CENG 371 - Scientific Computing

Fall 2022

Homework 4

Anıl Utku Ilgın
2260073

January 23, 2023

Question 2

a) Plot of relative errors of approximate SVD and SVDS for cameraman and fingerprint images are in figure (1) and figure (2).

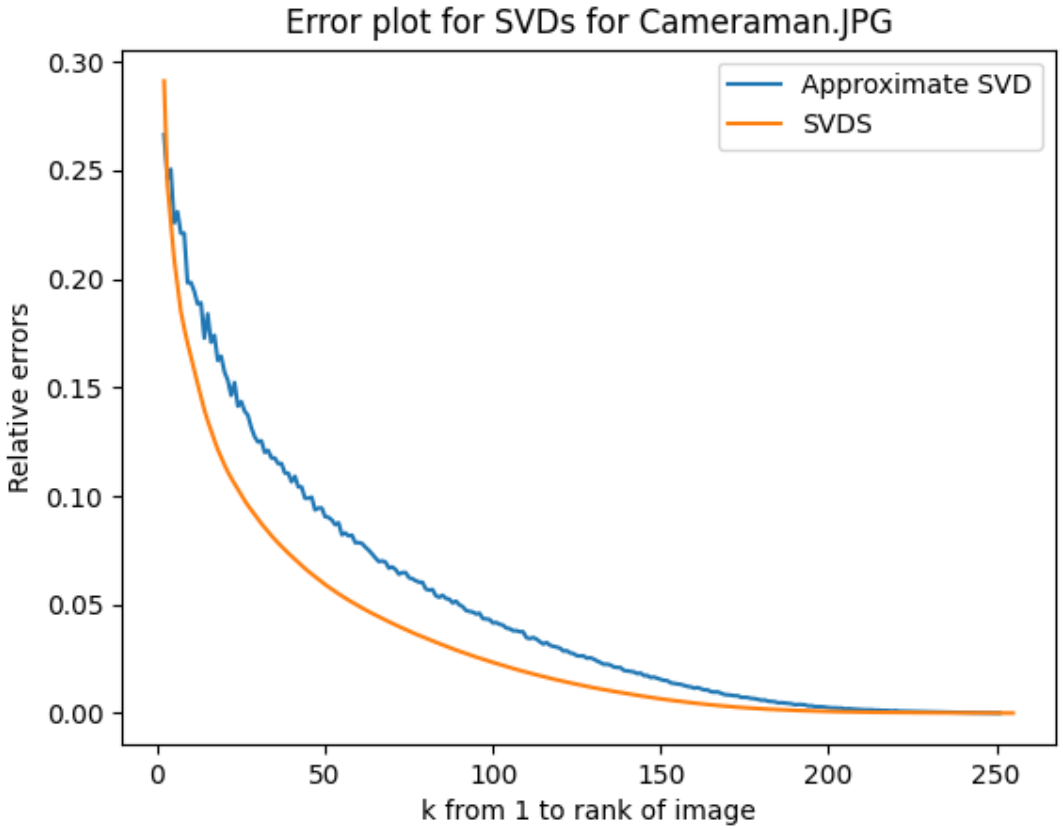


Figure 1 : Errors of approximate SVD vs SVDS algorithm for cameraman.jpg

Figure 2 : Errors of approximate SVD vs SVDS algorithm for fingerprint.jpg

In both images, it's seen that SVDS results in a lower error rate, since Approximate SVD also introduces noise into the system because of the gaussian matrix, and most importantly we don't get to chose the biggest singular values like SVDS does, instead we choose randomly therefore approximate SVD has a higher error rate in both cases. There is also a jittering in the approximate SVD errors, the randomness of the gaussian matrix is the reason for that.

b) Plot of run-times of approximate SVD and SVDS for cameraman and fingerprint images are in figure (3) and figure (4).

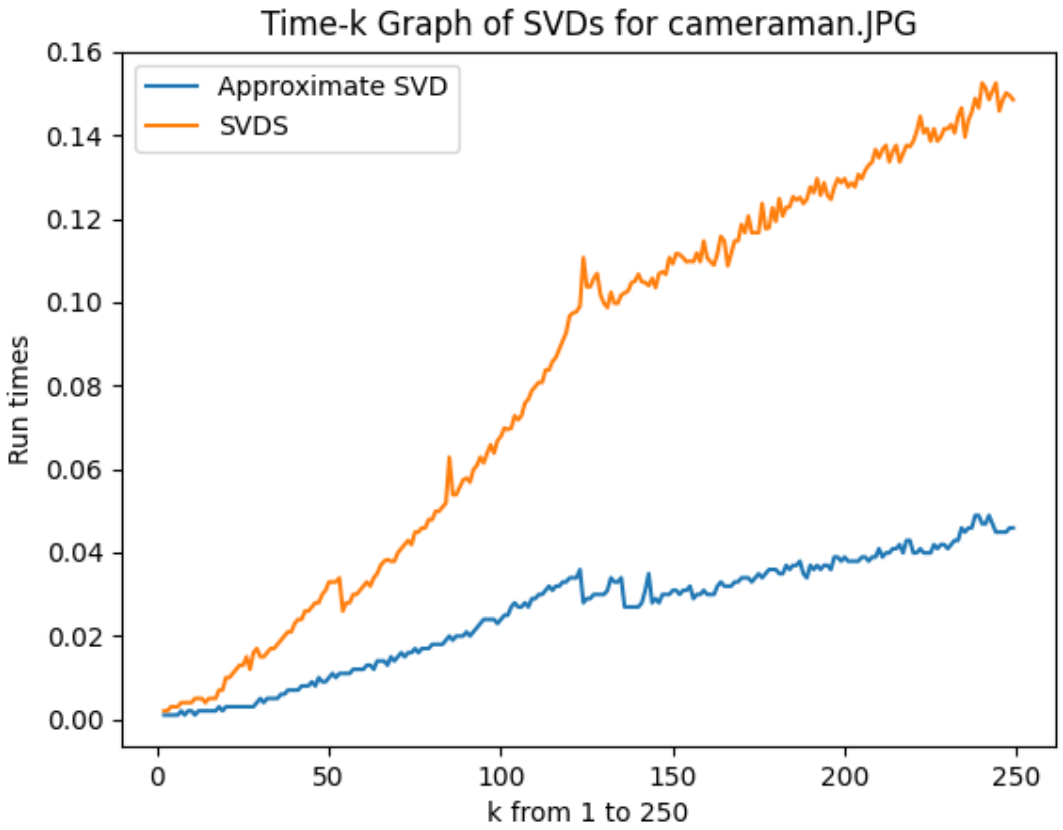


Figure 3 : k-time graph of SVDs for cameraman image.

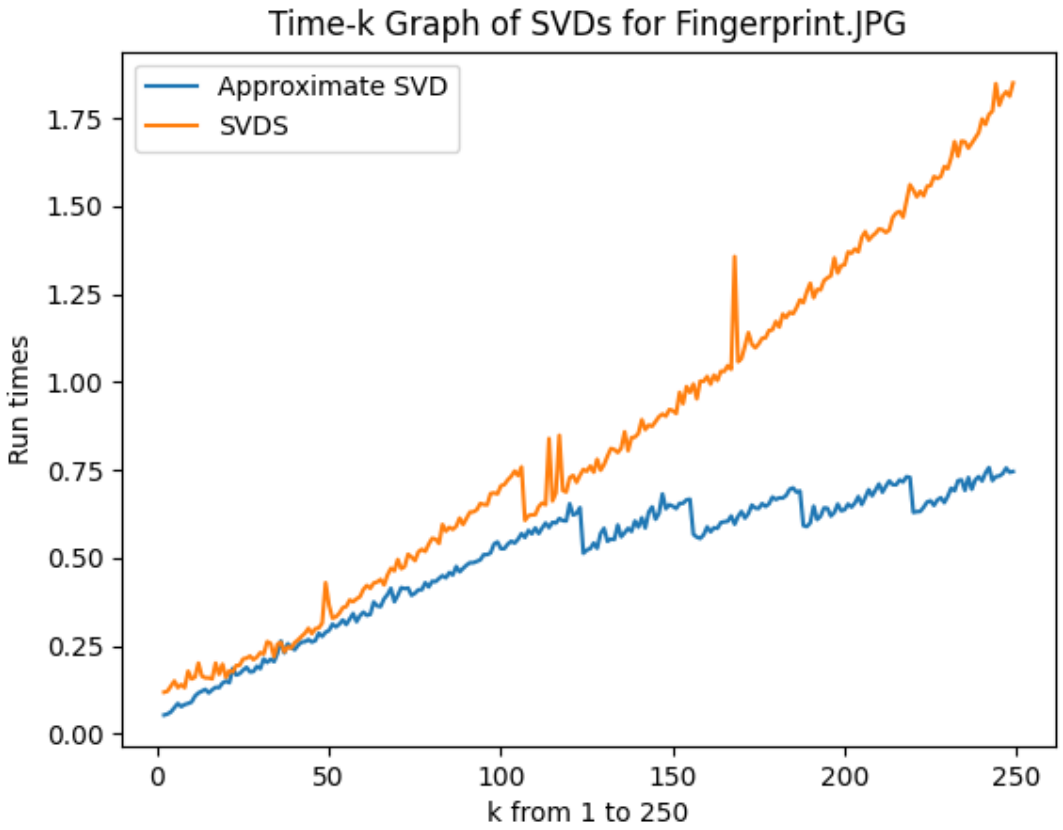


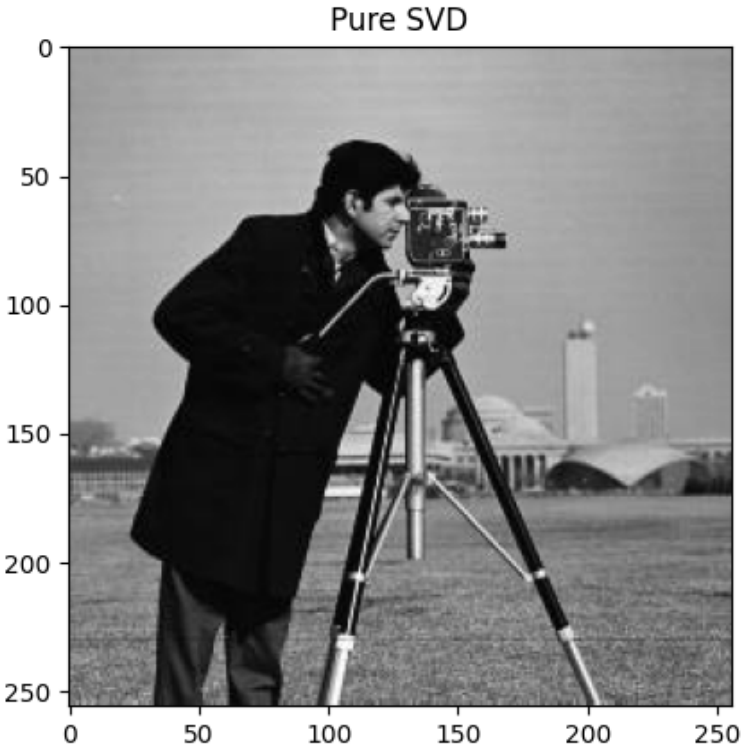
Figure 4: Figure 3 : k-time graph of SVDs for fingerprint image.

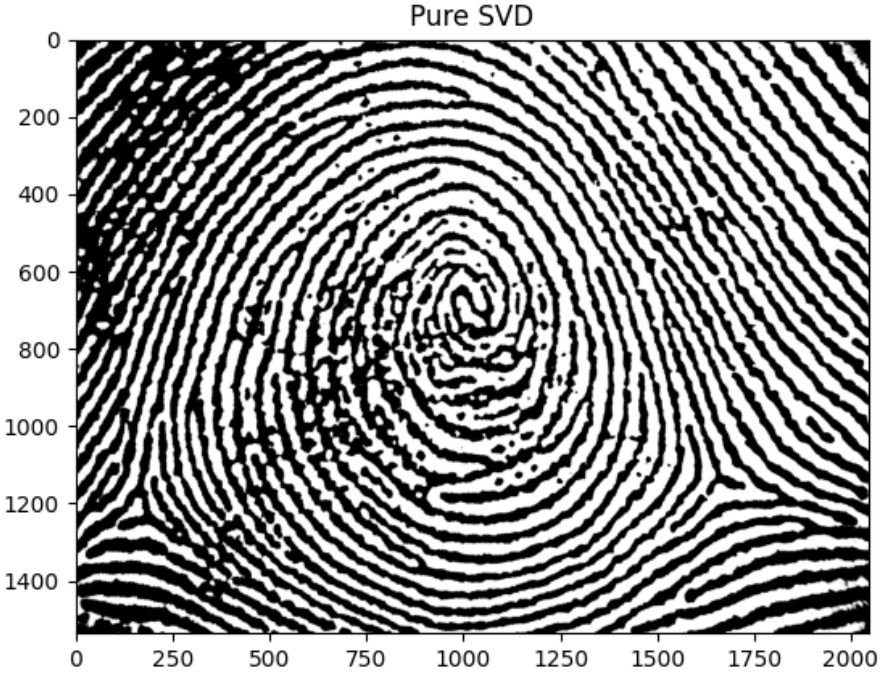
As seen in both figure 3 and 4, approximate SVD yields a better performance in terms of speed than the SVDS algorithm as k increases. Reason for this is that SVDS is computing the singular values in from in importance order (bigger to smaller singular values) and this slows down the progress as k increases, resulting in an exponential like growth. Approximate SVD is more close to linear and bounded around a limit as seen in the figures. Hence, approximate SVD is significantly faster than the SVDS, especially in higher k rates.

c) Qualitative Comparison between SVD, Approximate SVD and SVDS

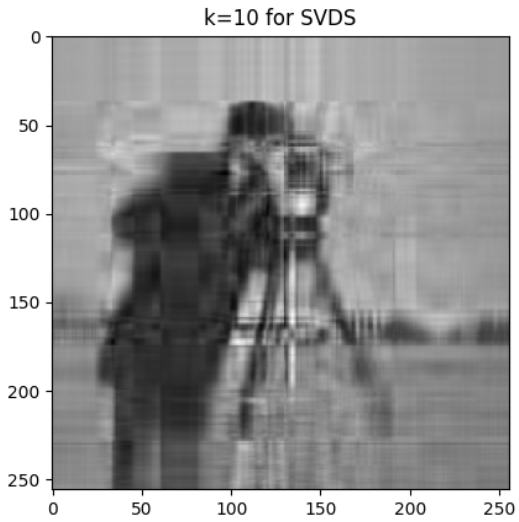
Cameraman Images

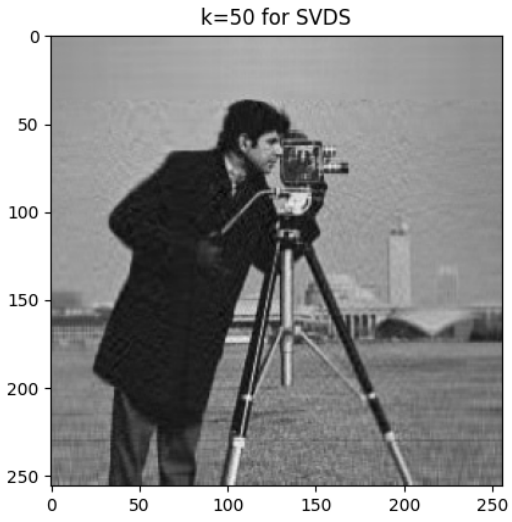
Pure SVD:

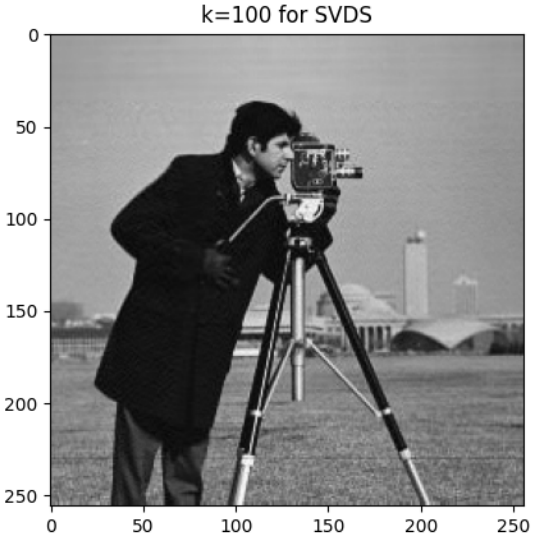


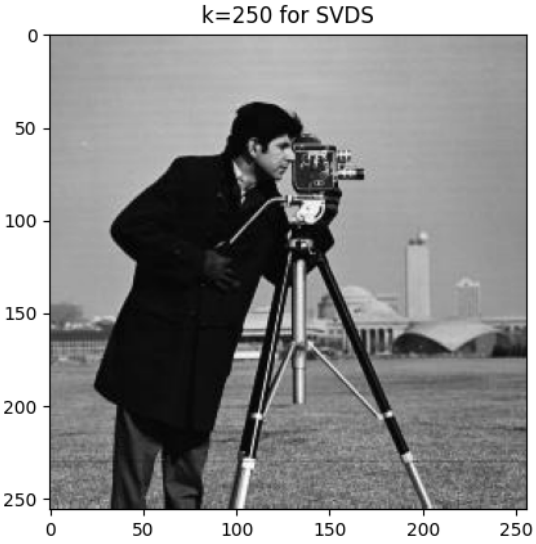


SVDS

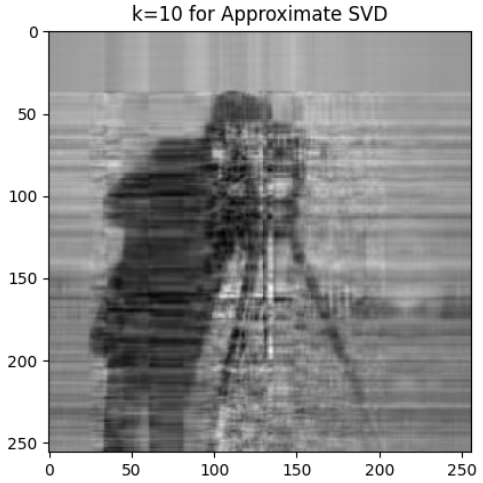

k=10 for SVDS
0
50
100
150
200
250
0
50
100
150
200
250

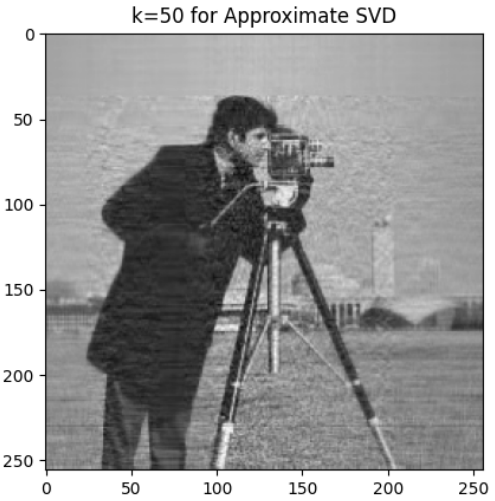

k=50 for SVDS
0
50
100
150
200
250
0
50
100
150
200
250

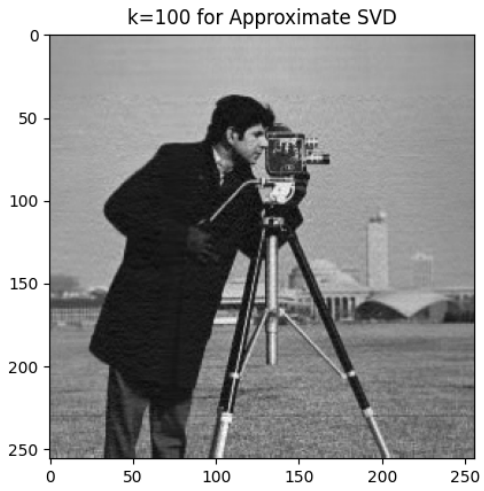

k=100 for SVDS
0
50
100
150
200
250
0
50
100
150
200
250

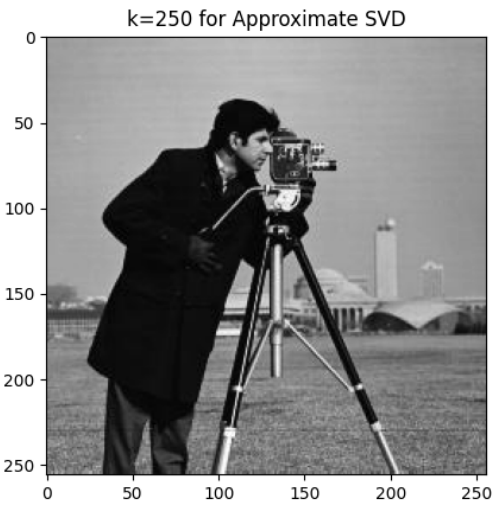

k=250 for SVDS
0
50
100
150
200
250
0
50
100
150
200
250

Approximate SVD

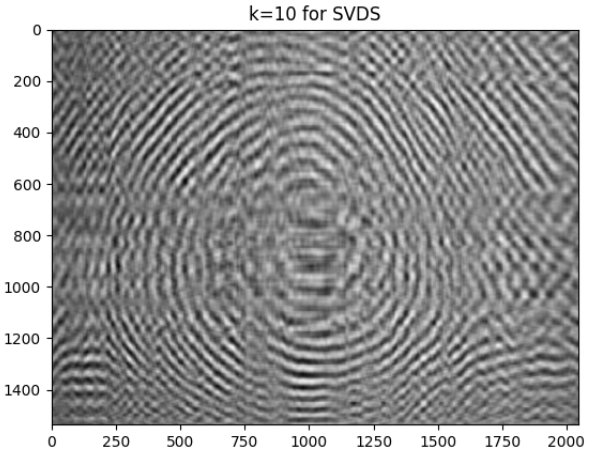


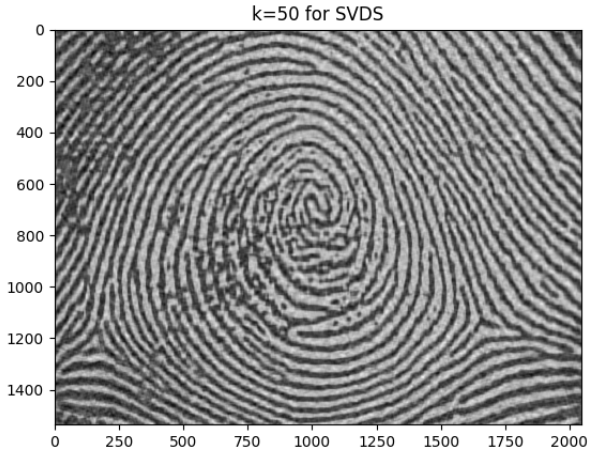


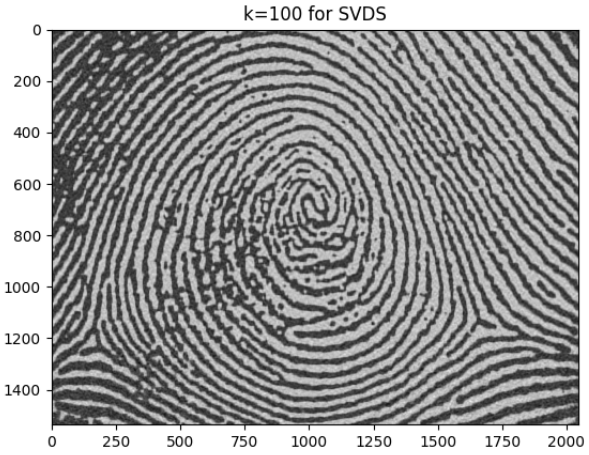


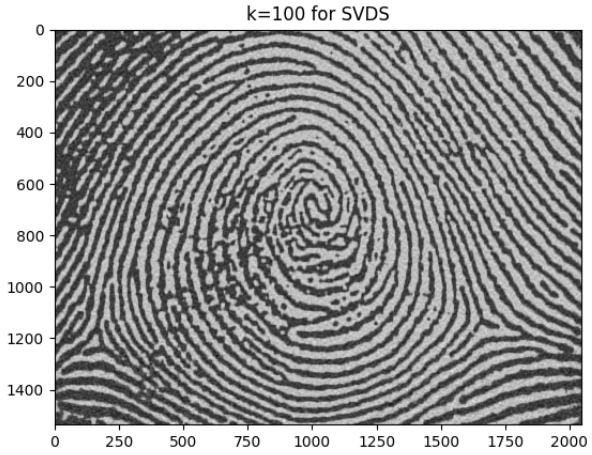


SVDS

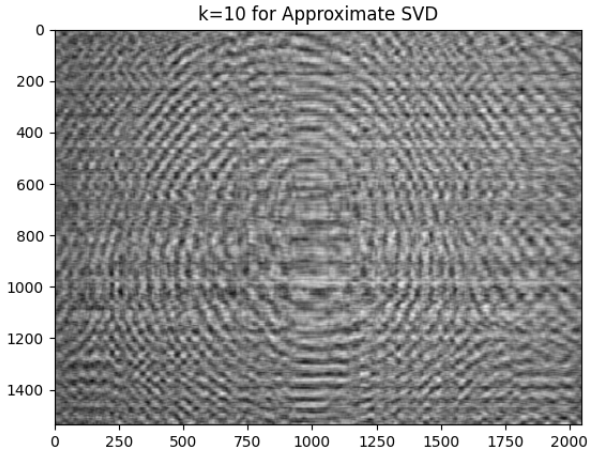

k=10 for SVDS
0
200
400
600
800
1000
1200
1400
0
250
500
750
1000
1250
1500
1750
2000

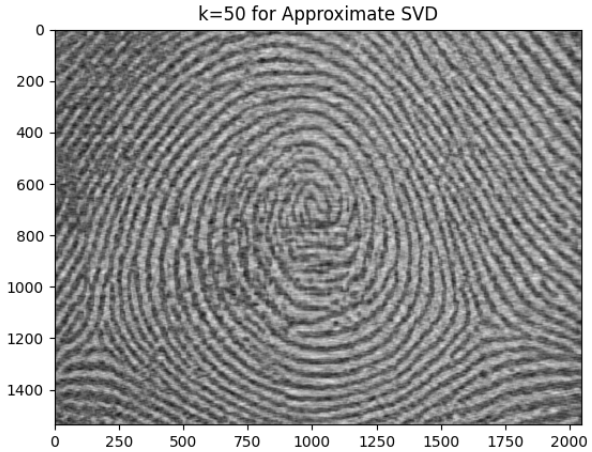

k=50 for SVDS
0
200
400
600
800
1000
1200
1400
0
250
500
750
1000
1250
1500
1750
2000

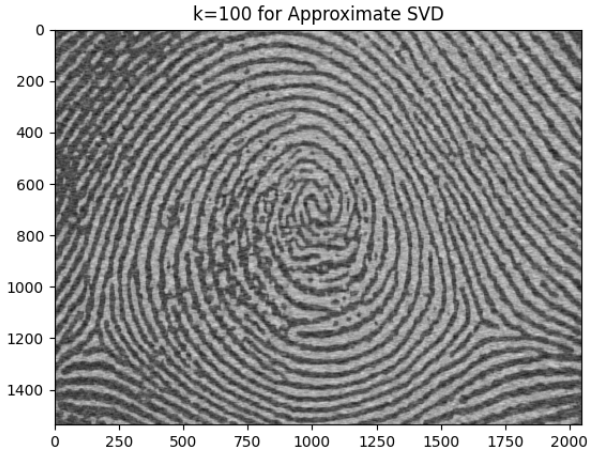

k=100 for SVDS
0
200
400
600
800
1000
1200
1400
0
250
500
750
1000
1250
1500
1750
2000

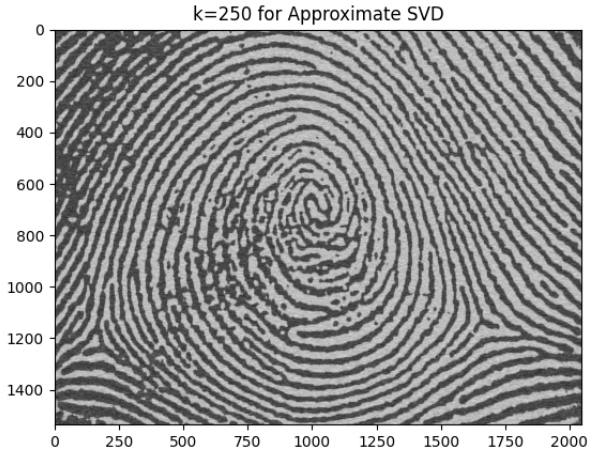

k=100 for SVDS
0
200
400
600
800
1000
1200
1400
0
250
500
750
1000
1250
1500
1750
2000

Approximate SVD









c) After k=100, I believe there is little to none difference between algorithm outputs in terms of quality and can compete with pure SVD algorithm as well.

However, as we lower the k value, even at the k=100 we can see there starts a quality drop in both SVDS and Approx. SVD algorithm.
In the approximate SVD algorithm a noticeable noise appears in images, and it's getting worse as k goes down, but it's negligible at k=100 and 250 for cameraman images, SVDS and Approx SVD are similar in terms of quality for that.

Yet, SVDS produces a much better quality image compared to approximate SVD, and we can deduce this idea by looking at fingerprint images. Approximate SVD algorithm's fingerprint images are much more cloudy and noiseful compared to SVDS even at k=100 and k=250 values.

For the singular values, The ones in SVDS are much more significant and contain more information than the Approx SVD at the low ranks, since we choose randomly in approx and choose in order for SVDS. This difference shows itself in the quality.

d) Use cases for approximate SVD

Approx. SVD's powerhouse is its speed. As seen in figure 3 and 4, approximate SVD is significantly faster than the SVDS values, and for k values > 100 we see that quality also gets better. Therefore, we can use it in the cases where speed is much more important than the quality. However, as seen in the fingerprints it's susceptible to cloud fine-details. Therefore, I think we can't use it for such cases.

- Therefore, I think we can use Approx SVD for the cases where fine details such as biometrics are not important, but bigger details are enough. Such as downsampling the images for training a CNN for distinct objects, such as different types of cars etc. It will significantly increase the speed without losing that much of accuracy.

- In addition, it can be used in maps as well. buildings and roads are distinct from each other and details can still be intact even after an SVD operation applied. This can speed up the real-time satellite views for viewing the earth.
- Finally, we can use it to display thumbnails of photos in galleries and where speed is important, such as Instagram etc.