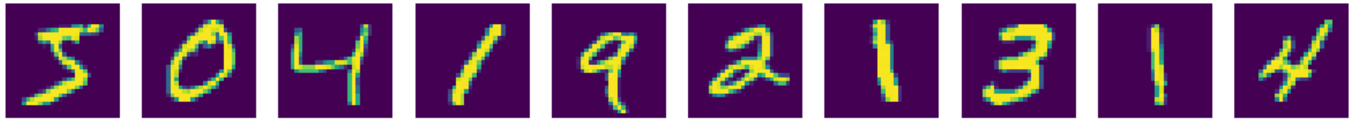
Explained variance by first 10 components (deskewed): 0.7328

Reconstructed images using median filter

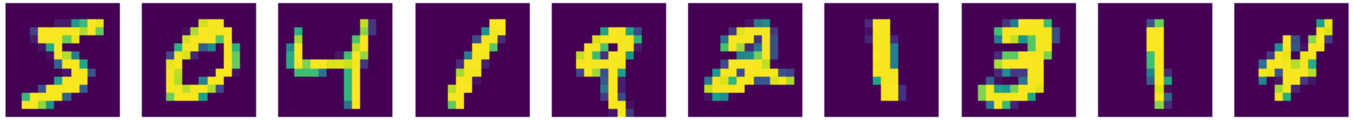


Max Pooling

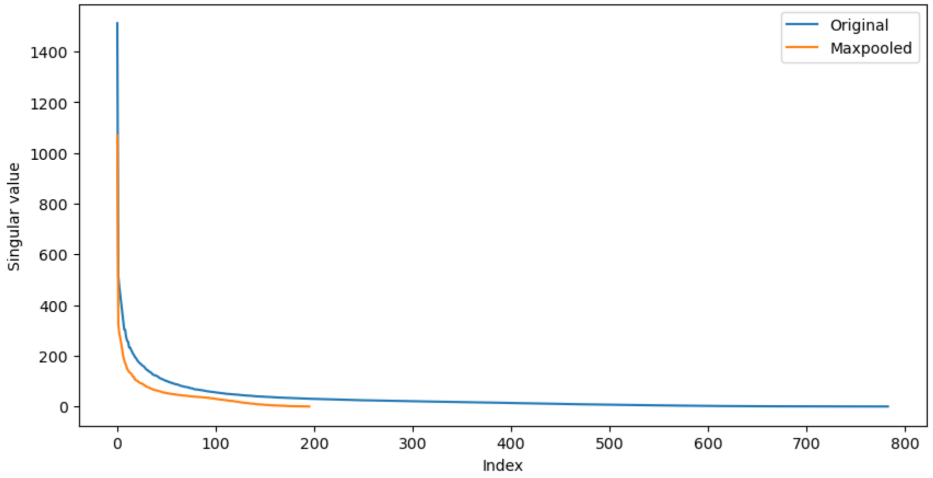
Original images



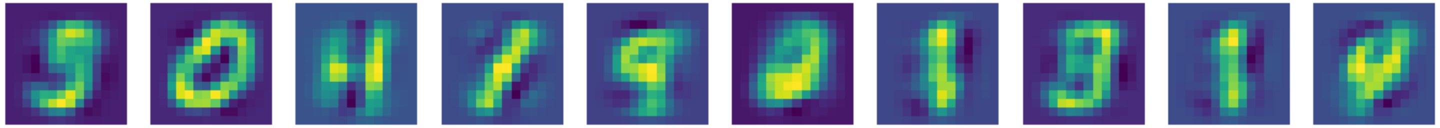
Maxpooled images



Singular values of the MNIST dataset



Reconstructed images using maxpooling



Kernel (2*2)
Stride = 2

Image size (28*28) -> (14*14)

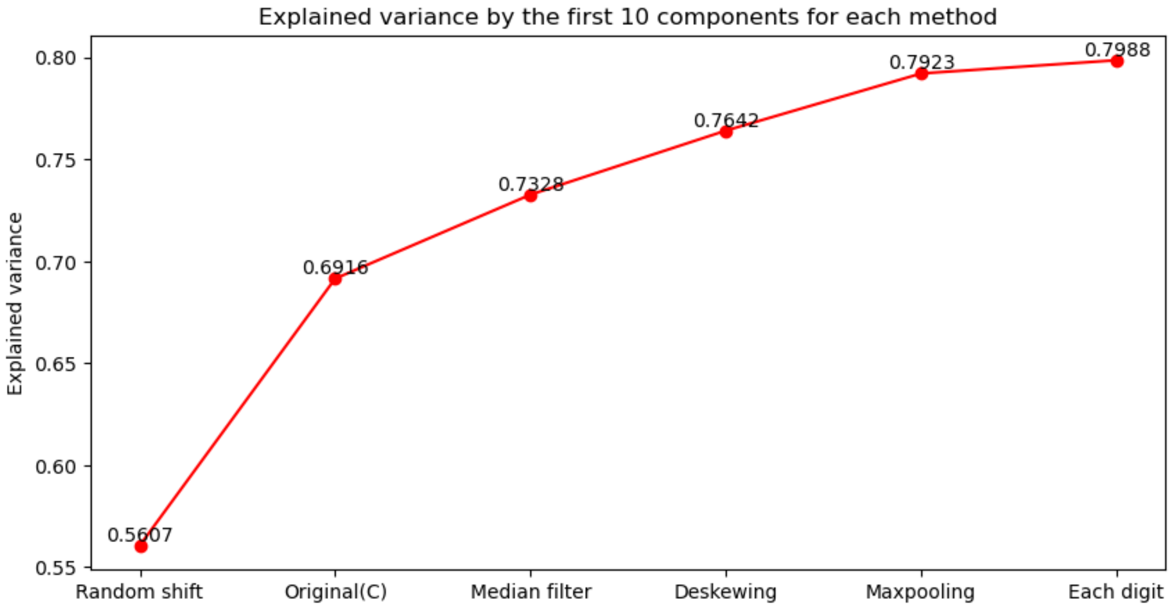
Explained variance by first 10 components (original): 0.6916

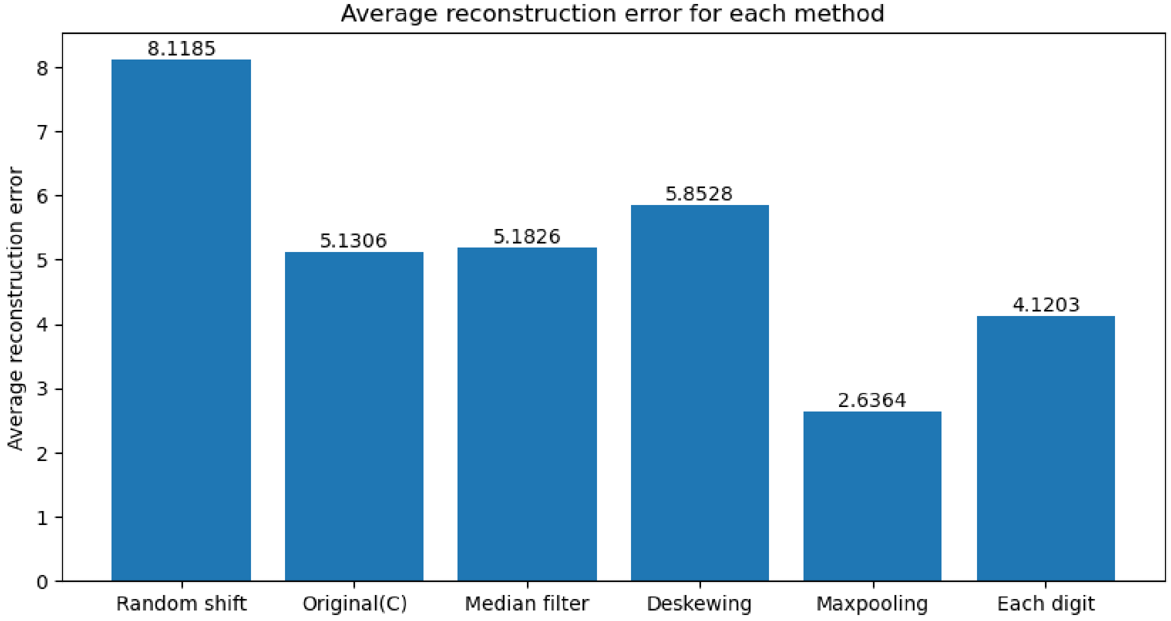
Explained variance by first 10 components: 0.7923

Conclusion

- Calculate reconstruction loss
- L2 distance between original image and reconstructed image
- Take mean of all images

- Except for Max Pooling transformation
  - Image size (28*28) -> (14*14)
  - L2 distance between max pooled images and reconstructed images





Questions?

uiowa.edu

Roy Huang
University of Iowa Electrical and Computer Engineering Department

Vincent Cai
University of Iowa Computer Science Department

Mingwei Xi
University of Iowa Mathematics Department

Reference

YouTube video:

SVD Visualized, Singular Value Decomposition explained by Visual Kernel (2022). Retrieved from https://www.youtube.com/watch?v=vSczTbgc8Rc&t=53s

Website:

Erik Storrs (2021). Explained: Singular Value Decomposition (SVD). Retrieved from https://storrs.io/svd/

Website:

Dibya Ghosh and Alvin Wan (). Deskewing. Retrieved from https://fsix.github.io/mnist/Deskewing.html