EE-636 : Assignment 5

SVD based Image Compression

Presented by :- Sayali R. Duragkar(203070025)

1) Images with different singular values:-

Image output using 1 singular values

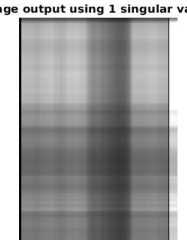

Image output using 2 singular values

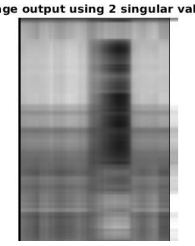

Image output using 3 singular values

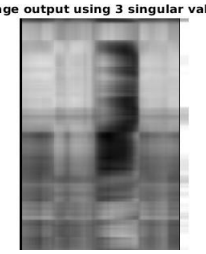

Image output using 4 singular values

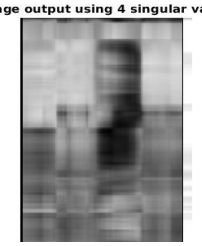

Image output using 5 singular values

Image output using 6 singular values

Image output using 7 singular values

Image output using 8 singular values

Image output using 9 singular values

Image output using 10 singular values

Image output using 11 singular values

Image output using 12 singular values

Image output using 13 singular values

Image output using 14 singular values

Image output using 15 singular values

Image output using 16 singular values

Image output using 50 singular values

Image output using 100 singular values

Image output using 150 singular values

Original Image

2) Pseudocode:-

Taking image for which we have to do svd based image compression.

Converting that color image to grayscale image .

Converting image data into matrix format(A).

Doing svd and splitting matrix i.e. A= U*S*V'

Where S is a diagonal matrix having singular values as its leading diagonal element.

We have also found out size of U,S,V matrix by using size() command

So by taking different no of singular values we are reconstructing image which is compressed.

```
    for N=1
      C=S;
      C(N+1:end,:)=0;   % instead of 1st row making other row elements 0
      C(:,N+1:end)=0;   %instead of 1st column making other column elements 0
      % i.e. overall only (1,1) entry of S matrix is non-zero other entries are 0
      D = U*C*V'
      %And for this matrix we are reconstructing image
      Figure;
      imshow(uint8(D));
  end
    for N=2:1:16
      C=S;
      C(N+1:end,:)=0;   % instead of 1st row making other row elements 0
      C(:,N+1:end)=0;   %instead of 1st column making other column elements 0
      % i.e. overall only (1,1) entry of S matrix is non-zero other entries are 0
      D = U*C*V'
      %And for this matrix we are reconstructing image
      Figure;
      imshow(uint8(D));
  end
    for N=50:50:150
      C=S;
      C(N+1:end,:)=0;   % instead of 1st row making other row elements 0
```

C(:,N+1:end)=0; %instead of 1st column making other column elements 0

% i.e. overall only (1,1) entry of S matrix is non-zero other entries are 0

D = U*C*V'

%And for this matrix we are reconstructing image

Figure;

imshow(uint8(D));

end

3)Code:-

```
close all;
clc;
inImage=imread('sayali123.jpg');
inImage=rgb2gray(inImage);
inImageD=double(inImage);
imwrite(uint8(inImageD), 'original.jpg');
[U,S,V]=svd(inImageD);
size(U)
size(S)
size(V)
figure;
imshow(inImage);
title('Original Image'); for
N = 1
    C = S;
    C(N+1:end,:)=0;
    C(:,N+1:end)=0;
    D = U*C*V';
    figure;
    buffer = sprintf('Image output using %d singular values', N);
    imshow(uint8(D));
    imwrite(uint8(D), sprintf('%dbw.jpg', N));
    title(buffer);
end
for N = 2:16
    C = S;
    C(N+1:end,:)=0;
    C(:,N+1:end)=0;
    D = U*C*V';
    figure;
    buffer = sprintf('Image output using %d singular values', N);
    imshow(uint8(D));
    imwrite(uint8(D), sprintf('%dbw.jpg', N));
```

```
    title(buffer);
end
for N = 50:50:150 C =
    S;
    C(N+1:end,:)=0;
    C(:,N+1:end)=0;
    D = U*C*V';
    figure;
    buffer = sprintf('Image output using %d singular values', N);
    imshow(uint8(D));
    imwrite(uint8(D), sprintf('%dbw.jpg', N));
    title(buffer);
end
```

```
ans =

   659   659


ans =

   659   450


ans =

   450   450
```