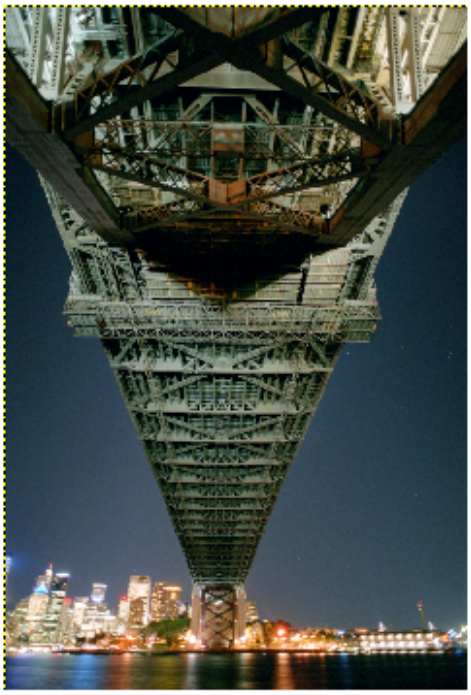
```
%WORK DONE BY GROUP - 1

close all
clear all
clc
inImage=imread('bridge.png');
X=im2gray(inImage); % Convert RBG to gray, 256 bit to double.
Y=im2double(X);
filename = 'bridge.png';
inImage=rgb2gray(inImage);
inImageD=double(inImage);
[a,b,c] = svd(Y);
 [X, map] = imread(filename);
 figure('Name','ORIGINAL component of the imported image');
 imshow(X);
 imwrite(X, '!original.png');
 R = X(:,:,1);
 G = X(:,:,2);
 B = X(:,:,3);
 Rimg = cat(3, R, zeros(size(R)), zeros(size(R)));
 Gimg = cat(3, zeros(size(G)), G, zeros(size(G)));
 Bimg = cat(3, zeros(size(B)), zeros(size(B)), B);
 %figure('Name','RED component of the imported image');
 %imshow(Rimg);
 %imwrite(Rimg, '!red.jpg');
 %figure('Name','GREEN component of the imported image');
 %imshow(Gimg);
 %imwrite(Gimg, '!green.jpg');
 %figure('Name','BLUE component of the imported image');
 %imshow(Bimg);
 %imwrite(Bimg, '!blue.jpg');
Red =double(R);
Green = double(G);
Blue = double(B);
dispEr = [];
numSVals = [];
N = 1;
% Compute values for the red image
[U,S,V]=svd(Red);
C = S;
C(N+1:end,:)=0;
C(:,N+1:end)=0;
Dr=U*C*V';
% Rebuild the data back into a displayable image and show it
%figure;
%buffer = sprintf('Red image output using %d singular values', N);
Rimg = cat(3, Dr, zeros(size(Dr)), zeros(size(Dr)));
%imshow(uint8(Rimg));
%imwrite(uint8(Rimg), sprintf('%dred.jpg', N));
%title(buffer);
% Compute values for the green image
[U2, S2, V2]=svd(Green);
```

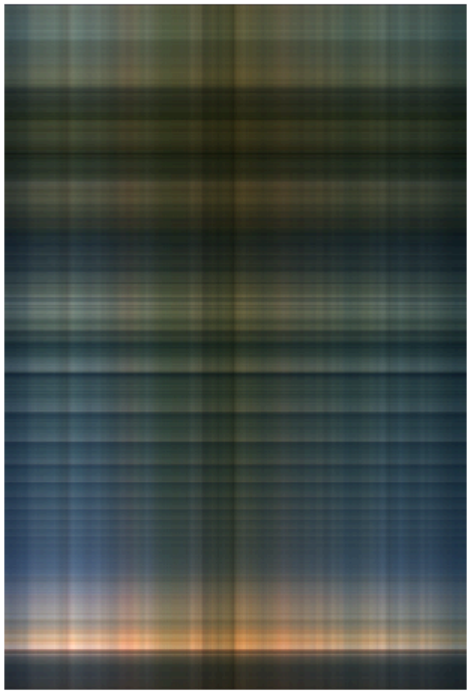
```
C = S2;
C(N+1:end,:)=0;
C(:,N+1:end)=0;
Dg=U2*C*V2';
% Rebuild the data back into a displayable image and show it
%figure;
%buffer = sprintf('Green image output using %d singular values', N);
Gimg = cat(3, zeros(size(Dg)), Dg, zeros(size(Dg)));
%imshow(uint8(Gimg));
%imwrite(uint8(Gimg), sprintf('%dgreen.jpg', N));
%title(buffer);
% Compute values for the blue image
[U3, S3, V3]=svd(Blue);
C = S3;
C(N+1:end,:)=0;
C(:,N+1:end)=0;
Db=U3*C*V3';
% Rebuild the data back into a displayable image and show it
%figure;
%buffer = sprintf('Blue image output using %d singular values', N);
Bimg = cat(3, zeros(size(Db)), zeros(size(Db)), Db);
%imshow(uint8(Bimg));
%imwrite(uint8(Bimg), sprintf('%dblue.jpg', N));
%title(buffer);
% Thake the data from the Red, Green, and Blue image
% Rebuild a colored image with the corresponding data and show it
figure;
buffer = sprintf('Colored image output using %d singular values', N);
Cimg = cat(3, Dr, Dg, Db);
imshow(uint8(Cimg));
imwrite(uint8(Cimg), sprintf('%dcolor.png', N));
title(buffer);
error=sum(sum((inImageD-Db).^2));
dispEr = [dispEr; error];
 numSVals = [numSVals; N];

for N=10:10:100
 % Recompute modes for the red image - already solved by SVD above
 C = S;
 C(N+1:end,:)=0;
 C(:,N+1:end)=0;
 Dr=U*C*V';
 % Rebuild the data back into a displayable image and show it
 %figure;
 %buffer = sprintf('Red image output using %d singular values', N);
 Rimg = cat(3, Dr, zeros(size(Dr)), zeros(size(Dr)));
 %imshow(uint8(Rimg));
 %imwrite(uint8(Rimg), sprintf('%dred.jpg', N));
 %title(buffer);
 % Recompute modes for the green image - already solved by SVD above
 C = S2;
 C(N+1:end,:)=0;
 C(:,N+1:end)=0;
 Dg=U2*C*V2';
```

```
 % Rebuild the data back into a displayable image and show it
 %figure;
 %buffer = sprintf('Green image output using %d singular values', N);
 Gimg = cat(3, zeros(size(Dg)), Dg, zeros(size(Dg)));
 %imshow(uint8(Gimg));
 %imwrite(uint8(Gimg), sprintf('%dgreen.jpg', N));
 %title(buffer);
 % Recompute modes for the blue image - already solved by SVD above
 C = S3;
 C(N+1:end,:)=0;
 C(:,N+1:end)=0;
 Db=U3*C*V3';
 % Rebuild the data back into a displayable image and show it
 %figure;
 %buffer = sprintf('Blue image output using %d singular values', N);
 Bimg = cat(3, zeros(size(Db)), zeros(size(Db)), Db);
 %imshow(uint8(Bimg));
 %imwrite(uint8(Bimg), sprintf('%dblue.jpg', N));
 %title(buffer);
 % Thake the data from the Red, Green, and Blue image
 % Rebuild a colored image with the corresponding data and show it
 figure;
 buffer = sprintf('Colored image output using %d singular values', N);
 Cimg = cat(3, Dr, Dg, Db);
 imshow(uint8(Cimg));
 imwrite(uint8(Cimg), sprintf('%dcolor.png', N));
 title(buffer);
 error=sum(sum((inImageD-Db).^2));
 dispEr = [dispEr; error];
 numSVals = [numSVals; N];

end
for N=100:50:300
 % Recompute modes for the red image - already solved by SVD above
 C = S;
 C(N+1:end,:)=0;
 C(:,N+1:end)=0;
 Dr=U*C*V';
 % Rebuild the data back into a displayable image and show it
 %figure;
 %buffer = sprintf('Red image output using %d singular values', N);
 Rimg = cat(3, Dr, zeros(size(Dr)), zeros(size(Dr)));
 %imshow(uint8(Rimg));
 %imwrite(uint8(Rimg), sprintf('%dred.jpg', N));
 %title(buffer);
 % Recompute modes for the green image - already solved by SVD above
 C = S2;
 C(N+1:end,:)=0;
 C(:,N+1:end)=0;
 Dg=U2*C*V2';
 % Rebuild the data back into a displayable image and show it
 %figure;
 %buffer = sprintf('Green image output using %d singular values', N);
 Gimg = cat(3, zeros(size(Dg)), Dg, zeros(size(Dg)));
```

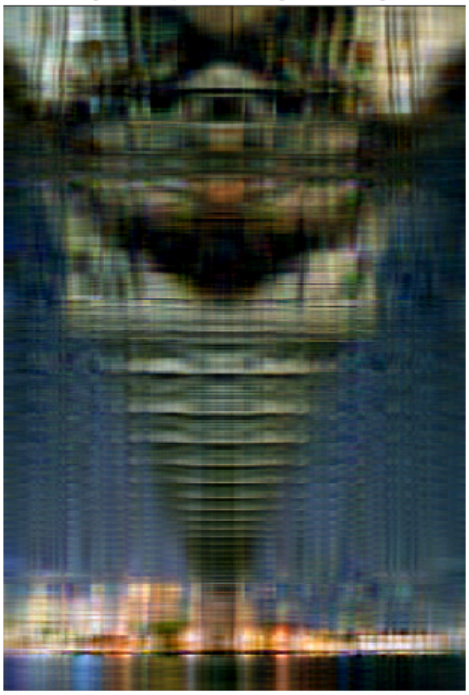
```
 %imshow(uint8(Gimg));
 %imwrite(uint8(Gimg), sprintf('%dgreen.jpg', N));
 %title(buffer);
 % Recompute modes for the blue image - already solved by SVD above
 C = S3;
 C(N+1:end,:)=0;
 C(:,N+1:end)=0;
 Db=U3*C*V3';
 % Rebuild the data back into a displayable image and show it
 %figure;
 %buffer = sprintf('Blue image output using %d singular values', N);
 Bimg = cat(3, zeros(size(Db)), zeros(size(Db)), Db);
 %imshow(uint8(Bimg));
 %imwrite(uint8(Bimg), sprintf('%dblue.jpg', N));
 %title(buffer);
 % Thake the data from the Red, Green, and Blue image
 % Rebuild a colored image with the corresponding data and show it
 figure;
 buffer = sprintf('Colored image output using %d singular values', N);
 Cimg = cat(3, Dr, Dg, Db);
 imshow(uint8(Cimg));
 imwrite(uint8(Cimg), sprintf('%dcolor.png', N));
 title(buffer);
 error=sum(sum((inImageD-Db).^2));
 % store vals for display
 dispEr = [dispEr; error];
 numSVals = [numSVals; N];
end
figure;
title('Error in compression');
plot(numSVals, dispEr);
grid on
xlabel('Number of Singular Values used');
ylabel('Error between compress and original image');

r=1:450;
   for i=1:450
       Xap=a(:,1:i)*b(1:i,1:i)*c(:,1:i)';
       %error(i)=immse(Xap,Y);
       psne(i)=psnr(Xap,Y);
   end
   figure
   plot(r,psne);
   hold on
   ylabel('PSNR')
   xlabel('r value')
   title('r vs PSNR')
```



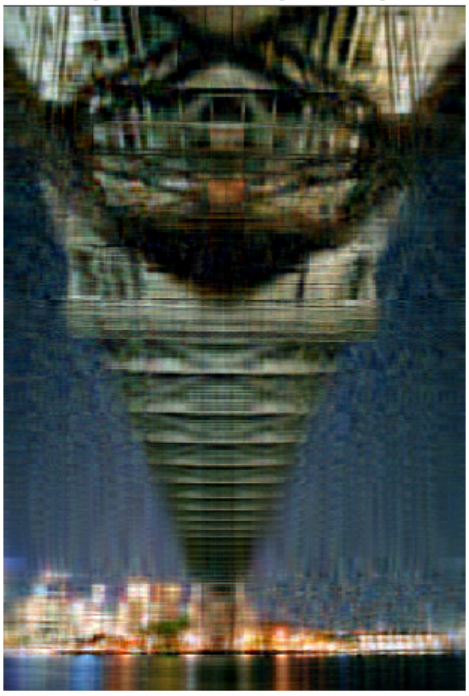
Colored image output using 1 singular values



Colored image output using 10 singular values



Colored image output using 20 singular values



Colored image output using 30 singular values

Colored image output using 40 singular values

Colored image output using 50 singular values

Colored image output using 60 singular values

Colored image output using 70 singular values

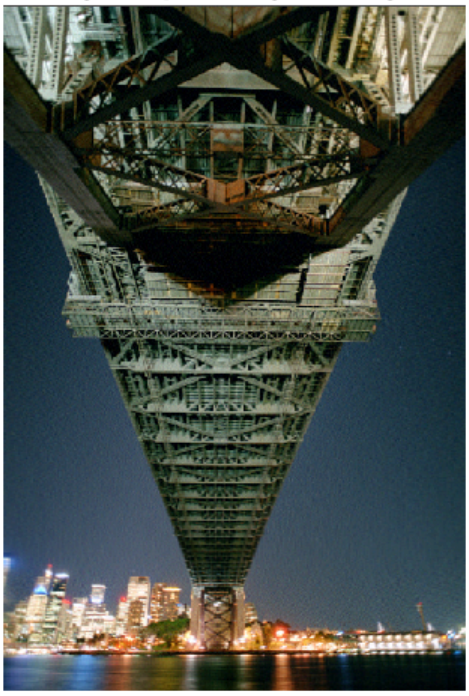
Colored image output using 80 singular values

Colored image output using 90 singular values

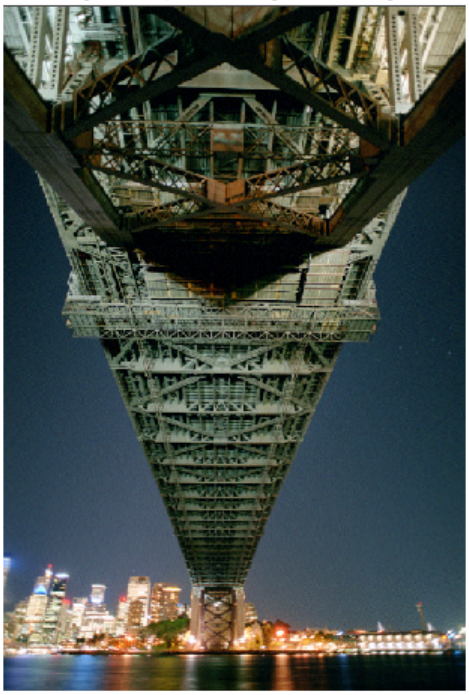
Colored image output using 100 singular values

Colored image output using 100 singular values

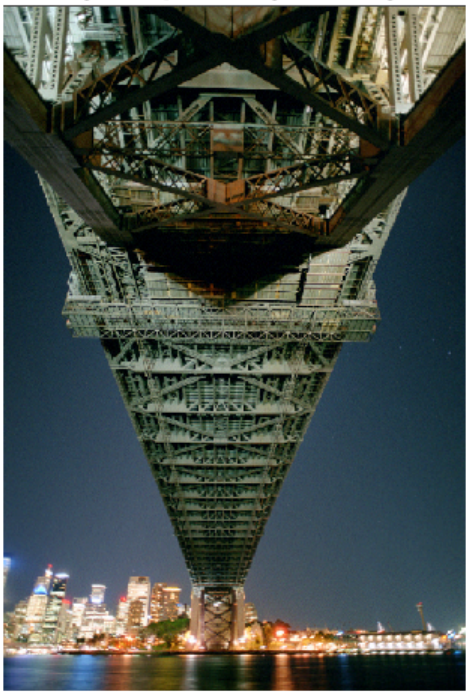
Colored image output using 150 singular values



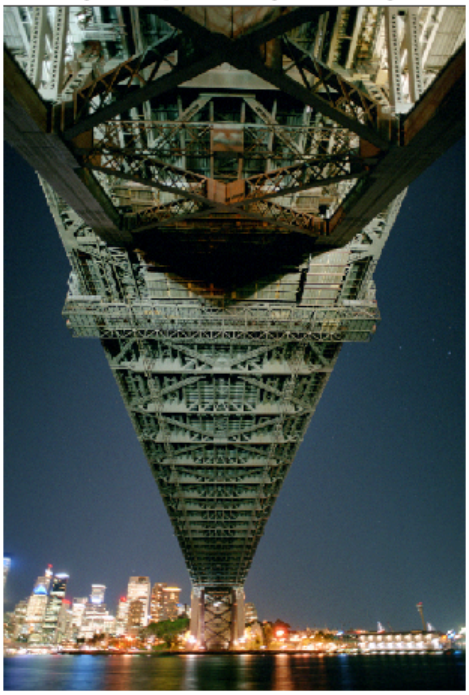
Colored image output using 200 singular values

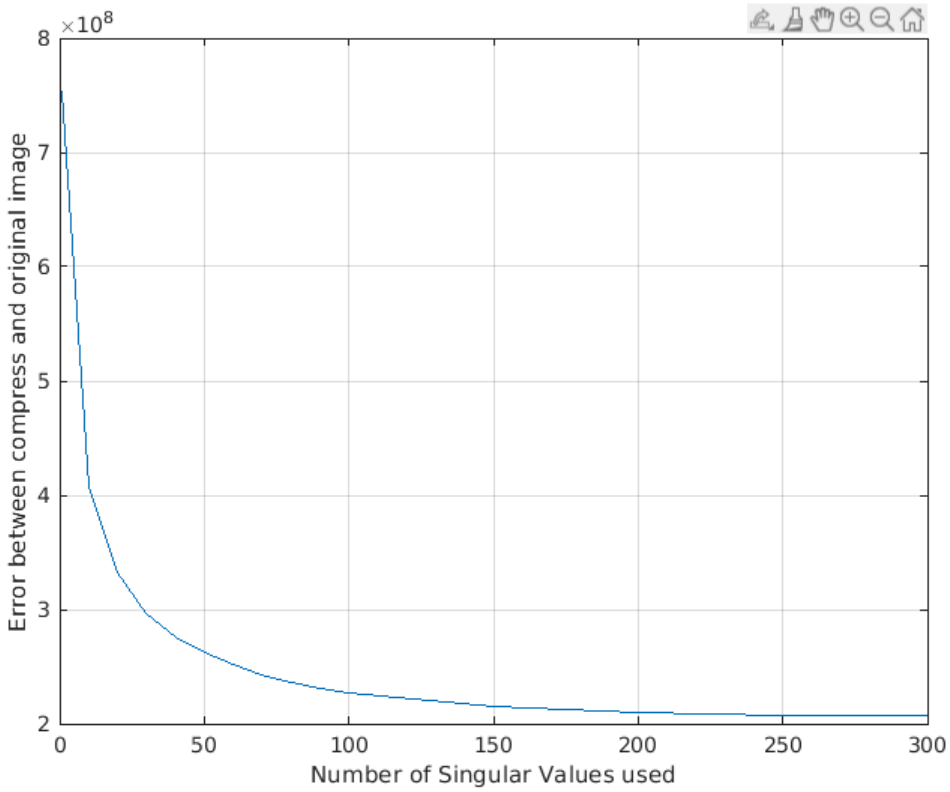


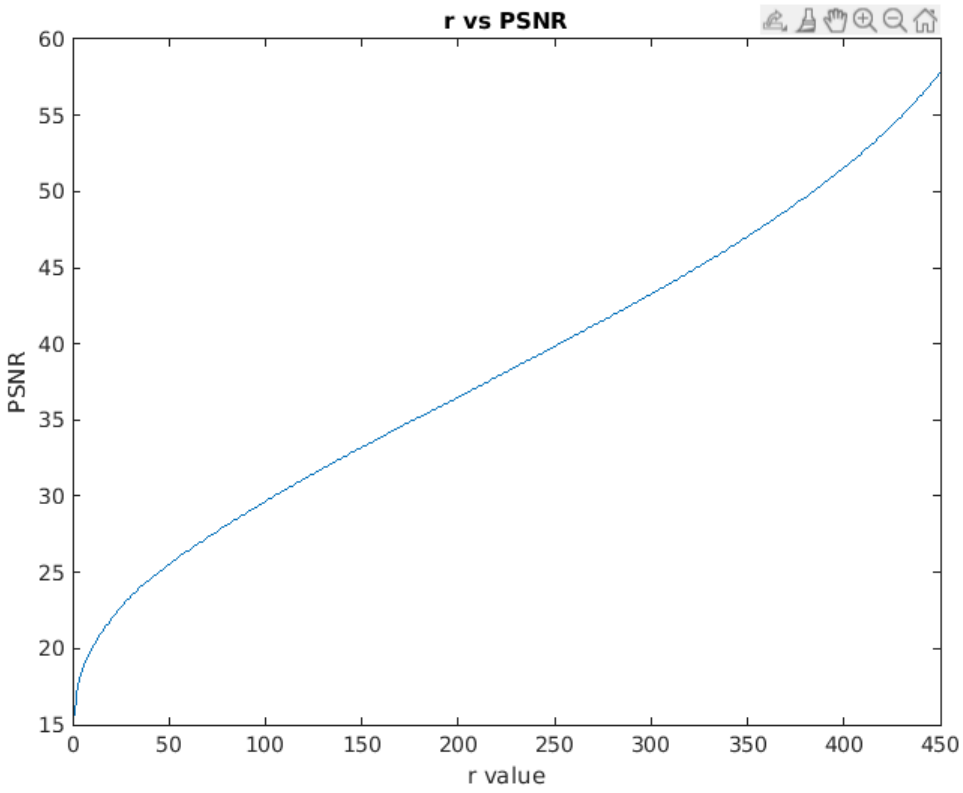
Colored image output using 250 singular values



Colored image output using 300 singular values







Published with MATLAB® R2021a