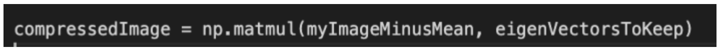
CIS 365 Principal Components Analysis

Dr. Denton Bobeldyk

In this assignment, you will programmatically implement the PCA algorithm as discussed in class. Use the image provided in blackboard.

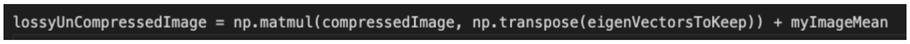
Algorithm:

1. Find the mean of the image
2. Subtract the mean of the image
3. Calculate the covariance matrix
4. Find the Eigenvectors and Eigenvalues of the covariance matrix.
5. Sort the Eigenvectors by their associated Eigenvalues (if not done already)
6. Set the numberOfEigenVectorsToKeep = 15
7. Output the percentage of variance that those eigenvectors account for
8. Create a variable 'eigenVectorsToKeep' that only contains that number of eigenVectors
9. Create a variable 'compressedImage' it should be calculated by using:



```
compressedImage = np.matmul(myImageMinusMean, eigenVectorsToKeep)
```

10. Create a lossy uncompressed image from the above variable:



```
lossyUnCompressedImage = np.matmul(compressedImage, np.transpose(eigenVectorsToKeep)) + myImageMean
```

11. Output the compressed image

Repeat the above steps with the variable numberOfEigenVectorsToKeep = {15, 100, 200}

Hand-in:

1. A screenshot of the lossy image for each of the above values (numberOfEigenVectorsToKeep)
2. The source code file used to generate the images (no zip files please)

Grading Rubric:

	0	50%	100%
Screenshots of lossy compressed images	No screenshots of the lossy compressed images	2 screenshots of the lossy compressed images	3 screenshots of the lossy compressed images
Algorithm implementation	Not clearly implemented	Partially implemented	Fully and correctly implemented
Percent variance output	Not output	output for some or incorrectly output	Correctly output for all

EXTRA: Update the above algorithm to work for any non-square image and document (in your code) what you did and why you did it that way.