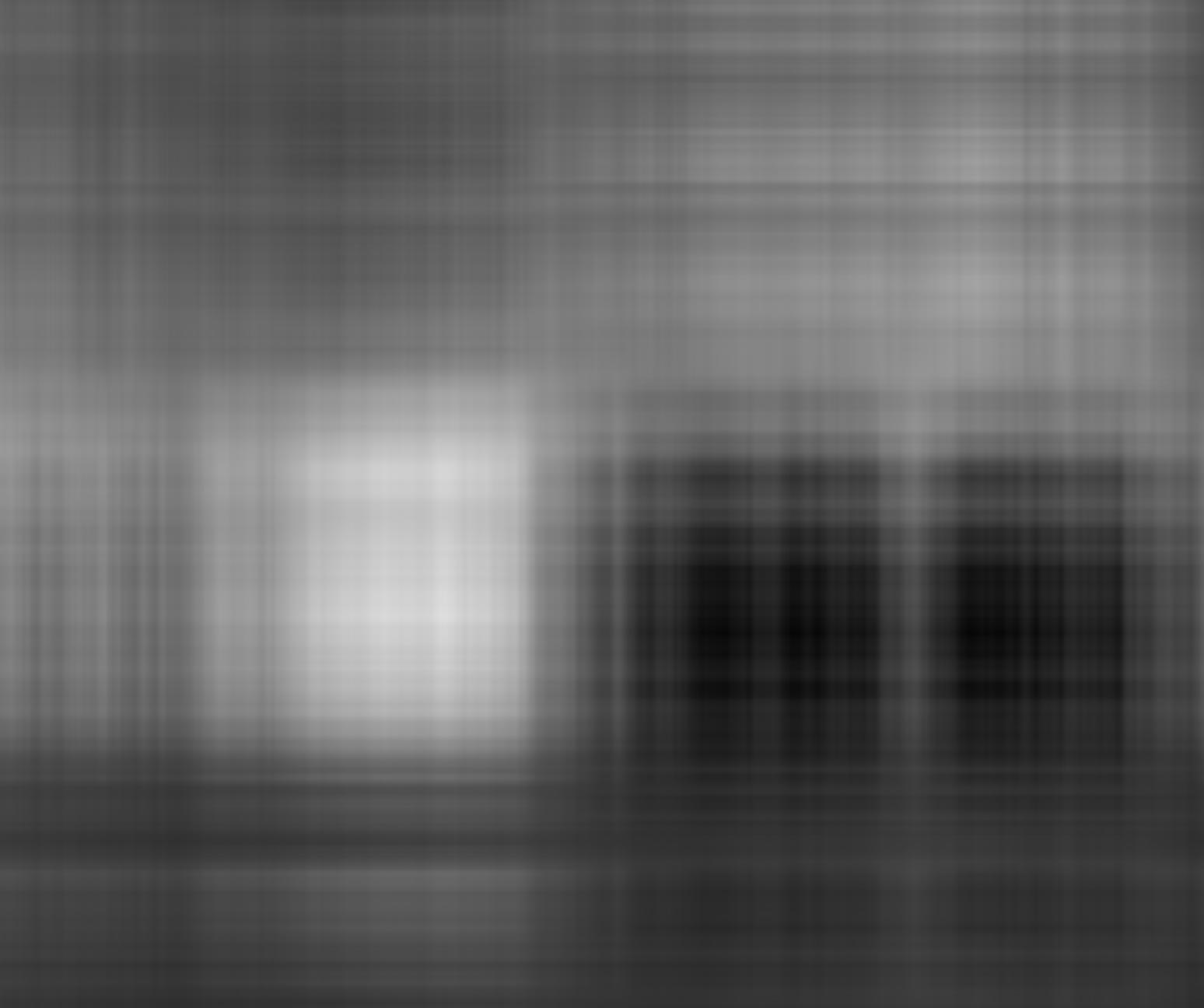
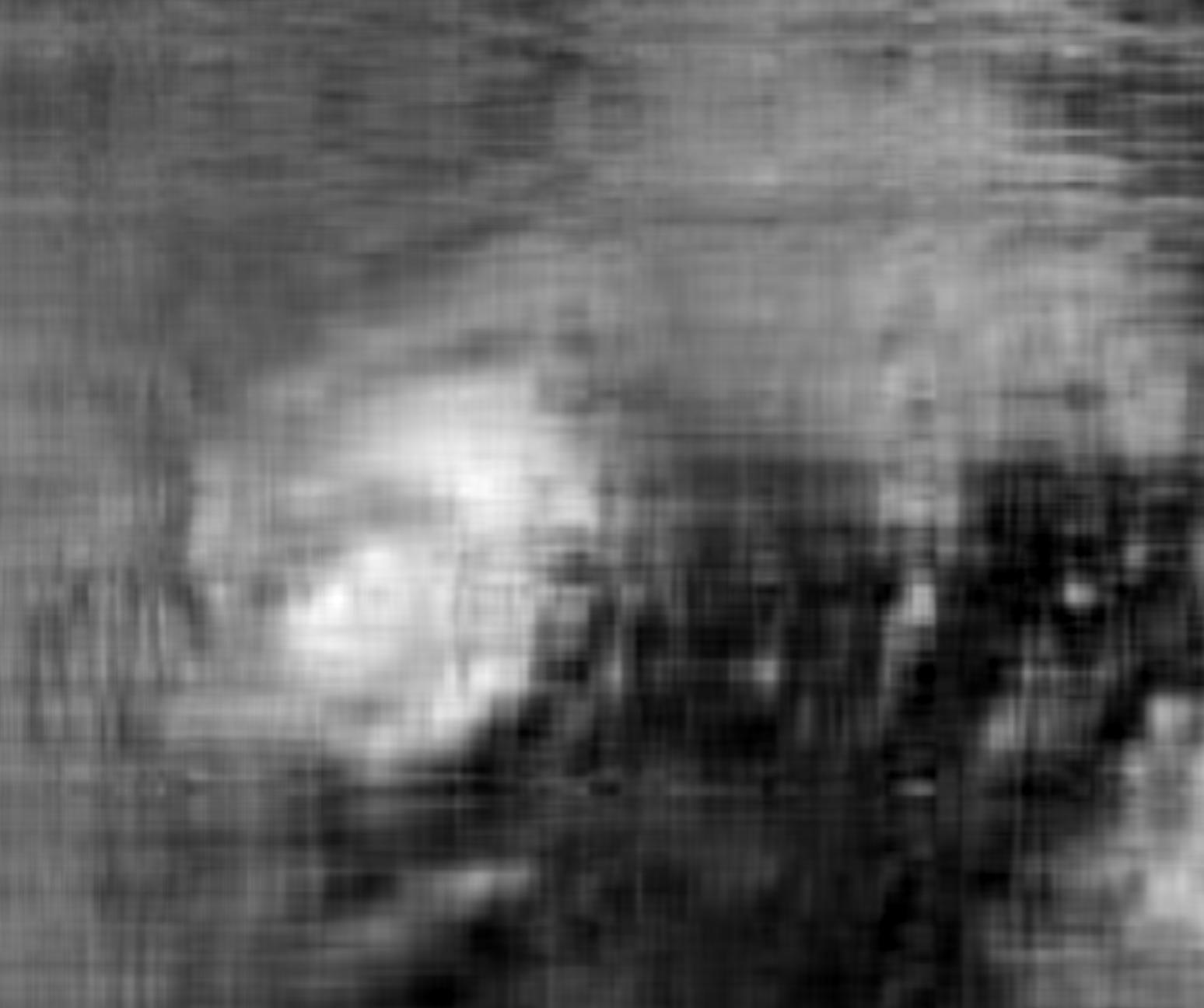
5. Programming

1. After performing SVD on this matrix and zero out all but top k(2,10,40)  singular values to form an approximation. Result images are shown as below:

* k = 2



* k = 10



* K = 40



result = [0.28258451691929531, 0.15928758406875393, 0.084079335040156966]

when k = 2, result = 0.28258451691929531

when k = 10, result = 0.15928758406875393

when k = 40, result = 0. 084079335040156966

1. How many numbers do you need to describe the approximation

Result = [5690, 28450, 113800]

when k = 2, result = 5690

when k = 10, result = 28450

when k = 40, result = 113800

My code is shown as below:

*import numpy as np*

*from PIL import Image*

*################################# Part A ######################################*

*# Load Image*

*image = Image.open('harvey-saturday-goes7am.jpg')*

*grey = image.convert("L")*

*# Image to array*

*X = np.asarray(grey)*

*# k value*

*k\_list = [2,10,40]*

*result\_list = []*

*partb\_list = []*

*for k in k\_list:*

*# SVD*

*U,s,Vt = np.linalg.svd(X, full\_matrices = False)*

*s[k:] = 0*

*S = np.diag(s)*

*X\_app = np.dot(np.dot(U,S), Vt)*

*# Show and save approximate\_image*

*img = Image.fromarray(X\_app)*

*if(img.mode != 'RGB'):*

*img = img.convert('RGB')*

*img.save(str(k)+'.jpg')*

*# calculate ||X-X\_app||f / ||X||f*

*temp = X - X\_app*

*result = np.linalg.norm(temp,'fro') / np.linalg.norm(X,'fro')*

*result\_list.append(result)*

*################################# Part B ######################################*

*# How many numbers do you need to describe the approximation*

*left = U[:,:k]*

*left\_row = left.shape[0]*

*left\_column = left.shape[1]*

*num\_left = left\_row \* left\_column*

*right = Vt[:k,:]*

*right\_row = right.shape[0]*

*right\_column = right.shape[1]*

*num\_right = right\_row \* right\_column*

*partb = num\_left + num\_right + k*

*partb\_list.append(partb)*