4 Question 4

4.1 Computing mean, covariance and principal mode of variation

The mean is calculated by computing sum of each dimension through all samples using vector multiplication and then divide by total number of samples of each number.
The covariance is calculated by computing $1/n \sum_{j=1}^{n}(x_j - \bar{x})(x_j - \bar{x})^T$ where n is the sample size for each number.
The principal mode of variation is found by finding the maximum eigenvalue and corresponding eigenvector from the covariance matrix computed.
The mean, covariance and the principal mode of variation has been documented in q4report.txt under q4 under results directory

4.2 Plotting the eigenvalues

The plots have been collected and recorded in the q4 section under results directory. I have calculated the threshold to be 10% of the maximum eigenvalue.

4.2.1 Number of principal modes of variation

digit	principal modes
0	11
1	5
2	18
3	17
4	16
5	12
6	12
7	12
8	19
9	14

4.2.2 Inference

We can learn from the plot that the significant modes of variation is quite less. We can learn that only a few dimensions in each number actually contribute towards some information about how digits drawn vary with people and some do not contribute at all to information of how numbers drawn are different. Hence we can reduce the dimension by a great amount for easy visualization as rest of the eigen values do not contribute any good information since the variation is low. The significant eigen values are less because the data doesn't have any valuable information to present when we over dimension it. The information presented by those eigenvalues becomes redundant as it is common in almost all cases of handwritten digit. The number of principal modes also signify the complexity of the number since for digit one which is just a single straight line, it is less and for a comples digti like 8 which has lot of curves, the modes needed to represent this is high.

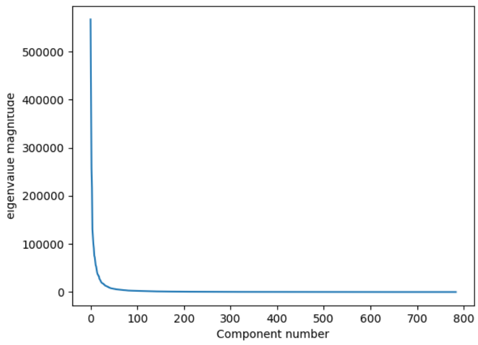

Figure 12: distribution of eigenvalues for 0 digit

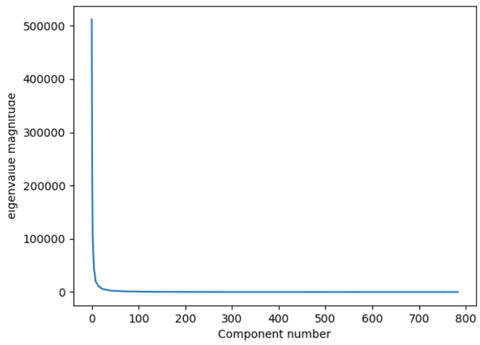

Figure 13: distribution of eigenvalues for 1 digit

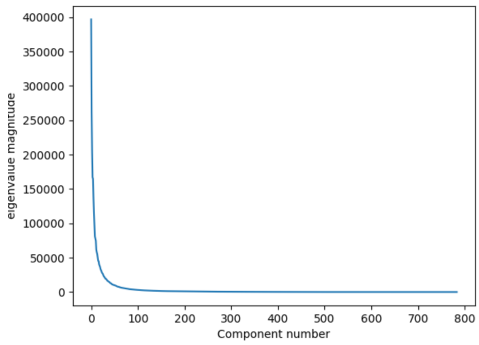

Figure 14: distribution of eigenvalues for 2 digit

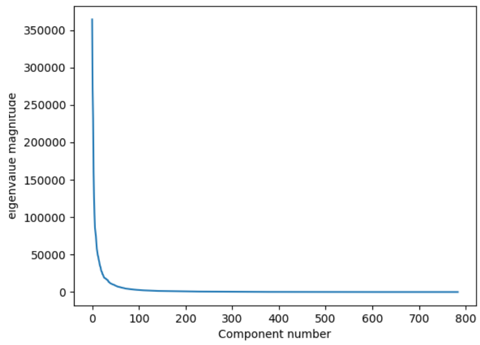

Figure 15: distribution of eigenvalues for 3 digit

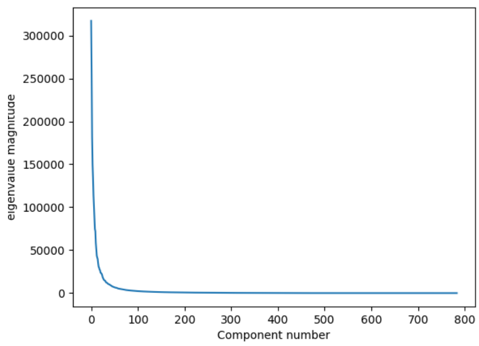

Figure 16: distribution of eigenvalues for 4 digit

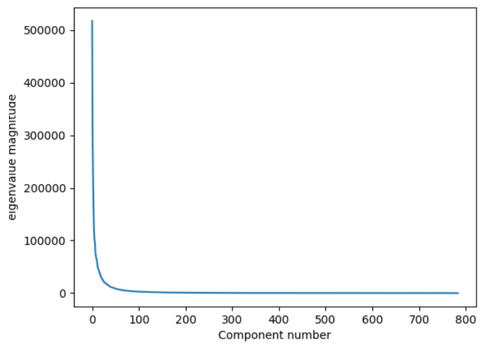

Figure 17: distribution of eigenvalues for 5 digit

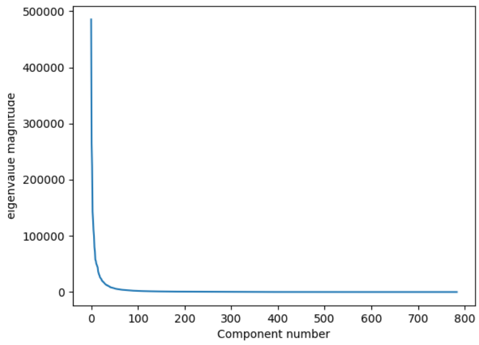

Figure 18: distribution of eigenvalues for 6 digit

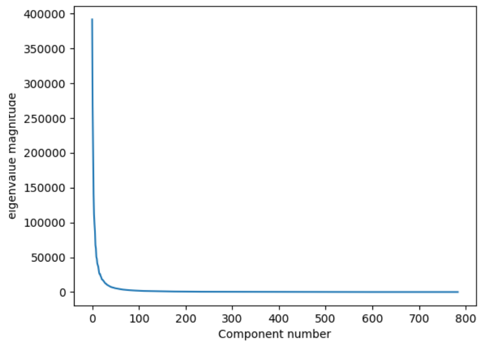

Figure 19: distribution of eigenvalues for 7 digit

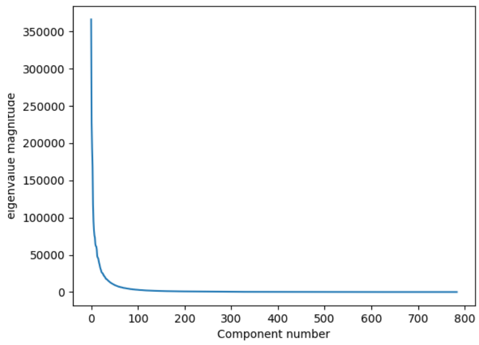

Figure 20: distribution of eigenvalues for 8 digit

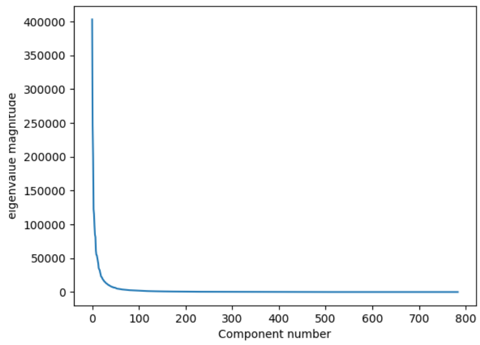

Figure 21: distribution of eigenvalues for 9 digit

4.3 Plotting the images

I have plotted the 3 images side by side and it has been stored under q4 under results directory

We can learn from the image plots that the principal mode of variation signifies in almost all the digits the angle with the horizontal that they have been written with. This is indeed the expected prinicpal mode of variation as analyzing through the dataset, one can find that the difference in angle is more acute and characterising for hand written numbers and the algorithm predicts exactly that. For the digit 1, We see that almost no one writes the digit in a angle more than 90 degree and it is common to write it slanted. We see that as we go away from the mean the slant decreases or increases along the principal mode of variation.We can clearly see that it is common to write a slanted 1, and people tend to make it more slanted or straight but almost never greater than 90 degree.

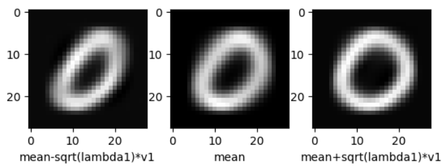

Figure 22: Digit 0

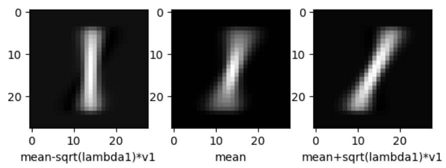

Figure 23: Digit 1

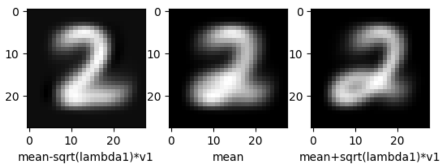

Figure 24: Digit 2

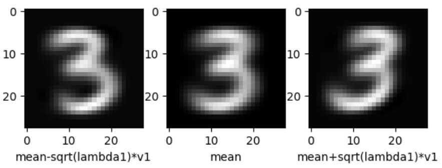

Figure 25: Digit 3

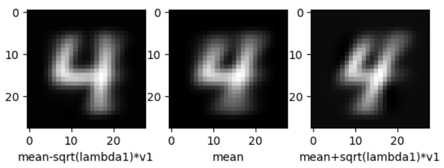

Figure 26: Digit 4

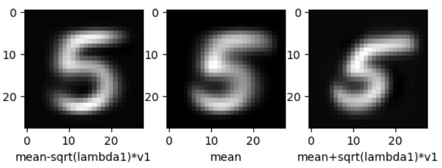

Figure 27: Digit 5

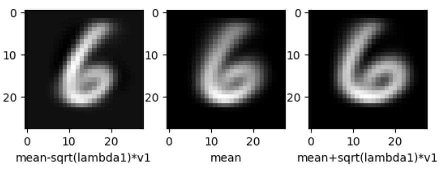

Figure 28: Digit 6

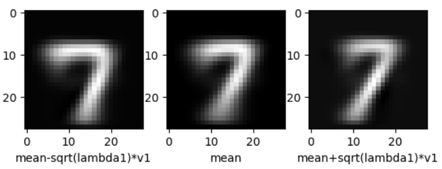

Figure 29: Digit 7

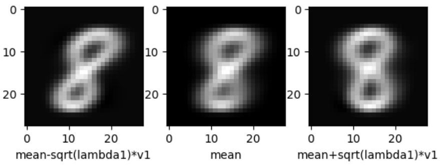

Figure 30: Digit 8

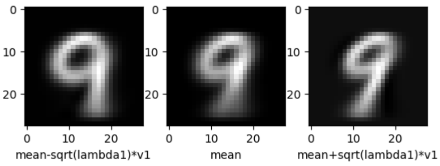

Figure 31: Digit 9

5 Question 5

We obtain the needed 84 co-ordinates by choosing the 84 dimensions that best represent the information. We can achieve this by obtaining the eigenvalues that display the maximum variation. We then project the 28*28 co-ordinated onto the chosen 84 eigenvectors thus obtaining the images in a 84 dimensional space.
In order to reconstruct, we must place the 84 coordinates back in to the 28*28 dimension and we can do so through the below algorithm -

$$PC = X * V$$

Now reconstruction,

$$PC * V^{-1} = X$$

(multiplying V inverse on both sides in above equation)

$$PC * V^T = X'$$

(since V is orthonormal(eigen vectors) V inverse is V transpose)

where PC denotes the Principal components (reduced dimension eigenvectors), V denotes the 84 eigen vectors corresponding to the 84 eigenvalues with maximal variance and X denotes the original dataset as (N,28*28) and X' denotes reconstructed datapoint as (N,28*28)

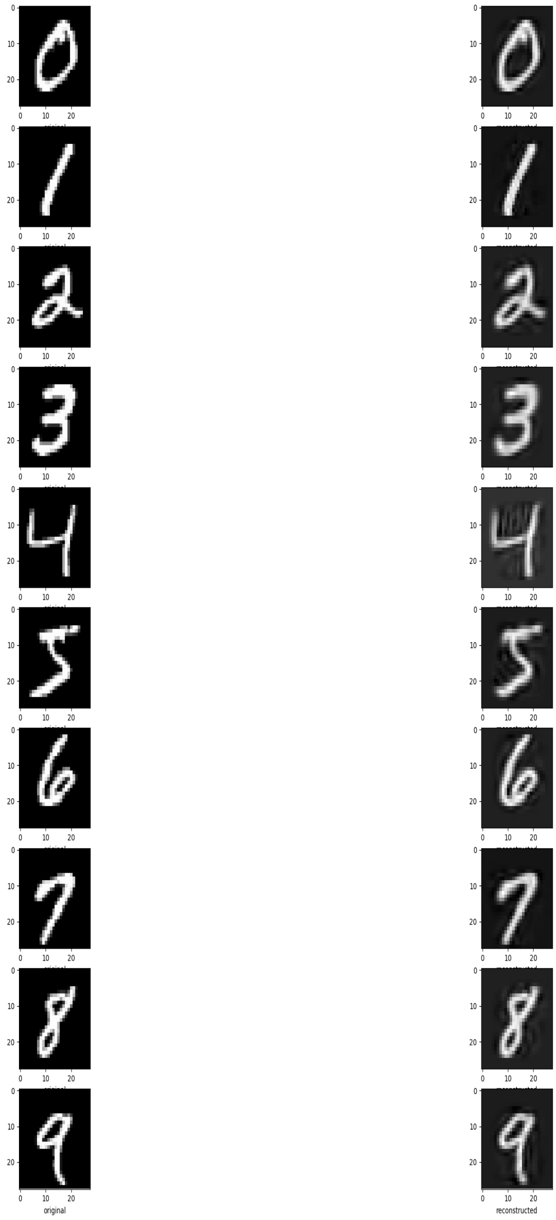

Figure 32: Reconstruction of digits

6 Question 6

6.1 Mean and Eigen-vectors

Data of all 16 fruit images was imported and stored in an array named $fruits^{16x19200}$.
$Mean^{1x19200}$, $standard^{16x19200}$ and covariance $\mathbf{C}^{\mathbf{19200x19200}}$ matrices are calculated. 10 biggest eigen values of C and corresponding eigen-vectors are found and sorted. Plots of the mean and top 4 eigen values and the mean are displayed as images below.

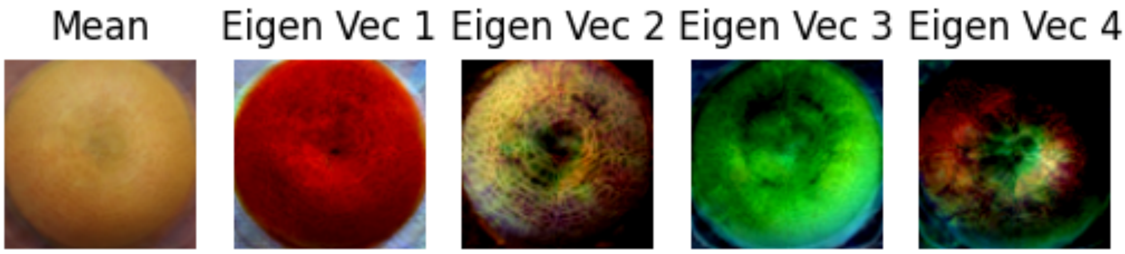

Figure 33: Visualising Mean and Principal Eigen Vectors

6.2 Closest Representation Reconstruction

R (closest representation reconstruction of image I with 4 principle eigen-vectors) can be written as linear combination of mean vector and 4 principle eigen-vectors. For any matrix, Frobenius norm

$$\|A_{\text{frob norm}}\| = \sqrt{\sum_i \sum_j (A_{\text{ij}})^2} \tag{2}$$

Now we change the eigen-basis of covariance matrix to:-

$$\hat{v}_1{}^{19200\text{x}1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ ... \\ 0 \end{bmatrix} \hat{v}_2{}^{19200\text{x}1} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ ... \\ 0 \end{bmatrix} \quad ... \quad \hat{v}_{19200}{}^{19200\text{x}1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ ... \\ 1 \end{bmatrix} \tag{3}$$

$$Now,\ R = a_1\hat{mean} + a_2\hat{eig}_1 + a_3\hat{eig}_2 + a_4\hat{eig}_3 + a_5\hat{eig}_4 \tag{4}$$

$$\hat{mean} = \sum_i u_{\text{i}}\hat{v}_{\text{i}} \quad where\ u_{\text{i}}\ is\ a\ real\ number \tag{5}$$

$$I = \sum_i r_{\text{i}}\hat{v}_{\text{i}} \quad where\ r_{\text{i}}\ is\ a\ real\ number \tag{6}$$

Frobenius norm on matrix of difference between R & I is used.

$$\Delta = I - R \tag{7}$$

$$\|\Delta_{\text{frob}}\| = (r_1 - u_1a_1 - a_2)^2 + (r_2 - u_2a_1 - a_3)^2 + (r_3 - u_3a_1 - a_4)^2 + (r_4 - u_4a_1 - a_5)^2 + \sum_{i=5}^{19200} (r_{\text{i}} - u_{\text{i}}a_1)^2 \tag{8}$$

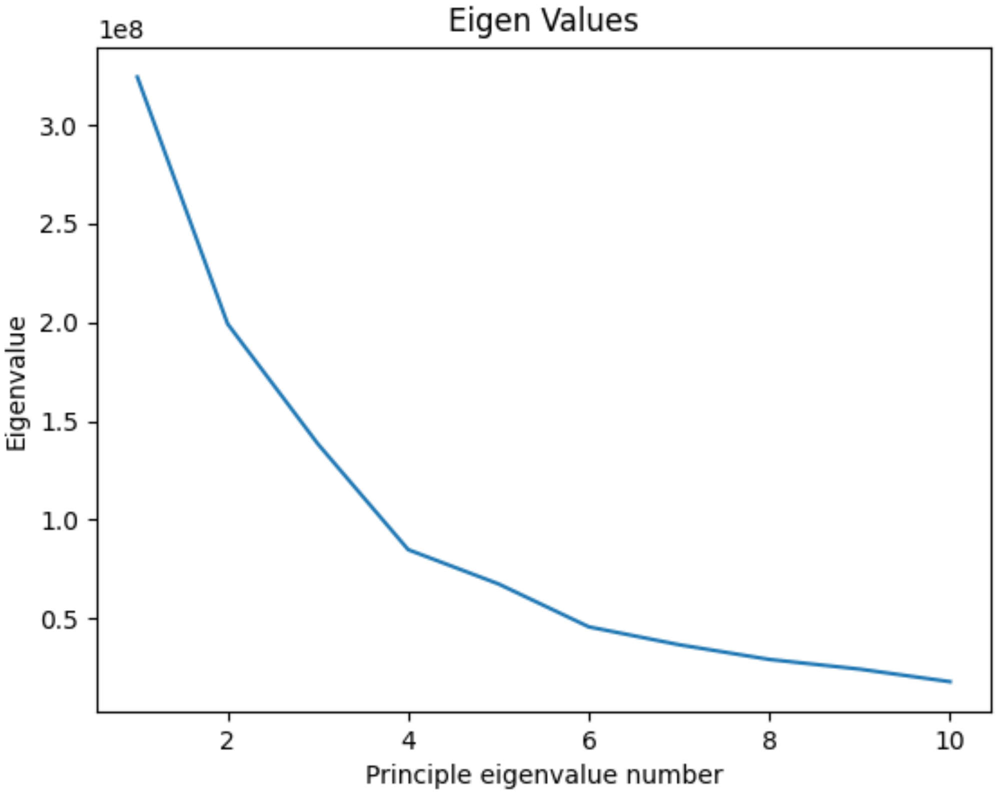

Figure 34: 10 Largest Eigen-values

To make$\|\Delta_{\text{frob}}\|_{\text{ min}}$, first 4 terms of can be made 0.i.e.

$$a_2 = r_1 - u_1a_1,\ a_3 = r_2 - u_2a_1 \ ...\ a_5 = r_4 - u_4a_1 \tag{9}$$

Then,

$$\frac{d}{da_1}\sum_{i=5}^{19200}(r_{\text{i}} - u_{\text{i}}a_1)^2 = \sum_{i=5}^{19200} 2(r_{\text{i}} - u_{\text{i}}a_1)(-u_{\text{i}}) = 0 \tag{10}$$

$$\therefore a_1 = \frac{\sum_{i=5}^{19200} r_{\text{i}}u_{\text{i}}}{\sum_{i=5}^{19200} {u_{\text{i}}}^2} \tag{11}$$

Hence we get all a_{i} and finally...

$$\bar{R} = a_1 m\hat{e}an + \sum_{i=1}^{4}(r_{\text{i}} - u_{\text{i}}a_1)e\hat{i}g_{\text{i}} \tag{12}$$

6.3 Sampling Random Images

To sample random images with the mean and 4 principle eigenvectors, first we define

$$Gen\ (generated\ image\ vector) = m\hat{e}an + Aw \tag{13}$$

$$AA^{\text{T}} = C\ (Covariance\ matrix) \tag{14}$$

$$A = \sqrt{C} \tag{15}$$

w is a vector consisting of independent and identically distributed variables from standard univariate gaussian distribution. We chose it randomly with the np.random.randn() function in numpy.

C is a 19200x19200 matrix, calculating square root of C with built-in functions is difficult. Hence we use the method we had used in Q2 to determine the matrix A.

$$A = QDQ^{\text{T}} \tag{16}$$

where Q is the 4x19200 matrix containing the principle eigen-vectors and
D^{4x4} is a diagonal matrix with square roots of eigen-values in on the diagonal. After A is calculated, `Gen` is calculated easily. Then `Gen` is reshaped and rescaled to the $[0, 1]$ range and the generated images are displayed.

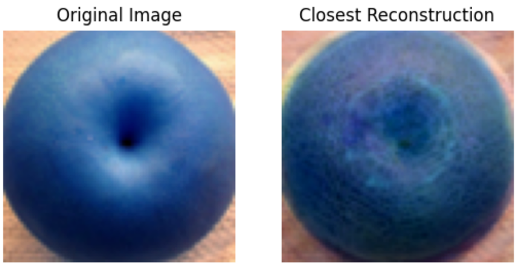

Figure 35: Reconstructed Fruit 1

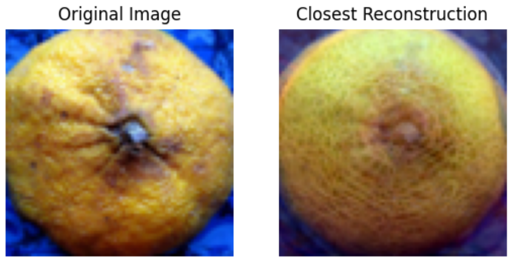

Figure 36: Reconstructed Fruit 2

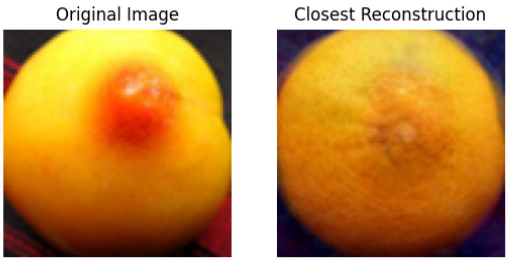

Figure 37: Reconstructed Fruit 3

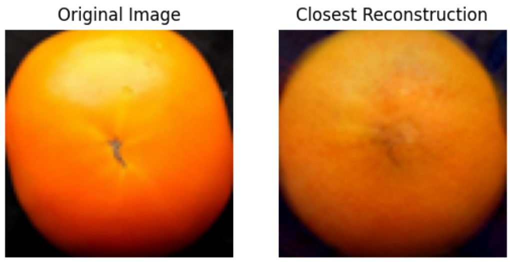

Figure 38: Reconstructed Fruit 4

Figure 39: Reconstructed Fruit 5

Figure 40: Reconstructed Fruit 6

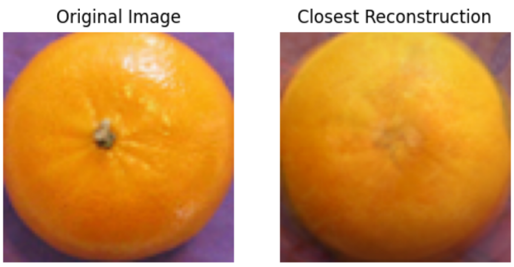

Figure 41: Reconstructed Fruit 7

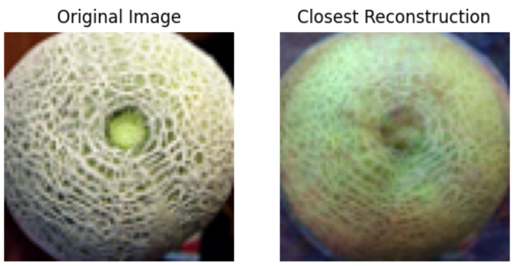

Figure 42: Reconstructed Fruit 8

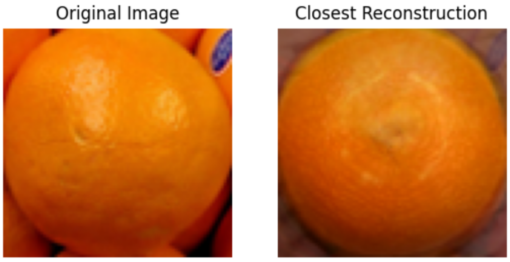

Figure 43: Reconstructed Fruit 9

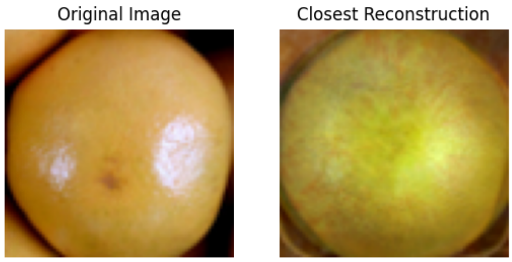

Figure 44: Reconstructed Fruit 10

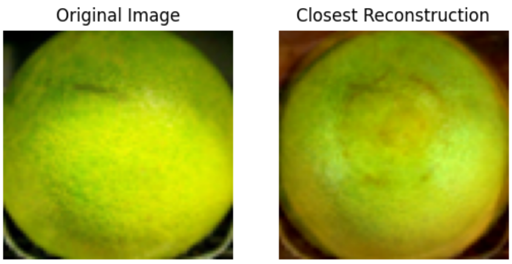

Figure 45: Reconstructed Fruit 11

Figure 46: Reconstructed Fruit 12

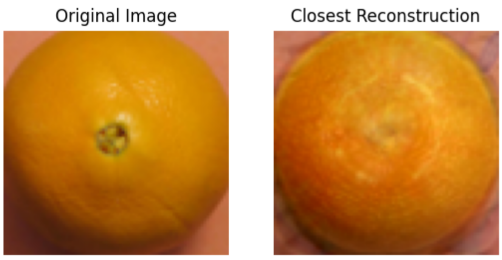

Figure 47: Reconstructed Fruit 13

Figure 48: Reconstructed Fruit 14

Figure 49: Reconstructed Fruit 15

Figure 50: Reconstructed Fruit 16

Figure 51: Generated Fruit Images with 4 Eigen vectors