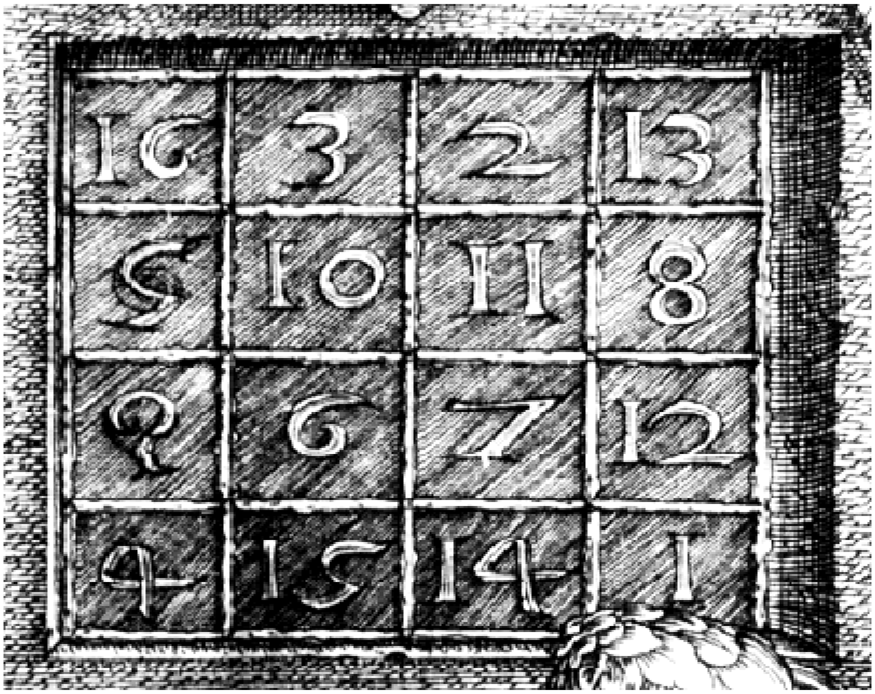
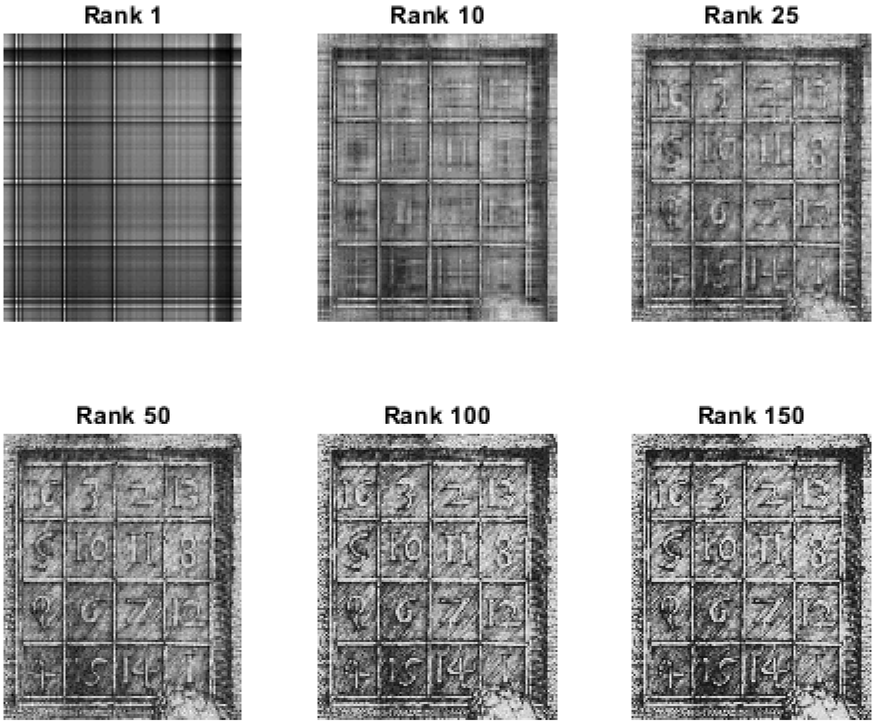
```
% Image Compression
% The following code uses Singular Value Decomposition to analyze
 image
% compression. We begin by calculating a few low-ranking
 approximations to
% explore the rankings at which the image becomes more recognizable.

imagedemo;
load(imglist{5},'X','map'); imagesc(X); colormap(map); axis off

[U, Sigma, V] = svd(X);
V = V';
X1 = U(:,1)*Sigma(1,1)*V(1,:);
X10 = U(:,1:10)*Sigma(1:10,1:10)*V(1:10,:);
X25 = U(:,1:25)*Sigma(1:25,1:25)*V(1:25,:);
X50 = U(:,1:50)*Sigma(1:50,1:50)*V(1:50,:);
X100 = U(:,1:100)*Sigma(1:100,1:100)*V(1:100,:);
X150 = U(:,1:150)*Sigma(1:150,1:150)*V(1:150,:);

figure('Name', 'Rank Approximations')
subplot(2,3,1)
imagesc(X1); colormap(map); axis off
title('Rank 1')
subplot(2,3,2)
imagesc(X10); colormap(map); axis off
title('Rank 10')
subplot(2,3,3)
imagesc(X25); colormap(map); axis off
title('Rank 25')
subplot(2,3,4)
imagesc(X50); colormap(map); axis off
title('Rank 50')
subplot(2,3,5)
imagesc(X100); colormap(map); axis off
title('Rank 100')
subplot(2,3,6)
imagesc(X150); colormap(map); axis off
title('Rank 150')

% The image is the most clear with Rank 150.
% However, the image becomes recognizable at Ranks 50 or 100 which are
 less
% expensive to calculate.
```

```
ans =

   256     3


ColorMapIndex =
        37


ans =

        0.3854            0            0


ans =

          1     0     0
          1     1     1
          0     0     1
          0     0     0
```





Published with MATLAB® R2019b