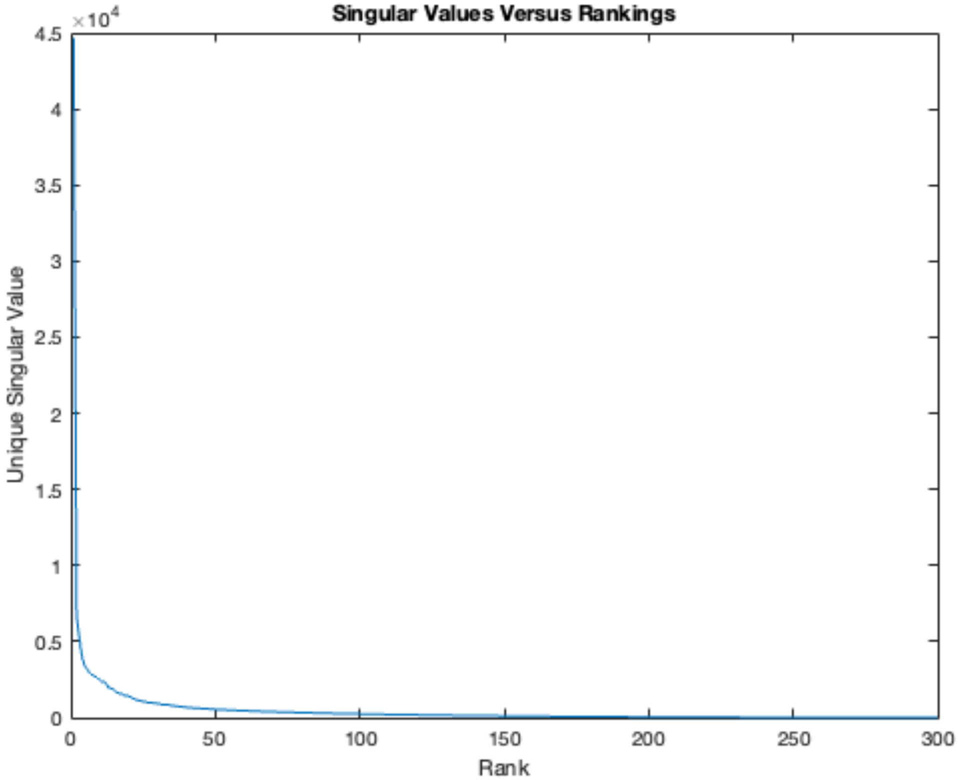
4.
A

```
% Reading the flower.bmp image and converting it into double format and grayscale.
Image = imread('/Users/bharath/Documents/MATLAB/flower.bmp');
Image = double(im2gray(Image));

% Perform SVD on the flower.bmp image
[U, S, V] = svd(Image);

% Extract the top 10 singular values from the flower.bmp image
top_10_singular_values = diag(S(1:10,1:10));

% Plot each unique singular value against its rating.
figure;
plot(1:length(diag(S)), diag(S));
xlabel('Rank');
ylabel('Unique Singular Value');
title('Singular Values Versus Rankings');

% Print the top 10 singular values
fprintf('Top 10 singular values: \n');
disp(top_10_singular_values);
```

```
Top 10 singular values:
   1.0e+04 *

    4.4672
    0.6623
    0.5166
    0.3875
    0.3328
    0.3069
    0.2853
    0.2773
    0.2651
    0.2455
```



Published with MATLAB® R2022a

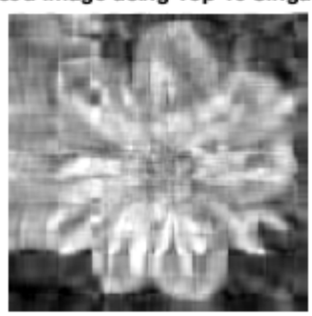
The graphs show the amount of energy captured by the initial r modes, indicating that a significant number of images can be depicted using only a limited number of predominant designs. These dominant features can be applied for tasks such as compression of images.

4.
B

```matlab
% Read the flower.bmp image and convert it to grayscale and double format
Image = imread('/Users/bharath/Documents/MATLAB/flower.bmp');
Image = double(im2gray(Image));

% Perform SVD on the image
[U, S, V] = svd(Image);

% Reconstruct and display the original image using SVD matrices
Image_reconstructed = U * S * V';
figure;
imshow(uint8(Image_reconstructed));
title('Reconstructed Image using SVD');

% Compress the image using top k singular values and corresponding left/right singular vectors
k_values = [10, 50, 100];
for i = 1:length(k_values)
    k = k_values(i);
    Image_compressed = S;
    Image_compressed(k+1:end,k+1:end) = 0;
    Image2_compressed = U * Image_compressed * V';
    figure;
    imshow(uint8(Image2_compressed));
    title(sprintf('Compressed Image using Top %d Singular Values', k));
end
```

Reconstructed Image using SVD

Compressed Image using Top 10 Singular Values



Compressed Image using Top 50 Singular Values

Compressed Image using Top 100 Singular Values

When compressing an image with a specific value "j," only the essential features are retained and the others are discarded. To obtain a higher resolution similar to a real image, the value "j" must be increased. With lower values of "j," the loss of details in the image can result in added noise.

4.
C

```
% Read the flower.bmp image and convert it to grayscale and double format
Image = imread('/Users/bharath/Documents/MATLAB/flower.bmp');
Image = double(im2gray(Image));

% Perform SVD on the image
[U, S, V] = svd(Image);

% Reconstruct and display the original image using SVD matrices
Image_reconstructed = U * S * V';
figure;
imshow(uint8(Image_reconstructed));
title('Reconstructed Image using SVD');

% Compress the image using top k singular values and corresponding left/right singular vectors
k_values = 200;
for i = 1:length(k_values)
    k = k_values(i);
    Image_compressed = S;
    Image_compressed(k+1:end,k+1:end) = 0;
    Image2_compressed = U * Image_compressed * V';
    figure;
    imshow(uint8(Image2_compressed));
    title(sprintf('Compressed Image using Top %d Singular Values', k));
end
```

[redacted]

The dimensions of the image are dependent on the singular value decomposition (SVD) values, which are stored as a diagonal matrix. The size of the matrix is determined by the dimensions of the original image. For example, if k=200, the matrix size would be 200x200.

With k=200. The original image is of size m x n, while the left singular value matrix has a size of m x min(m x n), and the right singular value matrix has a size of min(m x n) x n. As we have taken k=200 the dimensions of the image will be transformed to m x 200 + 200 x 200 + 200 x n, significantly smaller than its original size m x n. The value of k being set to 200 presents a chance to reduce the size of the image, but this may not always be the best option. It's crucial to choose the value of k in a way that strikes a balance between the size of the compressed or reconstructed image for optimal results.