Assignment 2

Harshvardhan Agarwal 200050050

Parth Dwivedi 200050100

October 2021

1 Euclidean Plane Sampling

1.1 Uniform Distribution Inside Ellipse

Here, it is easier to work with polar coordinates rather than cartesian coordinates, so let us write each point inside the ellipse as a function of paramters r and θ as :-

$$x = r\cos\theta$$
$$y = \frac{rb}{a}\sin\theta$$

where $r \in [0, a]$ and $\theta \in [0, 2\pi)$ and a and b are the semi-major and semi-minor axis of the ellipse and b is the semi-minor axis of the ellipse.

Since the distribution of points is uniform in the ellipse and 0 everywhere else, the joint probability density $f(x, y)$ must be the same at all points within the ellipse and 0 elsewhere.

Since we have using the polar coordinate system for calculation, but we are still taking the infinitesimally small area ($dxdy$) in the Cartesian System, we shall require the Jacobian for transformation:-

$$\begin{aligned}\mathbf{J} &= \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} \\ &= \begin{bmatrix} cos\theta & -r\sin\theta \\ \frac{b}{a}\sin\theta & \frac{rb}{a}\cos\theta \end{bmatrix} \\ \therefore |\mathbf{J}| &= \frac{rb}{a}\end{aligned}$$

Since, area of ellipse $= \pi ab$ and integration of joint PDF over the real plane must be 1, we have :-

$$f(x, y) = \begin{cases} \frac{1}{\pi ab} & \frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1 \\ 0 & otherwise \end{cases}$$

Since $dxdy = |\mathbf{J}|\, drd\theta$, we can write the joint pdf in polar coordinate as :-

$$f(r, \theta) = \begin{cases} \frac{r}{\pi a^2} & 0 \leq r \leq a,\ 0 \leq \theta < 2\pi \\ \\ 0 & otherwise \end{cases}$$

Let us assume the Random Variable R and the Random Variable Θ to be independent random variables such that :-

$$R = a\sqrt{U}$$
$$\Theta = 2\pi U$$

where U is the Uniformly Distributed Random Variable from 0 to 1.

We shall now prove that the joint PDF resulting from this is indeed the PDF that we require ($f(r, \theta)$).

For Random Variable R,

$$\begin{aligned}
\mathbb{P}(R \leq r) &= \mathbb{P}(a\sqrt{U} \leq r) \\
&= \mathbb{P}(U \leq \frac{r^2}{a^2}) \\
&= \begin{cases} 1 & r > a \\ \frac{r^2}{a^2} & 0 \leq r \leq a \\ 0 & r < 0 \end{cases} \\
\text{(PDF of R)} \quad f_R(r) &= \begin{cases} \frac{2r}{a^2} & 0 \leq r \leq a \\ 0 & otherwise \end{cases}
\end{aligned}$$

For Random Variable Θ,

$$\begin{aligned}
\mathbb{P}(\Theta \leq \theta) &= \mathbb{P}(2\pi U \leq \theta) \\
&= \mathbb{P}(U \leq \frac{\theta}{2\pi}) \\
&= \begin{cases} 1 & \theta > 2\pi \\ \frac{\theta}{2\pi} & 0 \leq \theta \leq 2\pi \\ 0 & \theta < 0 \end{cases} \\
\text{(PDF of } \Theta) \quad f_\Theta(\theta) &= \begin{cases} \frac{1}{2\pi} & 0 \leq \theta \leq 2\pi \\ 0 & otherwise \end{cases}
\end{aligned}$$

As we have assumed R and Θ to be independent,

$$\begin{aligned}
\mathbb{P}(R \leq r, \Theta \leq \theta) &= \mathbb{P}(R \leq r)\mathbb{P}(\Theta \leq \theta) \\
\implies f(r, \theta) &= f_R(r) f_\Theta(\theta)
\end{aligned}$$

So, we get :-

$$f(r, \theta) = \begin{cases} \frac{r}{\pi a^2} & 0 \leq r \leq a,\ 0 \leq \theta < 2\pi \\ \\ 0 & otherwise \end{cases}$$

Which is exactly the probability distribution that we wanted to achieve, and so our initial assumption was correct.

Therefore, the algorithm to generate points uniformly distributed on an ellipse is -

1. Generate an instance of random variable R from an instance of uniformly distributed Random Variable U, using $R = a\sqrt{U}$.
2. Generate an instance of random variable Θ from another instance of uniformly distributed Random Variable U, using $\Theta = 2\pi U$.

3. Use the instances of R and Θ to get the coordinates of the point using $x = r\cos\theta$ and $y = \frac{rb}{a}\sin\theta$
4. Now we have generated an instance of the Random Variable that is uniformly distributed over an ellipse.

The correctness of this Algorithm is further supported by the empirical evidence below, where the points are uniformly distributed.

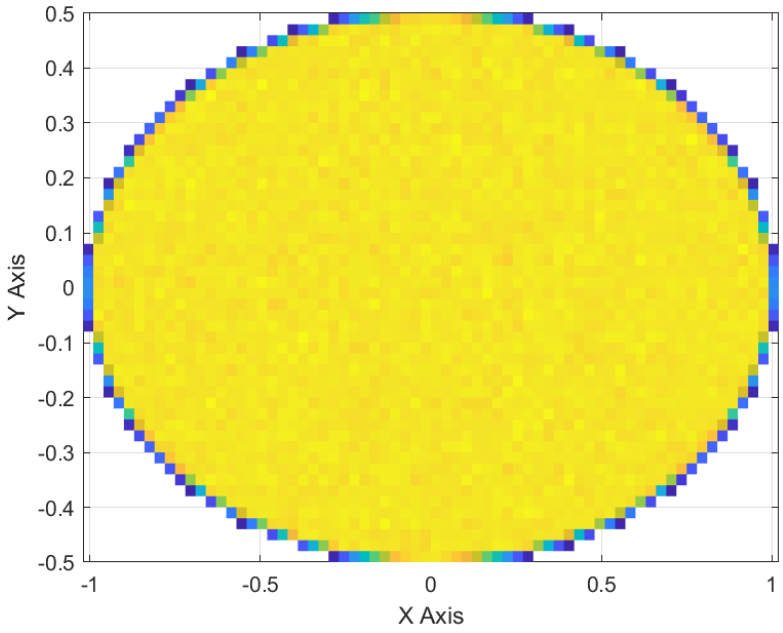

Figure 1: Histogram of 10^7 Independent Draws

1.2 Uniform Distribution Inside Triangle

Let us consider one of the vertices of the Triangle to be at Origin, and the two vectors forming the adjacent sides of the triangle as $\vec{u}$ and $\vec{v}$, such that both of them start from origin.

Therefore, any point inside the triangle can be written as:-

$$a\vec{u} + b\vec{v}$$

where $0 \leq a \leq 1, 0 \leq b \leq 1, a + b \leq 1$.

Considering $\vec{u}$ to be parallel to X axis (since in any case axes can be rotated appropriately), we have:-

$$x = a|\vec{u}| + b\left|\frac{\vec{v}\cdot\vec{u}}{|\vec{u}|}\hat{u}\right|$$
$$y = b\left|\vec{v} - \frac{\vec{v}\cdot\vec{u}}{|\vec{u}|}\hat{u}\right|$$

Therefore, the Jacobian in this case is:-

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial a} & \frac{\partial x}{\partial b} \\ \frac{\partial y}{\partial a} & \frac{\partial y}{\partial b} \end{bmatrix}$$
$$= \begin{bmatrix} |\vec{u}| & \left|\frac{\vec{v}\cdot\vec{u}}{|\vec{u}|}\hat{u}\right| \\ 0 & \left|\vec{v} - \frac{\vec{v}\cdot\vec{u}}{|\vec{u}|}\hat{u}\right| \end{bmatrix}$$
$$\therefore |\mathbf{J}| = |\vec{u} \times \vec{v}|$$

Since area of Triangle $= \frac{|\vec{u}\times\vec{v}|}{2}$, and integration of Joint PDF over the real plane is 1, we have:-

$$f(x,y) = \begin{cases} \frac{2}{|\vec{u}\times\vec{v}|} & \text{if (x,y) is inside triangle} \\ 0 & otherwise \end{cases}$$

Writing in terms of a and b, we get :-

$$f(a,b) = \begin{cases} 2 & 0 \leq a \leq 1, 0 \leq b \leq 1, a+b \leq 1 \\ 0 & otherwise \end{cases}$$

This is the final PDF that we require.

Let us first try to find a method of generating uniformly distributed points inside the parallelogram formed by $\vec{u}$ and $\vec{v}$.

Since area of Parallelogram $= |\vec{u} \times \vec{v}|$, and integration of Joint PDF over the real plane is 1, we have:-

$$f(x,y) = \begin{cases} \frac{1}{|\vec{u}\times\vec{v}|} & \text{if (x,y) is inside parallelogram} \\ 0 & otherwise \end{cases}$$

Writing in terms of a and b, we get :-

$$f(a,b) = \begin{cases} 1 & 0 \leq a \leq 1, 0 \leq b \leq 1 \\ 0 & otherwise \end{cases}$$

Let us assume the Random Variable A and the Random Variable B to be independent random variables such that :-

$$A = U$$
$$B = U$$

where U is the Uniformly Distributed Random Variable from 0 to 1.

We shall now prove that the joint PDF resulting from this is indeed the PDF that we require ($f(a,b)$).

We know that :-

$$\text{(PDF of } A) \quad f_A(a) = \begin{cases} 1 & 0 \leq a \leq 1 \\ 0 & otherwise \end{cases}$$
$$\text{(PDF of } B) \quad f_B(b) = \begin{cases} 1 & 0 \leq b \leq 1 \\ 0 & otherwise \end{cases}$$

As we have assumed A and B to be independent,

$$\mathbb{P}(A \leq a, B \leq b) = \mathbb{P}(A \leq a)\mathbb{P}(B \leq b)$$
$$\implies f(a,b) = f_A(a) f_B(b)$$

Hence :-

$$f(a,b) = \begin{cases} 1 & 0 \leq a \leq 1, 0 \leq b \leq 1 \\ \\ 0 & otherwise \end{cases}$$

Which matches exactly with the parallelogram probability density, hence, this is a valid method of producing uniformly distributed random points inside a parallelogram.

To generate uniformly distributed points inside a triangle, we first define a Bijection :-

$$T : [0,1] \times [0,1] \longrightarrow [0,1] \times [0,1] \text{ defined by}$$
$$T(a,b) = (1-a, 1-b)$$

Clearly, T is a Bijection, and a property of it is that it maps (a,b) such that $a+b>1$ to points (a_1,b_1) such that $a_1+b_1<1$ and vice-versa.

Let us consider the probability distribution we get if we apply T to instances of points (a,b) that have $a+b>1$.

Then in the Original PDF, f(a,b) would be increased by f(1-a,1-b) for all $0 \leq a \leq 1, 0 \leq b \leq 1$ such that $a+b<1$ and f(a,b) for all $0 \leq a \leq 1, 0 \leq b \leq 1$ such that $a+b>1$ would become 0.

So final PDF would become:-

$$f(a,b) = \begin{cases} 2 & 0 \leq a \leq 1, 0 \leq b \leq 1, a+b \leq 1 \\ \\ 0 & otherwise \end{cases}$$

Which is the density function that we wanted. Hence, this is a valid method of generating uniformly distributed points inside a Triangle.

Final Algorithm :-

1. Generate instances of Random Variables A and B (which are same as instances of Uniformly Distributed Random Variable from 0 to 1)
2. If the sum of generated instances is greater than 1, apply function T to the generated instance.
3. Find out the Cartesian Coordinates of the point through $a\vec{u} + b\vec{v}$.
4. The final point will be an instance of a Random Variable that is uniformly distributed over the triangle formed by $\vec{u}$ and $\vec{v}$.

The correctness of this Algorithm is further supported by the empirical evidence below, where the points are uniformly distributed.

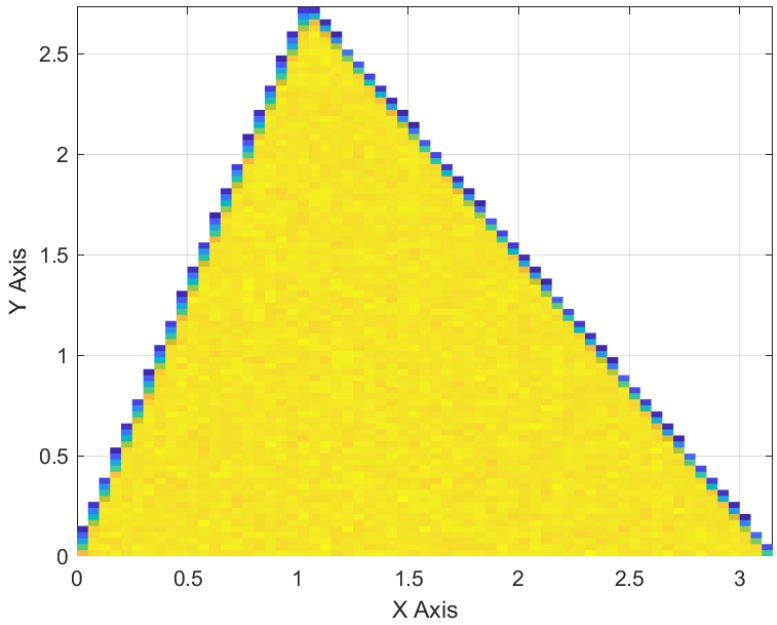

Figure 2: Histogram of 10^7 Independent Draws

2 Multivariate Gaussian

2.1 Generating Sample Points from 2D Gaussian

We know that for Random Vector $X := AW + \mu$,

$$E[X] = \mu$$
$$Cov(X) = AA^T$$

As defined in the class slides.

We are given Covariance Matrix (C) and Mean (μ), therefore, we need to find out A such that $AA^T = C$. For ease of calculation, let us assume A to be a Lower Triangular $D \times D$ Matrix, where D is the number of dimensions, 2 in this case. So, let :-

$$A = \begin{bmatrix} a & 0 \\ b & c \end{bmatrix}$$

So, we have :-

$$\begin{aligned} C &= AA^T \\ &= \begin{bmatrix} a & 0 \\ b & c \end{bmatrix} \times \begin{bmatrix} a & b \\ 0 & c \end{bmatrix} \\ &= \begin{bmatrix} a^2 & ab \\ ab & b^2 + c^2 \end{bmatrix} \end{aligned}$$

This implies :-

$$a = \sqrt{C_{1,1}}$$
$$b = \frac{C_{1,2}}{a}$$
$$c = \sqrt{C_{2,2} - b^2}$$

We can confirm that $AA^T = C$ since C is symmetric.

Now we can generate sample points as follows :-

1. Use the randn() function to generate $W \sim \mathcal{N}(0_{2,1}, I_{2,2})$
2. Using Matrices A and μ, generate instance of $X = AW + \mu$
3. This method will generate instances of distribution having mean = μ and Covariance = $AA^T = C$

2.2 BoxPlot Error Measure of Mean

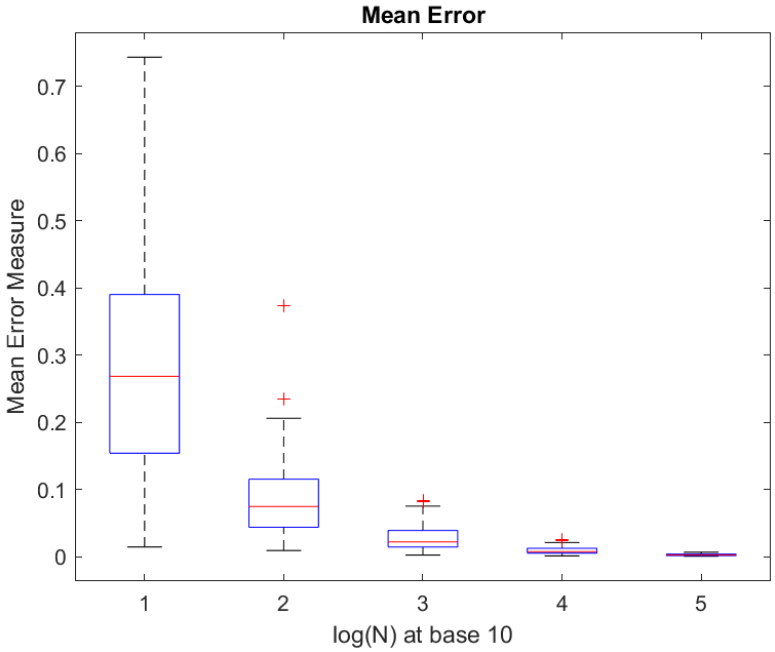

Figure 3: Error Measure of Mean

2.3 BoxPlot Error Measure of Covariance Matrix

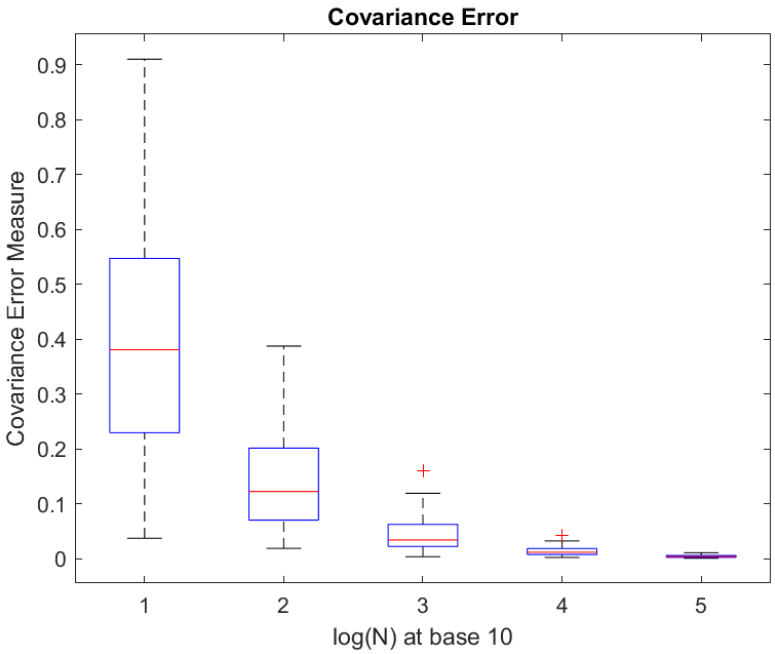

Figure 4: Error Measure of Covariance Matrix

2.4 Scatter Plots and Principal Modes of Variation

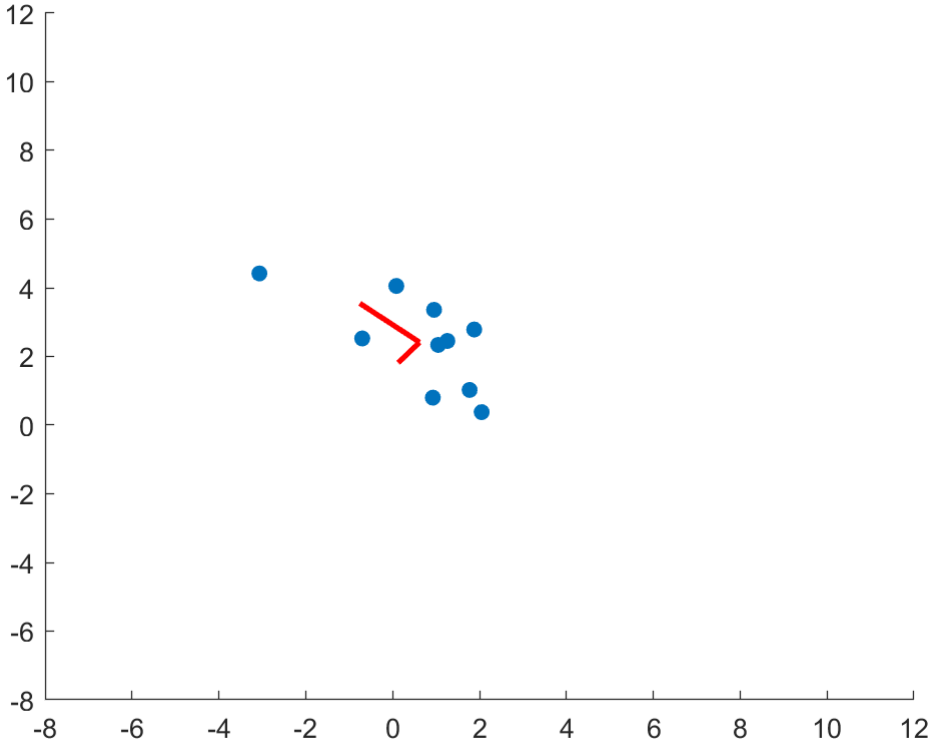

Figure 5: Scatter Plot and Principal Modes of Variation for N = 10 Draws

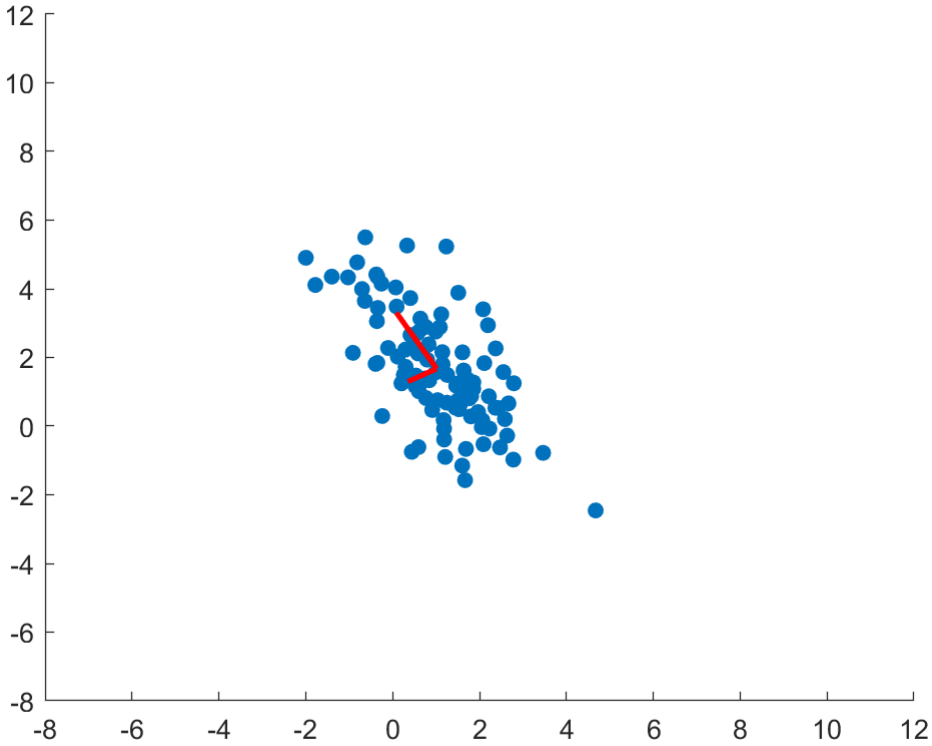

Figure 6: Scatter Plot and Principal Modes of Variation for N = 10^2 Draws

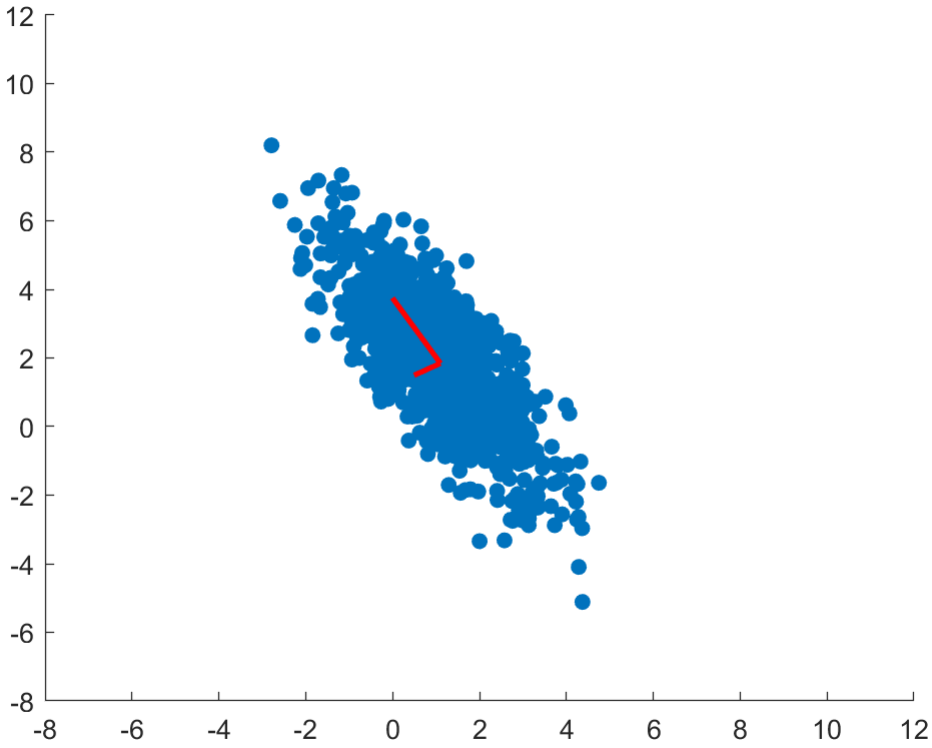

Figure 7: Scatter Plot and Principal Modes of Variation for N = 10^3 Draws

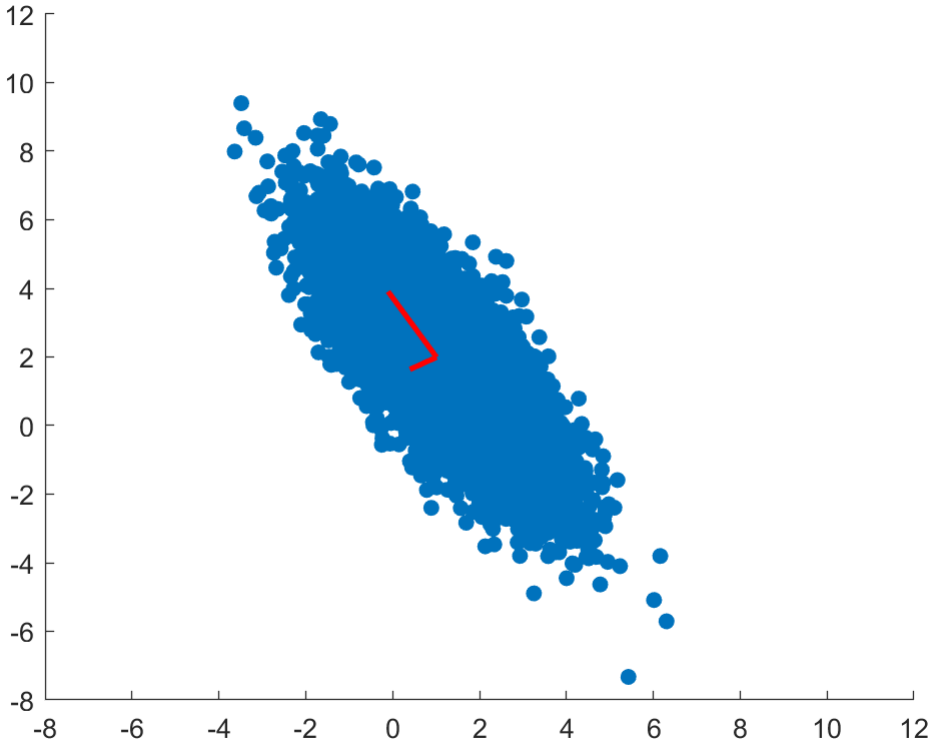

Figure 8: Scatter Plot and Principal Modes of Variation for N = 10^4 Draws

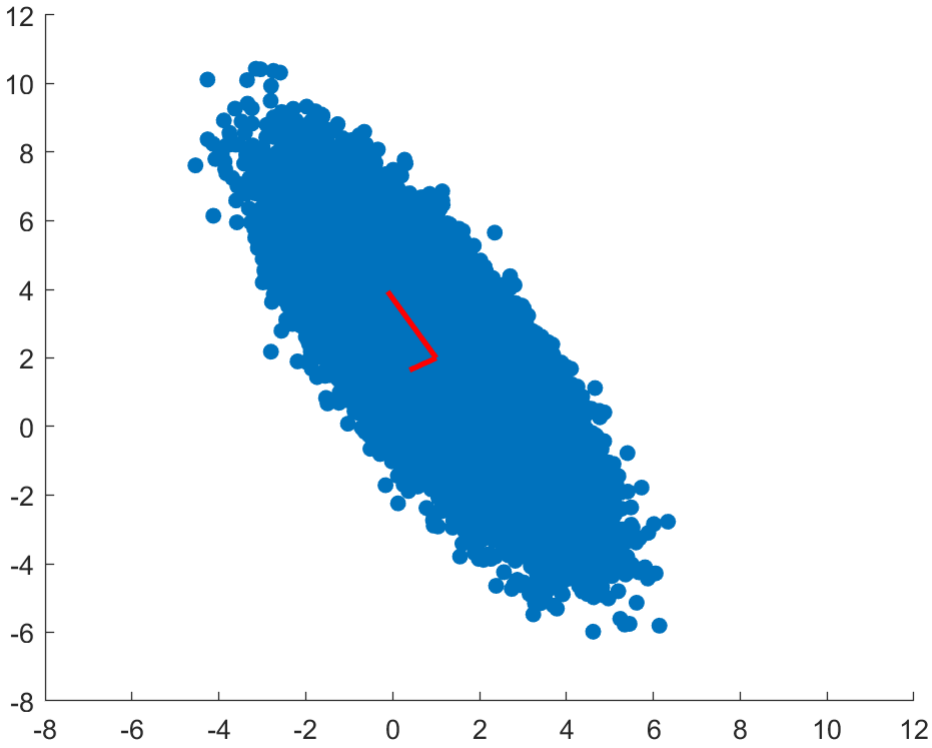

Figure 9: Scatter Plot and Principal Modes of Variation for N = 10^5 Draws

3 PCA and Hyperplane Fitting

3.1 Principal Component Analysis

To establish an approximate linear relation, we find the line that best fits the data. Here, the best-fitting line is defined as one that minimizes the average squared distance from the points to the line. We assume our best fit line to pass through sample mean with slope v where $||v|| = 1$ and let $z = [x\ y]^T$.

$$\begin{aligned} argmin_v \sum_{i=1}^{N}(||z||^2 - < z_i, v >^2) &= argmax_v \sum_{i=1}^{N} < z_i, v >^2 \\ &= argmax_v \sum_{i=1}^{N}(z_i^T v)^2 \\ &= argmax_v \sum_{i=1}^{N} v^T z_i z_i^T v \\ &= argmax_v\ N \cdot v^T C v \end{aligned}$$

Here C is a symmetric positive definite matrix. Thus C can be diagnolised as $C = QDQ^T$ where Q is an orthogonal matrix composed of eigenvectors of C and D is a diagnol matrix with diagnol elements as corresponding eigenvalues.

$$\begin{aligned} argmin_v \sum_{i=1}^{N}(||z||^2 - < z_i, v >^2) &= argmax_v\ N \cdot v^T QDQ^T v \\ &= Q\hat{e}_i \end{aligned}$$

where i is such that D_{ii} is the maximum eigenvalue.
Thus, the solution is $v =$ eigenvector of C corresponding to maximum eigenvalue.

3.2 Scatter Plot and Best Fit Line

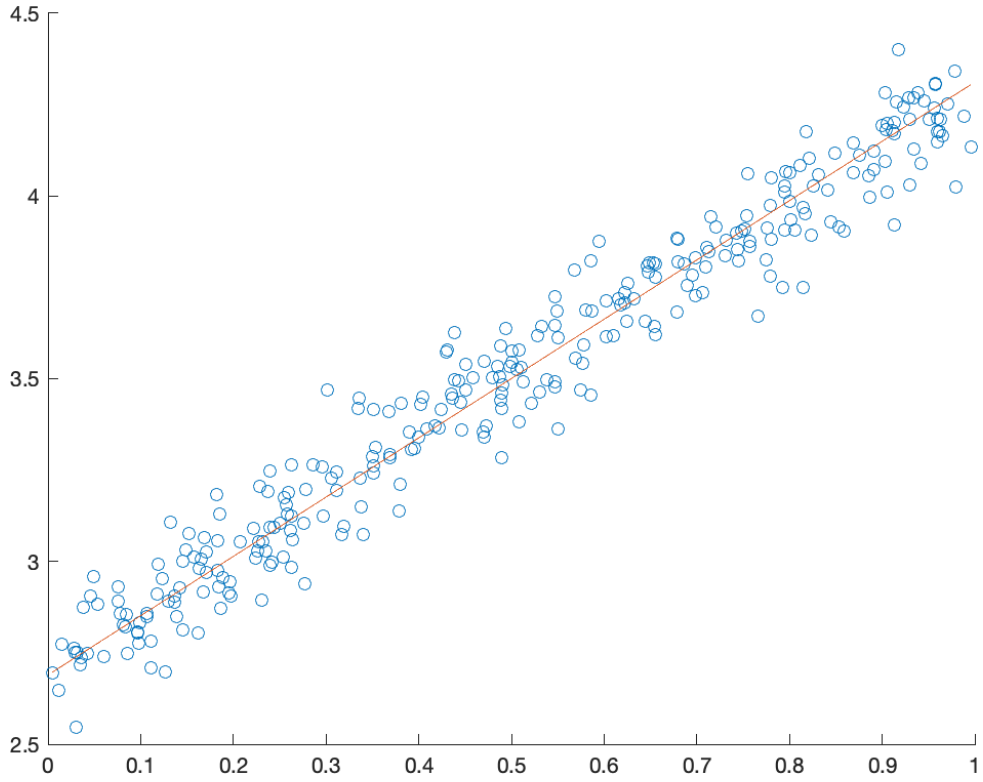

Figure 10: Scatter Plot Set 1 and best fit line

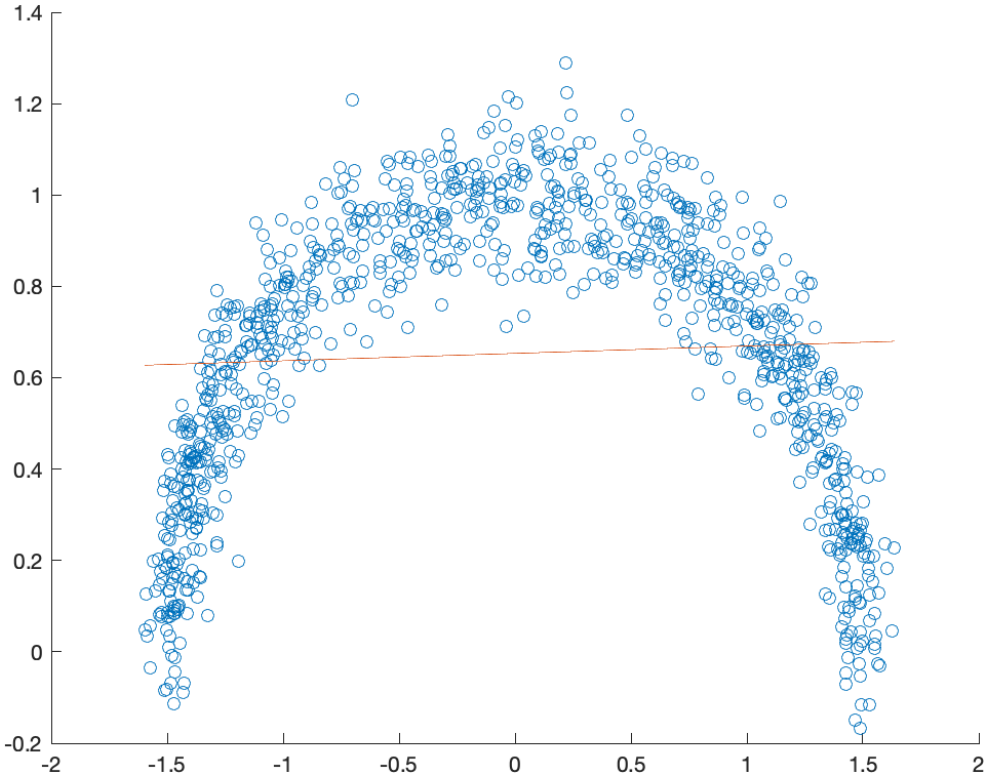

Figure 11: Scatter Plot Set 2 and best fit line

3.3 Linear Approximation Error

Clearly, the linear approximation between X and Y is not a good estimate for Set 2 since the dataset is not distributed linearly while on the other hand the estimate is correct for Set 1. We calculate the error from the best fit line as

$$Error = \frac{\sum_{i=1}^{N}(||z||^2 - < z_i, v >^2)}{N}$$

Set 1 : Error = 0.0025
Set 2 : Error = 0.0976

4 PCA and MNIST Digits

4.1 Significant Modes of Variation

The significant modes of variation (i.e. number of eigenvalues greater than 1) for each digit is around 10 (far less than 784). Here's the data

Digits	Number of eigenvalues greater than 1
0	10
1	4
2	11
3	10
4	10
5	10
6	9
7	8
8	9
9	8

Table 1: Significant Modes of Variation

This is because the basic structure of image remains the same. All the pictures of a digit have similar pixel data.

4.2 Principal Mode of Variation

The principal mode of variation describes the most significant variation around the mean. The difference in features in mean, $\mu + \sqrt{\lambda_1}v_1$ and $\mu - \sqrt{\lambda_1}v_1$ shows how digits are drawn in a different fashion.
For digit 1, the mean looks blurred as it combines both the straight and slant version of 1. The image of $\mu + \sqrt{\lambda_1}v_1$ shows that majority draw 1 in a slant fashion.

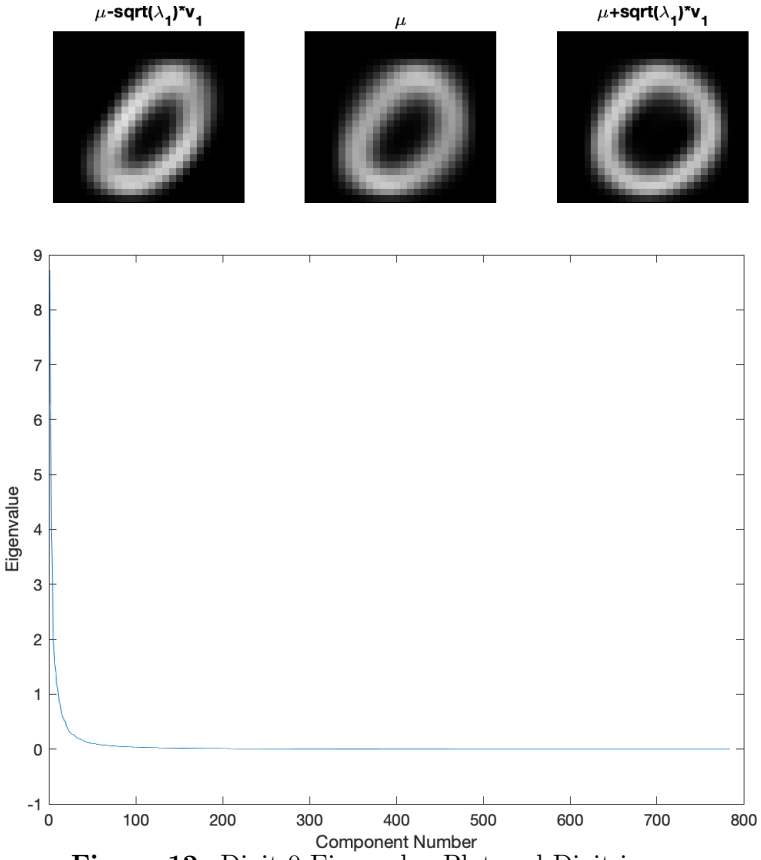

Figure 12: Digit-0 Eigenvalue Plot and Digit images

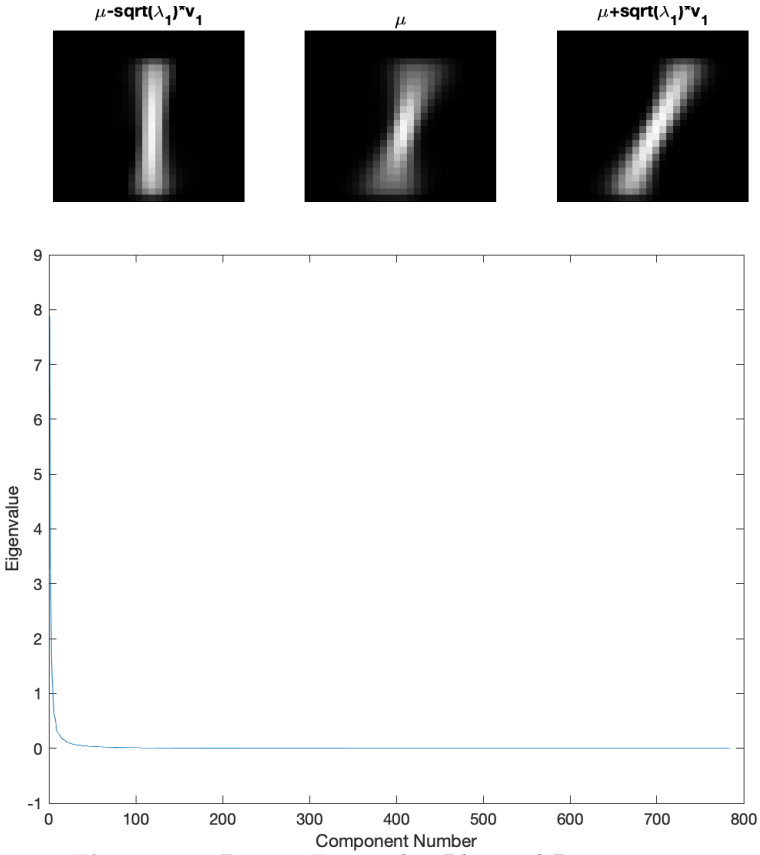

Figure 13: Digit-1 Eigenvalue Plot and Digit images

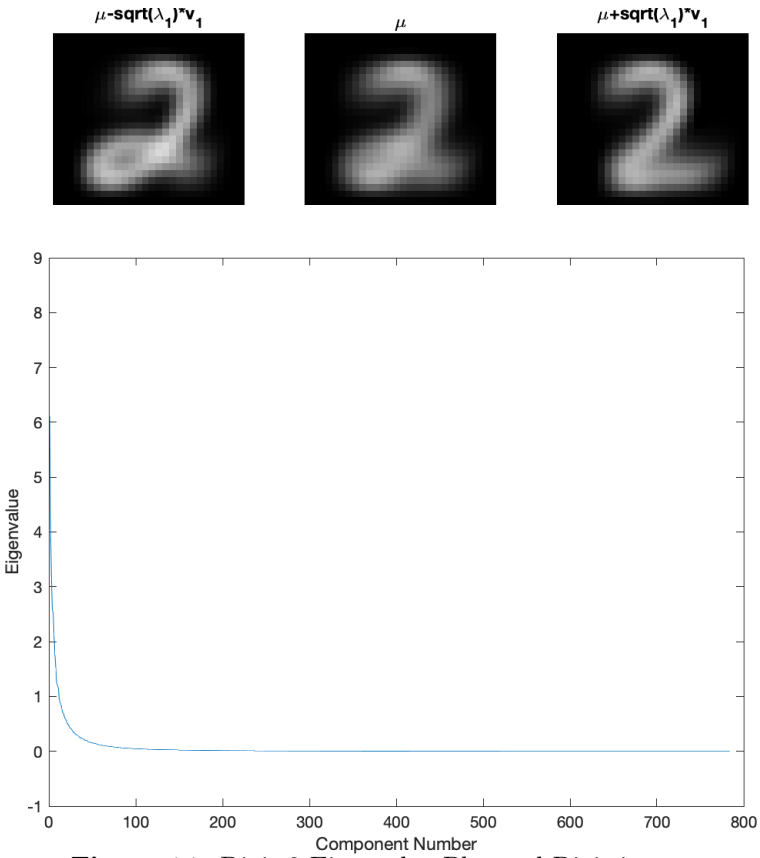

Figure 14: Digit-2 Eigenvalue Plot and Digit images

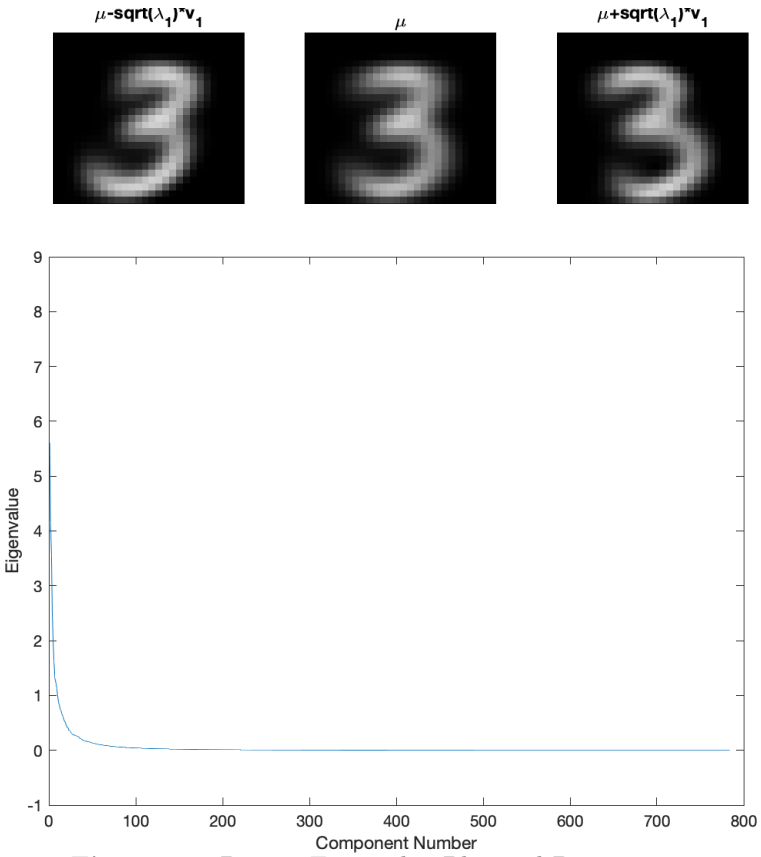

Figure 15: Digit-3 Eigenvalue Plot and Digit images

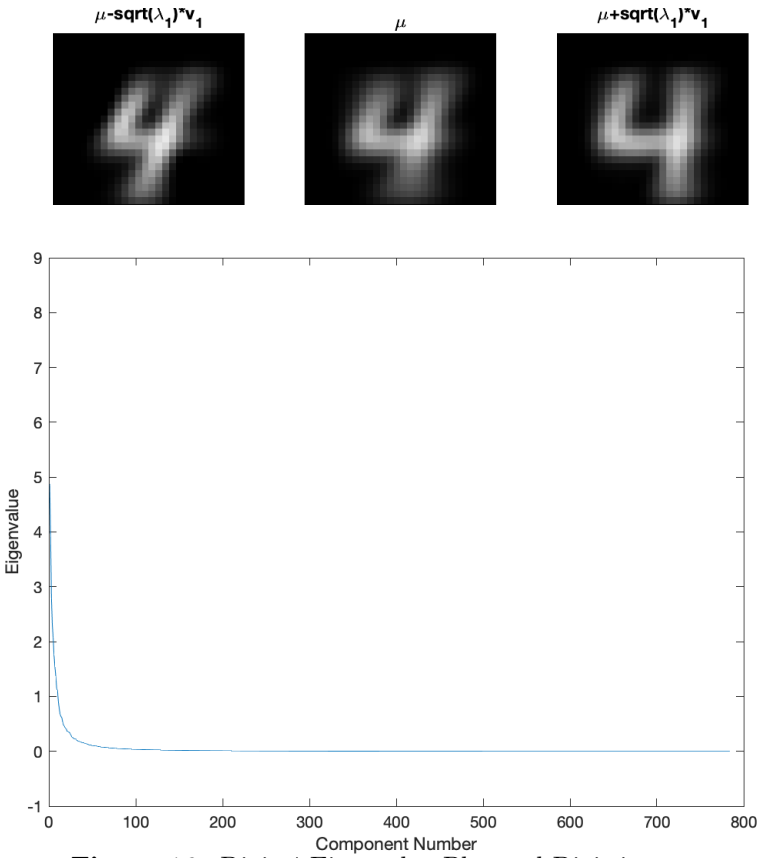

Figure 16: Digit-4 Eigenvalue Plot and Digit images

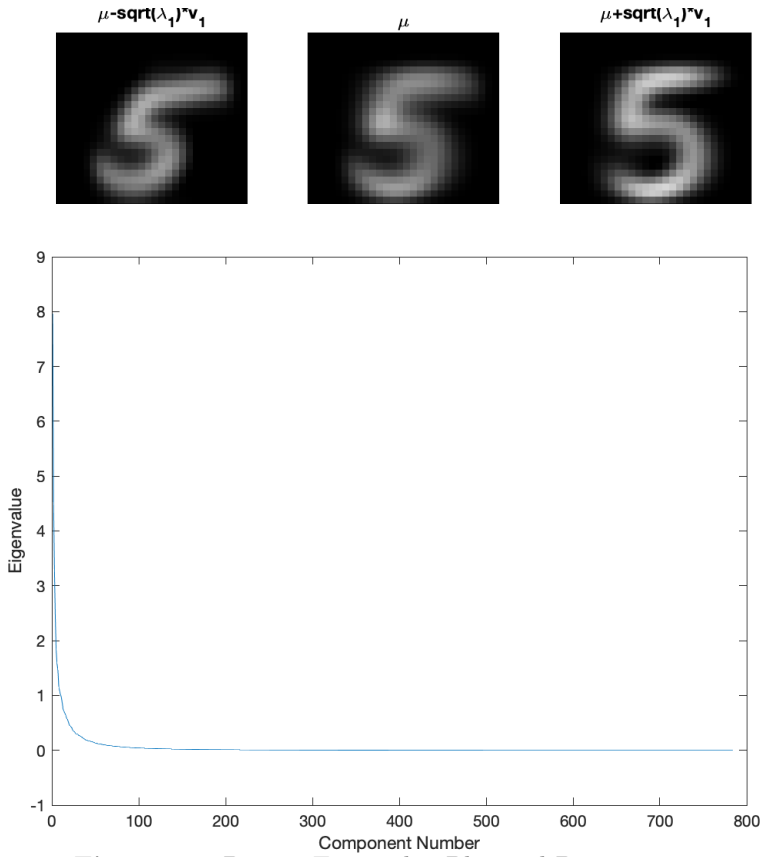

Figure 17: Digit-5 Eigenvalue Plot and Digit images

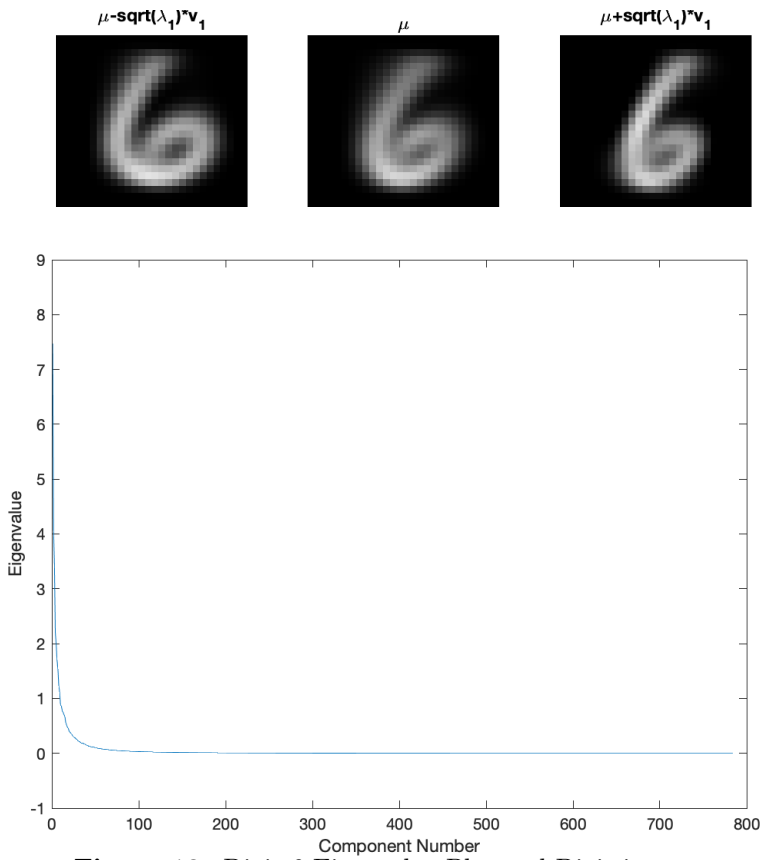

Figure 18: Digit-6 Eigenvalue Plot and Digit images

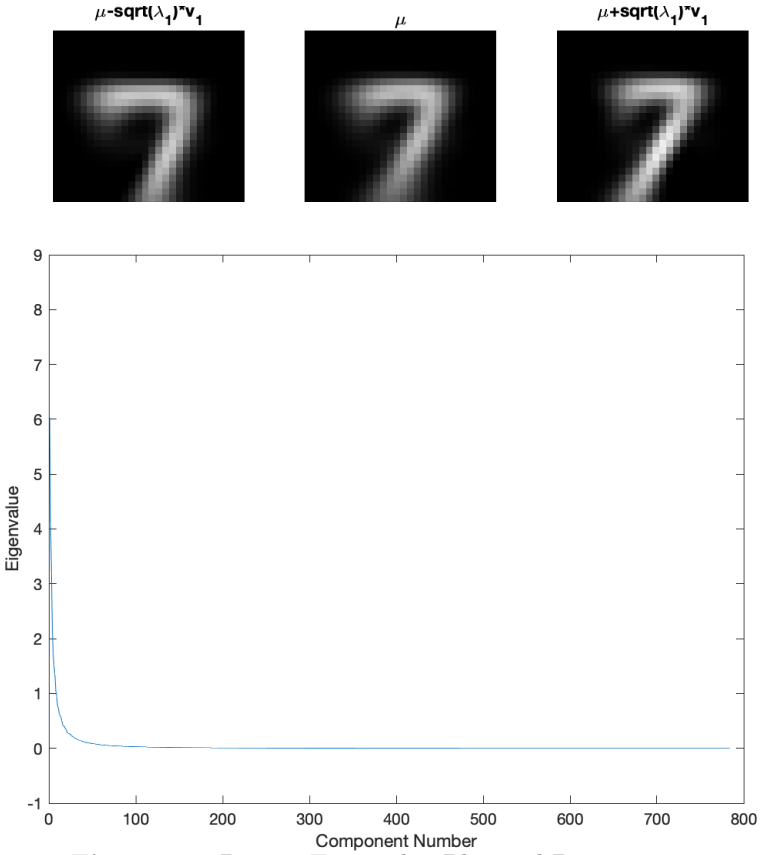

Figure 19: Digit-7 Eigenvalue Plot and Digit images

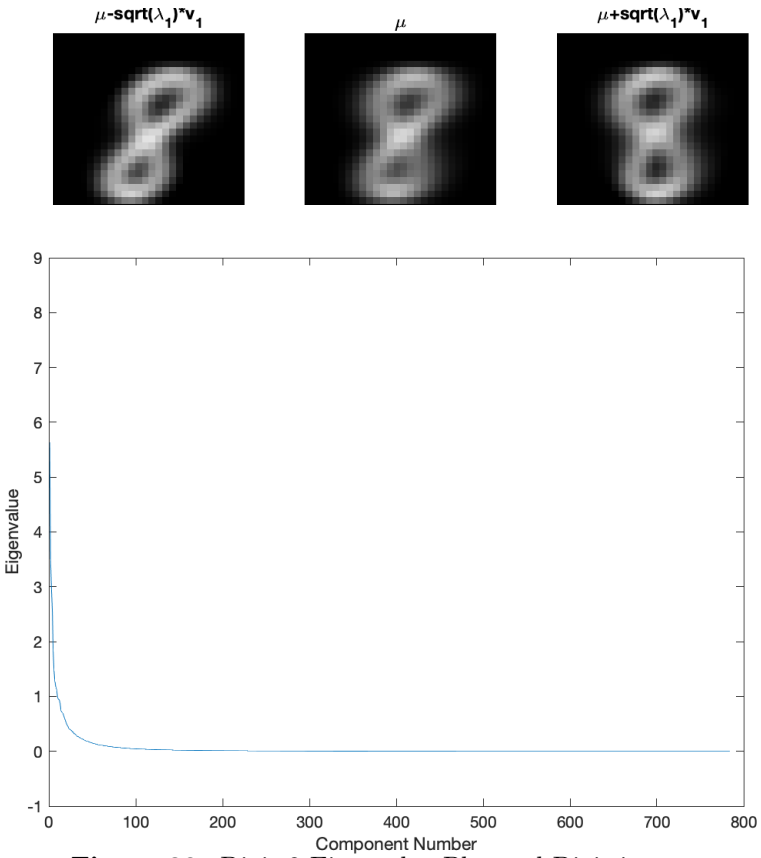

Figure 20: Digit-8 Eigenvalue Plot and Digit images

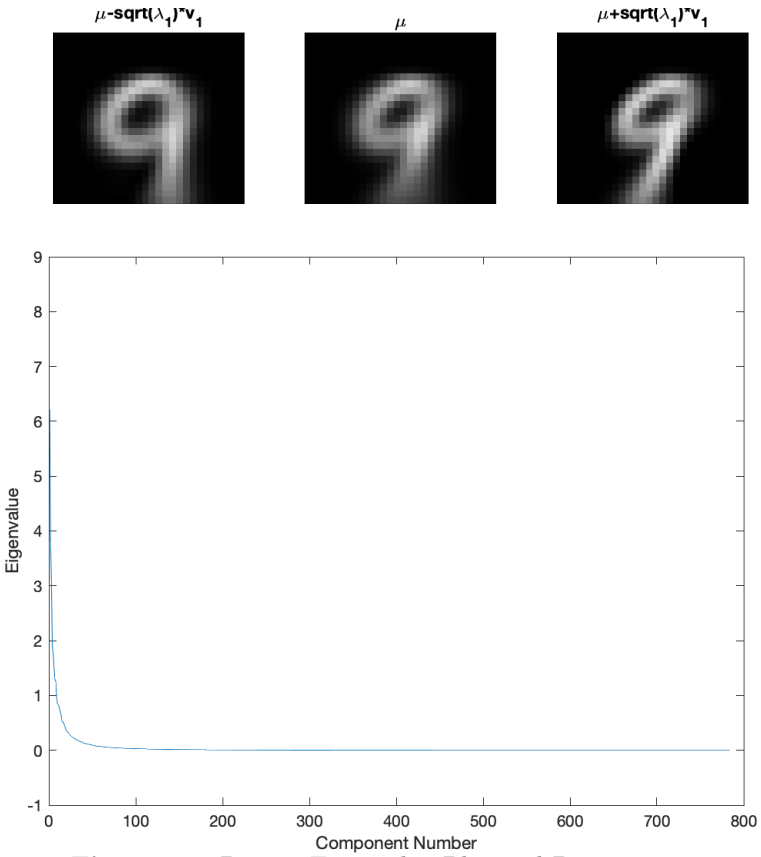

Figure 21: Digit-9 Eigenvalue Plot and Digit images

5 PCA and Dimensionality Reduction

5.1 Algorithm

1. First, we find the mean of the given data sample.
2. We then find the Sample Covariance Matrix of the data with mean shifted to zero.
3. We then find the 84 largest eigenvalues and corresponding eigenvectors of the Sample Covariance Matrix.
4. Then we "project" the data onto the 84 dimensional hyperplane given by the eigenvectors of these eigenvalues by multiplying the product of eigenvector matrix and its transpose, with the data (after subtracting mean), and then adding back the mean to obtain the final image.

5.2 84 Dimensional Hyperplane

We find 84 principal eigenvectors of the covariance matrix which will represent the 84 dimensional hyperplane. We project the original data onto the hyperplane to get the reconstructed image. The results show that the image formed is a good representation of the original.

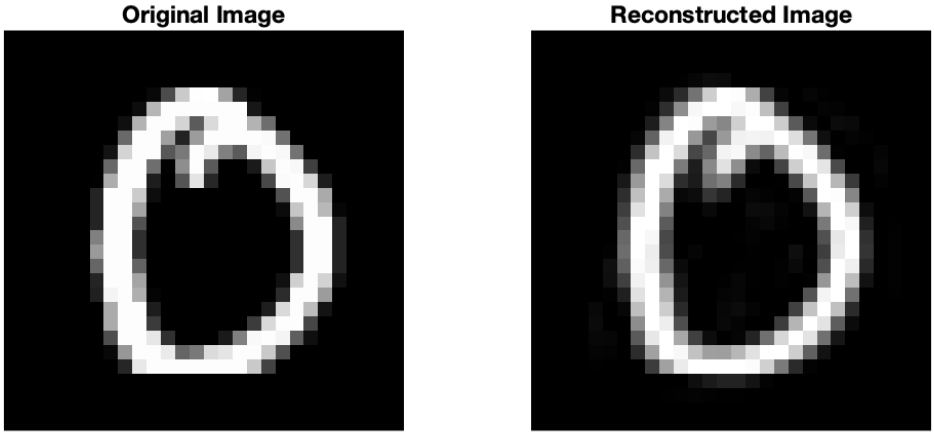

Figure 22: Image of Digit 0

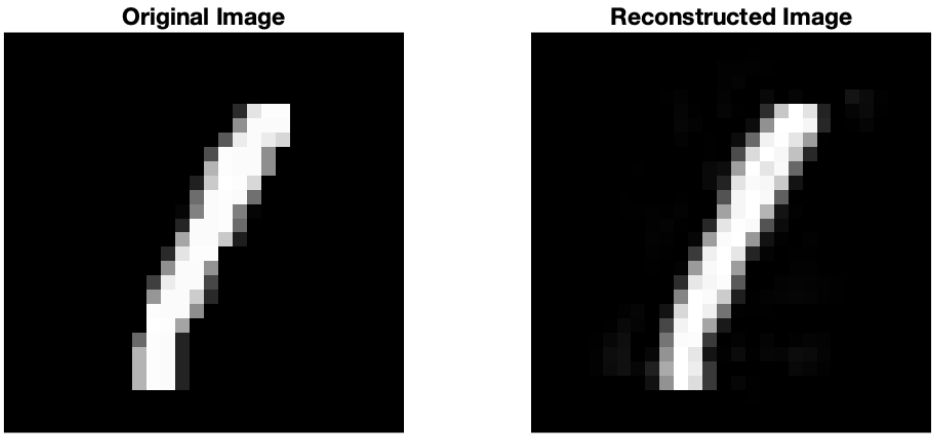

Figure 23: Image of Digit 1

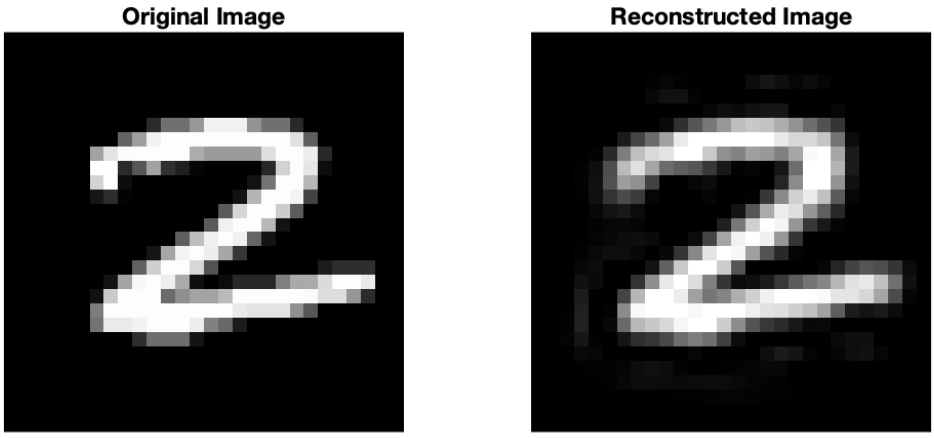

Figure 24: Image of Digit 2

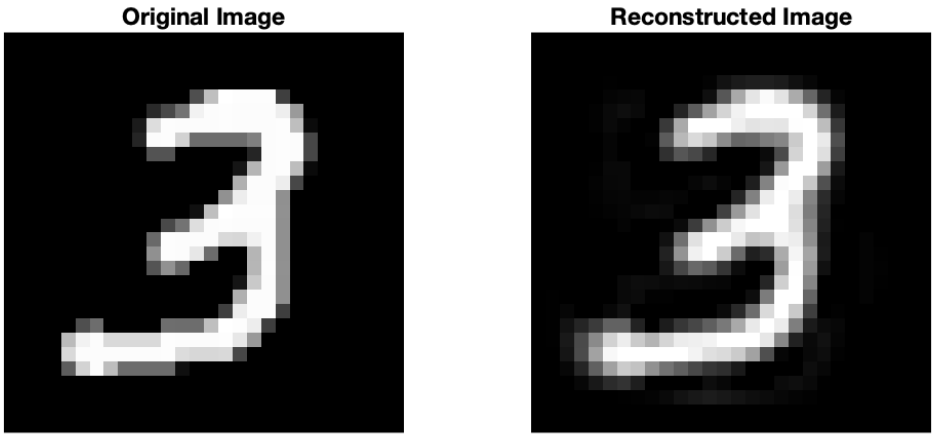

Figure 25: Image of Digit 3

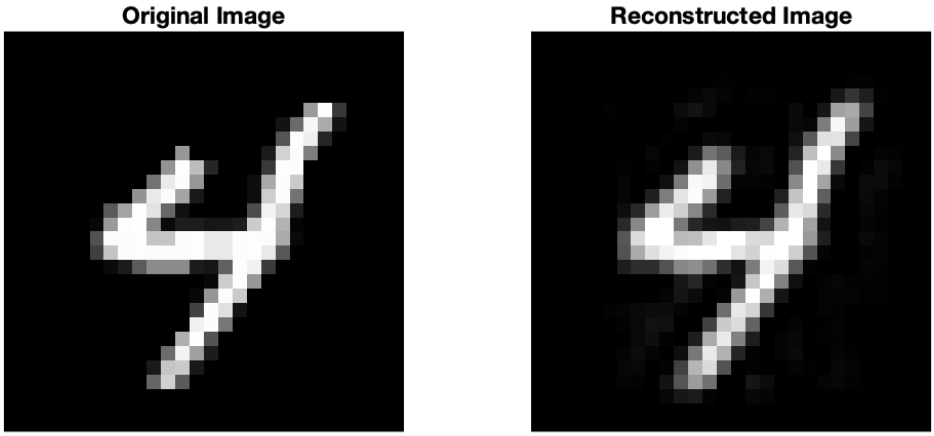

Figure 26: Image of Digit 4

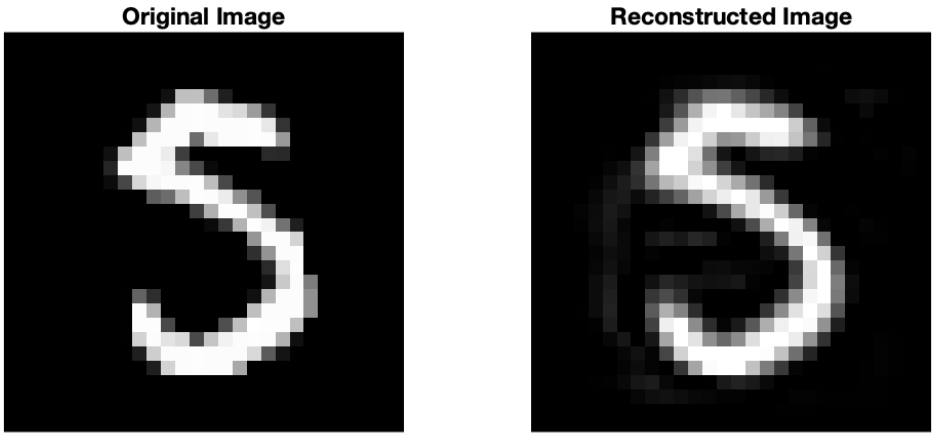

Figure 27: Image of Digit 5

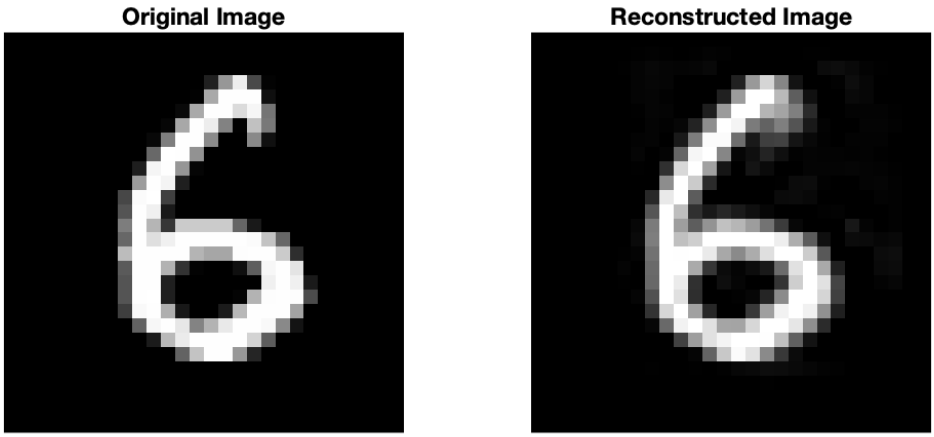

Figure 28: Image of Digit 6

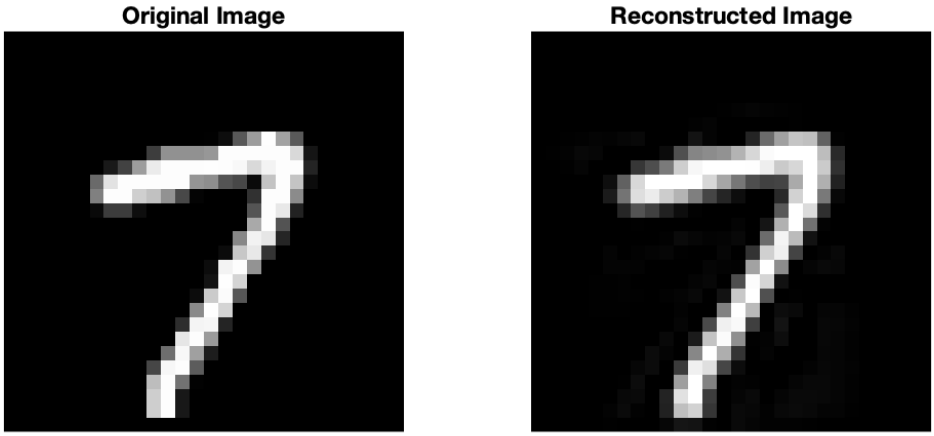

Figure 29: Image of Digit 7

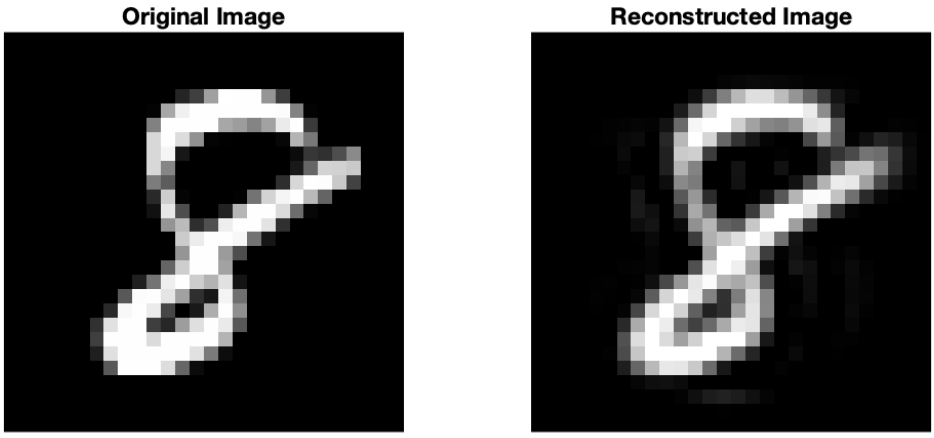

Figure 30: Image of Digit 8

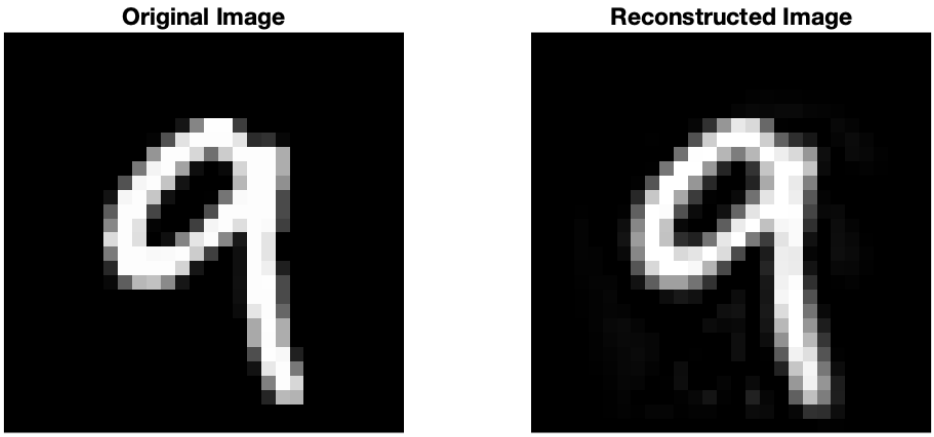

Figure 31: Image of Digit 9

6 PCA and Fruit Image Dataset

6.1 Mean and Principle Eigenvectors

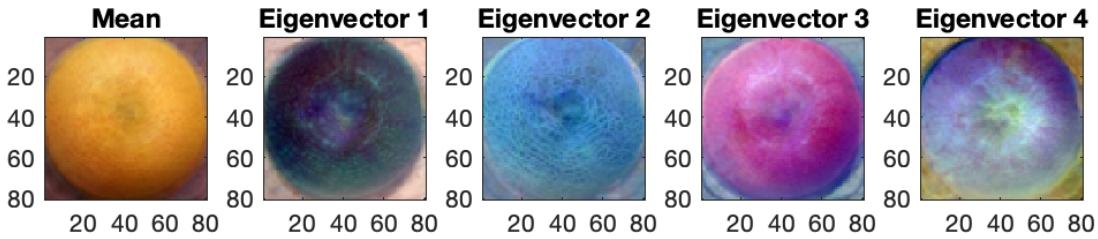

Figure 32: Mean and 4 Principle Eigenvectors

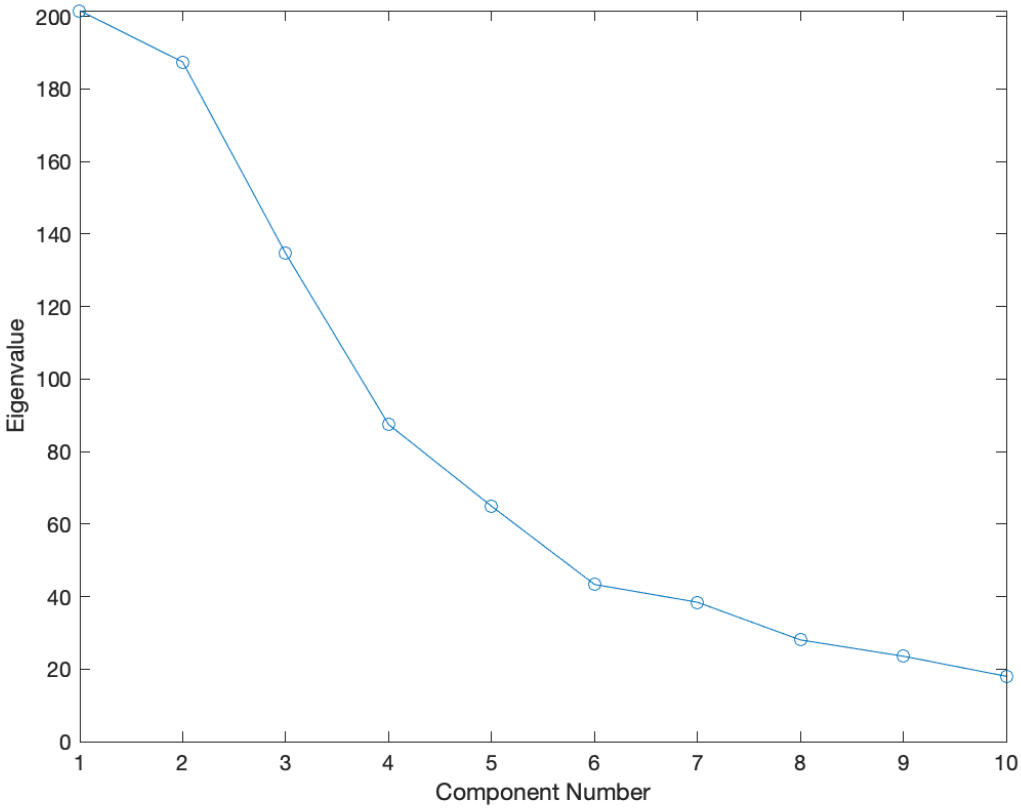

Figure 33: 10 Principle Eigenvalues

6.2 Closest Representation

Let the original image be given as y and the reconstructed image be represented as $y' = \mu + \sum_{i=1}^{4} \lambda_i v_i$ where μ is the mean and v_i's are the 4 principal eigenvectors.

We have to minimize the Frobenius norm of the difference between y and y'. That is we have to minimize

$$\begin{aligned} ||y - y'||^2 &= ||y - \mu - \sum_{i=1}^{4} \lambda_i v_i||^2 \\ &= ||\tilde{y} - \sum_{i=1}^{4} \lambda_i v_i||^2 \\ &= ||\tilde{y}||^2 + ||\sum_{i=1}^{4} \lambda_i v_i||^2 - 2 \cdot \tilde{y} \cdot \sum_{i=1}^{4} \lambda_i v_i \\ &= ||\tilde{y}||^2 + \sum_{i=1}^{4} (\lambda_i^2 ||v_i||^2 - 2\lambda_i \cdot \tilde{y} \cdot v_i) \end{aligned}$$

This is minimized when $\forall\, i, \;\; i \in \{1, 2, 3, 4\}$

$$\begin{aligned} \lambda_i &= \frac{\tilde{y} \cdot v_i}{||v_i||^2} \\ \lambda_i v_i &= (\frac{\tilde{y} \cdot v_i}{||v_i||^2}) v_i \end{aligned}$$

That is, to get the closest representation, we project the original image along the eigenvectors and add them up along with the mean.

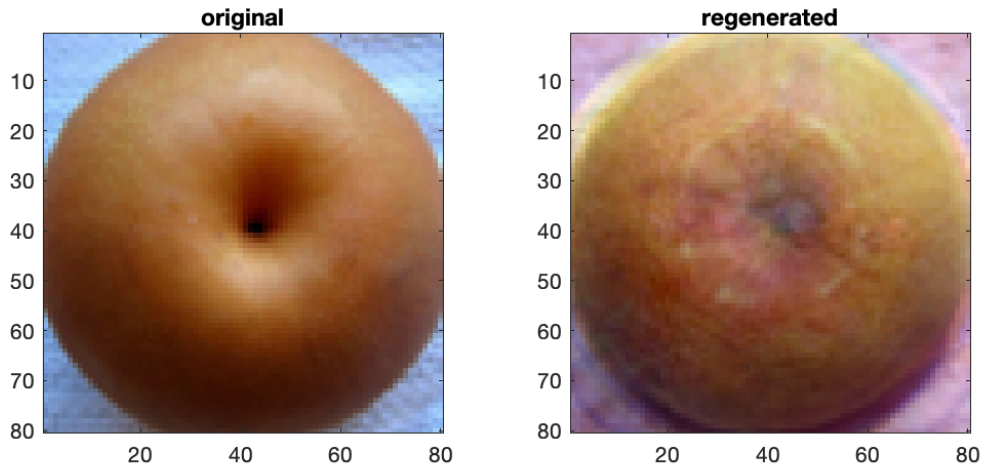

Figure 34: Image of Fruit 1

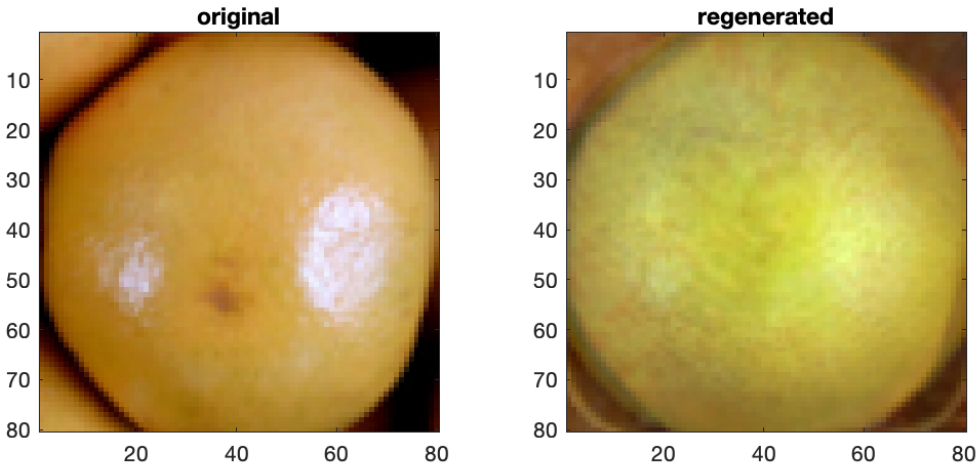

Figure 35: Image of Fruit 2

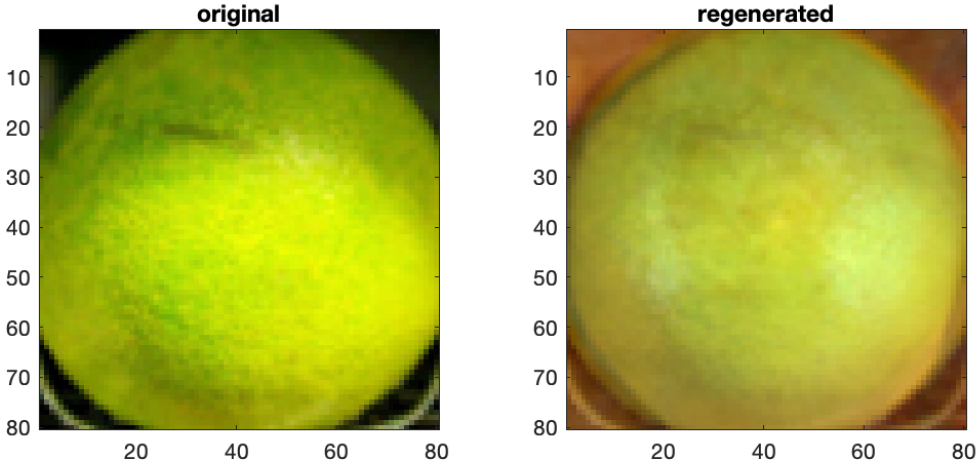

Figure 36: Image of Fruit 3

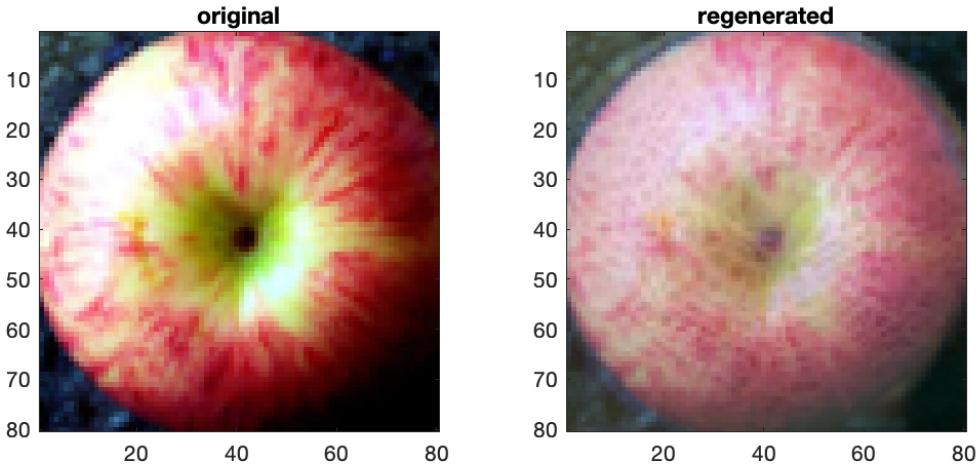

Figure 37: Image of Fruit 4

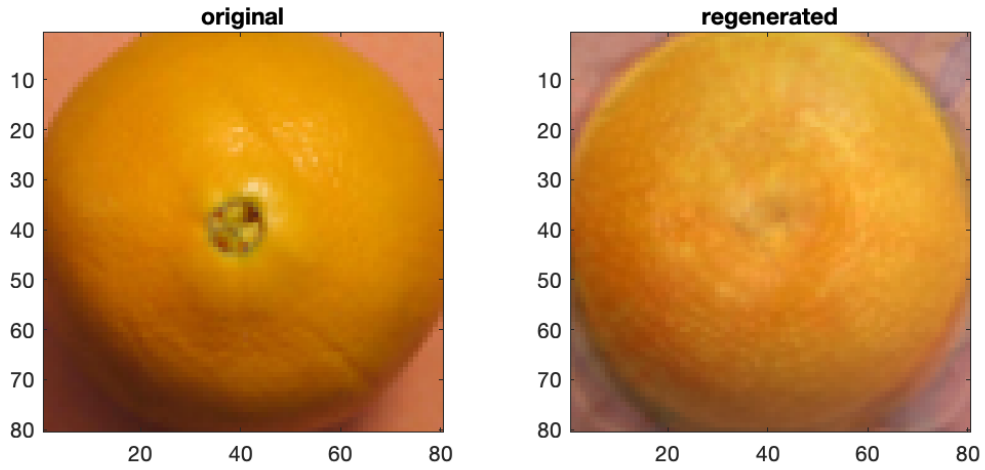

Figure 38: Image of Fruit 5

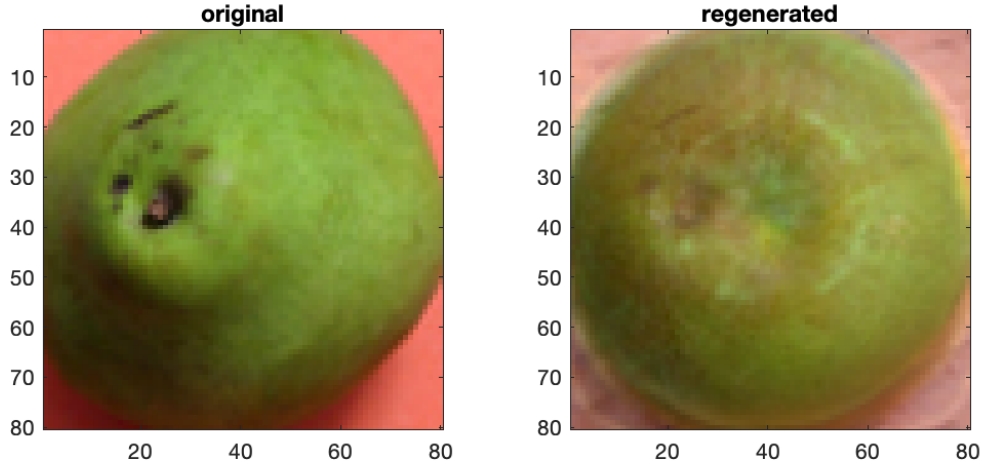

Figure 39: Image of Fruit 6

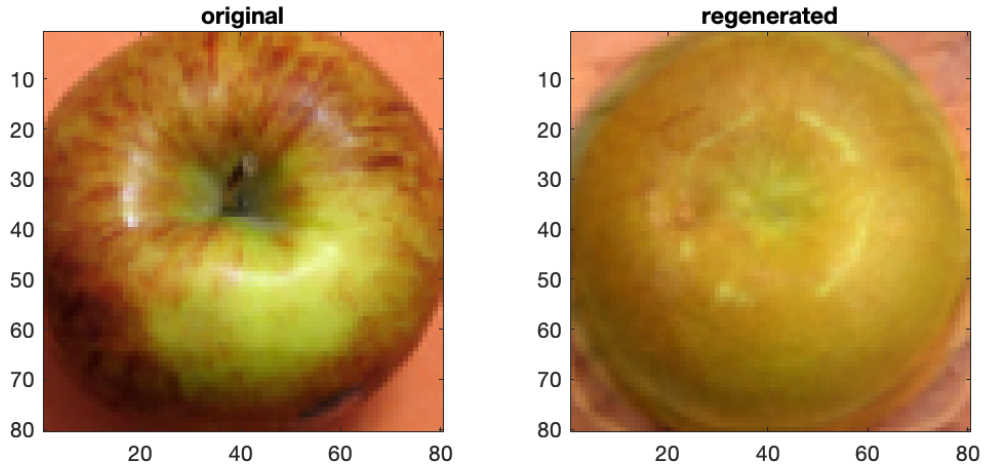

Figure 40: Image of Fruit 7

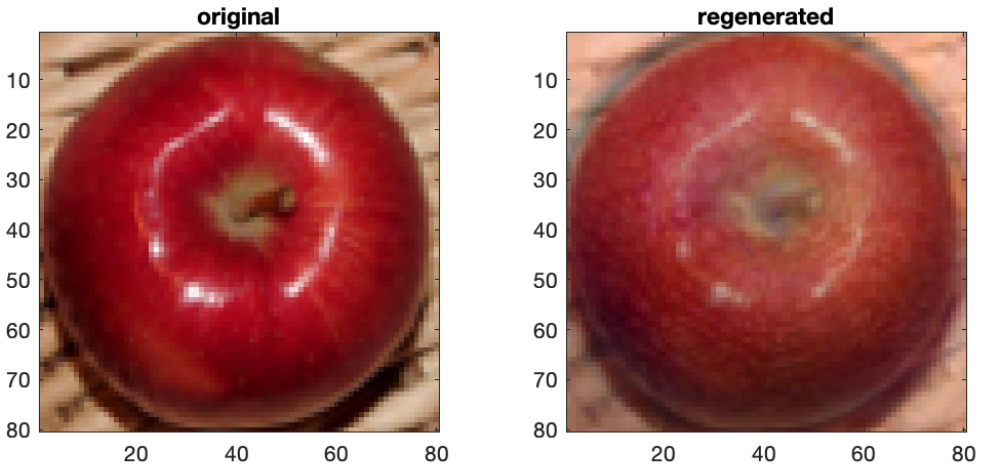

Figure 41: Image of Fruit 8

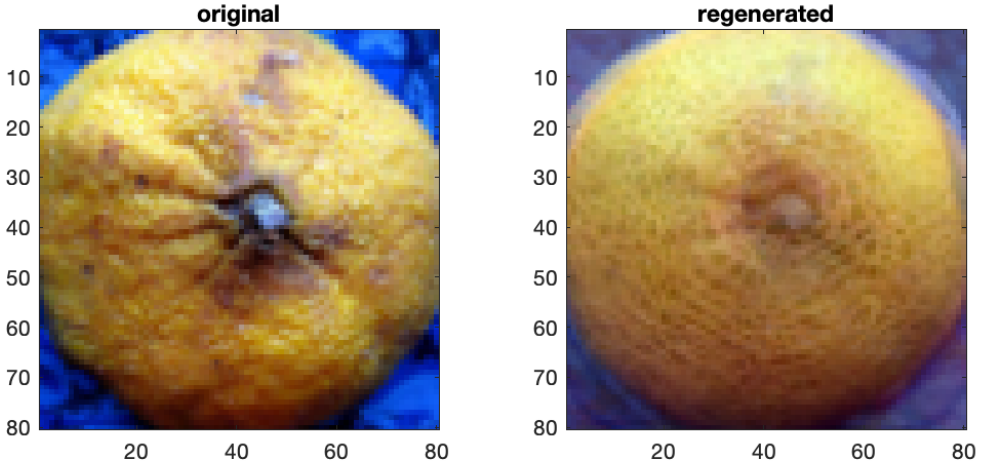

Figure 42: Image of Fruit 9

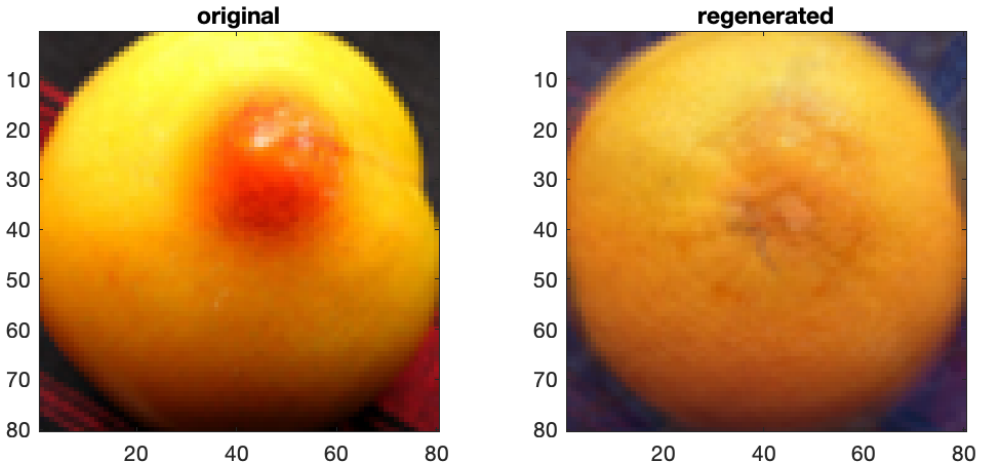

Figure 43: Image of Fruit 10

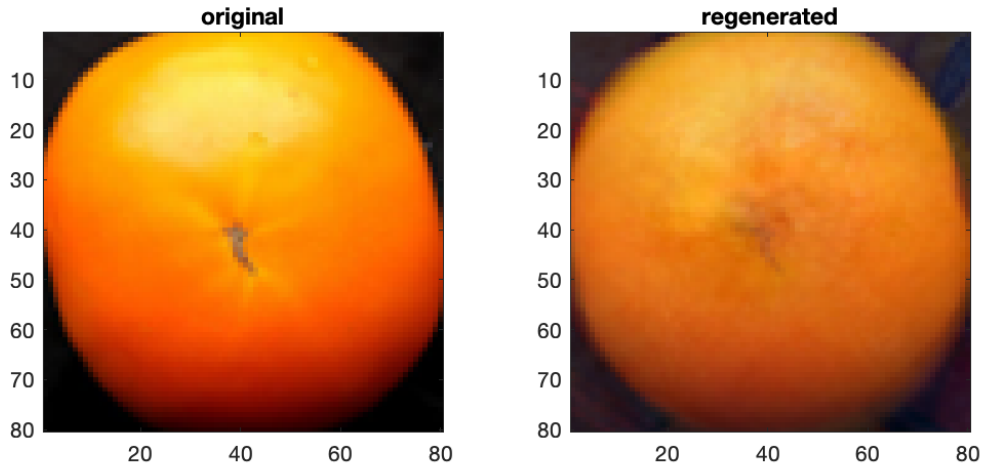

Figure 44: Image of Fruit 11

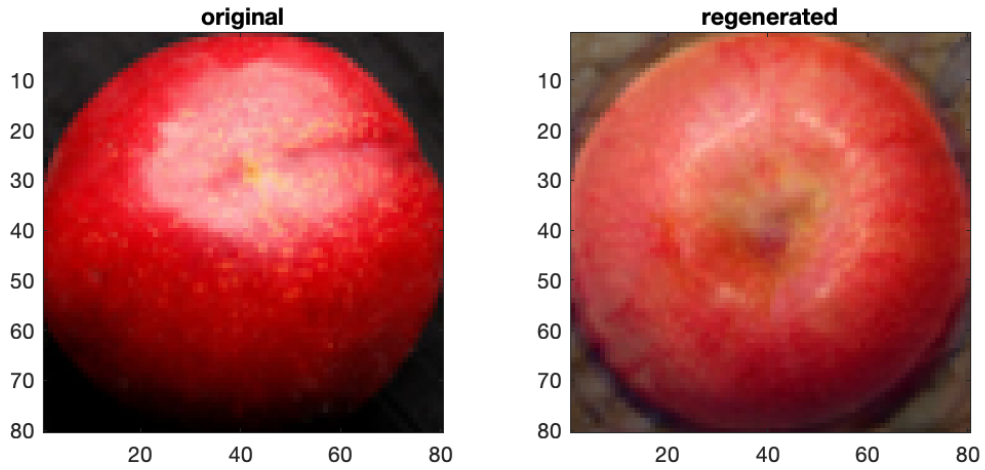

Figure 45: Image of Fruit 12

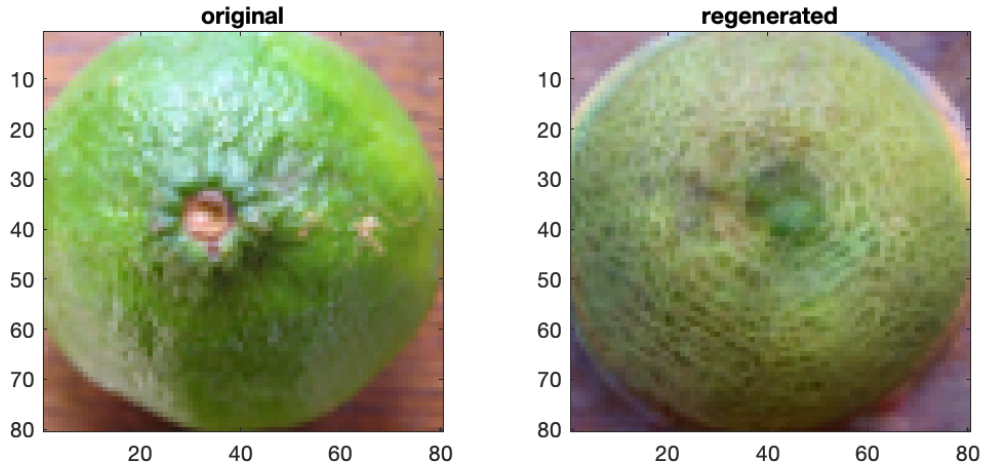

Figure 46: Image of Fruit 13

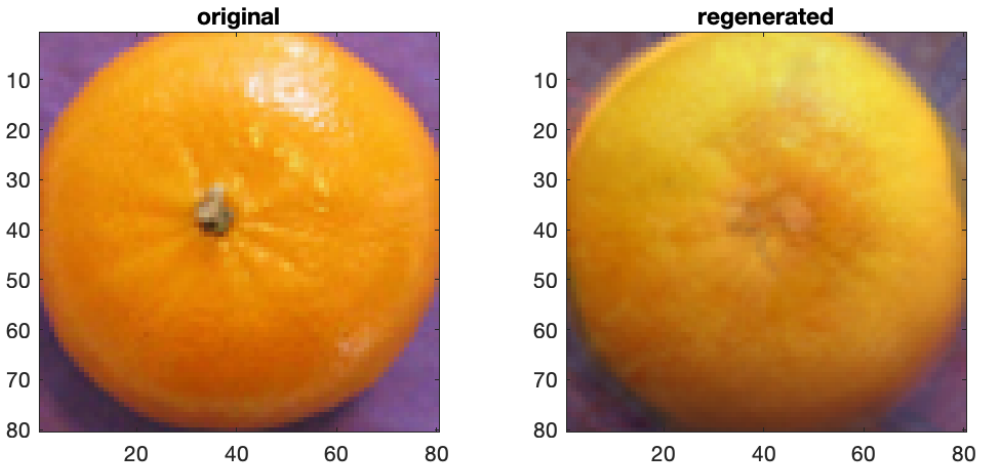

Figure 47: Image of Fruit 14

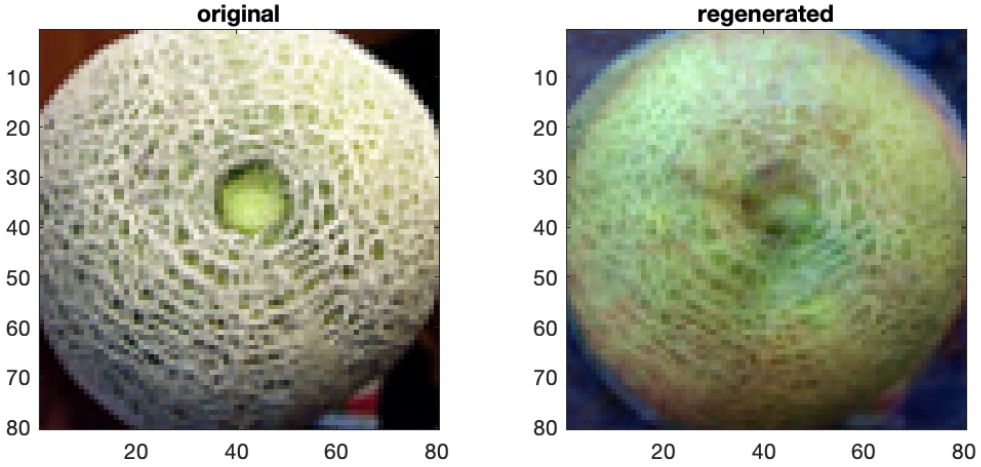

Figure 48: Image of Fruit 15

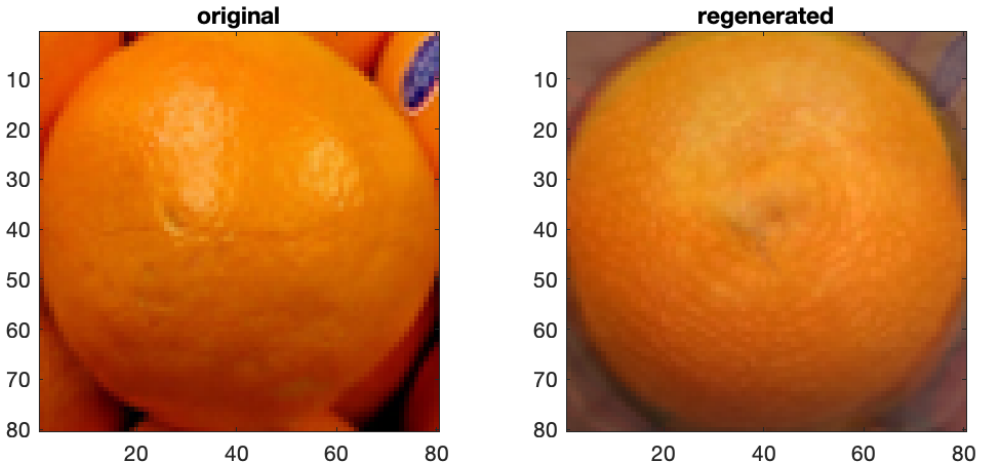

Figure 49: Image of Fruit 16

6.3 Generate New Fruits

1. Generate D instances of Gaussian Random Variable using randn() function, where D represents the number of eigenvectors being used to construct images (in this case, 4).

2. New fruit images are generated by adding a linear combination of eigenvectors multiplied by square root of corresponding eigenvalues to mean with coefficients generated in previous step.

$$y' = \mu + \sum_{i=1}^{4} randn \cdot (\sqrt{\lambda_i} v_i)$$

3. The image represented by y' will be an example of a new/generated fruit.

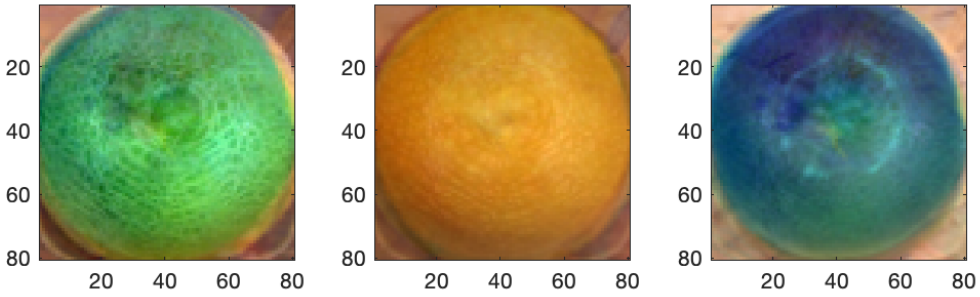

Figure 50: Generated Fruits