CS 215: Assignment02 Report

Poorna Teja and Karthikeya

October 2021

Contents

1 Sampling within a Euclidean Plane **2**
1.1 Running the code . . . 2
1.2 Algorithm to generate uniformly distributed random points(UDRPs) 2
1.2.1 In an ellipse . . . 2
1.2.2 In a triangle . . . 2
1.3 Histograms of the sample points . . . 3
1.3.1 Ellipse . . . 3
1.3.2 Triangle . . . 3

2 Multivariate Gaussian **4**
2.1 Running the code . . . 4

3 PCA and Hyperplane Fitting **5**
3.1 Running the code . . . 5
3.2 Approximate linear relationship between X and Y . . . 5
3.3 Plots . . . 5

4 Principal Component Analysis **6**
4.1 Running the code . . . 6
4.2 Plot of eigen values . . . 6

5 Principal Component Analysis (PCA) for Dimensionality Reduction **16**
5.1 Running the code . . . 16
5.2 Comparision of original and compressed images . . . 16

6 Principal Component Analysis (PCA) for Another Image Dataset **20**
6.1 Running the code . . . 20
6.2 Mean and eigen vectors . . . 20
6.3 Closest representations . . . 20
6.4 New fruits . . . 21

1 Sampling within a Euclidean Plane

1.1 Running the code

1.2 Algorithm to generate uniformly distributed random points(UDRPs)

1.2.1 In an ellipse

Consider an ellipse in a polar coordinate system(r,θ).A point (r,θ)'s occurrence in the UDRP generator has a probability density proportional to r which can be intuitively understood from the fact that the infinitesimally small area around it is $\sqrt{\frac{b}{a}}rdrd\theta$.The random point generator can be divided into two independent parts,

- The θ generator which is $U(0,2\pi)$.
- The r generator has the range $(0,a)$ and probability density proportional to r.

The random point is calculated from the output of the θ generator(θ_1), the r generator(r_1) as $(r_1cos(\theta_1),\frac{b}{a}r_1sin(\theta_1))$.The task is to construct the r generator.Consider two distributions R_1 and R_2 where R_1 has a range [0,1] and it's probability density is proportional to x, R_2 is $U(0,1)$.Let $y=f(x)$ be an unknown function from $[0,1]$ to $[0,1]$ such that applying the transformation of f on R_1 results in R_2.Let's assume that f is monotonic and use the formula for transformation of RV's $R_1(x) = R_2(f(x))\frac{d}{dx}(f(x))$.
$f(x) = x^2$ is a valid transformation from R1 to R2 so we can construct the r generator using $\mathbf{a*sqrt(U(0,1))}$.

1.2.2 In a triangle

Consider a rectangle encompassing the triangle in consideration.The idea behind generating UDRPs in a rectangle is elementary, consider the rectangle (0,0),(a,0),(0,b),(a,b) the UDRP generator is simply (U(0,a),U(0,b)).We can construct the UDRP generator of the triangle using the generator of the rectangle as

Algorithm 1: UDRP$generator_{trl}$

```
point p = UDRPgenerator_rect()
if p is outside triangle then
    return UDRPgenerator_trl()
return p
```

All the points have the same probability density in UDRP$generator_{rect}()$ and $\frac{A_{trl}}{A_{rect}}$ is the probability for that point to be inside the triangle, clearly the UDRP generator constructed for the triangle has the same probability density for all the points inside the triangle proving it's correctness.

1.3 Histograms of the sample points

1.3.1 Ellipse

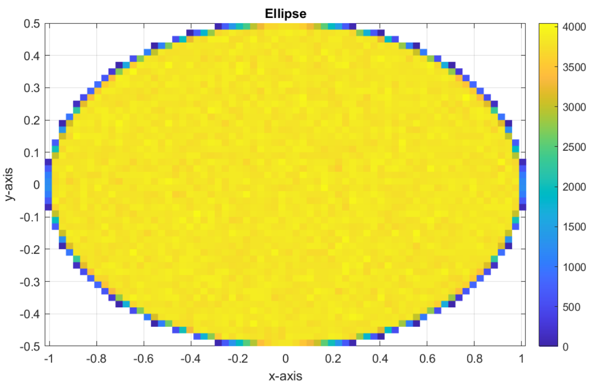

Figure 1: Histogram of the sample points

1.3.2 Triangle

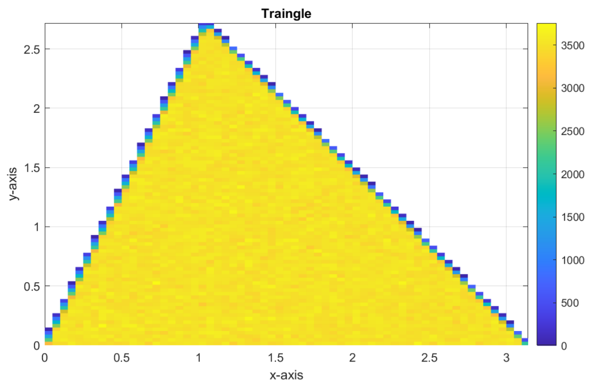

Figure 2: Histogram of the sample points

2 Multivariate Gaussian

2.1 Running the code

3 PCA and Hyperplane Fitting

3.1 Running the code

3.2 Approximate linear relationship between X and Y

3.3 Plots

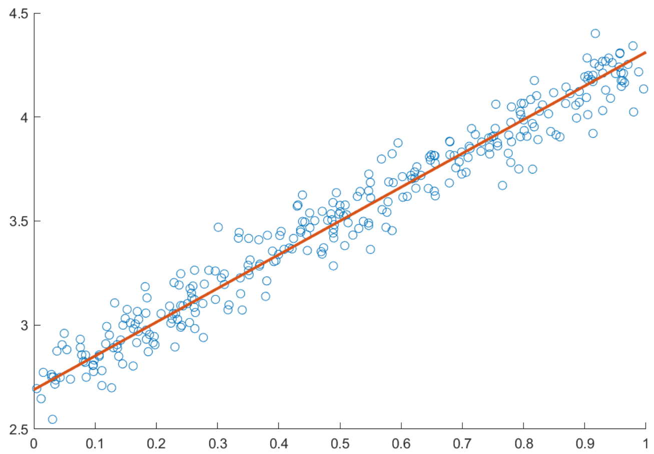

Figure 3: Set1

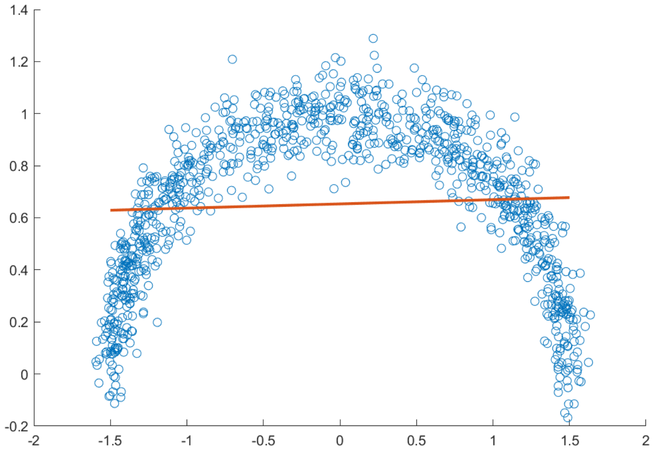

Figure 4: Set2

4 Principal Component Analysis

4.1 Running the code

4.2 Plot of eigen values

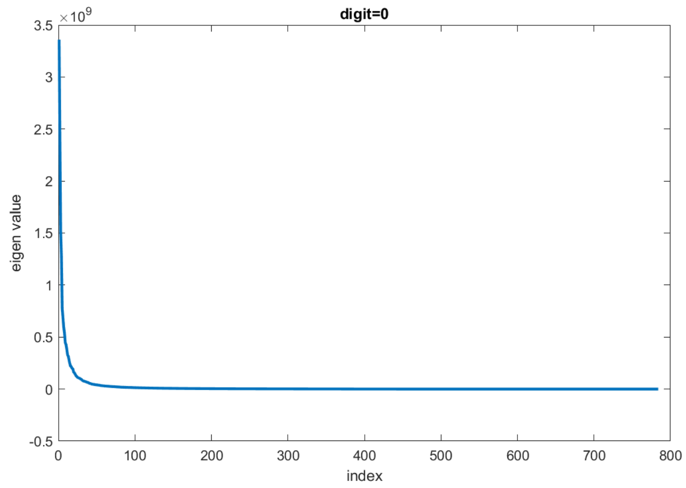

Figure 5: digit 0

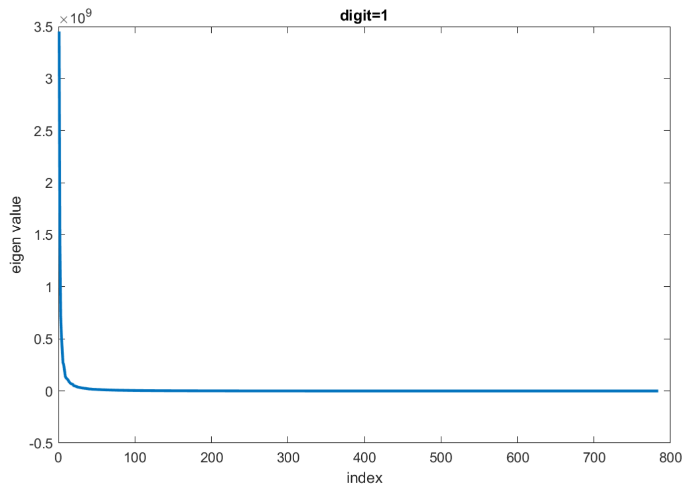

Figure 6: digit 1

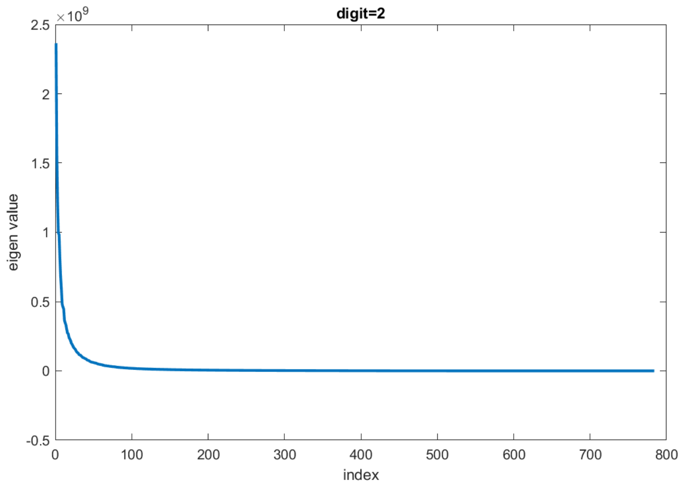

Figure 7: digit 2

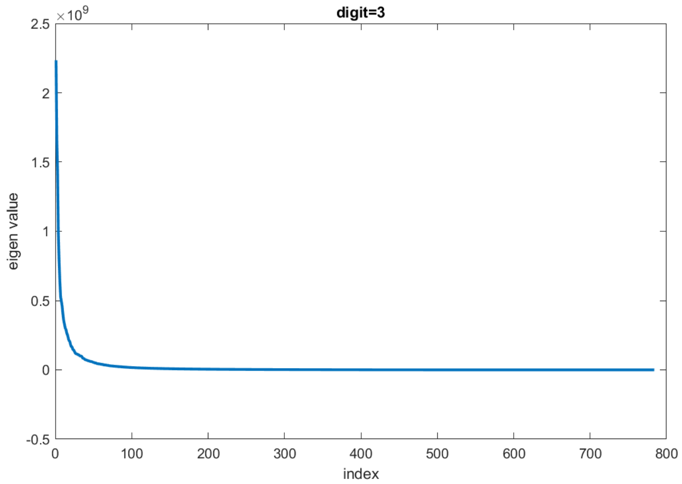

Figure 8: digit 3

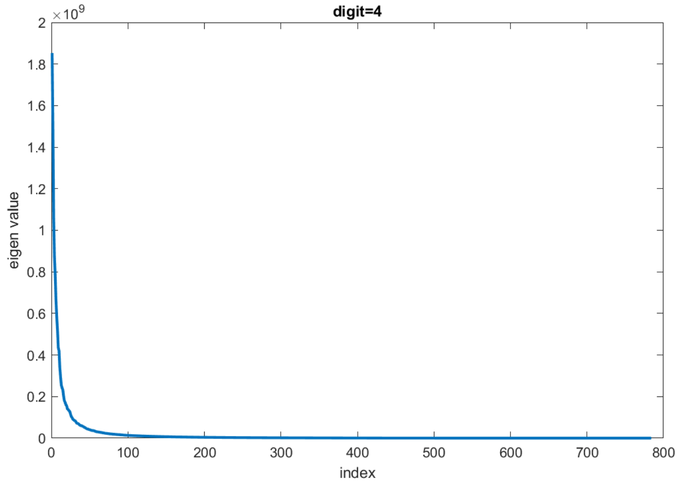

Figure 9: digit 4

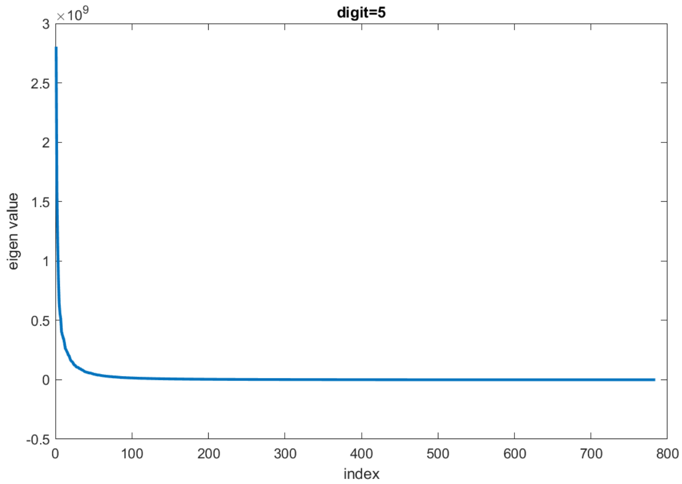

Figure 10: digit 5

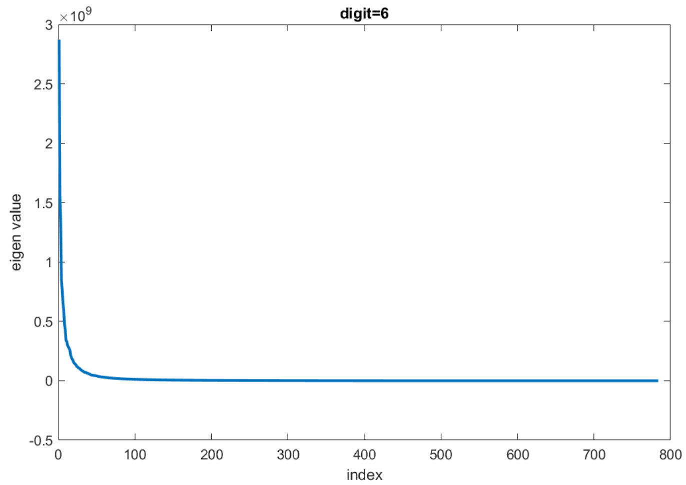

Figure 11: digit 6

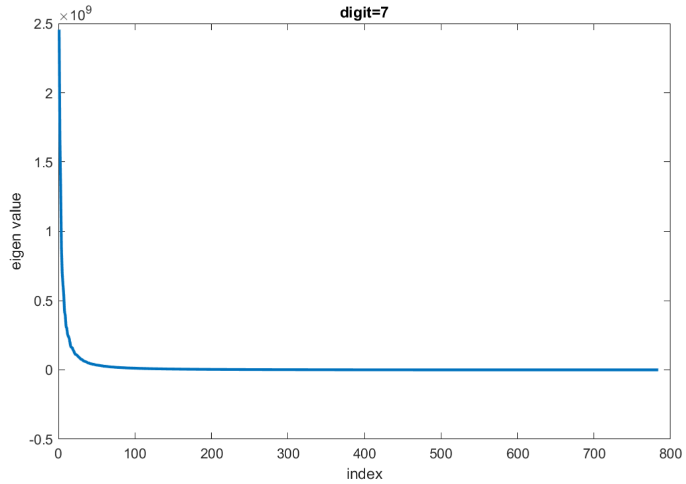

Figure 12: digit 7

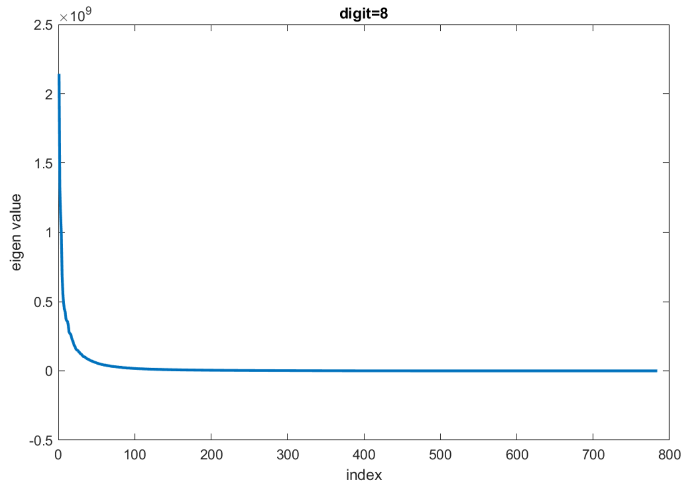

Figure 13: digit 8

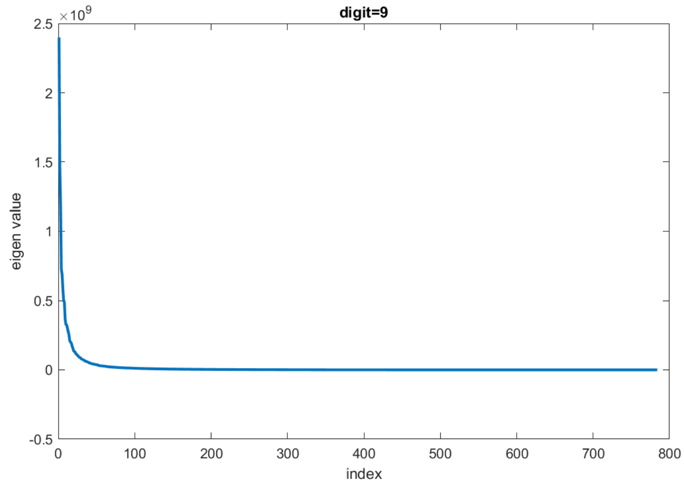

Figure 14: digit 9

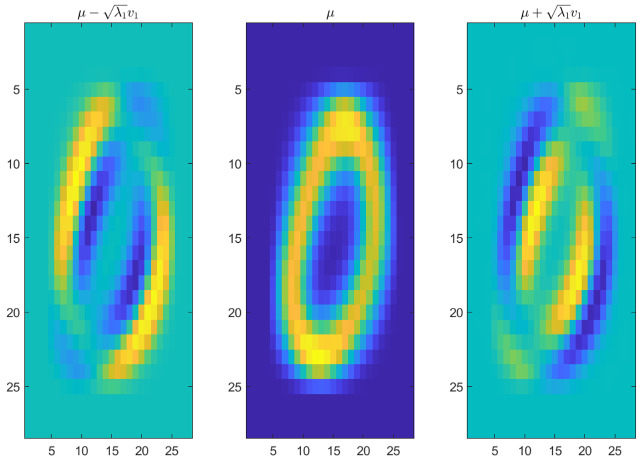

Figure 15: digit 0

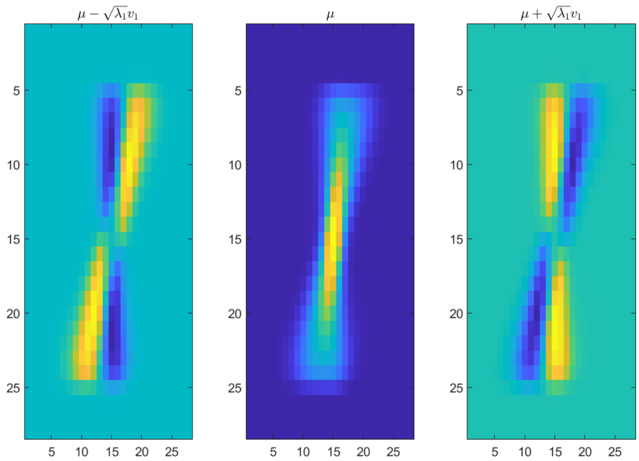

Figure 16: digit 1

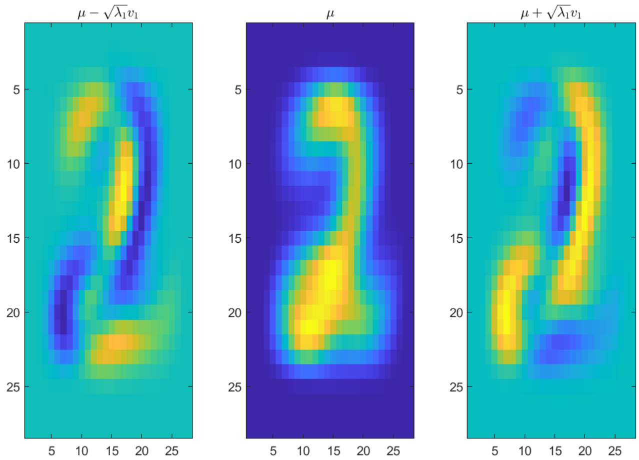

Figure 17: digit 2

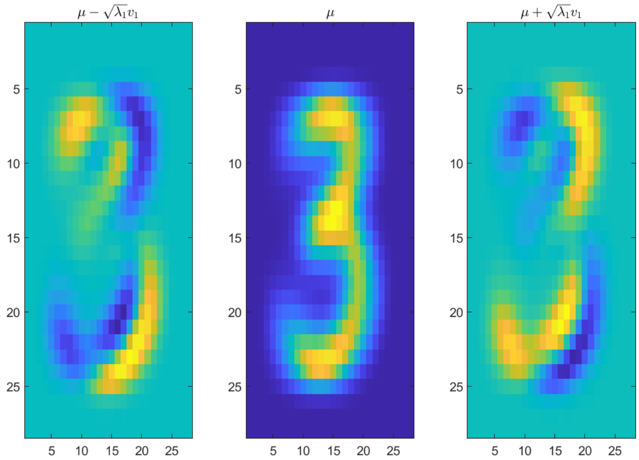

Figure 18: digit 3

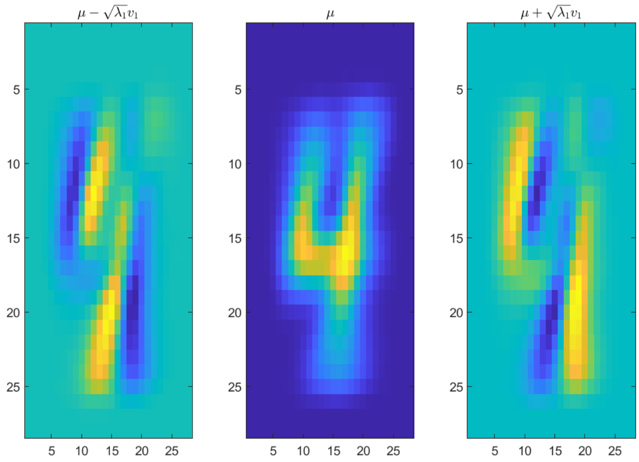

Figure 19: digit 4

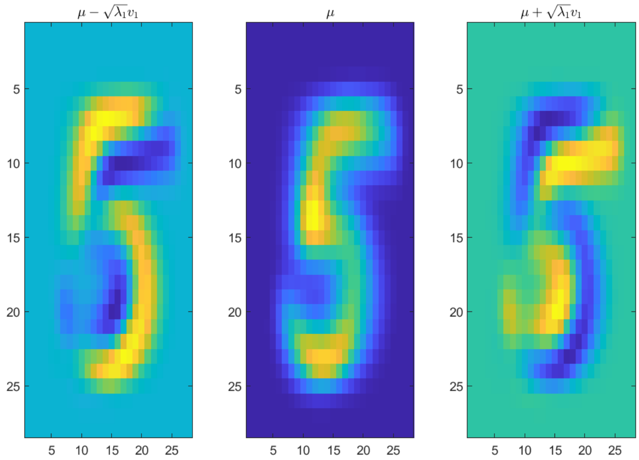

Figure 20: digit 5

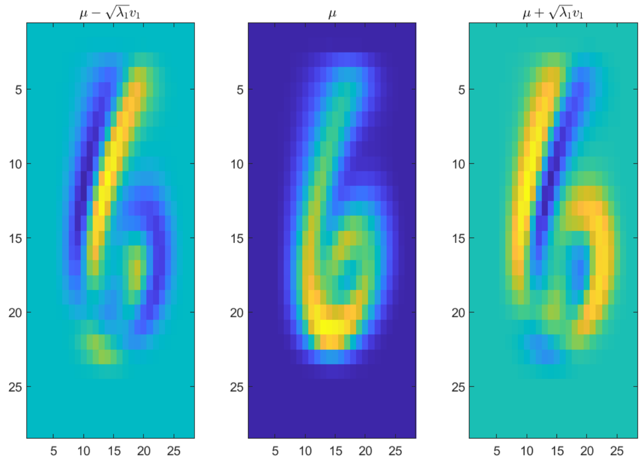

Figure 21: digit 6

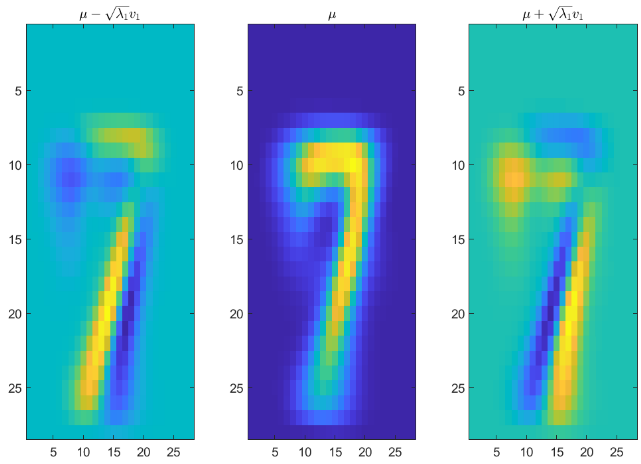

Figure 22: digit 7

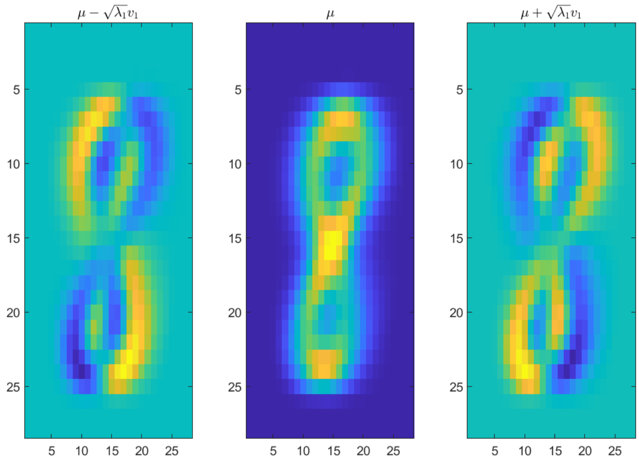

Figure 23: digit 8

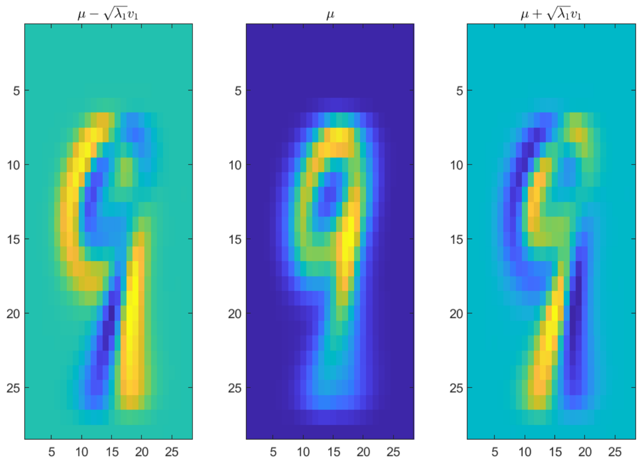

Figure 24: digit 9

5 Principal Component Analysis (PCA) for Dimensionality Reduction

5.1 Running the code

5.2 Comparision of original and compressed images

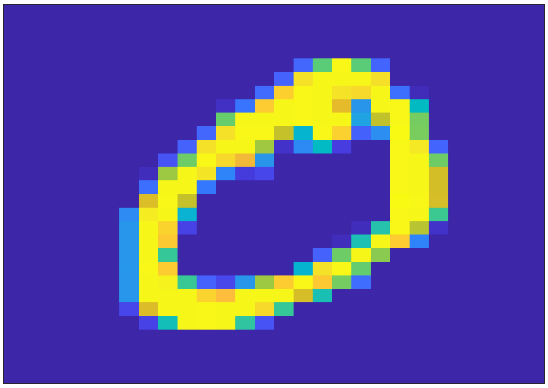

Figure 25: Original

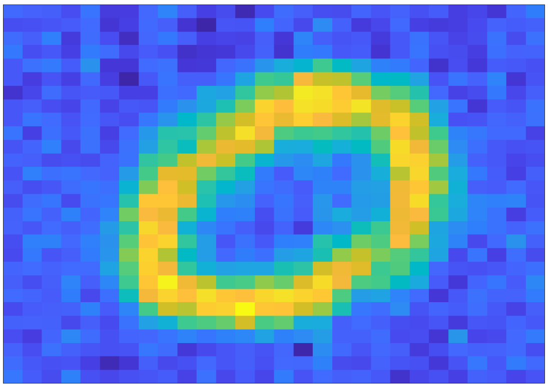

Figure 26: Compressed

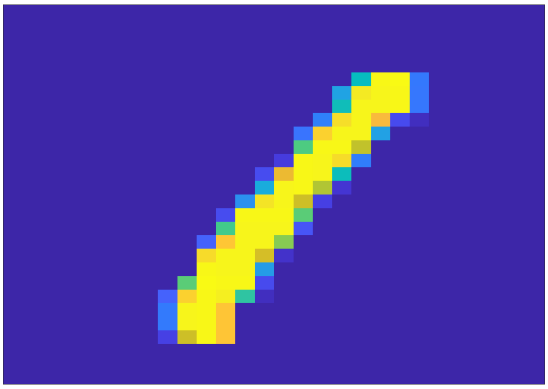

Figure 27: Original

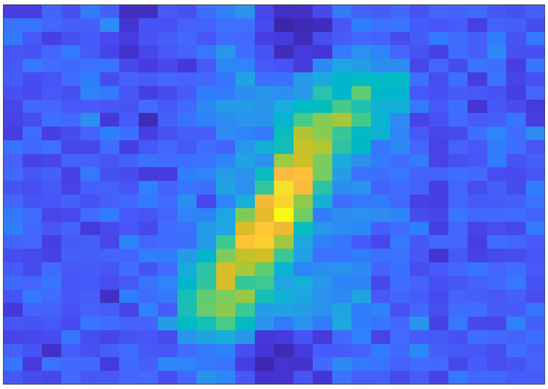

Figure 28: Compressed

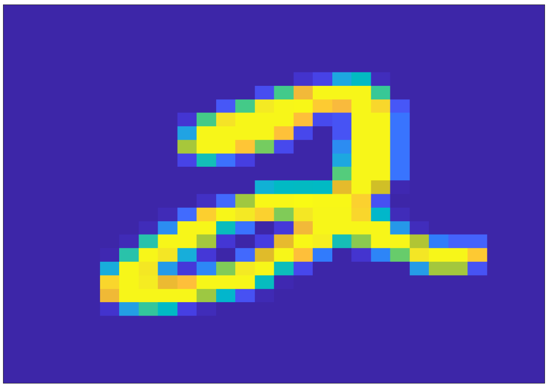

Figure 29: Original

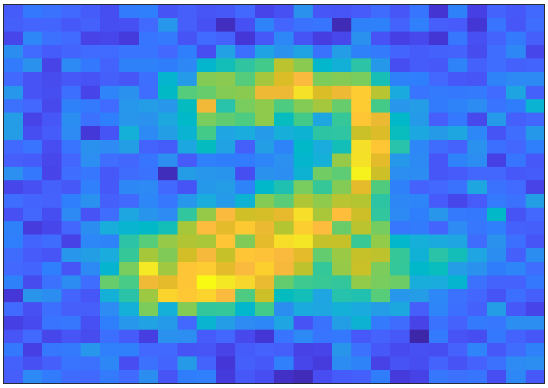

Figure 30: Compressed

Figure 31: Original

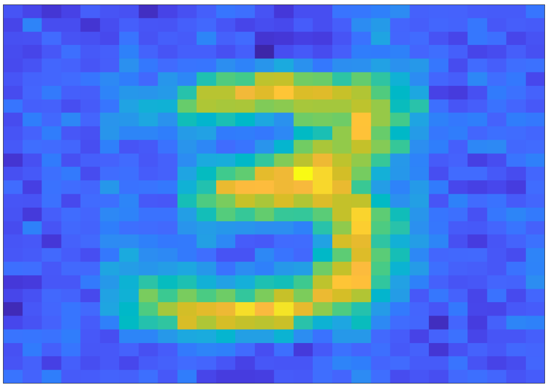

Figure 32: Compressed

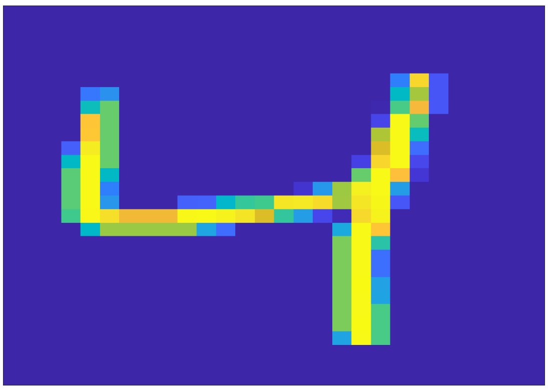

Figure 33: Original

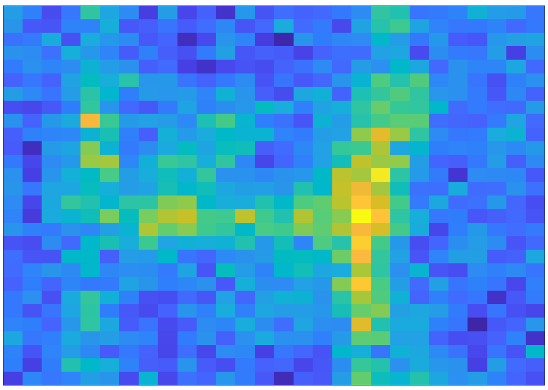

Figure 34: Compressed

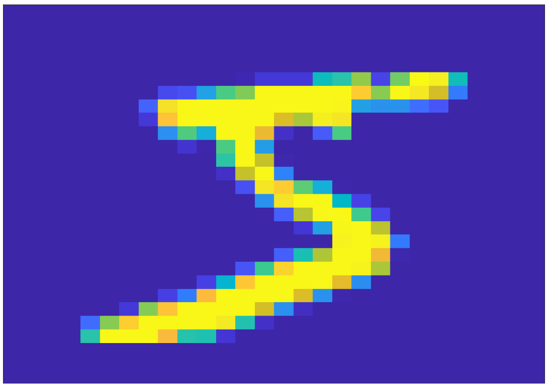

Figure 35: Original

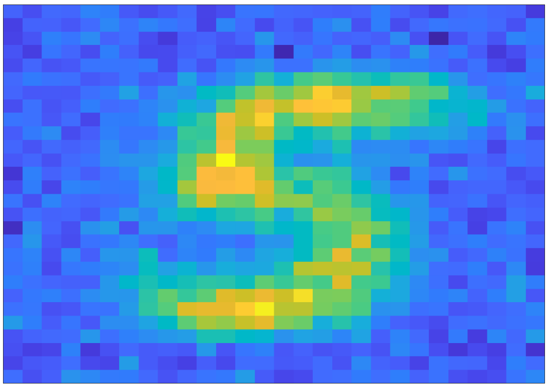

Figure 36: Compressed

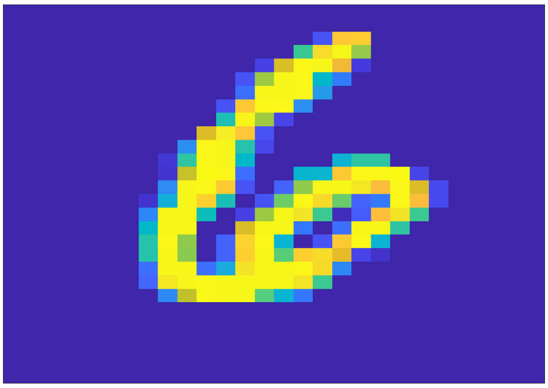

Figure 37: Original

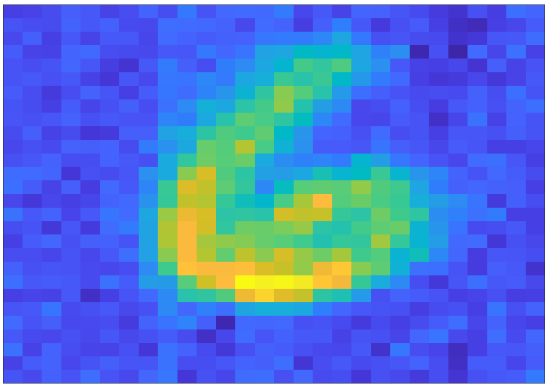

Figure 38: Compressed

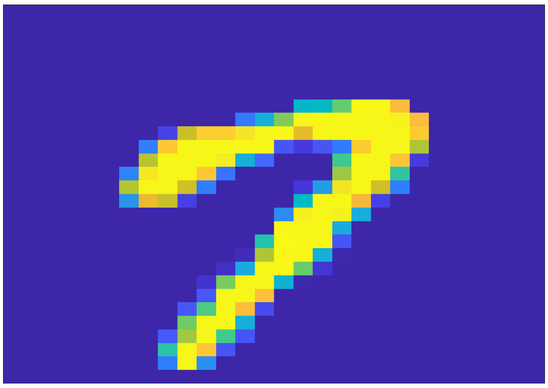

Figure 39: Original

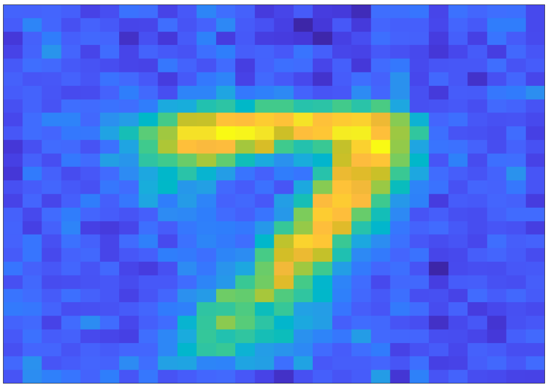

Figure 40: Compressed

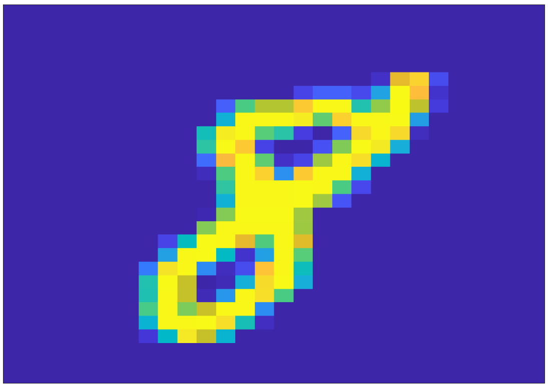

Figure 41: Original

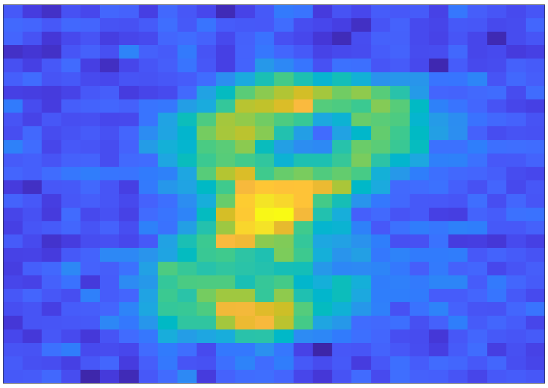

Figure 42: Compressed

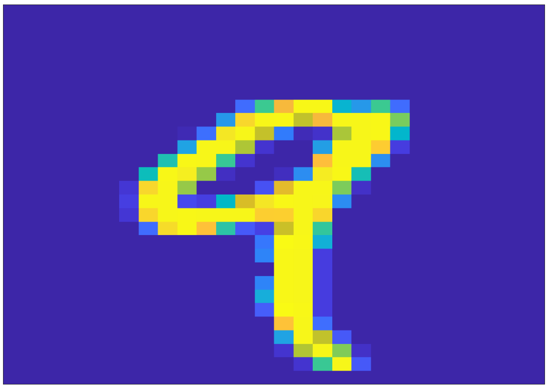

Figure 43: Original

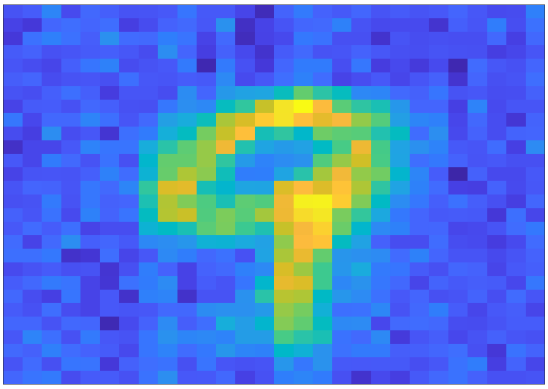

Figure 44: Compressed

6 Principal Component Analysis (PCA) for Another Image Dataset

6.1 Running the code

6.2 Mean and eigen vectors

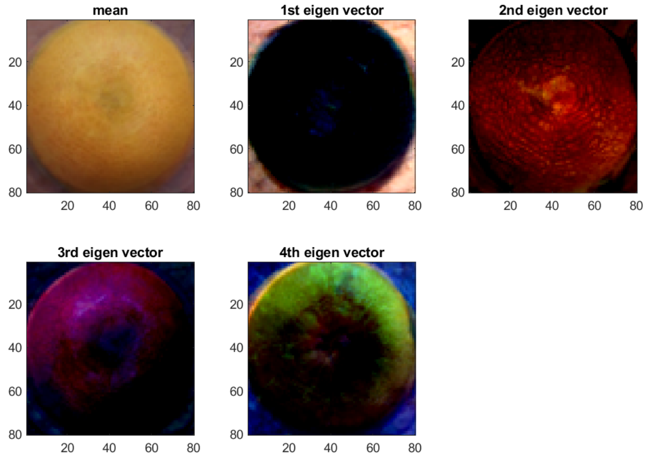

Figure 45: Mean and eigen vectors

6.3 Closest representations

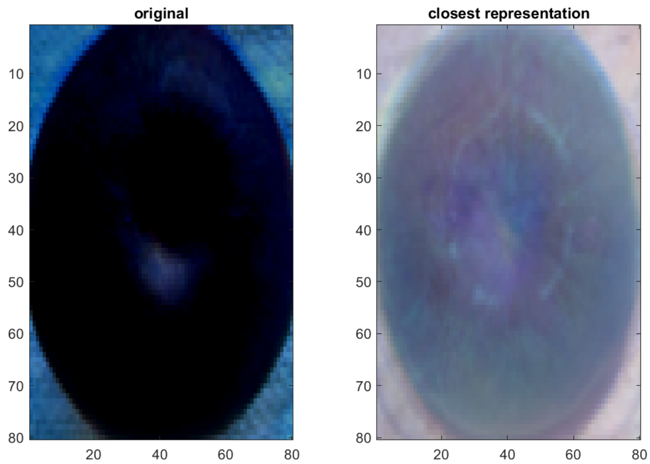

Figure 46: Image 1

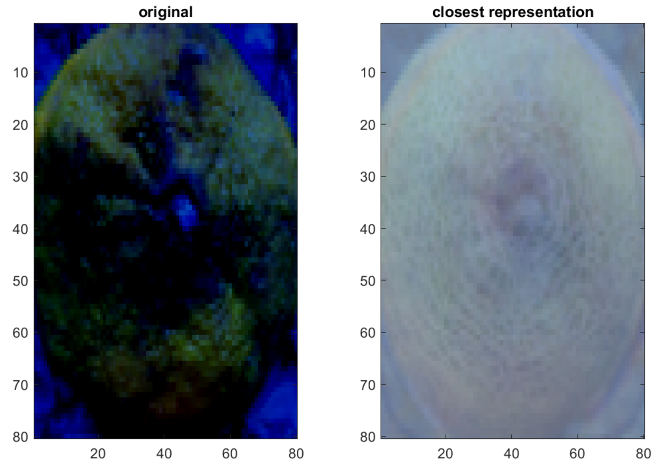

Figure 47: Image 2

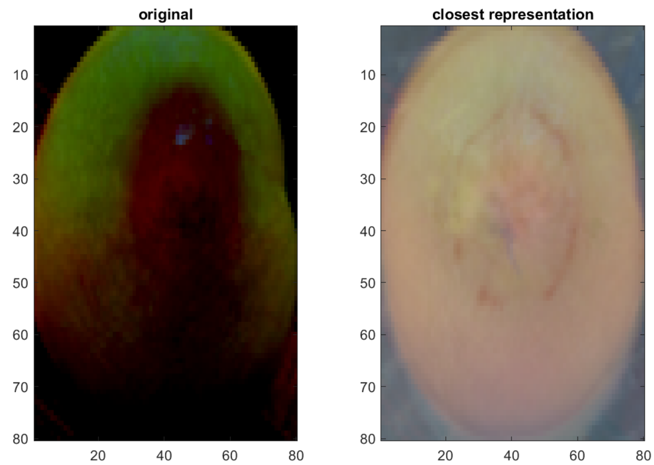

Figure 48: Image 3

6.4 New fruits

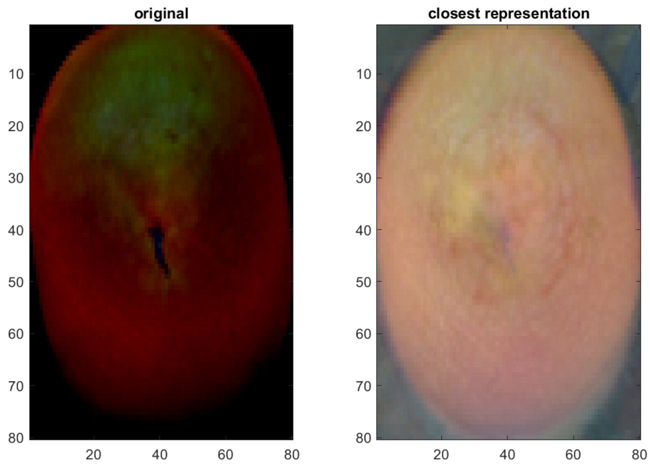

Figure 49: Image 4

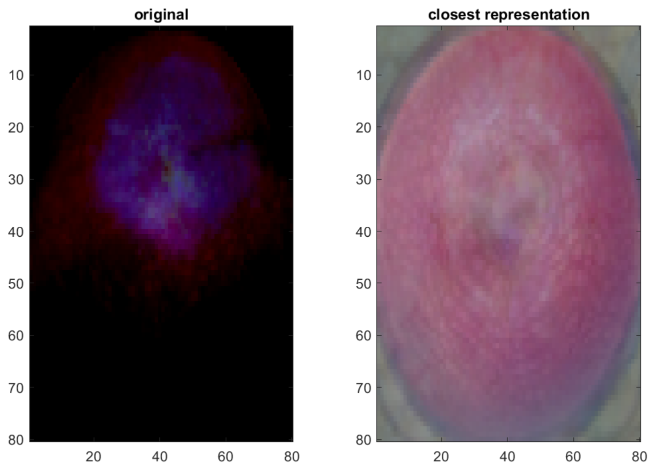

Figure 50: Image 5

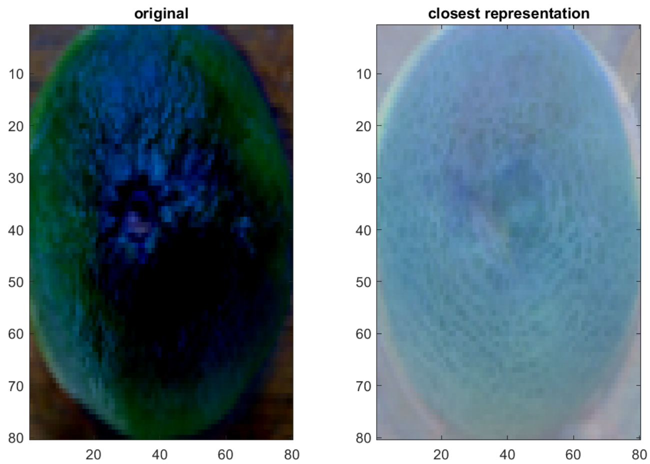

Figure 51: Image 6

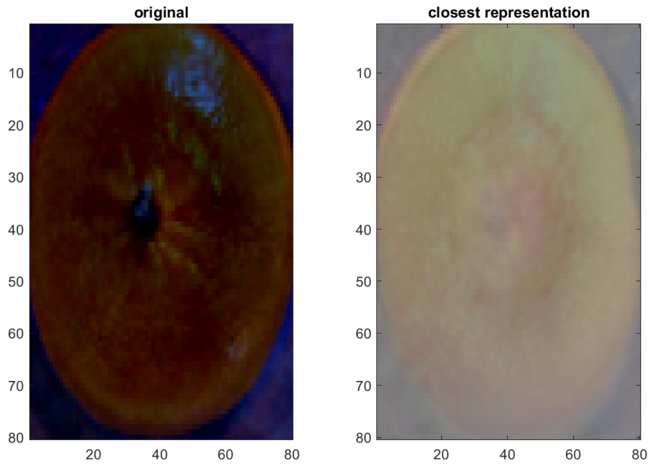

Figure 52: Image 7

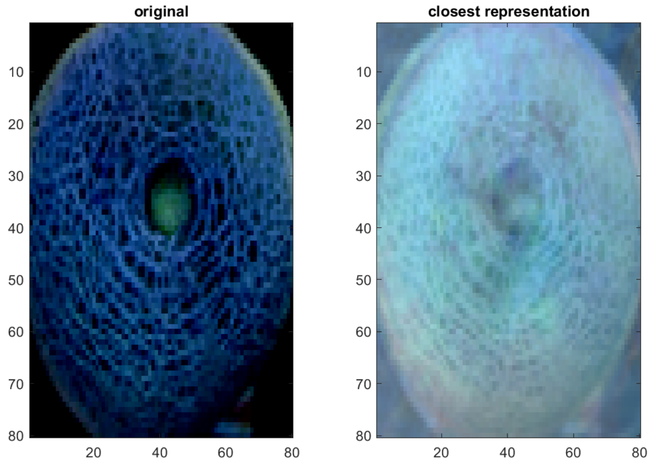

Figure 53: Image 8

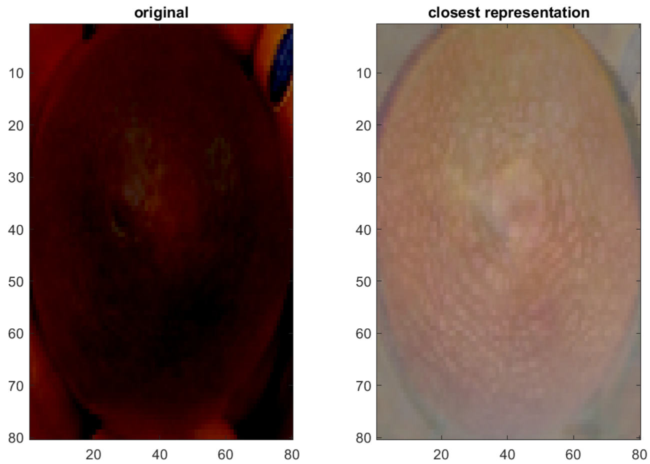

Figure 54: Image 9

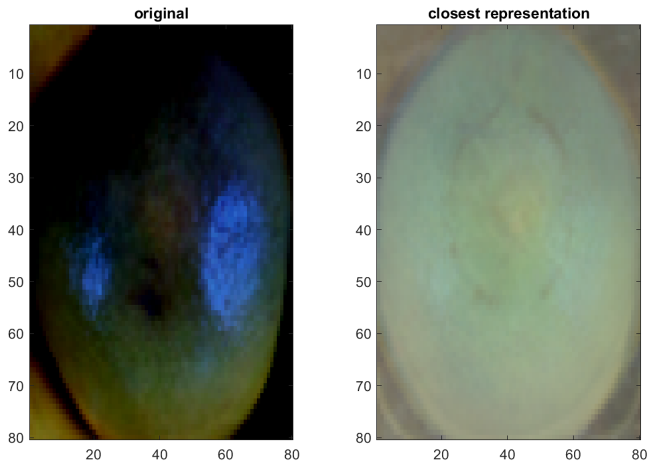

Figure 55: Image 10

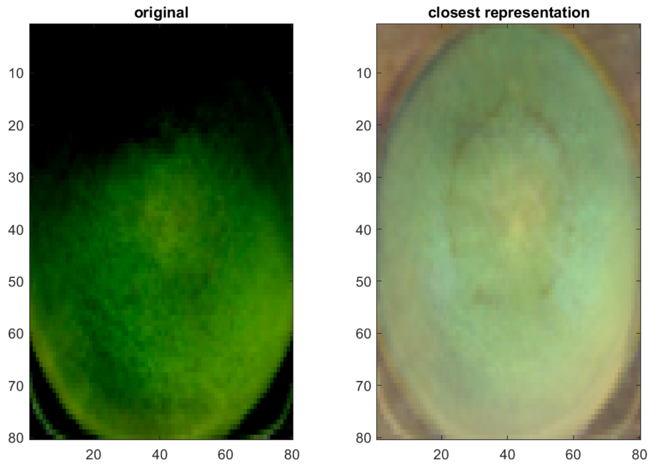

Figure 56: Image 11

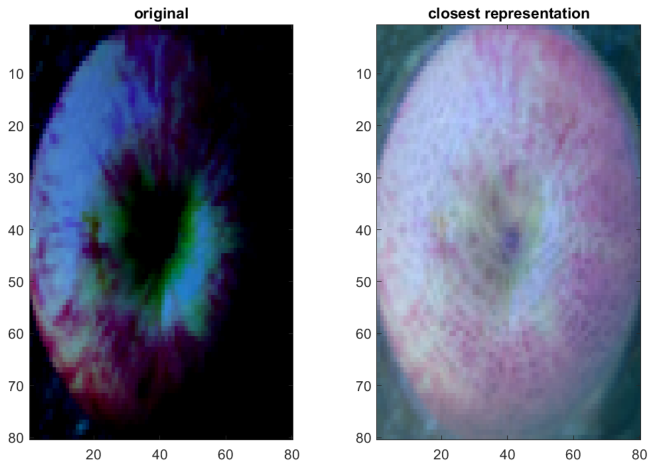

Figure 57: Image 12

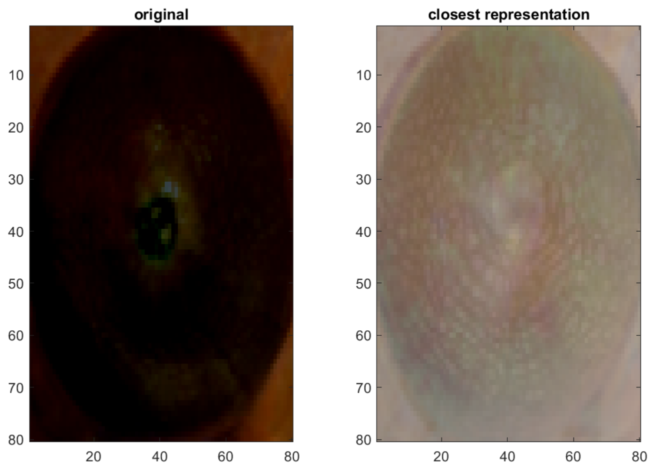

Figure 58: Image 13

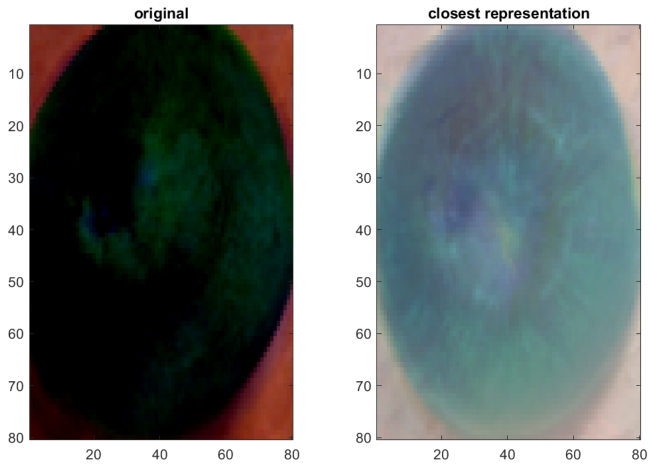

Figure 59: Image 14

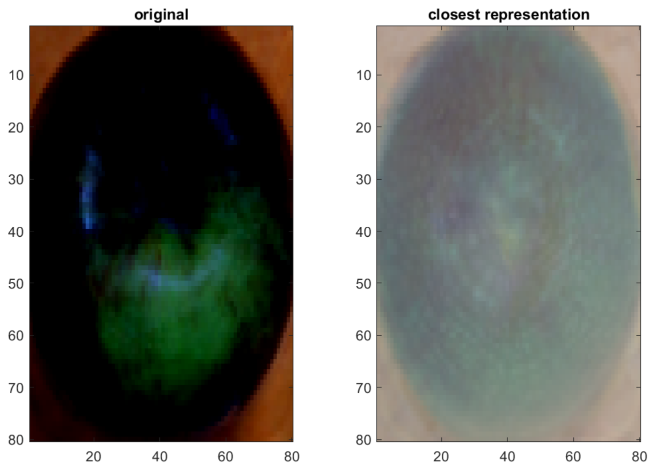

Figure 60: Image 15

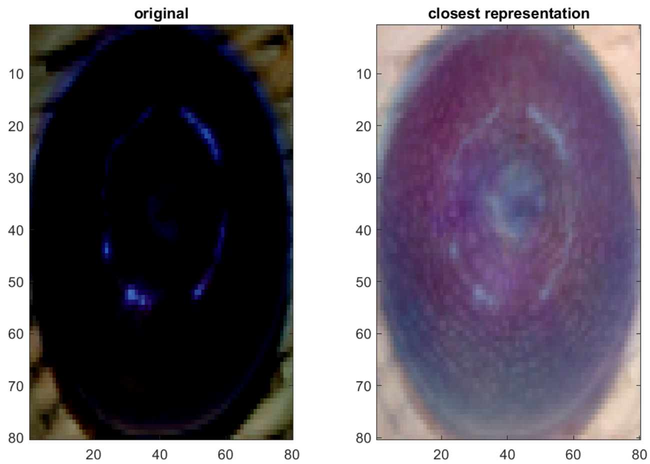

Figure 61: Image 16

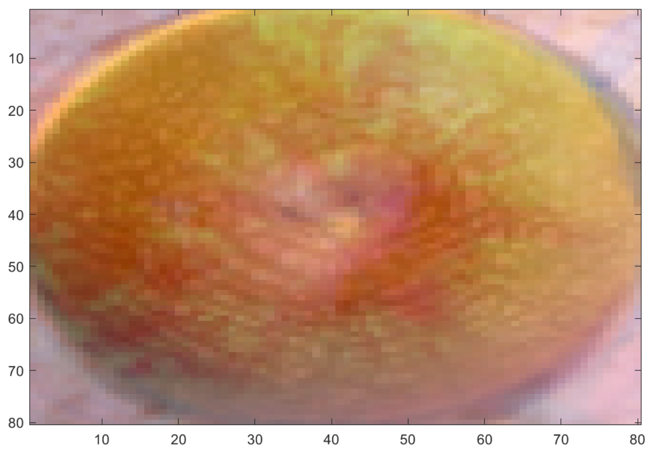

Figure 62: NewFruit 1

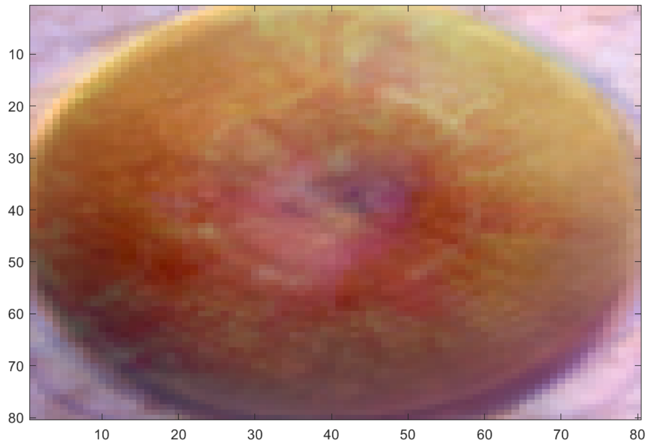

Figure 63: NewFruit 2

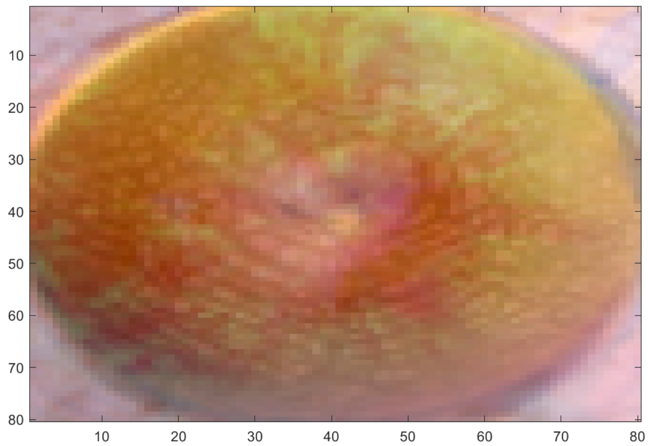

Figure 64: NewFruit 3