```matlab
%%This class gives tools for image compressing via Principal%%
%%Component Analysis (KLT)%%
%%written by Tim Jaschek as a part of his bachelor thesis%%

%%Used to generate FIGURE 6 %%
%%...to generate it, type the following in your MATLAB command:
%%Image;
%%Image.program();
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

classdef Image
   properties (Constant)
   end
   methods (Static)
       function X = load_space()
           %load an image
           X = imread('bluemarbel.jpg');
           %show the image
           figure
           image(X)
       end
       function B = blurBW(X)
           %function to blur an B/W image
           [m,n] = size(X);
           X = double(X);
           %%to 25%
           mm=m/2;
           nn = n/2;
           B=zeros(mm,nn);
           for i=1:mm
               for j=1:nn
                   B(i,j) = 1/4 *(X(2*i-1,2*j-1) + X(2*i,2*j-1) + X(2*i-1,2*j) + X(2*i,2*j));
               end
           end
           B = uint8(B);
       end
       function Z = blur(X)
           %function to blur an R/G/B image
           XR = X(:,:,1);
           XG = X(:,:,2);
           XB = X(:,:,3);
           RN = Image.blurBW(XR);
           GN = Image.blurBW(XG);
```

```
            BN = Image.blurBW(XB);
            Z = cat(3, RN, GN, BN);
        end
        function X_trans = PCA(X,k)
            %function for image compression via KLT of a B/W image
            [m,n]=size(X);
            X = double(X);
            X_hat = X;
            mean = zeros(1,n);
            K = zeros(n,n);
            %%correct mean%%%
            for i=1:n
                mean(i) = sum(X(:,i))/m ;
                X_hat(:,i) = X_hat(:,i)-mean(i);
            end
            %%covariance matrix%%%
            for i=1:n
                for j=1:n
                     K(i,j) = (1/(n-1))*dot((X_hat(:,i)),(X_hat(:,j)));
                end
            end
            %%Eigenvalues and Eigenvectors%%%
            [V,D]=eig(K);
            [lambda,ind] = sort(diag(D),'descend');
            Phi = V(:,ind);
            Phi = Phi(:,1:k);
            PhiT = Phi.';
            %%%Transform X %%%
            Y = PhiT*(X_hat);
            X_trans = Phi*Y;
            for i=1:n
                X_trans(:,i) = X_trans(:,i)+mean(i);
            end
            X_trans = uint8(X_trans);
        end
        function Z = PCA_RGB(X,k);
            %function for image compressing via KLT of an R/G/B image
            XR = X(:,:,1);
            XG = X(:,:,2);
            XB = X(:,:,3);
            RN = Image.PCA(XR,k);
            GN = Image.PCA(XG,k);
```

```matlab
           BN = Image.PCA(XB,k);
           Z = cat(3, RN, GN, BN);
       end
       function program()
          %generates FIGURE 6
          disp('loading image')
          X = Image.load_space();
          figure
          set(gca, 'XTickLabel', [],'XTick',[])
          h = subplot(2,2,1)
          imshow(X)
          disp('data compressing...')
          Z = Image.PCA_RGB(X,400);
          hh = subplot(2,2,2)
          imshow(Z)
          Z = Image.PCA_RGB(X,200);
          hhh = subplot(2,2,3)
          imshow(Z)
          Z = Image.PCA_RGB(X,100);
          hhhh = subplot(2,2,4)
          imshow(Z)
          p = get(h, 'pos');
          pp = get(hh, 'pos');
          ppp = get(hhh, 'pos');
          pppp = get(hhhh, 'pos');
          p([3,4]) = p([3,4]) + [0.1 0.1];
          set(h, 'pos', p);
          pp([3,4]) = pp([3,4]) + [0.1 0.1];
          set(hh, 'pos', pp);
          ppp([3,4]) = ppp([3,4]) + [0.1 0.1];
          set(hhh, 'pos', ppp);
          pppp([3,4]) = pppp([3,4]) + [0.1 0.1];
          set(hhhh, 'pos', pppp);
       end
   end
end
```