CENG 371 SCIENTIFIC COMPUTING

HW4

Name: Andaç Berkay Seval

ID: 2235521

Q1) Function is implemented in approximate_svd.m file.

Q2) Rank of the matrix representing cameraman.jpg is 256, and rank of the matrix representing fingerprint.jpg is 1536. Thus, k = 1:256 and k = 1:1536 for cameraman.jpg and fingerprint.jpg respectively.

a) For cameraman.jpg:

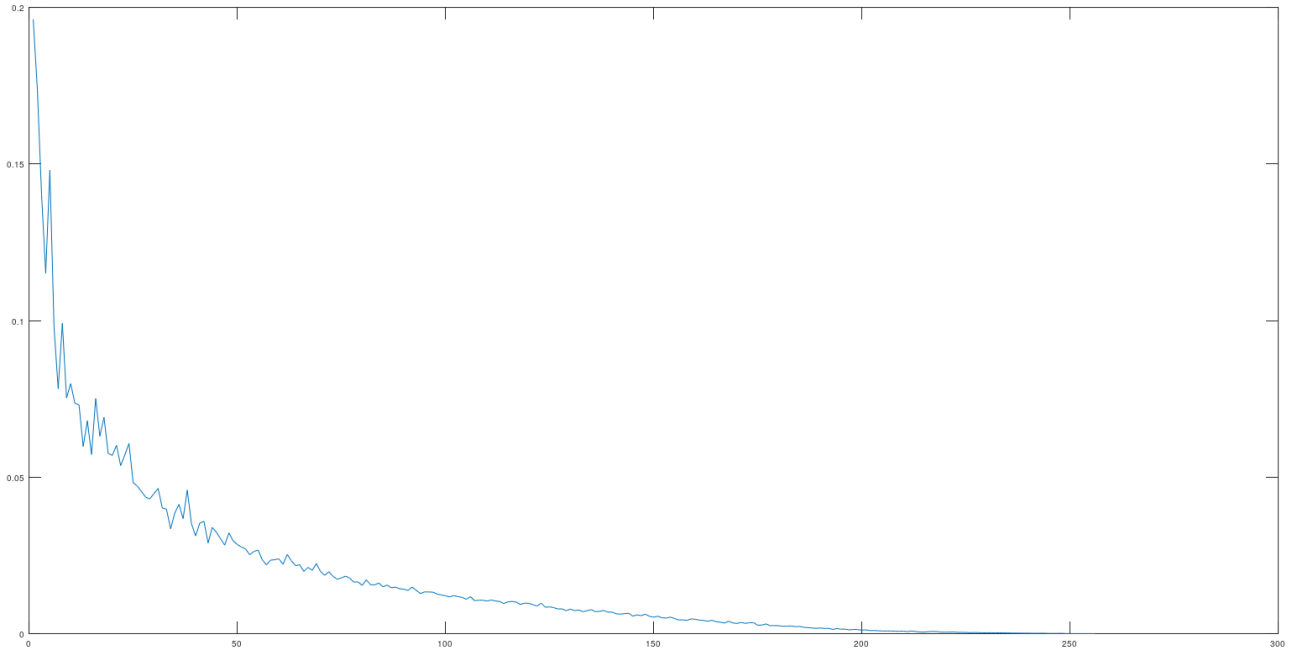

Figure 1: Relative error vs k with the matrices returned by approximate_svd (u_k, σ_k, v_k)

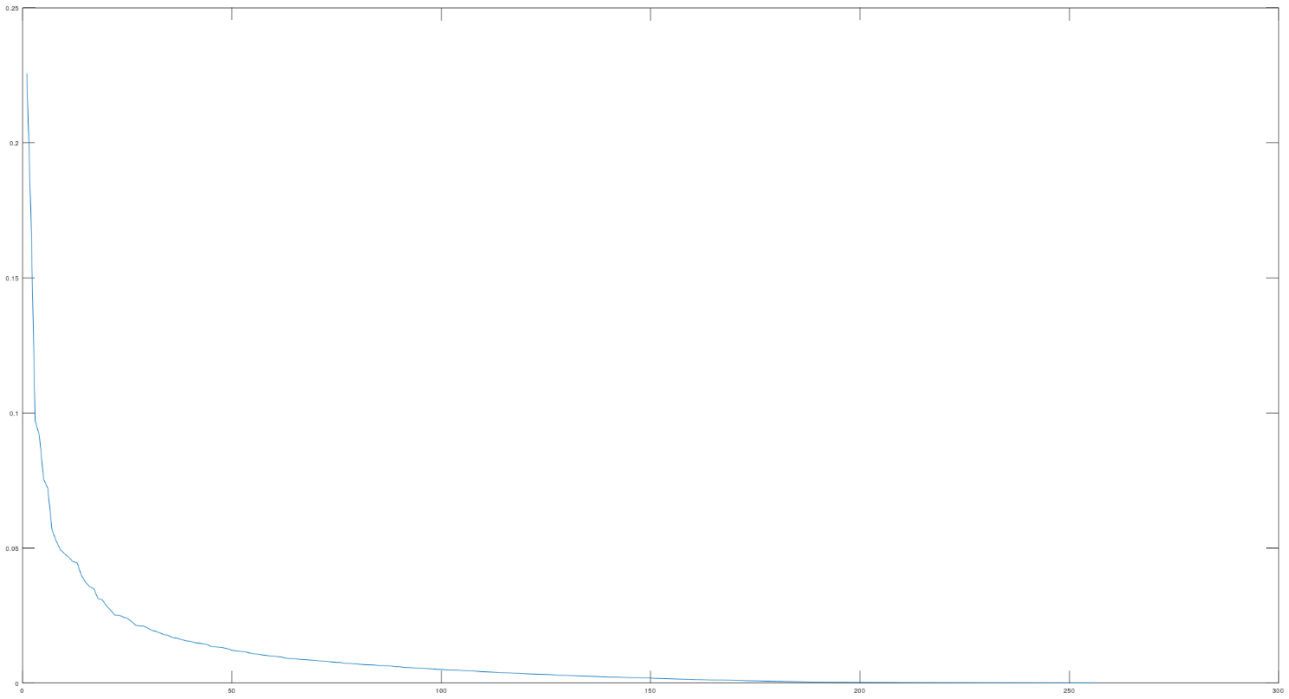

Figure 2: Relative error vs k with the matrices returned by svds (u'_k, σ'_k, v'_k)

Both graphs are similar. While k value is increasing, relative error converges to 0. However, second graph with the matrices returned by svds (u'_k, σ'_k, v'_k) converges smoother and faster compared to first graph. Therefore, first graph with the matrices returned by approximate_svd (u_k, σ_k, v_k) has more zigzags in smaller k values (up to 100) and has more relative error for the same k values compared to the second graph.

For fingerprint.jpg:

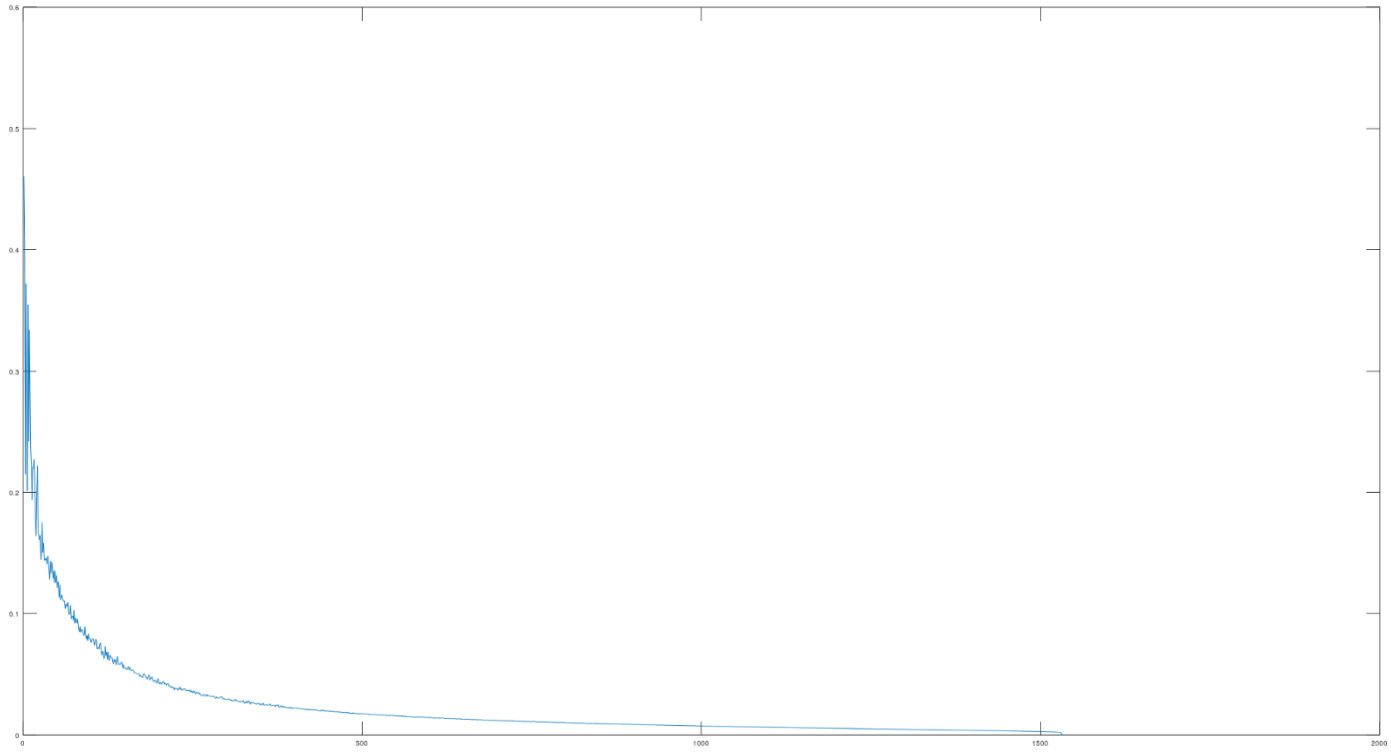

Figure 3: Relative error vs k with the matrices returned by approximate_svd (u_k, σ_k, v_k)

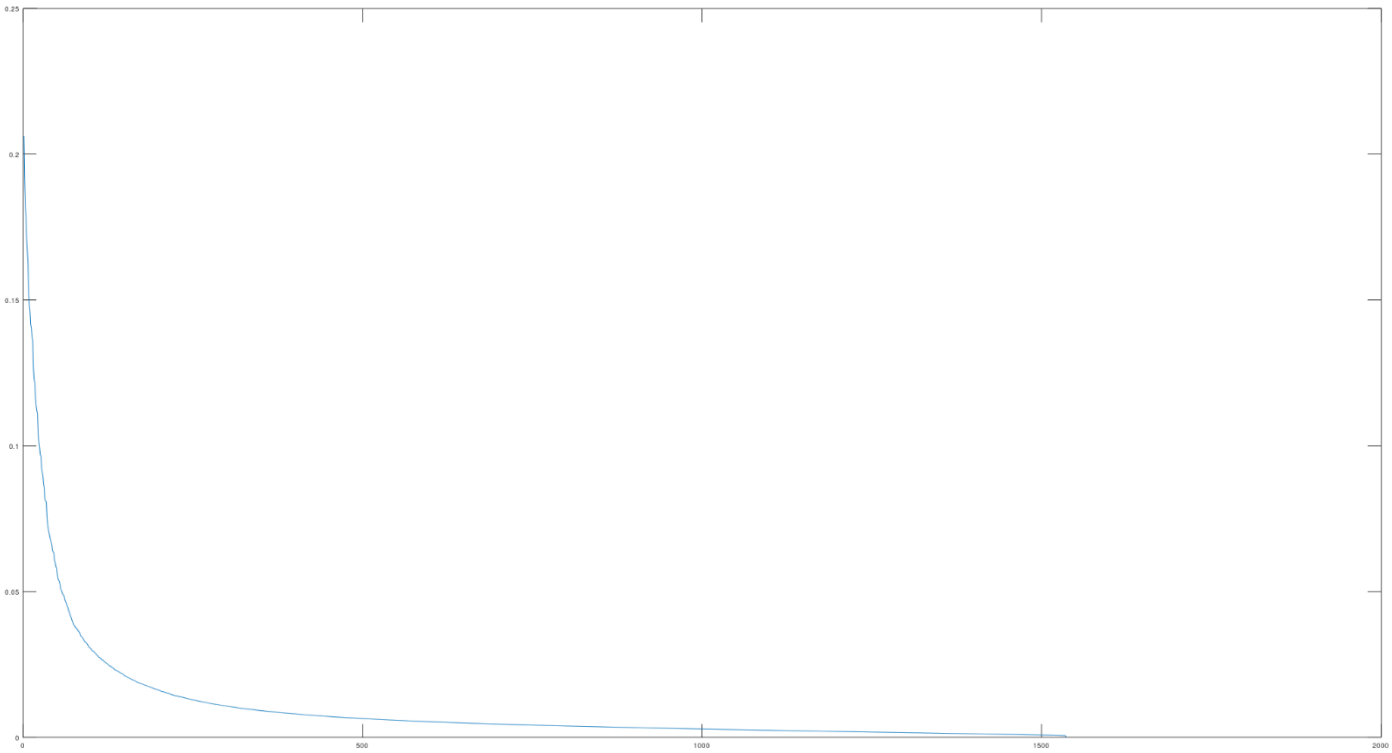

Figure 4: Figure 2: Relative error vs k with the matrices returned by svds (u'_k, σ'_k, v'_k)

Both graphs are similar. While k value is increasing, relative error converges to 0. However, second graph with the matrices returned by svds (u'_k, σ'_k, v'_k) converges smoother and faster compared to first graph. Therefore, first graph with the matrices returned by approximate_svd (u_k, σ_k, v_k) has more zigzags in smaller k values (up to 500) and has more relative error for the same k values compared to the second graph.

b) For cameraman.jpg:

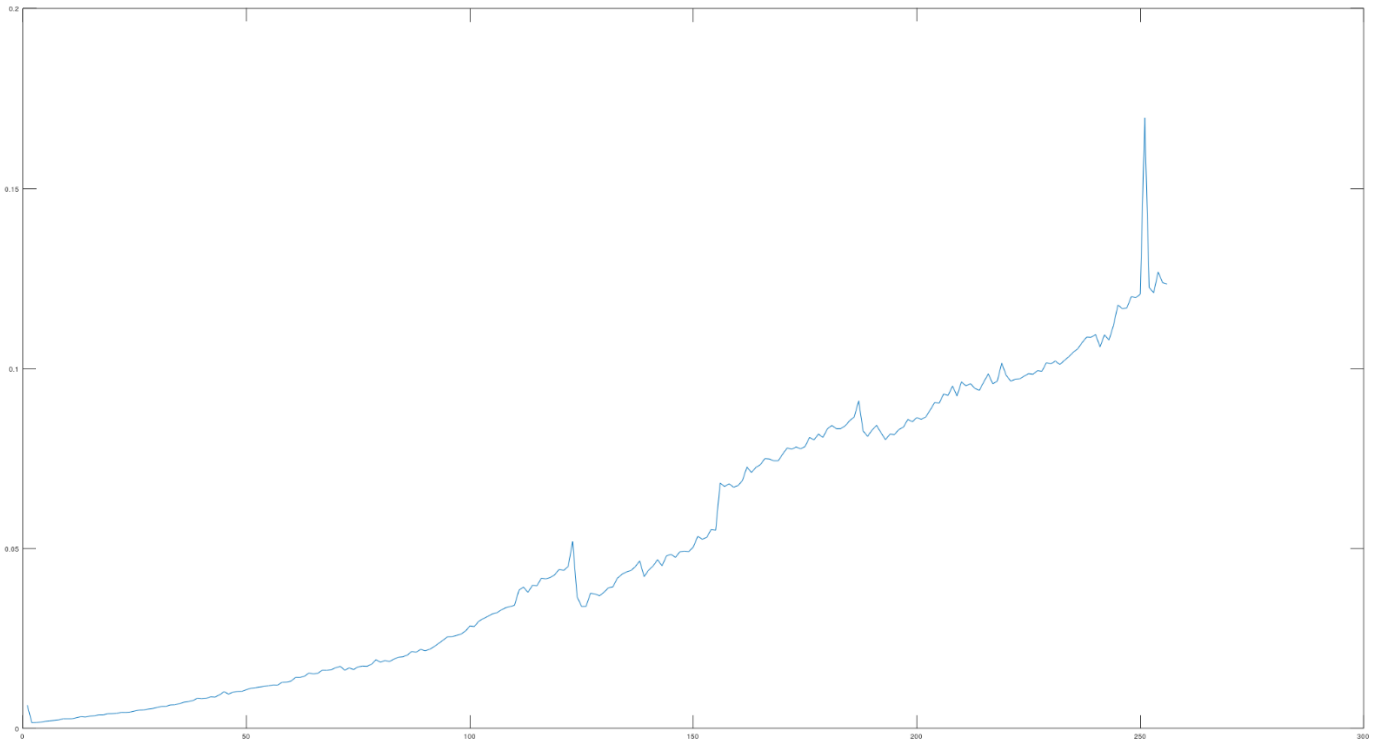

Figure 5: Run time of approximate_svd vs k

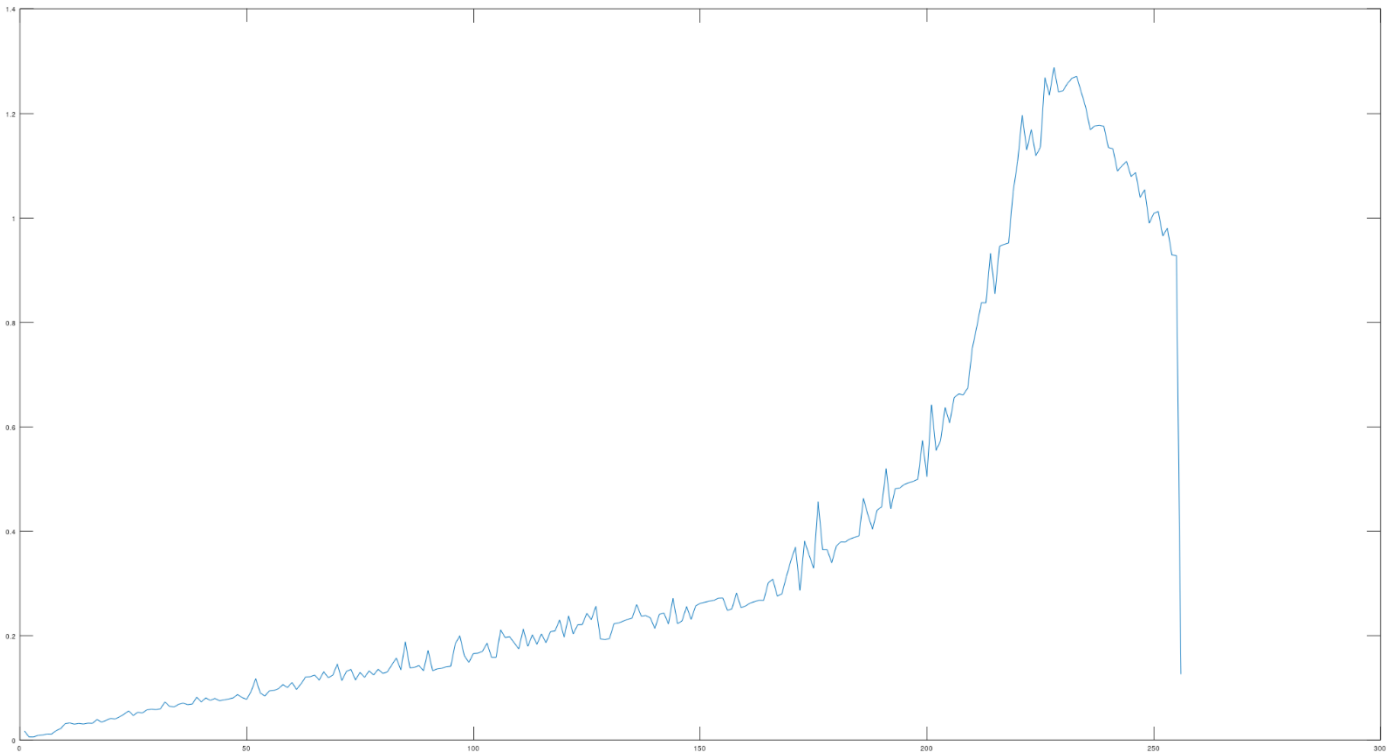

Figure 6: Run time of svds vs k

Run time of the approximate_svd is shorter for almost all k values compared to the run time of svds. In both graphs, there are some zigzags (while k is increasing, run time increases but then drops). Nevertheless, generally, run time increases while k is increasing in both graphs. At some point in the second graph, after k value is nearly 250, there is a huge drop in the run time of svds.

For fingerprint.jpg:

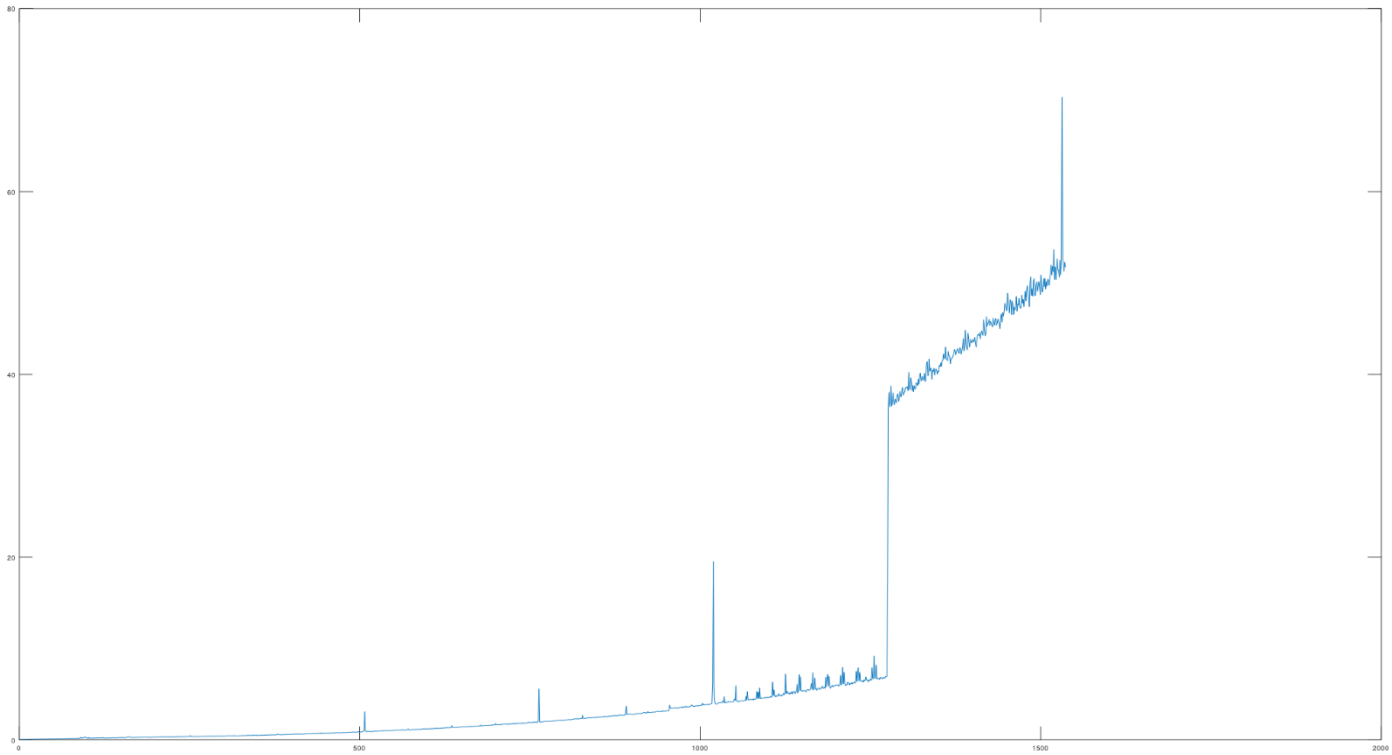

Figure 7: Run time of approximate_svd vs k

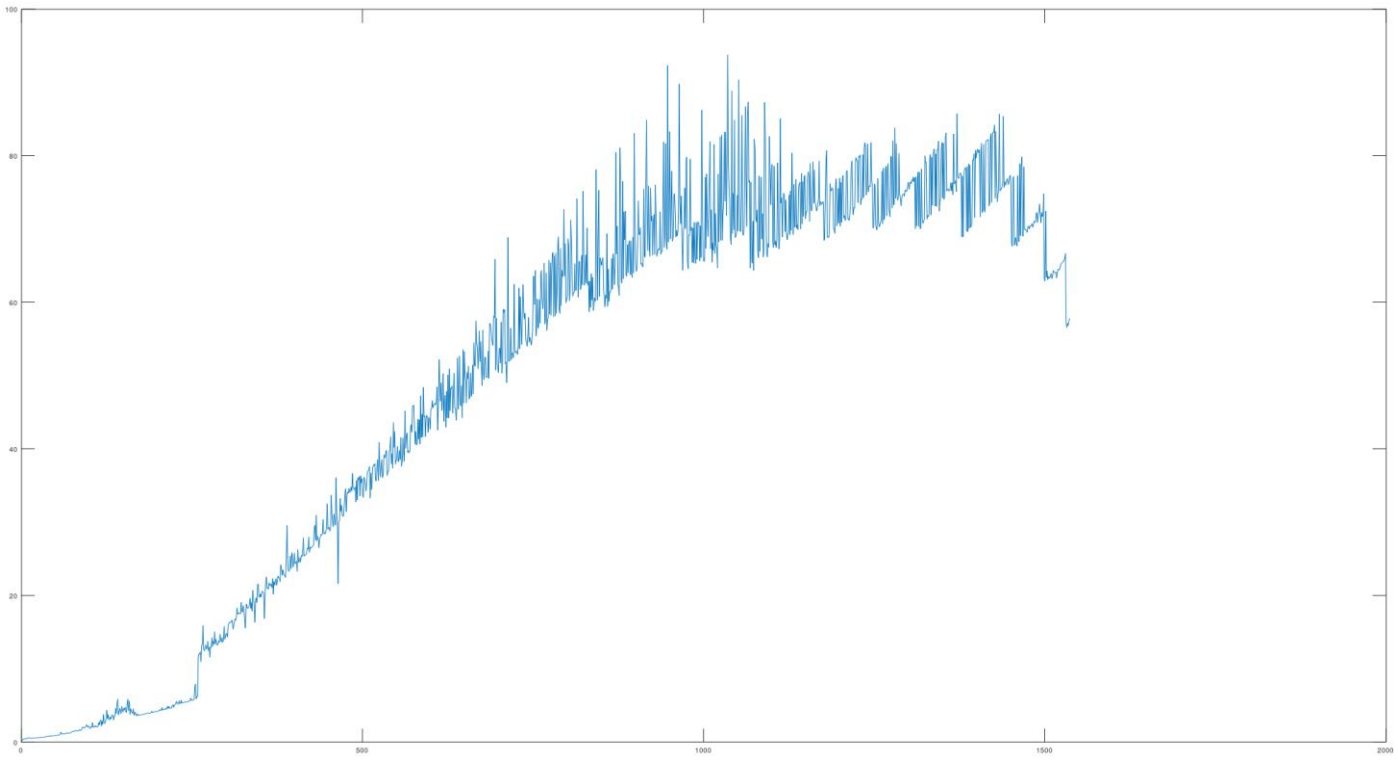

Figure 8: Run time of svds vs k

Run time of the approximate_svd is shorter for almost all k values compared to the run time of svds. In both graphs, there are some zigzags (while k is increasing, run time increases but then drops). Nevertheless, generally, run time increases while k is increasing in both graphs. At some point in the second graph, after k value is nearly 1500, there is a drop in the run time of svds.

c) For cameraman.jpg:

Figure 9: Image resulting from low-rank approximation by svd ($U \Sigma V^T$)

Figure 10: Image resulting from low-rank approximation by approximate_svd with k = 10

Figure 11: Image resulting from low-rank approximation by svds with k = 10

Figure 12: Image resulting from low-rank approximation by approximate_svd with k = 50

Figure 13: Image resulting from low-rank approximation by svds with k = 50

Figure 14: Image resulting from low-rank approximation by approximate_svd with k = 150

Figure 15: Image resulting from low-rank approximation by svds with k = 150

Figure 16: Image resulting from low-rank approximation by approximate_svd with k = 200

Figure 17: Image resulting from low-rank approximation by svds with k = 200

When we compare images resulting from different low-rank approximation functions, we can see that while k value is increasing, in other words, number of used singular values is increasing, image quality gets better and closer to the original image. Image resulting from svd function ($U \Sigma V^T$) is same as the original image. I chose k values 10, 50, 150 and 200 for approximate_svd and svds functions. If we look at the functions; for k = 10: both images are far from the original image since only 10 singular values are used while image construction, however, since the image constructed by approximate_svd ($u_{10}\ \sigma_{10}\ v_{10}^T$) has more relative error than image constructed by svds ($u'_{10}\ \sigma'_{10}\ v'_{10}^T$), $u'_{10}\ \sigma'_{10}\ v'_{10}^T$ looks slightly better than $u_{10}\ \sigma_{10}\ v_{10}^T$, for k = 50: both images are closer to the original image compared to when k is 10, however, since the image constructed by approximate_svd ($u_{50}\ \sigma_{50}\ v_{50}^T$) has more relative error than image constructed by svds ($u'_{50}\ \sigma'_{50}\ v'_{50}^T$), $u'_{50}\ \sigma'_{50}\ v'_{50}^T$ looks slightly better than $u_{50}\ \sigma_{50}\ v_{50}^T$, for k = 150: both images are closer to the original image compared to k is 50, however, this time, I could not see much difference between images constructed by approximate_svd ($u_{150}\ \sigma_{150}\ v_{150}^T$) and svds ($u'_{150}\ \sigma'_{150}\ v'_{150}^T$) since relative errors are very small for both of the functions, for k = 200: both images are nearly the same as the original image, it is really hard to identify the differences between $u_{200}\ \sigma_{200}\ v_{200}^T$, $u'_{200}\ \sigma'_{200}\ v'_{200}^T$ and $U \Sigma V^T$ for human eyes.

For fingerprint.jpg:

Figure 18: Image resulting from low-rank approximation by svd ($U \Sigma V^T$)

Figure 19: Image resulting from low-rank approximation by approximate_svd with k = 100

Figure 20: Image resulting from low-rank approximation by svds with k = 100

Figure 21: Image resulting from low-rank approximation by approximate_svd with k = 250

Figure 22: Image resulting from low-rank approximation by svds with k = 250

Figure 23: Image resulting from low-rank approximation by approximate_svd with k = 500

Figure 24: Image resulting from low-rank approximation by svds with k = 500

Figure 25: Image resulting from low-rank approximation by approximate_svd with k = 1000

Figure 26: Image resulting from low-rank approximation by svds with k = 1000

When we compare images resulting from different low-rank approximation functions, we can see that while k value is increasing, in other words, number of used singular values is increasing, image quality gets better and closer to the original image. Image resulting from svd function ($U \Sigma V^T$) is same as the original image. I chose k values 100, 250, 500 and 1000 for approximate_svd and svds functions. If we look at the functions; for k = 100: both images are far from the original image since only 100 singular values are used while image construction, however, since the image constructed by approximate_svd ($u_{100}\ \sigma_{100}\ v_{100}^T$) has more relative error than image constructed by svds ($u'_{100}\ \sigma'_{100}\ v'_{100}^T$), $u'_{100}\ \sigma'_{100}\ v'_{100}^T$ looks slightly better than $u_{100}\ \sigma_{100}\ v_{100}^T$, for k = 250: both images are closer to the original image compared to when k is 100, however, since the image constructed by approximate_svd ($u_{250}\ \sigma_{250}\ v_{250}^T$) has more relative error than image constructed by svds ($u'_{250}\ \sigma'_{250}\ v'_{250}^T$), $u'_{250}\ \sigma'_{250}\ v'_{250}^T$ looks clearly better than $u_{250}\ \sigma_{250}\ v_{250}^T$ in this one, for k = 500: both images are closer to the original image compared to k is 250, however, since the image constructed by approximate_svd ($u_{500}\ \sigma_{500}\ v_{500}^T$) has more relative error than image constructed by svds ($u'_{500}\ \sigma'_{500}\ v'_{500}^T$), $u'_{500}\ \sigma'_{500}\ v'_{500}^T$ looks still better than $u_{500}\ \sigma_{500}\ v_{500}^T$ and almost same as the original image, for k = 1000: both images are nearly the same as the original image, it is really hard to identify the differences between $u_{1000}\ \sigma_{1000}\ v_{1000}^T$, $u'_{1000}\ \sigma'_{1000}\ v'_{1000}^T$ and $U \Sigma V^T$ for human eyes.

d) Based on my discussions, approximate_svd could be used for image compression applications. With using approximate_svd, we could obtain digital image by $u_k\ \sigma_k\ v_k^T$. Moreover, depending on the k value, in other words, number of singular vales that are using for reconstructing image, one could get an image that preserves key features of the original image but has a less storage space in the memory. That acquired image might not look exactly like original image, however it could have useful and important features of it. Therefore, one could obtain an image with using approximate_svd that has the required features of the original image by adjusting the number of singular values and requires less memory storage. Hence, run time of approximate_svd is less than svds and relative errors are comparable for a reasonable k value, approximate_svd is preferable against svds.

References:

[1] Cao, Lijie. “Singular Value Decomposition Applied To Digital Image Processing”. Division of Computing Studies, Arizona State University Polytechnic Campus, Mesa Arizona 85212.