

## Lab 12: SVD

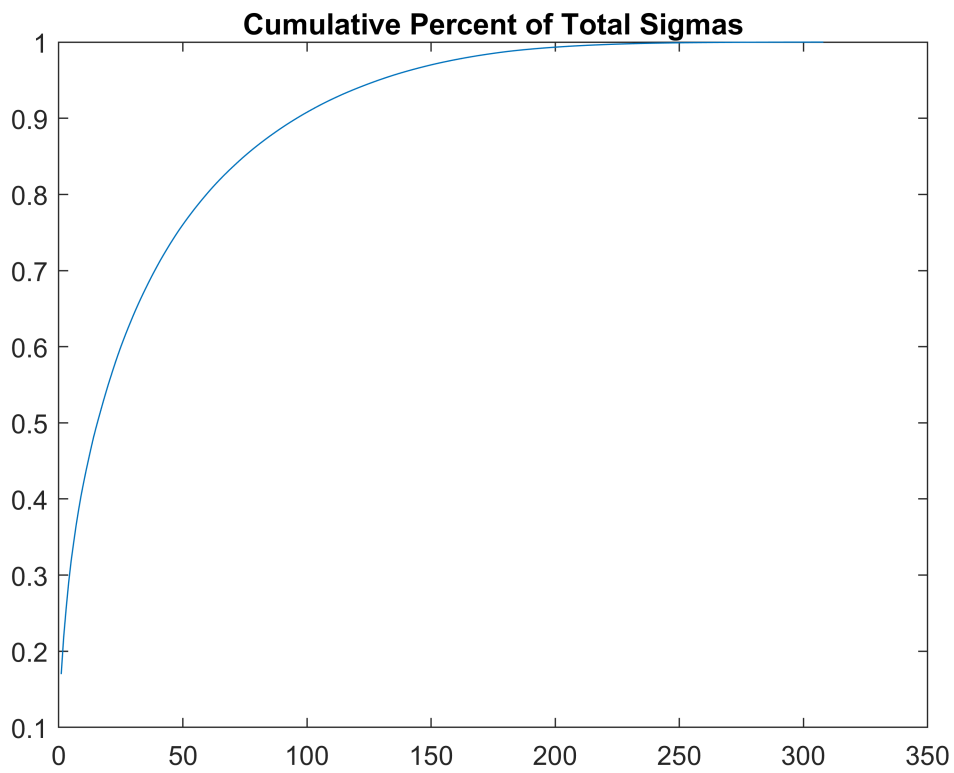
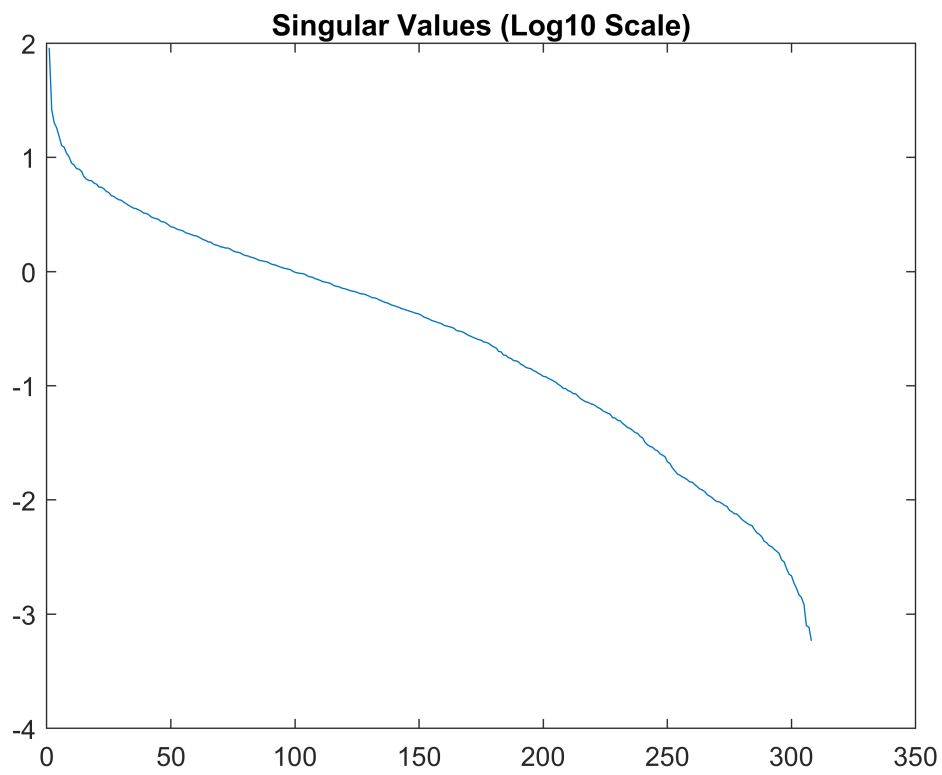
### Image compressing - Gorodetskii and Cell

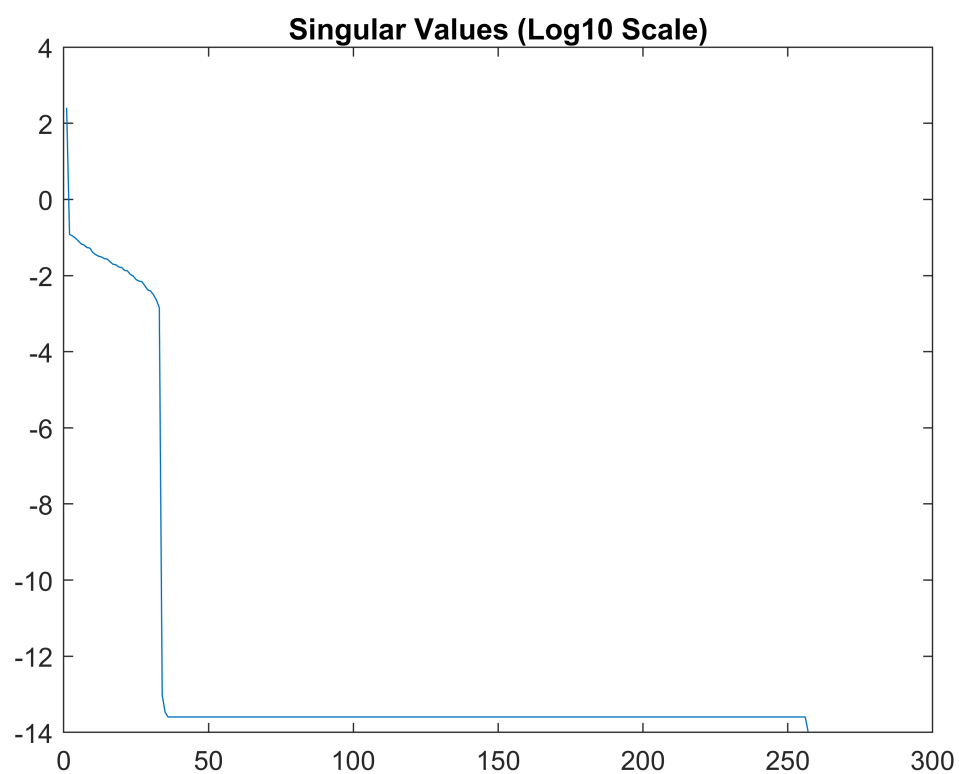
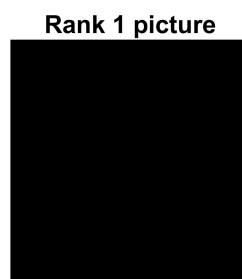
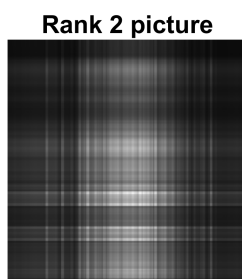
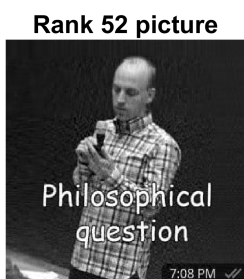
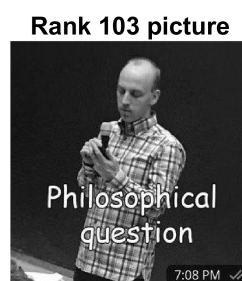
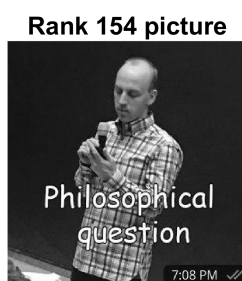
Cell is needed for understanding what does information means

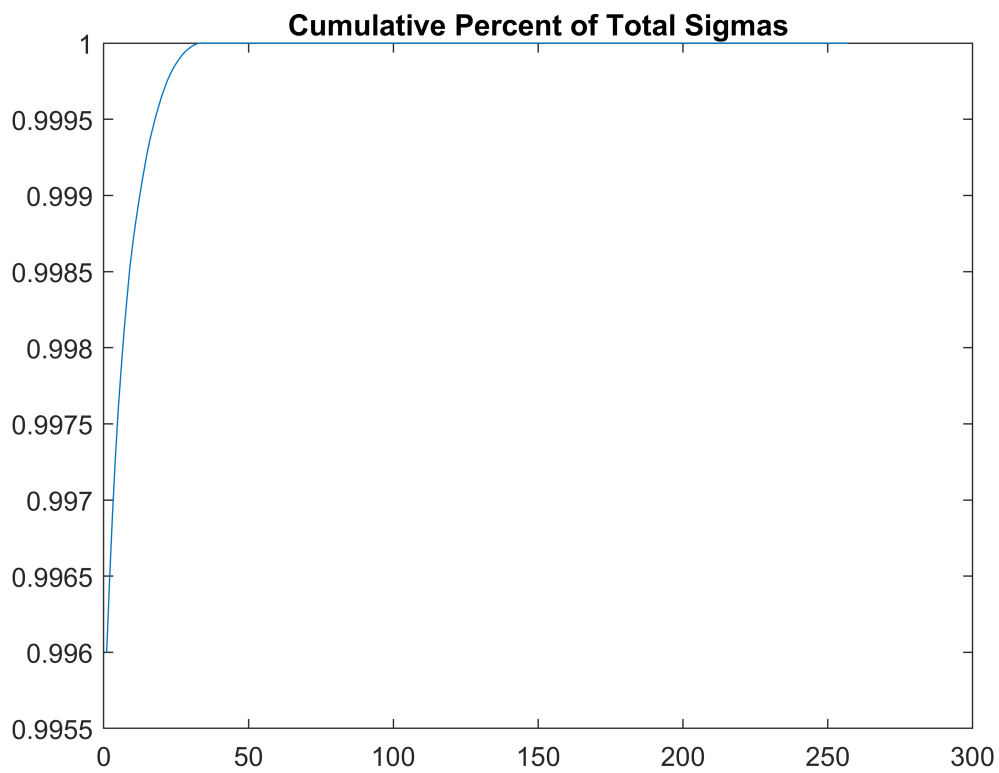
```
pic_name = ['tsar.jpg'; 'cell.jpg']
```

```
pic_name = 2x8 char array  
    'tsar.jpg'  
    'cell.jpg'
```

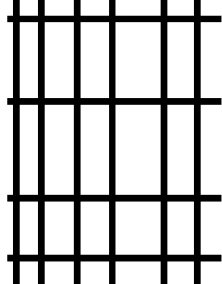
```
for pic_num=1:size(pic_name,1)  
    logo_num = im2double(rgb2gray(imread(pic_name(pic_num,:))));  
    [U, S, V] = svd(logo_num);  
    % Compute SVD of this picture  
    [U, S, V] = svd(logo_num);  
    S_myau = S;  
    % Plot the magnitude of the singular values (log scale)  
    sigmas = diag(S);  
    figure; plot(log10(sigmas)); title('Singular Values (Log10 Scale)');  
    % It shows how much information will be after redusing matrix rank  
    figure; plot(cumsum(sigmas) / sum(sigmas)); title('Cumulative Percent of Total Sigmas');  
  
    % Show full-rank picture  
    figure; subplot(2, 3, 1), imshow(logo_num), title('Full-Rank Logo');  
  
    % Compute low-rank approximations of the picture, and show them  
    ranks = [ceil(rank(S)/2), ceil(rank(S)/3), ceil(rank(S)/6), 2, 1];  
    for i = 1:length(ranks)  
        % Keep largest singular values, and nullify others.  
        approx_sigmas = sigmas; approx_sigmas(ranks(i):end) = 0;  
  
        % Form the singular value matrix, padded as necessary  
        ns = length(sigmas);  
        approx_S = S; approx_S(1:ns, 1:ns) = diag(approx_sigmas);  
  
        % Compute low-rank approximation by multiplying out component matrices.  
        approx_logo = U * approx_S * V';  
  
        % Plot approximation  
        subplot(2, 3, i + 1), imshow(approx_logo), title(sprintf('Rank %d picture', ranks(i)));  
    end  
end
```



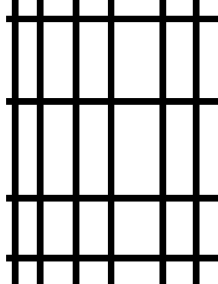




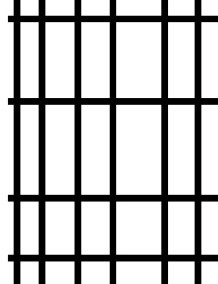
**Full-Rank Logo**



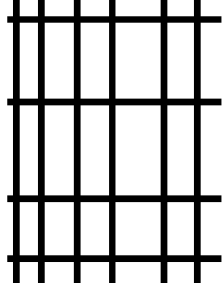
**Rank 17 picture**



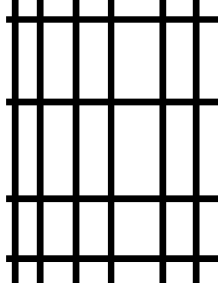
**Rank 11 picture**



**Rank 6 picture**



**Rank 2 picture**



**Rank 1 picture**

