
Table of Contents

.....	1
Exercise 1	1
Exercise 2	1
Exercise 3	1
Functions	3

```
clear
clc
```

Exercise 1

Exercise 2

```
rng(1)
img = randi(8, 4, 4) - 1;
[m, s, B] = BTC_encode(img);
img_decoded = BTC_decode(m, s, B);
```

Exercise 3

```
L1 = double(imread('cameraman.jpg'));
L2 = rand(256);

A = {L1, L2};
r = 2.^(0:7);
rel_error = zeros(2, length(r));
for i = 1:2
    img = A{i};
    [U, S, V] = svd(img);

    figure;
    for j = 1:length(r)
        img_r = reconstruct_r(U, S, V, r(j));

        subplot(3, 3, j)
        imshow(img_r, [])
        title(sprintf('r = %d', r(j)))

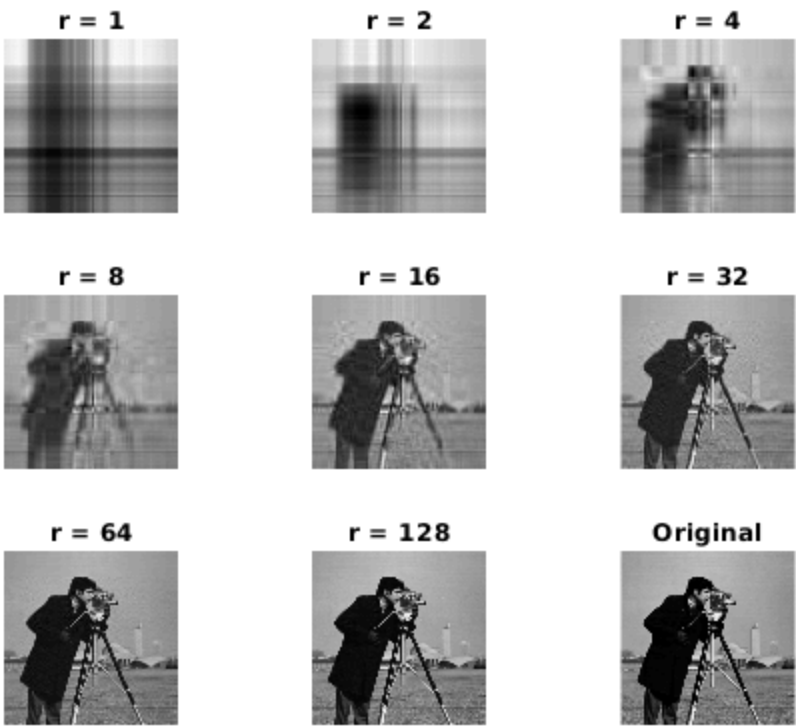
        rel_error(i, j) = frobenius(img - img_r) / frobenius(img);
    end
    subplot(3, 3, 9)
    imshow(img, [])
    title('Original')
end
```

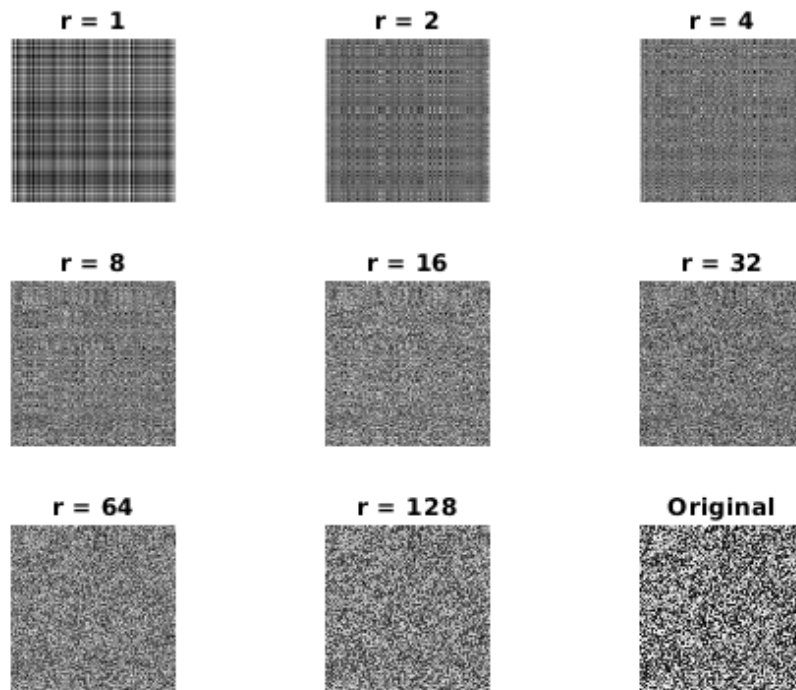
```
T = table('Size', [8, 2], 'VariableNames', {'L1', 'L2'}, ...
          'RowNames', {'1', '2', '4', '8', '16', '32', '64', '128'}, ...
          'VariableTypes', {'double', 'double'});
T(:, 1) = num2cell(rel_error(1, :));
T(:, 2) = num2cell(rel_error(2, :));
T
```

T =

8×2 table

	<i>L1</i>	<i>L2</i>
1	0.35943	0.49857
2	0.29108	0.49479
4	0.22583	0.4874
8	0.17715	0.47326
16	0.1299	0.44653
32	0.08538	0.39677
64	0.046534	0.30784
128	0.014282	0.16301





Functions

```
function [m, s, B] = BTC_encode(img)
    % Assume img is a 4x4 block
    m = mean(img(:));
    s = sqrt(sum((img(:) - m).^2)) / 4;
    B = img >= m;
end

function img = BTC_decode(m, s, B)
    % Assume B is a 4x4 block
    B_complement = 1 - B;
    Q = sum(B(:));
    P = sum(B_complement(:));
    A = sqrt(Q / P);
    img = (m + s / A) * B + (m - s * A) * B_complement;
end

function Ar = reconstruct_r(U, S, V, r)
    Ar = zeros(size(U, 1), size(V, 1));
    for i = 1:r
        Ar = Ar + S(i, i) * U(:, i) * V(:, i)';
    end
end

function fnorm = frobenius(X)
```

```
fnorm = 0;  
  
for i = 1:size(X, 1)  
    fnorm = fnorm + sum(X(i, :).^2);  
end  
  
fnorm = sqrt(fnorm);  
end
```

Published with MATLAB® R2019b