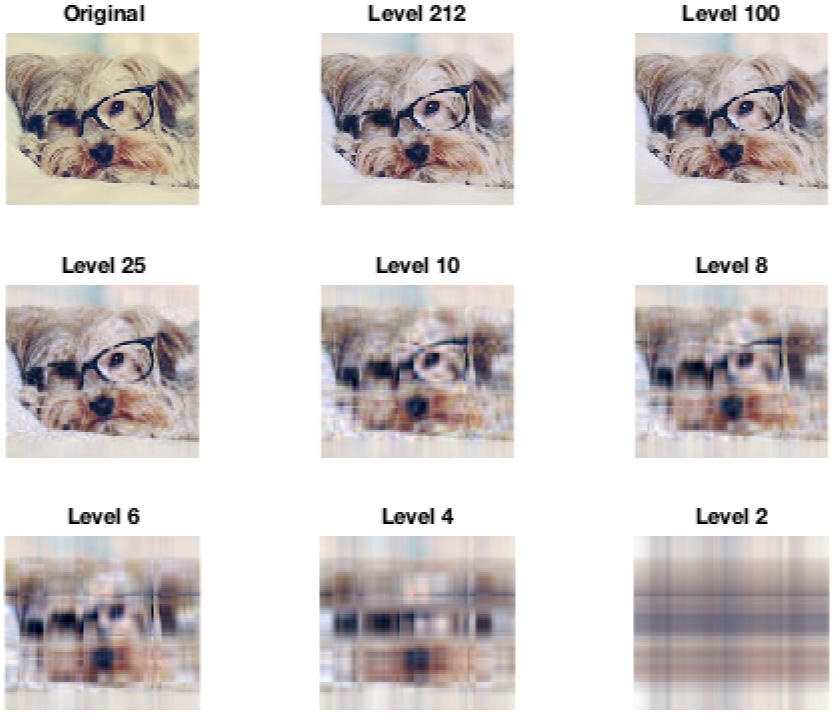
```
% Application for Single Value Decomposition in analysing
% the principle component of an image

% Prepare for command window, editor and work space
clc
clear
close all

% Read in the image 'Dog.png'
Dog_image = imread('Dog.png');

% Display the basic infomation of the picture
whos Dog_image

% Subtract the third dimension into three channels: red, green, blue
Dog_1 = im2double(Dog_image(:, :, 1));
Dog_2 = im2double(Dog_image(:, :, 2));
Dog_3 = im2double(Dog_image(:, :, 3));

% Plug the third dimension into a new image to compare the difference
% between the pre-processed image and the post-processed one
Dog_image2(:, :, 1) = Dog_1;
Dog_image2(:, :, 2) = Dog_2;
Dog_image2(:, :, 3) = Dog_3;

% Perform SVD on the matrix Dog_i and store the corresponding co-
domain orthogonal
% matrix in U_i, diagonal single value matrix in S_i, domain matrix in
 V_i
[U_1, S_1, V_1]=svd(Dog_1);
[U_2, S_2, V_2]=svd(Dog_2);
[U_3, S_3, V_3]=svd(Dog_3);

% Take out the diagonal of the single value matrix S_i and store it
 into the
% corresponding vector s_i
s_1 = diag(S_1);
s_2 = diag(S_2);
s_3 = diag(S_3);

% Initialize the approximation matrix of the Dog matrix
approximation_Dog = zeros(size(Dog_image));

% Construct a vector consisting of chosen levels of approximation in
% descending order
levels = [212, 100, 25, 10, 8, 6, 4, 2];

figure; imshow(Dog_image2), title('Image composed by three original
 copies of the third dimension');

figure; subplot(3, 3, 1), imshow(Dog_image), title('Original');
```

```
% Iterate through i to plot approximations of the Dog in different
% level ofaccuracy
for i = 1:length(levels)

    % Throw out the trifling terms in S
    approximation_s_1 = s_1;
    approximation_s_1(levels(i):end) = 0;

    approximation_s_2 = s_2;
    approximation_s_2(levels(i):end) = 0;

    approximation_s_3 = s_2;
    approximation_s_3(levels(i):end) = 0;

    % Construct the new single value matrix
    length_s_1 = length(s_1);
    approximation_S_1 = S_1;
    approximation_S_1(1:length_s_1, 1:length_s_1) =
 diag(approximation_s_1);

    length_s_2 = length(s_2);
    approximation_S_2 = S_2;
    approximation_S_2(1:length_s_2, 1:length_s_2) =
 diag(approximation_s_2);

    length_s_3 = length(s_3);
    approximation_S_3 = S_3;
    approximation_S_3(1:length_s_3, 1:length_s_3) =
 diag(approximation_s_3);

    % Construc the approximation of matrix Dog by using SVD definition
    approximation_Dog_1 = U_1 * approximation_S_1 * V_1';
    approximation_Dog_2 = U_2 * approximation_S_2 * V_2';
    approximation_Dog_3 = U_3 * approximation_S_3 * V_3';

    approximation_Dog(:, :, 1) = approximation_Dog_1;
    approximation_Dog(:, :, 2) = approximation_Dog_2;
    approximation_Dog(:, :, 3) = approximation_Dog_3;

    % Subplot the new approximation
    subplot(3, 3, i+1), imshow(approximation_Dog),
 title(sprintf('Level %d', levels(i)));
end
```

```
  Name             Size                Bytes  Class    Attributes

  Dog_image      212x238x3            151368  uint8
```

Image composed by three original copies of the third dimension



Original

Level 212

Level 100

Level 25

Level 10

Level 8

Level 6

Level 4

Level 2

Published with MATLAB® R2017a