CS215 Assignment 2

Aryan Mathe & Yash Ruhatiya

October 2022

Contents

1 Sampling within a Euclidean Plane

1.1 Uniform Points Generation Inside Ellipse

The distance of the point on the ellipse centred at the origin is given by the equation

$$r(\theta) = \frac{ab}{\sqrt{a^2 \sin^2\theta + b^2 \cos^2\theta}}$$

Now, we know that

$$\frac{dA}{A} = p(\theta)d\theta$$

$$\frac{\frac{1}{2}r^2(\theta)d\theta}{\pi ab} = p(\theta)d\theta$$

$$p(\theta) = \frac{ab}{2\pi(a^2 \sin^2\theta + b^2 \cos^2\theta)}$$

CDF of the following probability distribution can be found out as

$$CDF(\theta) = \int_0^\theta p(\theta)d\theta$$

Solving the above integral will give the following result,

$$CDF(x) = \begin{cases} \frac{1}{2\pi}\tan^{-1}(2\tan(x)) & \theta \in (0, \frac{\pi}{2}) \\ \frac{1}{2} - \frac{1}{2\pi}\tan^{-1}(2\tan(x)) & \theta \in (\frac{\pi}{2}, \pi) \\ \frac{1}{2} + \frac{1}{2\pi}\tan^{-1}(2\tan(x)) & \theta \in (\pi, \frac{3\pi}{2}) \\ 1 - \frac{1}{2\pi}\tan^{-1}(2\tan(x)) & \theta \in (\frac{3\pi}{2}, 2\pi) \end{cases}$$

Now solving for the **Inverse CDF** of the following probability distribution we get,

$$CDF^{-1}(x) = \begin{cases} \tan^{-1}(\frac{\tan(2\pi x)}{2}) & x \in (0, \frac{1}{4}) \\ \pi + \tan^{-1}(\frac{\tan(2\pi x)}{2}) & x \in (\frac{1}{4}, \frac{3}{4}) \\ 2\pi + \tan^{-1}(\frac{\tan(2\pi x)}{2}) & x \in (\frac{3}{4}, 1) \end{cases}$$

Now we have to again apply the same process to generate the sample points within the **radial line segment** at a particular θ as sampled by the above distribution.
Again,

$$\frac{dA}{A} = P(r)dr$$

$$\frac{2\pi r dr}{\pi R^2} = P(r)dr$$

$$P(r) = \frac{2r}{R^2}$$

where **r** is the variable and **R** is the Radius of the circle within which the point has to be generated.

$$CDF(r) = \int_0^r \frac{2r}{R^2}$$

$$CDF(r) = \frac{r^2}{R^2}$$

So clearly **Inverse CDF** is given as,

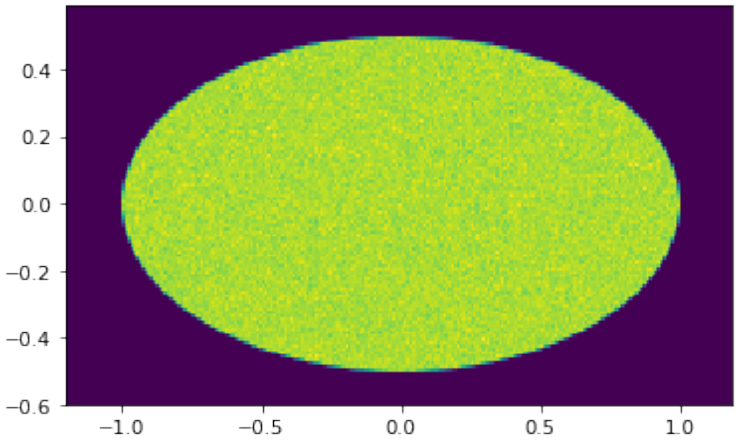
$$CDF^{-1}(x) = R\sqrt{x}$$

Huh! finally done with the math, it's time to sample points inside the ellipse. Using the **inverse CDF** for both the distributions we will generate **r** and θ Using parametric coordinates we will generate the points as,

$$x = r\cos(\theta)$$

$$y = r\sin(\theta)$$

Plotting the points on a **2D histogram** plot gives the following result.



1.2 Uniform Points Generation Inside Triangle

Let $\vec{A}, \vec{B}$ and $\vec{C}$ be three position vectors representing the vertex of the given triangle. First, we generated two vectors along two edges of the triangle,

$$\vec{v1} = \vec{B} - \vec{A}$$

$$\vec{v2} = \vec{C} - \vec{A}$$

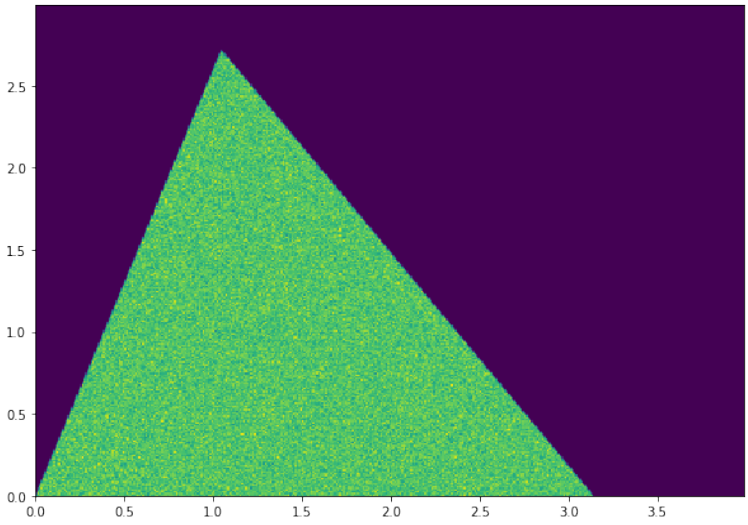
Then we generated two random numbers **a** and **b** in the range $[0, 1]$
Now, consider the **linear combination**

$$(a\vec{v1} + b\vec{v2}) + \vec{A}$$

It will generate the points uniformly in the region bound by the parallelogram with sides formed by the vectors $\vec{v1}, \vec{v2}$.

To get the desired result of uniform distribution of points inside the triangle we will just **reflect** all the points that will be generated outside of the required triangle just doing one **inequality** check using the equation of the line joining $\vec{B}$ and $\vec{C}$.

This algorithm will generate the desired points within the triangular region as depicted in the image below



2 Multivariate Gaussian

2.1 Sample Points Generation

We just followed general principles to sample random vectors from **gaussian distribution**. We know that,

$$X = AW + \mu$$

where X is the gaussian random vector with **Mean** as μ , and AA^T as **Covariance Matrix**.
Also, W is a standard gaussian random vector.
Let

$$A = RS$$

where R is the **rotation** matrix and S is the **scaling** matrix

$$\begin{aligned} C &= AA^T \\ C &= (RS)(RS)^T \\ C &= RSS^TR^T \\ C &= RS^2R^T \quad \text{(since S is diagonal)} \end{aligned}$$

Also for rotation matrix R we have, $RR^T = I$

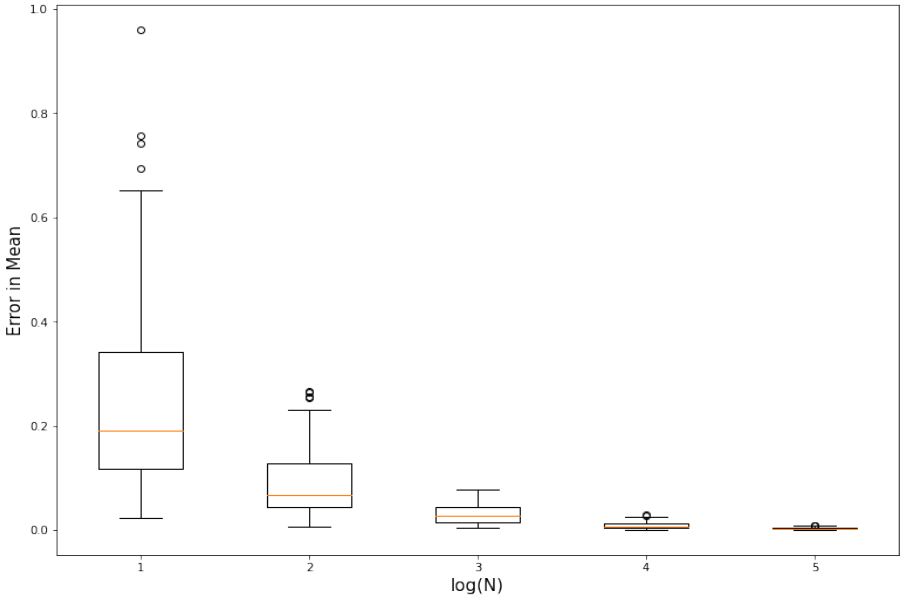
$$R^TCR = S^2$$

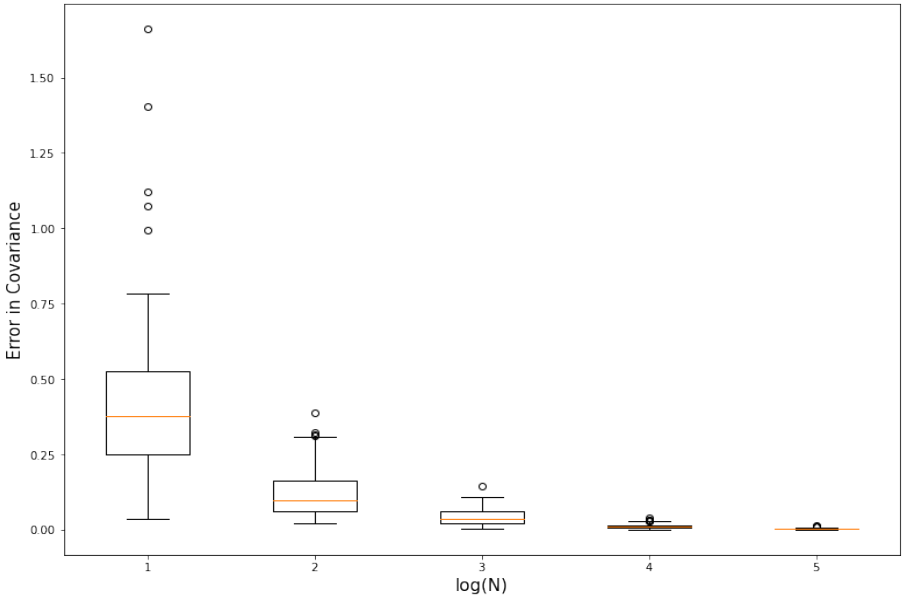
where S^2 is a diagonal matrix, and using **spectral theorem** we can clearly say that matrix R is the matrix consisting of eigenvectors of covariance matrix as its column vectors and S^2 as a diagonal matrix consisting of eigenvalues of Covariance matrix. Hence, we now have

$$\begin{aligned} X &= RSW + \mu \\ X &= R(R^TCR)^{1/2}W + \mu \end{aligned}$$

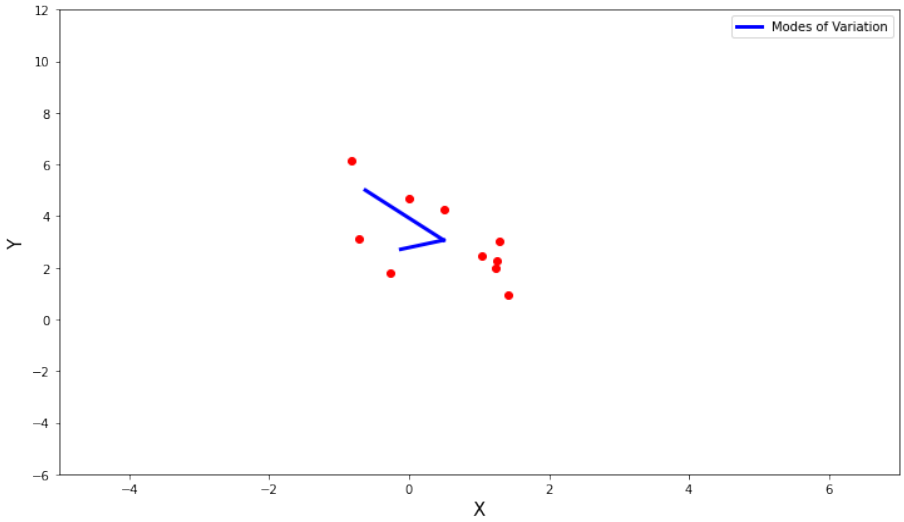
where matrix R^TCR is a diagonal matrix wit all entries as non-negative values, so its **sqrt** is defined.
Now we can generate a standard gaussian random vector and put that into this equation to ultimately sample the points according to the given **Covariance** and **Mean**.

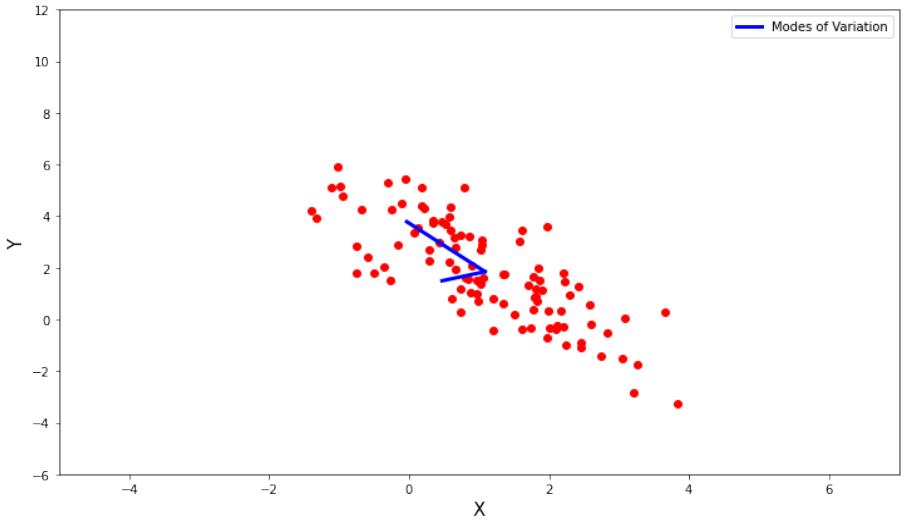
2.2 Mean and Covariance Error Plot

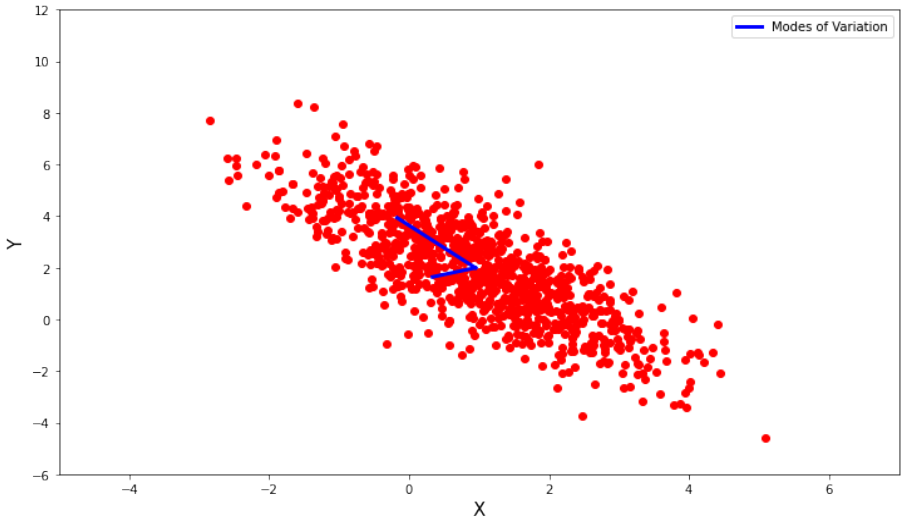


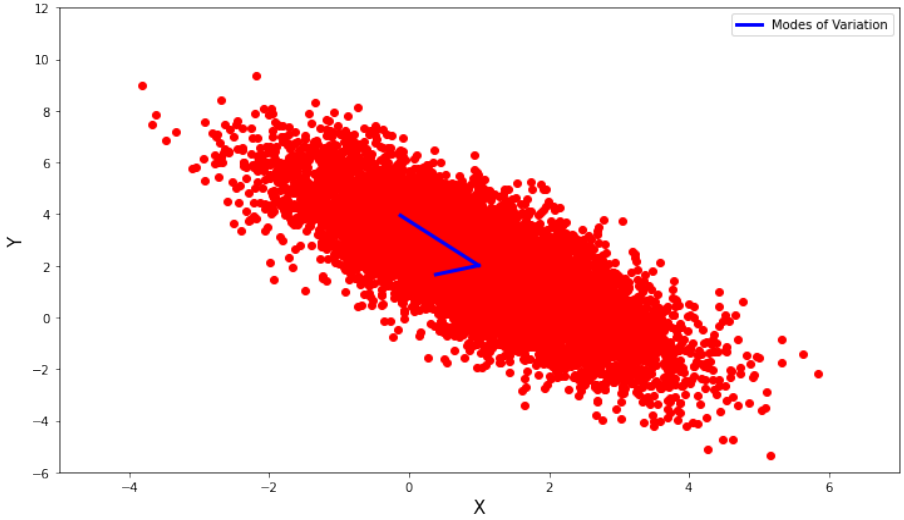


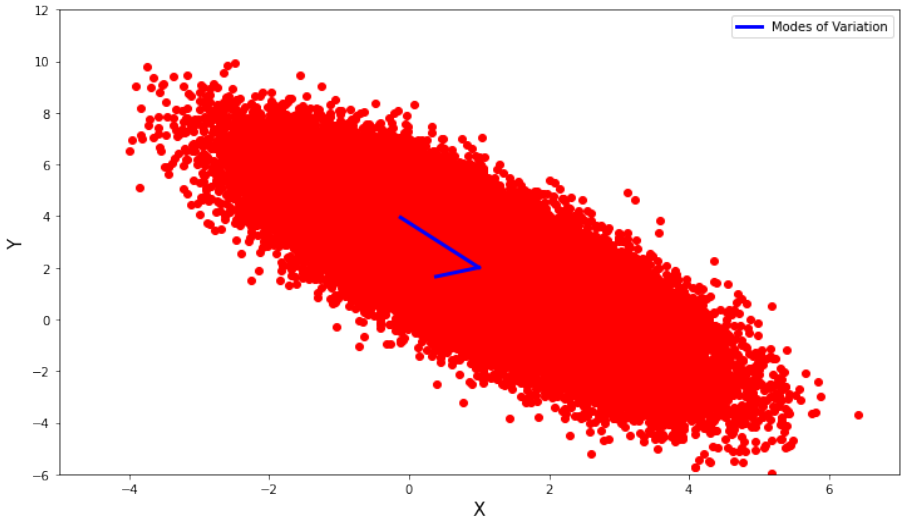
2.3 Comparison between Data and Modes of Variation






Modes of Variation
12
10
8
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6
Y
X


Modes of Variation
12
10
8
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6
Y
X



3 PCA and Hyperplane Fitting

3.1 Developing Linear Relationship via PCA

To develop a linear relationship between the two random variables we proceed as follows
Since the **Covariance** matrix is real and symmetric, we can use eigen decomposition to get its eigenvectors.

$$C = O^T DO$$

where O is an orthogonal matrix with its columns as eigenvectors and D is a diagonal matrix with its diagonal entries as the eigenvalues(σ^2).
So if we choose the eigenvector corresponding to the maximum eigenvalue or maximum σ^2 we will get the direction with maximal mode of variation.

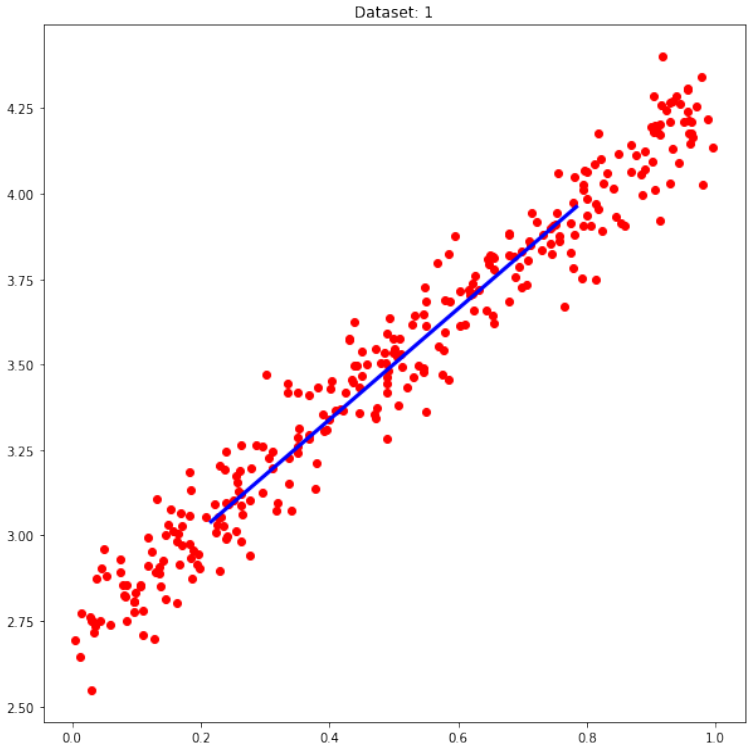
From the given set of data points, first, we calculated the sample covariance given by **MLE** as,

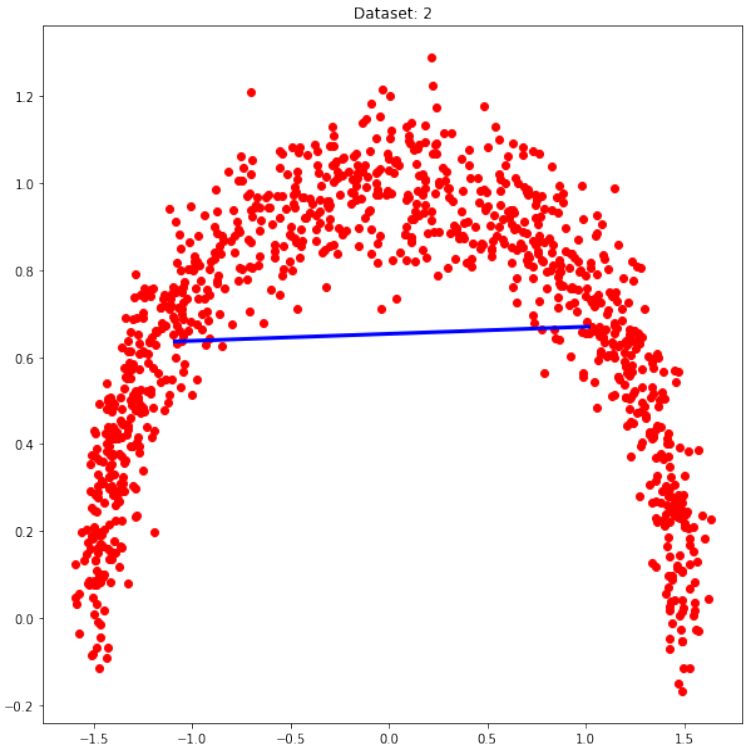
$$\frac{\sum_i^N (X_i - E[X]) \cdot (X_i - E[X])^T}{N}$$

Then calculated the eigenvector with **maximum eigenvalue** using the method as described above.

Using this eigenvector we generated a line segment in the direction of this eigenvector passing through the sample mean, which will give the best approximation for a linear relationship between random variables X and Y.

3.2 Comparison between the Linear Relations through PCA for different data sets





As we can clearly observe from the data plots the **PCA** gives a very good approximation for the first set of data points because they seem to follow an approximated linear relationship.

On the other hand, the second set of data points doesn't follow a linear relationship. Hence, our attempt to develop a linear relationship among those data points seems to have a signification **deviation** from the actual data.

Hence to have a good approximation for the second set of data points, we need to **tune** some parameters based on the structure of the plot to have a proper relation.

4 Principal Component Analysis

4.1 Eigenvalues of the covariance matrix

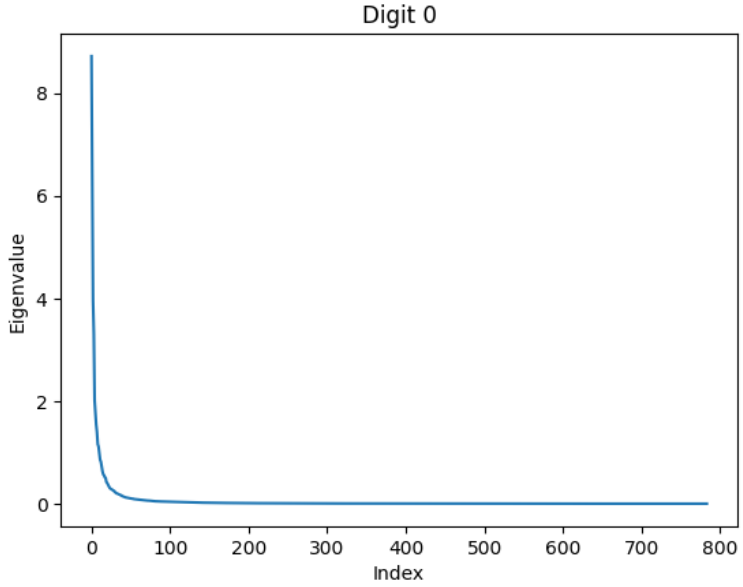
For each digit, it can be observed from the graph that the eigenvalues **decrease rapidly**. It is observed that for each of the digits eigenvalues at indexes greater than 100 are almost insignificant and tend to zero. So out of the 784 eigenvectors only almost **100** of them are significant.

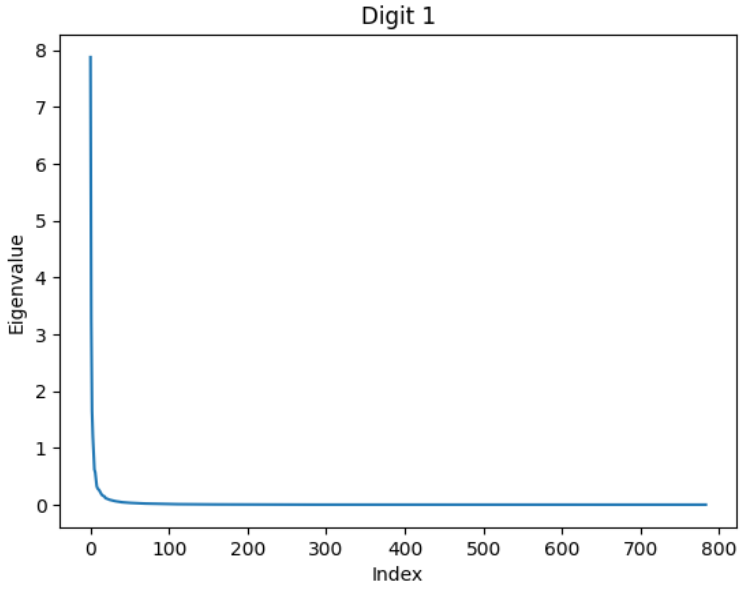
From these observations, it can be inferred that only some of the eigenvalues (and hence the corresponding eigenvectors) have major contributions in forming the dataset and hence this dataset can be effectively reconstructed by **dimensionality reduction**, as is done in Q5.

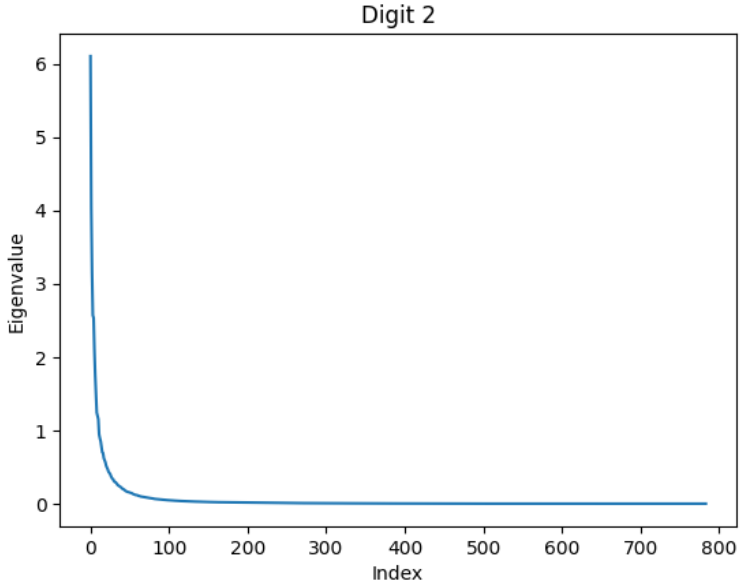
This happens because the vectors in these datasets are **not uncorrelated**, and there is a certain pattern observed in the data.

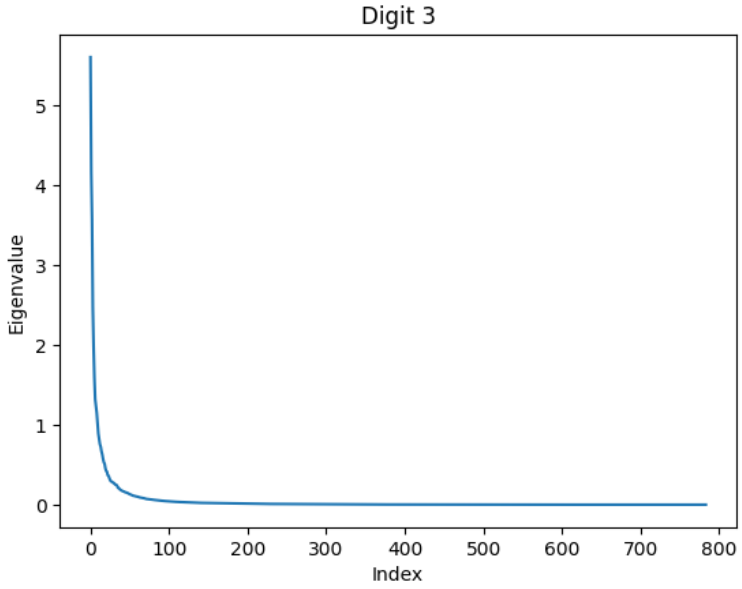
Here the pattern can be thought of as the structure of a digit which remains the same, but it is only the orientation and other slight changes that occur due to different handwritings of different people.

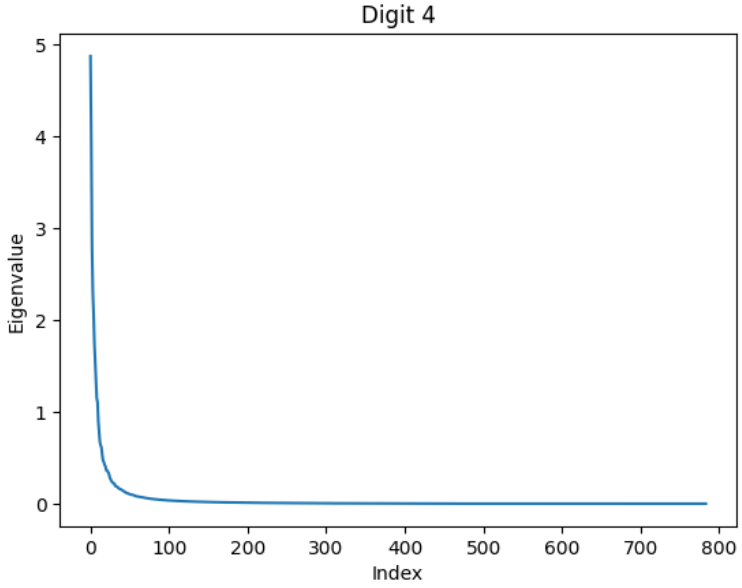
PCA helps us convert these correlated vectors into a smaller set of **independent principal components** while also retaining the variation in the original dataset.

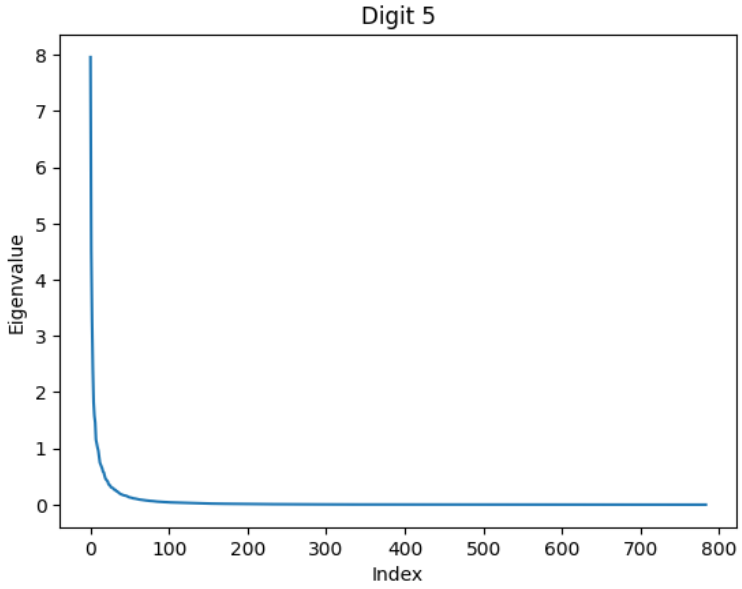


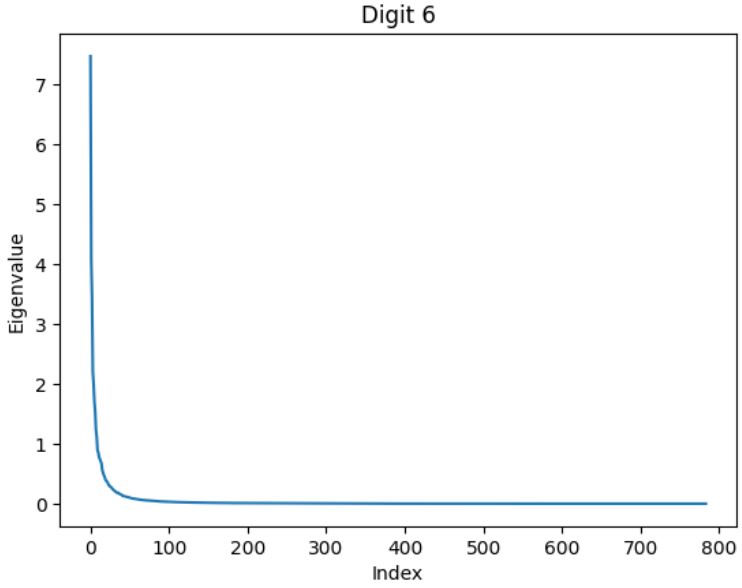

Digit 1
Eigenvalue
Index
0
1
2
3
4
5
6
7
8
0
100
200
300
400
500
600
700
800

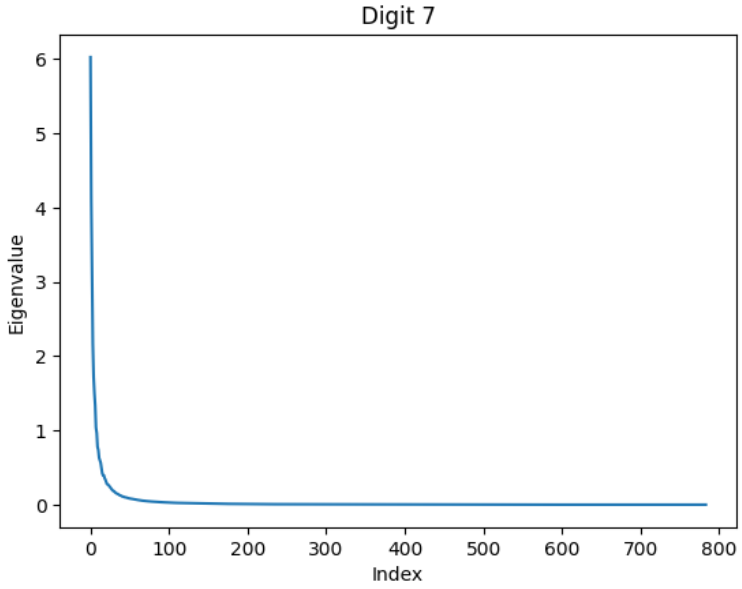

Digit 2
Eigenvalue
Index
0
1
2
3
4
5
6
0
100
200
300
400
500
600
700
800

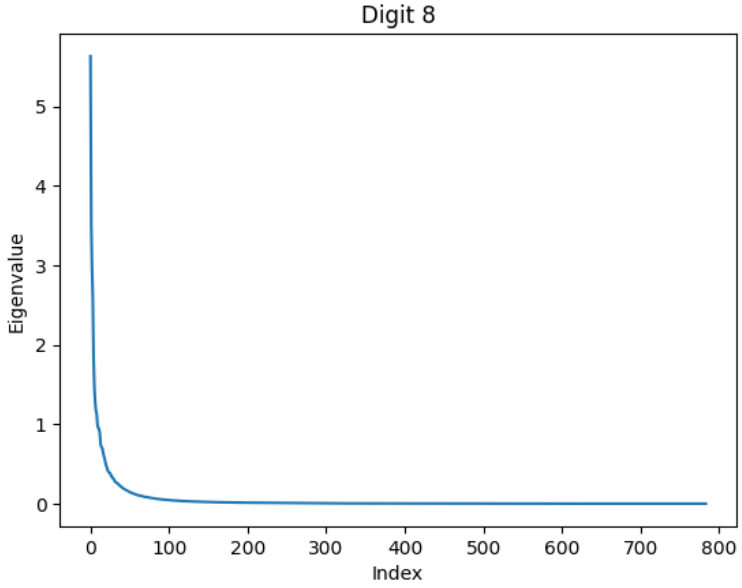

Digit 3
Eigenvalue
Index
0
1
2
3
4
5
0
100
200
300
400
500
600
700
800

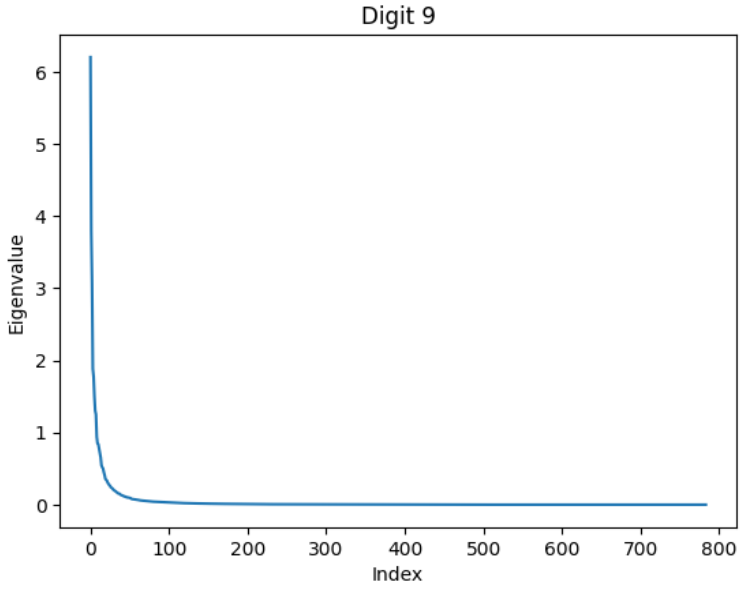

Digit 4
Eigenvalue
Index
0
1
2
3
4
5
0
100
200
300
400
500
600
700
800


Digit 5
Eigenvalue
Index
0
1
2
3
4
5
6
7
8
0
100
200
300
400
500
600
700
800


Digit 6
Eigenvalue
Index
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800


Digit 7
Eigenvalue
Index
0
1
2
3
4
5
6
0
100
200
300
400
500
600
700
800

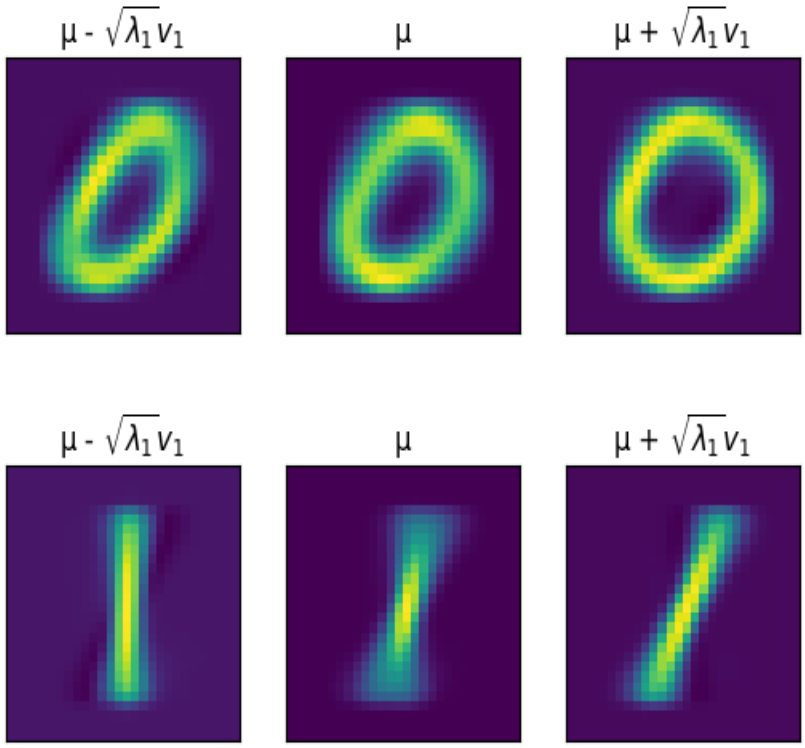

Digit 8
Eigenvalue
Index
0
1
2
3
4
5
0
100
200
300
400
500
600
700
800

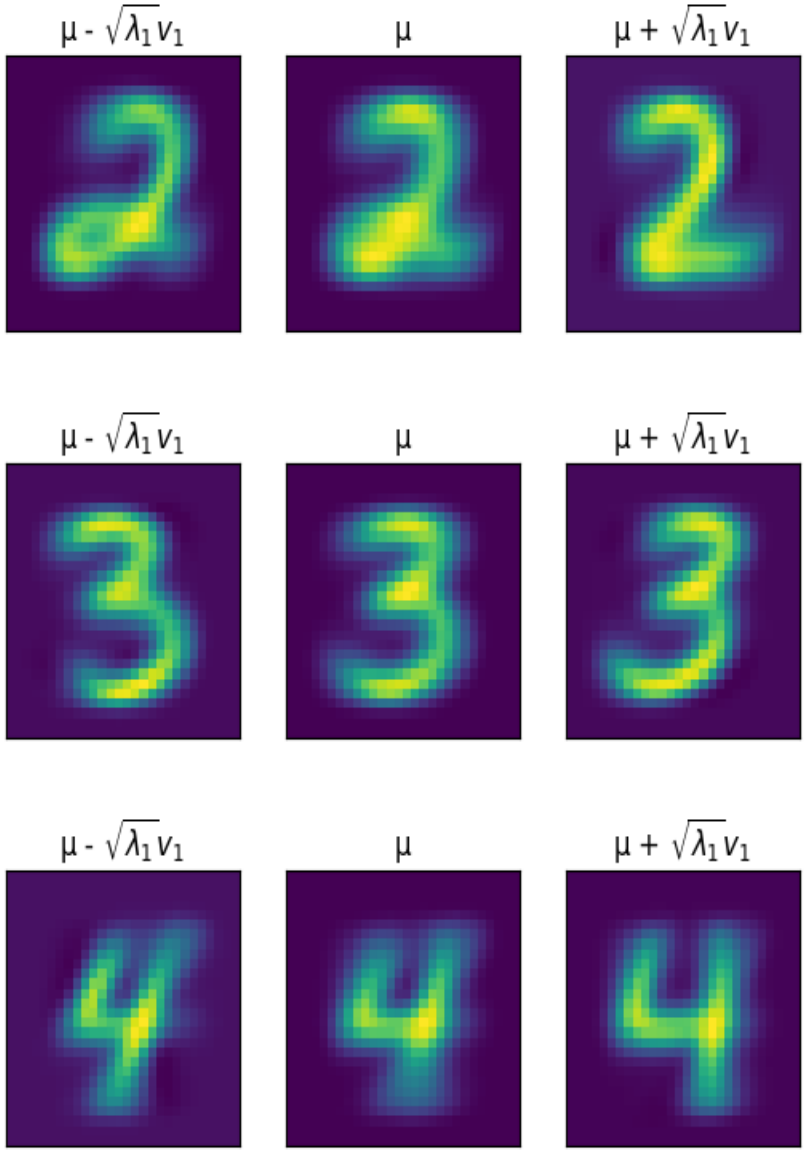

Digit 9
Eigenvalue
Index
0
1
2
3
4
5
6
0
100
200
300
400
500
600
700
800

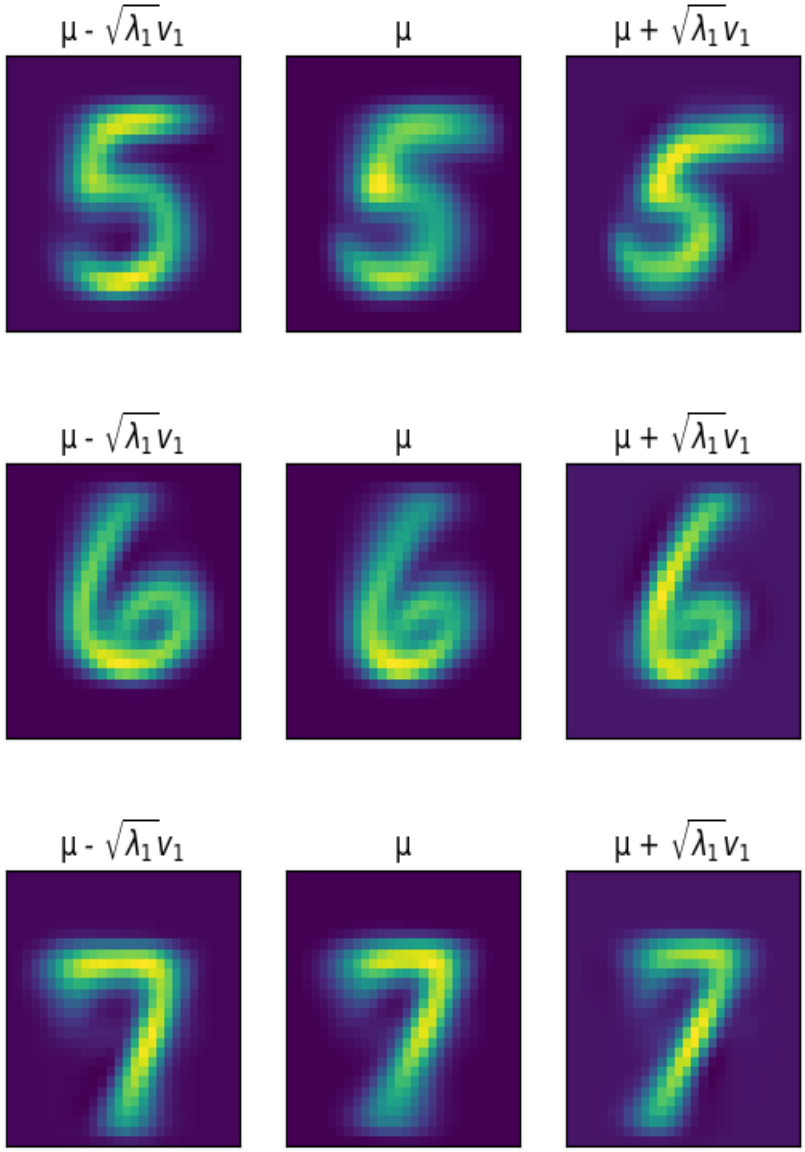
4.2 Mean and the mode of principal variations around the mean

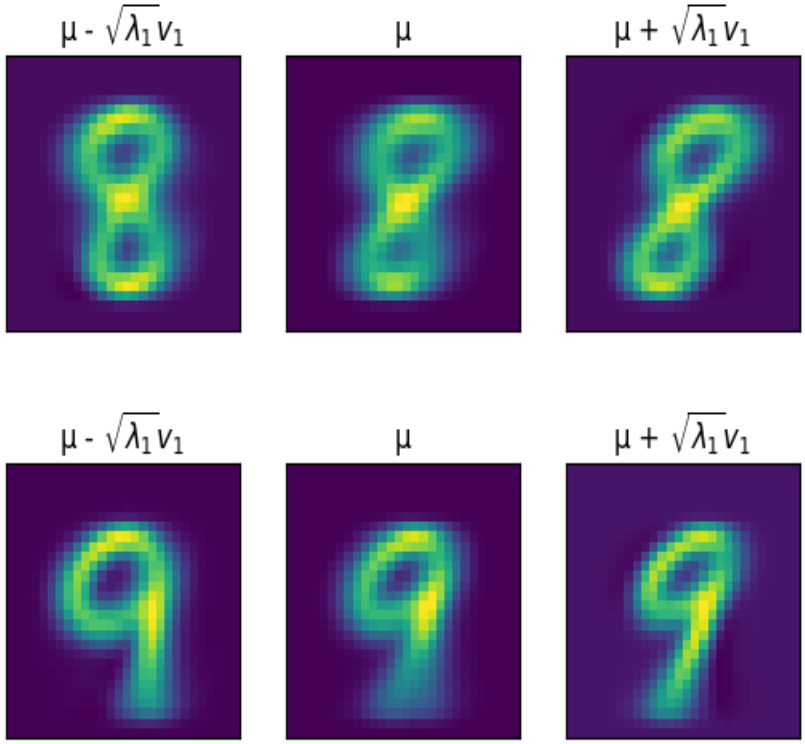
The three figures obtained for each digit are quite similar, which is justified because the basic structure of the digit should remain the same and small changes arise depending on the person.

As an example, in case of digit 1 it can be observed that in the principal modes of variation the digit 1 is either straight of right rotated/slanted and hence we can conclude that almost everyone writes 1 either straight or right rotated , but almost no one writes it left tilted.




$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$
$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$
$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$

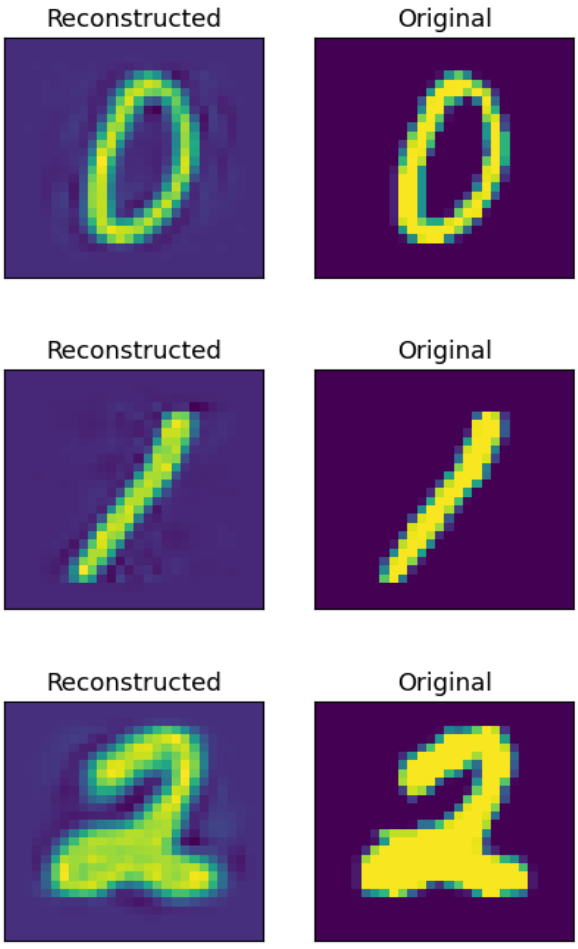

$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$
$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$
$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$

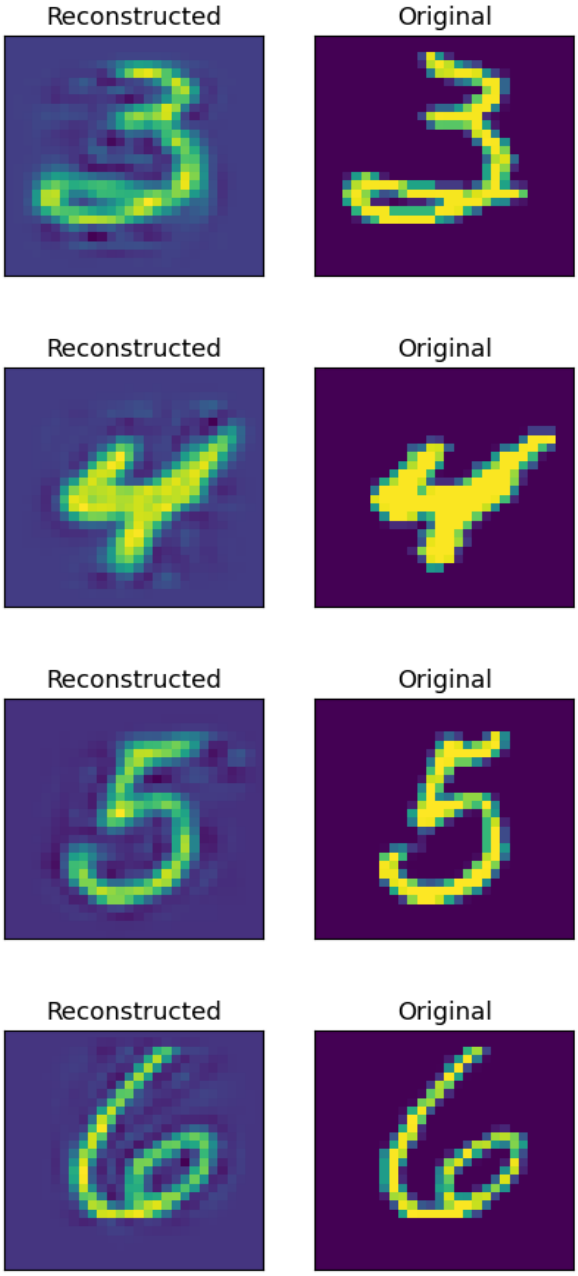

$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$
$\mu - \sqrt{\lambda_1} v_1$
μ
$\mu + \sqrt{\lambda_1} v_1$

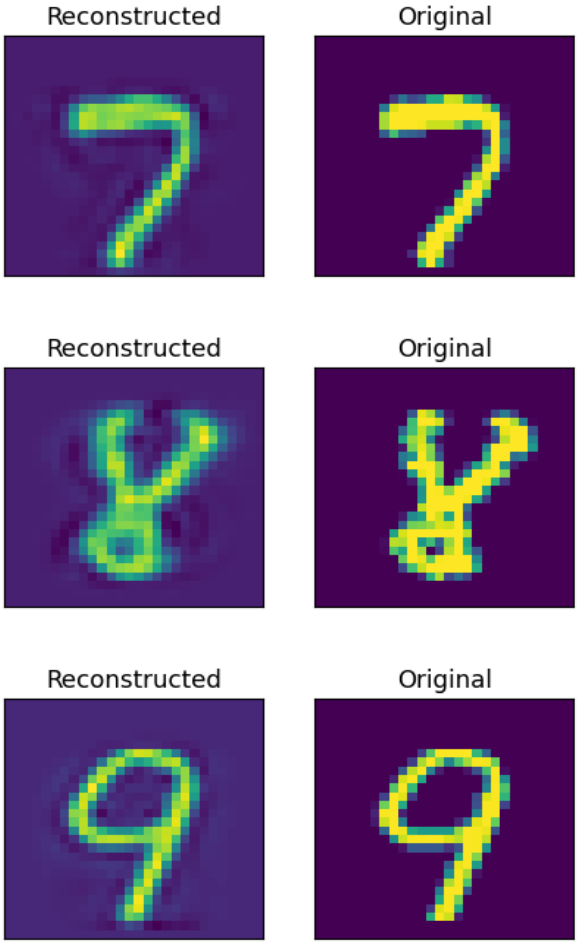
5 PCA for Dimensionality Reduction

Please refer to **part 2 of question 6** for the algorithm, as the same algorithm is being used in both places. The only difference being that the top 4 eigenvectors have been used there, while we are using 84 eigenvectors here that are forming the basis of an 84-dimensional set.

The problem here really reduces to finding the closest possible approximation of the original image as we are given with the basis of the 84-dimensional basis to which the image is to be reduced.

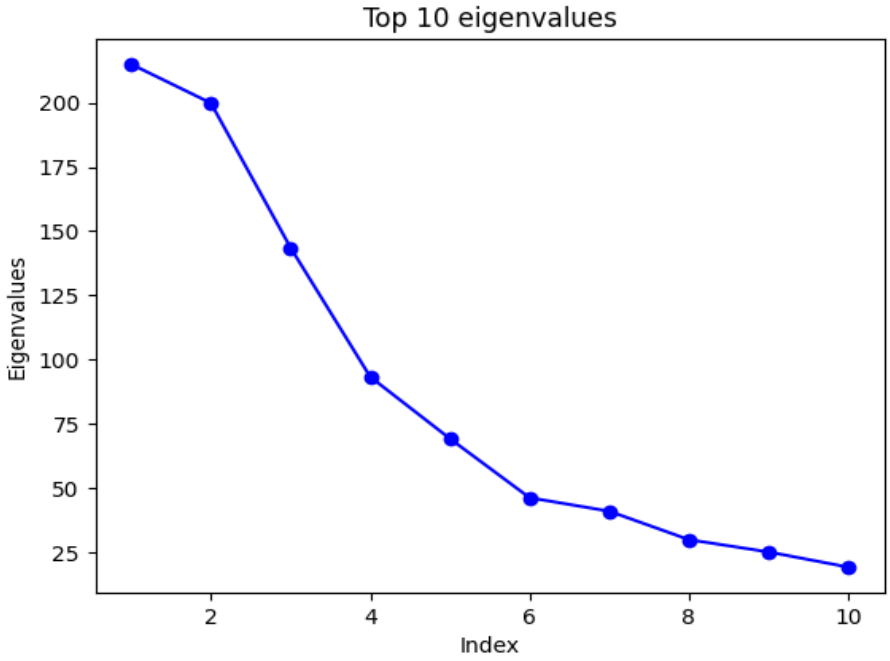


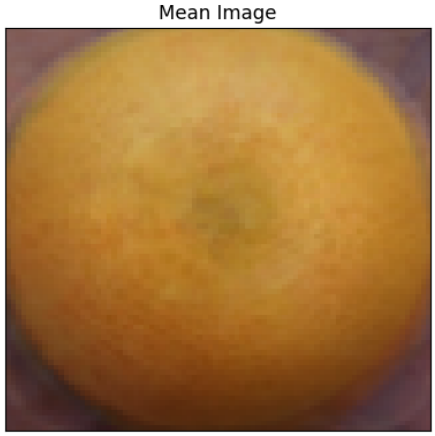

Reconstructed
Original
Reconstructed
Original
Reconstructed
Original
Reconstructed
Original


Reconstructed
Original
Reconstructed
Original
Reconstructed
Original

6 PCA for Another Image Data

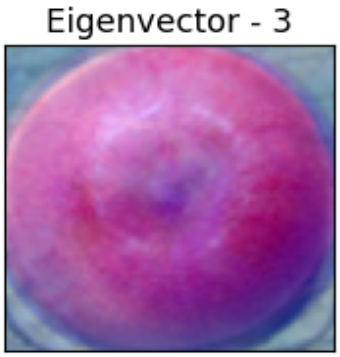
6.1 Mean, Eigenvectors and Eigenvalues

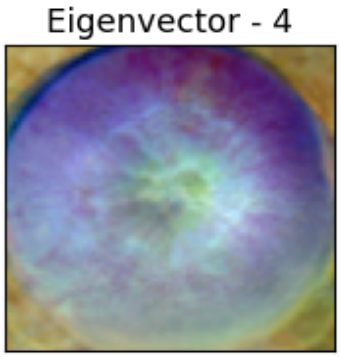



Mean Image


Eigenvector - 1


Eigenvector - 2


Eigenvector - 3


Eigenvector - 4

6.2 Image reconstruction and Dimensionality Reduction

The problem of finding the closest representation of the image using a linear combination of the eigenvectors added to the mean, taking the measure of closeness as Frobenius norm essentially reduces to the least square optimization problem once we reshape the matrices to a column vector.

We use this idea here to get the closest image. Given below is a detailed mathematical description of the analogy between the least square problem and this question. We can write our reconstructed image as,

$$AX + \mu$$

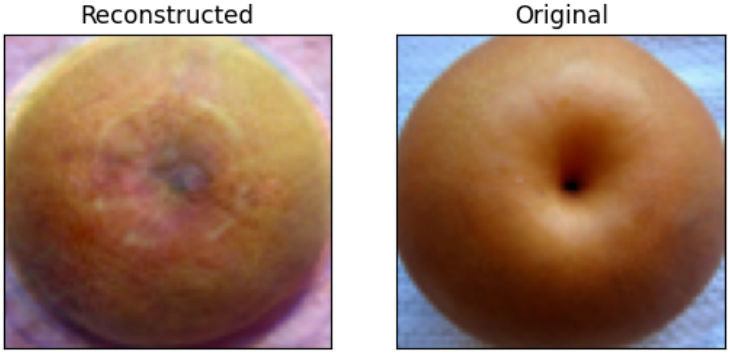
where A is the matrix consisting of the **4 eigenvectors** with maximum corresponding eigenvalue, X is the coefficients of their linear combination, and μ is the mean of the given data.
Now to get the closest approximation of this expression with the image, we can simply have an optimization problem,

$$min||AX + \mu - I||$$

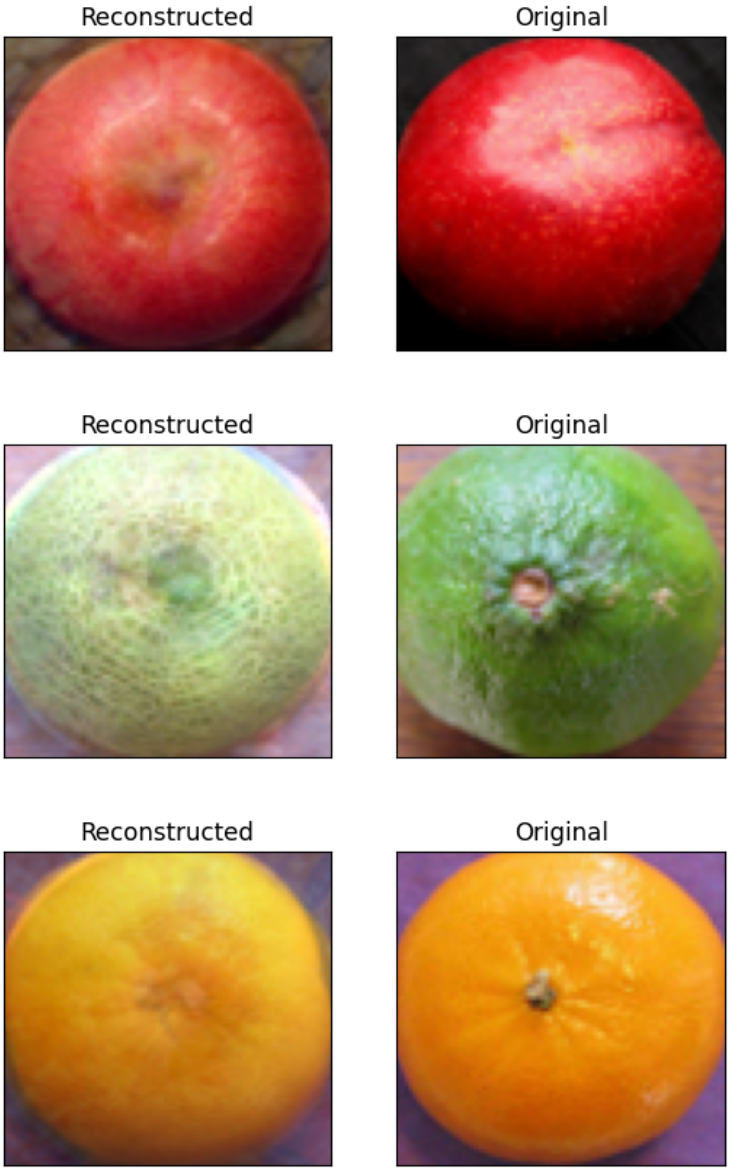
where I is the given **Image**.
Now, using the standard result of this optimization problem we have the following equation

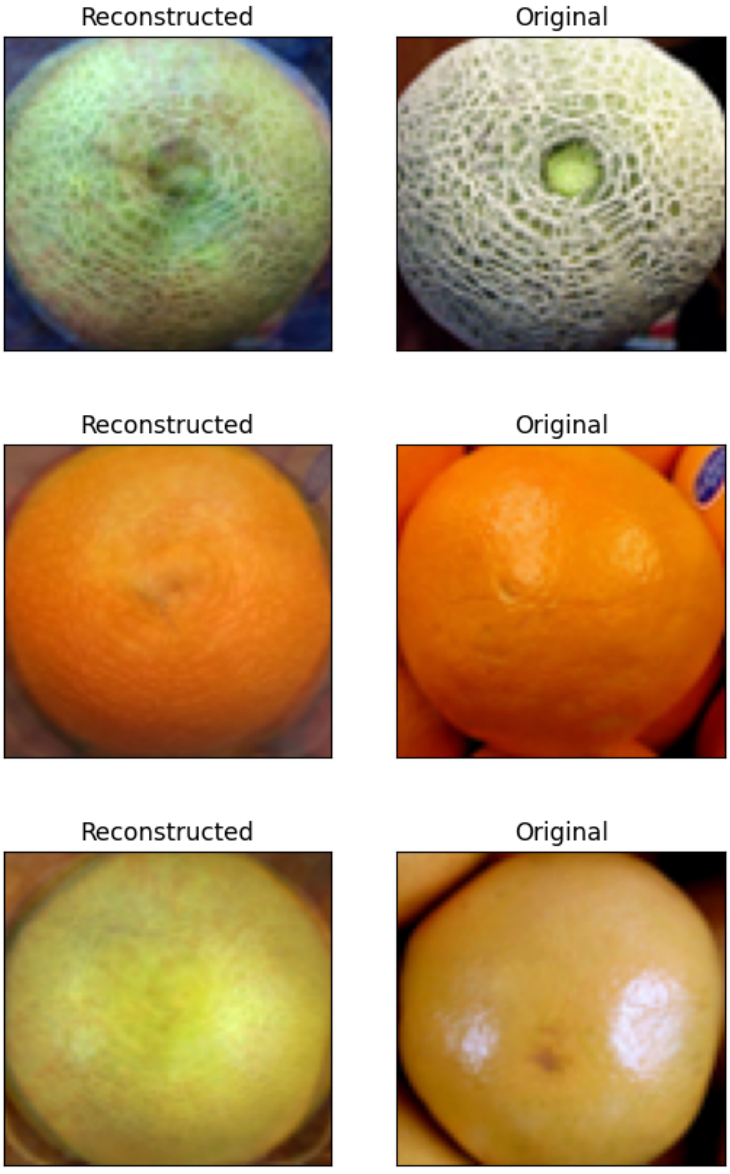
$$A^T AX = A^T(I - \mu)$$

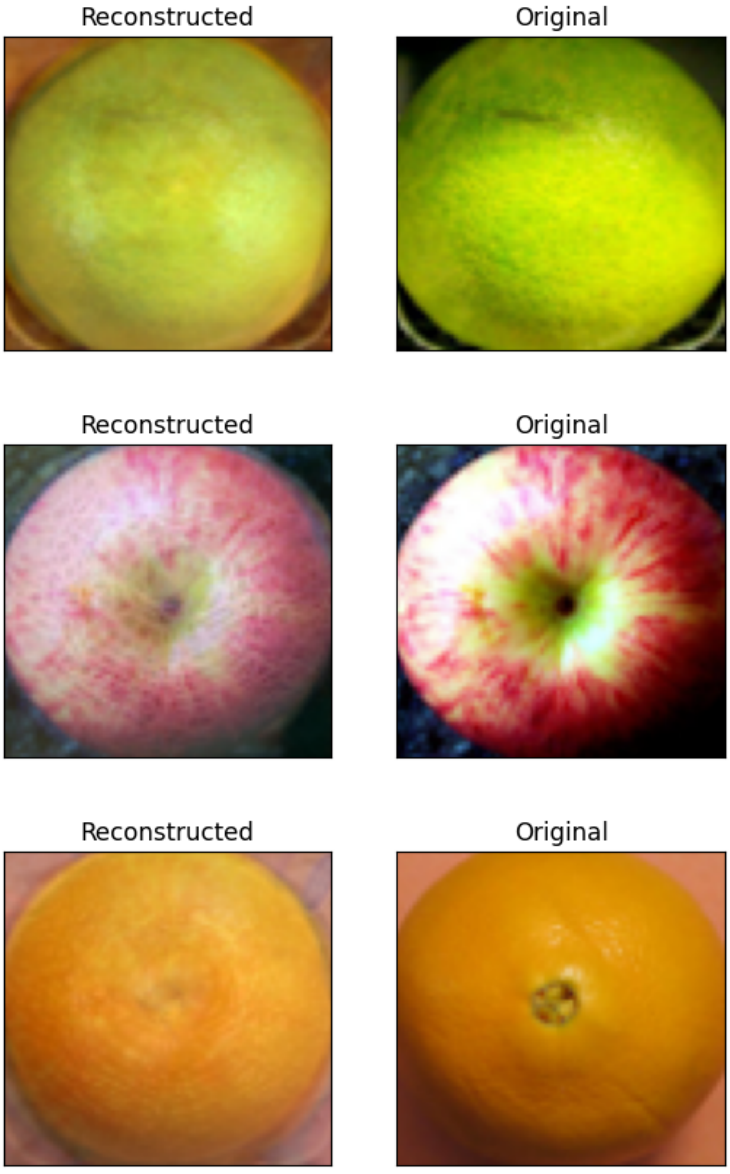
where X will be our coefficients of combination.
After calculating X we can simply find the reconstructed image as $AX + \mu$, and we are done!

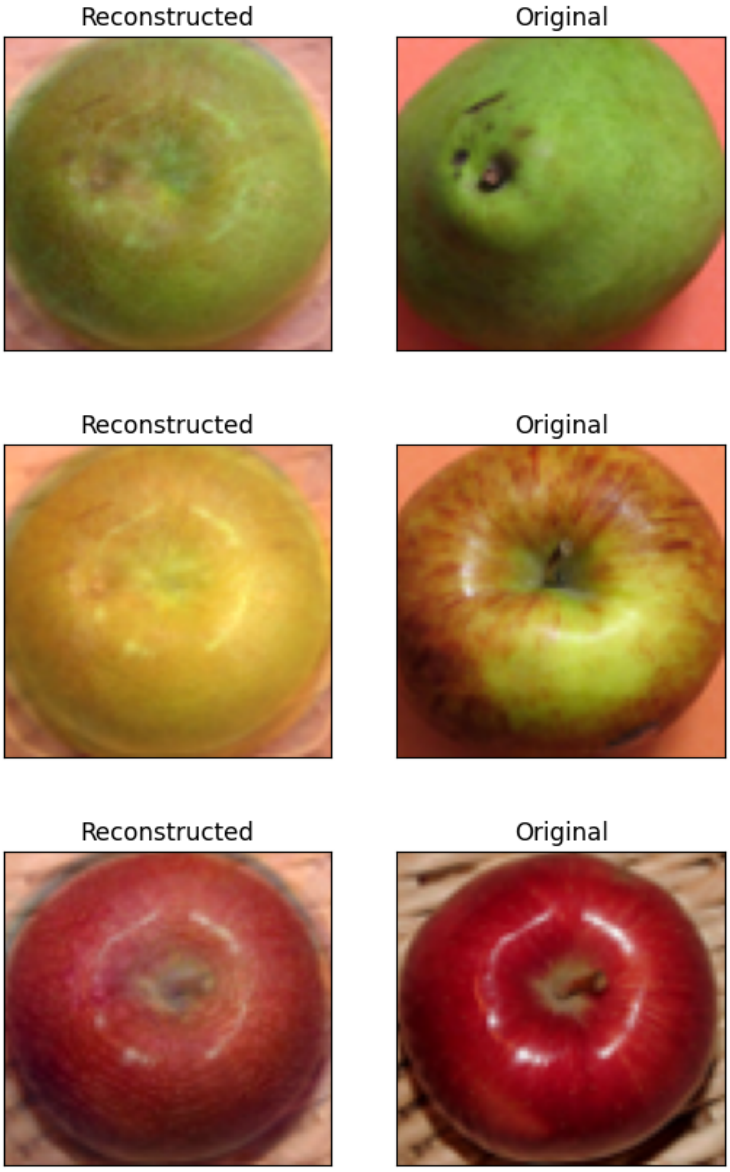



Reconstructed
Original
Reconstructed
Original
Reconstructed
Original


Reconstructed
Original
Reconstructed
Original
Reconstructed
Original


Reconstructed
Original
Reconstructed
Original
Reconstructed
Original


Reconstructed
Original
Reconstructed
Original
Reconstructed
Original


Reconstructed
Original
Reconstructed
Original
Reconstructed
Original

6.3 Sampling Random Images

First, we generated a gaussian random (4×1) vector X, and generated a new image using the below equation,

$$I_{new} = RSX + \mu$$

where R is the matrix with its columns as the top four eigenvectors of the covariance matrix of the given dataset, S is a diagonal matrix with its diagonal entries as the eigenvalues corresponding to those eigenvectors and μ is the mean of the data.

Here we are assuming that only these 4 eigenvectors are majorly responsible for generating the data and the others are effectively ignored.