MAT 343 Lab 6 - Jooho Kim

NOTE: for this problem you might want to watch the second video in the tutorial videos for this lab. Delete this note upon submission.

```
A=imread('gauss.jpg'); %load the picture
B=double(A(:,:,1));  %convert to double precision
B=B/255;             %scale the values of B
[U S V]=svd(B);      %compute the SVD decomposition of B
```

Problem 1

Compute the dimensions of U, S and V

```
size(U) % 258 * 258
```

```
ans = 1×2
   258   258
```

```
size(S) % 258 * 396
```

```
ans = 1×2
   258   396
```

```
size(V) % 396 * 396
```

```
ans = 1×2
   396   396
```

Problem 2

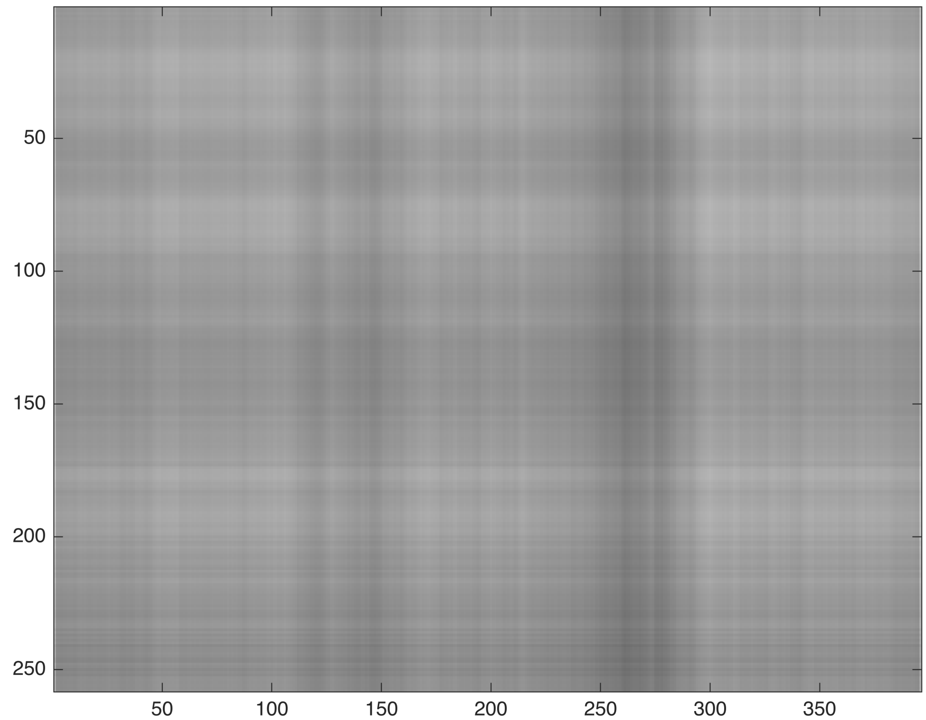
Compute the best rank-1 approximation and store it in `rank1`

```
[U S V] = svd(B);

rank1 = S(1,1)*U(:,1)*V(:,1)';

rank(rank1);

norm(B-rank1, 'fro');
```

Visualize rank1 by performing steps 3 -6

```
C = zeros(size(A));

C(:,:,1) = rank1;
C(:,:,2) = rank1;
C(:,:,3) = rank1;

C = max(0,min(1,C));
```

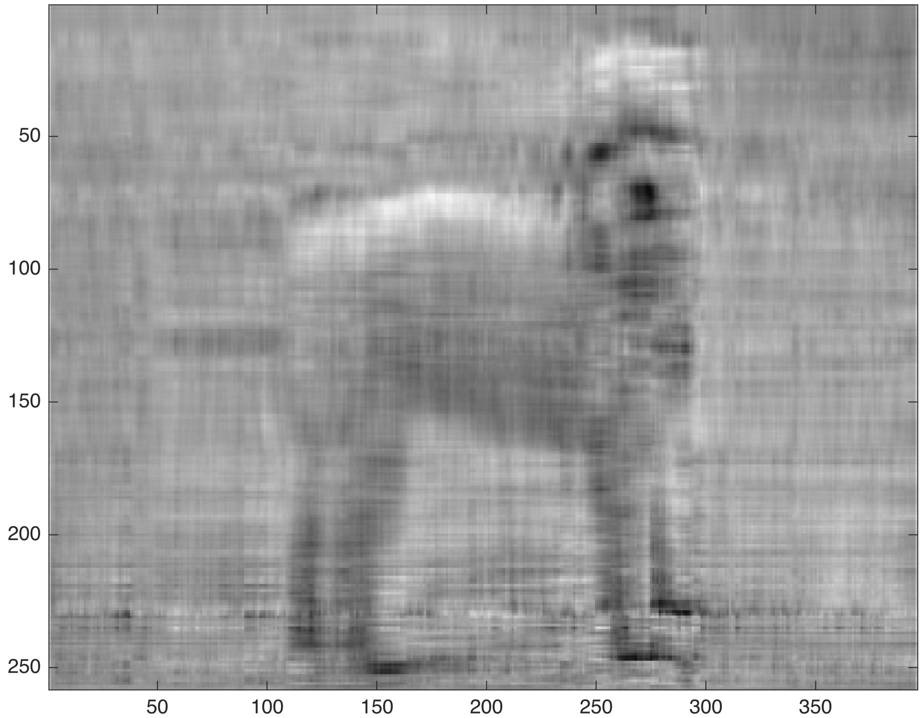
```
image(C)
```



Problem 3

Create and view a rank-10 approximation to the original picture

```
rank10 =  S(1,1)*U(:,1)*V(:,1)';

for i = 2:10
    rank10 = rank10 + S(i,i)*U(:,i)*V(:,i)';
end

rank(rank10);

norm(B - rank10, 'fro');

C = zeros(size(A));

C(:,:,1) = rank10;
C(:,:,2) = rank10;
C(:,:,3) = rank10;

C = max(0,min(1,C));

image(C)
```



Problem 4

Experiment with different ranks until you found one that gives, in your opinion, an acceptable approximation.

```
% rank 20
rank20 =  S(1,1)*U(:,1)*V(:,1)';

for i = 2:20
    rank20 = rank20 + S(i,i)*U(:,i)*V(:,i)';
end

rank(rank20);

norm(B - rank20, 'fro');

C = zeros(size(A));

C(:,:,1) = rank20;
C(:,:,2) = rank20;
C(:,:,3) = rank20;

C = max(0,min(1,C));

image(C)
```


```
% rank 30
rank30 =  S(1,1)*U(:,1)*V(:,1)';

for i = 2:30
    rank30 = rank30 + S(i,i)*U(:,i)*V(:,i)';
end

rank(rank30);

norm(B - rank30, 'fro');

C = zeros(size(A));

C(:,:,1) = rank30;
C(:,:,2) = rank30;
C(:,:,3) = rank30;

C = max(0,min(1,C));

image(C)
```


```
% rank 40
rank40 =  S(1,1)*U(:,1)*V(:,1)';

for i = 2:40
    rank40 = rank40 + S(i,i)*U(:,i)*V(:,i)';
end

rank(rank40);

norm(B - rank40, 'fro');

C = zeros(size(A));

C(:,:,1) = rank40;
C(:,:,2) = rank40;
C(:,:,3) = rank40;

C = max(0,min(1,C));

image(C)
```

I think a rank of 30 is the smallest acceptable approximation to the original picture.

Problem 5

What rank-r approximation exactly reproduces the original picture? Explain,

Answer: **Since A is 258 * 396, the max rank is 258. Therefore, 258 will re-produce the original picture.**

Problem 6

(i)

How much data is needed to represent a rank-k approximation? Explain.

Answer: **N = k(m + n + 1) = k * m + k * n + k. We need N much data to represent a rank-k approximation.**

(ii)

Find the compression rate for the value of the rank you determined in problem 4. Explain.

```
%m = 258; % Rows of A
%n = 396; % Columns of A
%k = 30;  % Rank of the approximation
%original_data_size = m * n;
%approximation_data_size = k * (m + n + 1);
%compression_rate = approximation_data_size / original_data_size * 100;
```

Answer: **This code will generate compression_rate = 19.2330 of the rank = 30.**

What does the compression rate represent? Explain.

Answer: **The compression rate of 19.24% means that the rank-30capproximation uses only 19.24% of the data**

compared to the original matrix. This represents significant data reduction, as 80.76% of the data is eliminated

while retaining a close approximation of the original picture.

Problem 7

Find the smallest value of k such that the rank-k approximation uses the same or more amount of data as the original picture. Explain how you obtained the answer.

Answer: **We want to find the smallest value of k such that the rank-k approximation data size**

equals or exceeds the original data size ($258 \cdot 396 = 102{,}168$). From Problem6, we obtained Rank-k approximation = $k \cdot (m+n+1)$.

Replace m and n with originia data size (258 * 396). Then we obtain $k \cdot (258+396+1) = k \cdot 655 \geq 102{,}168$.

Compute inequality, $k \geq 155.98$. Since k must be an integer (representing rank), round up the the result. Then we get k =156.

Thus, The smallest k = 156.