

(a) Source codes:

Functions:

```
function rgb = hsi2rgb(hsi)
    %read hsi
    hsi = im2double(hsi);
    H = hsi(:, :, 1) * 2 * pi;
    S = hsi(:, :, 2);
    I = hsi(:, :, 3);

    R = zeros(size(hsi, 1), size(hsi, 2));
    G = zeros(size(hsi, 1), size(hsi, 2));
    B = zeros(size(hsi, 1), size(hsi, 2));

    % RG sector (0 <= H < 2*pi/3)
    idx = find( (0 <= H) & (H < 2*pi/3));
    B(idx) = I(idx) .* (1 - S(idx));
    R(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx)) ./ ...
        cos(pi/3 - H(idx)));
    G(idx) = 3*I(idx) - (R(idx) + B(idx));
    % BG sector (2*pi/3 <= H < 4*pi/3)
    idx = find( (2*pi/3 <= H) & (H < 4*pi/3) );
    R(idx) = I(idx) .* (1 - S(idx));
    G(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx) - 2*pi/3) ./ ...
        cos(pi - H(idx)));
    B(idx) = 3*I(idx) - (R(idx) + G(idx));
    % BR sector
    idx = find( (4*pi/3 <= H) & (H <= 2*pi));
    G(idx) = I(idx) .* (1 - S(idx));
    B(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx) - 4*pi/3) ./ ...
        cos(5*pi/3 - H(idx)));
    R(idx) = 3*I(idx) - (G(idx) + B(idx));

    % output
    rgb = cat(3, R, G, B);
    rgb = max(min(rgb, 1), 0);
end
```

```

function hsi = rgb2hsi(rgb)
    %read image
    rgb = im2double(rgb);
    r = rgb(:, :, 1);
    g = rgb(:, :, 2);
    b = rgb(:, :, 3);

    % Implement the conversion equations.
    num = 0.5*((r - g) + (r - b));
    den = sqrt((r - g).^2 + (r - b).*(g - b));
    den(den == 0) = eps;
    theta = acos(num./(den));

    H = theta;
    H(b > g) = 2*pi - H(b > g);
    H = H/(2*pi);

    num = min(min(r, g), b);
    den = r + g + b;
    den(den == 0) = eps;
    S = 1 - 3.* num./den;

    H(S == 0) = 0;

    I = (r + g + b)/3;

    % Combine all three results into an hsi image.
    hsi = cat(3, H, S, I);
end

```

Main Program

```

%% Clear the environment and the command line
clc;
close all;
clear;

%% read image
rose_img = imread('LovePeace rose.tif');
[m,n] = size(rose_img);

```

```

%% Images of R, G, B
rose_r = rose_img(:,:,1);
rose_g = rose_img(:,:,2);
rose_b = rose_img(:,:,3);

%% Images of H,S,I
rose_hsi = rgb2hsi(rose_img);
rose_h = uint8(255*mat2gray(abs(rose_hsi(:,:,1)))));
rose_s = uint8(255*mat2gray(abs(rose_hsi(:,:,2)))));
rose_i = uint8(255*mat2gray(abs(rose_hsi(:,:,3)))));

%% RGB sharpening
laplacianKernel = [-1,-1,-1;-1,8,-1;-1,-1,-1];
rose_rgb_lap = imfilter(double(rose_img), laplacianKernel, "replicate");
rose_rgb_lap = uint8(255*mat2gray(abs(rose_rgb_lap))));
rose_rgb_sharpened = rose_img + rose_rgb_lap;

%% HSI sharpening
laplacianKernel = [-1,-1,-1;-1,8,-1;-1,-1,-1];
rose_i_lap = imfilter(double(rose_i), laplacianKernel, "replicate");
rose_i_lap = uint8(255*mat2gray(abs(rose_i_lap))));
rose_i_sharpened = rose_i + rose_i_lap;
rose_hsi_sharpened = cat(3, rose_h, rose_s, rose_i);
rose_hsi_sharpened = hsi2rgb(rose_hsi_sharpened);
rose_hsi_sharpened = uint8(255*mat2gray(abs(rose_hsi_sharpened))));

%% Difference betewwn two image
rose_diff = rose_rgb_sharpened - rose_hsi_sharpened + 255/2;

%% output
figure(1);
imshow(rose_img, []);

figure(2);
imshow(rose_r, []);
fig= gcf;
exportgraphics(fig, 'rose_r.png', 'Resolution', 200);

figure(3);
imshow(rose_g, []);
fig= gcf;
exportgraphics(fig, 'rose_g.png', 'Resolution', 200);

figure(4);
imshow(rose_b, []);

```

```
fig= gcf;
exportgraphics(fig, 'rose_b.png', 'Resolution', 200);

figure(5);
imshow(rose_h, []);
fig= gcf;
exportgraphics(fig, 'rose_h.png', 'Resolution', 200);

figure(6);
imshow(rose_s, []);
fig= gcf;
exportgraphics(fig, 'rose_s.png', 'Resolution', 200);

figure(7);
imshow(rose_i, []);
fig= gcf;
exportgraphics(fig, 'rose_i.png', 'Resolution', 200);

figure(9);
imshow(rose_rgb_sharpened, []);
fig= gcf;
exportgraphics(fig, 'rose_rgb_sharpened.png', 'Resolution', 200);

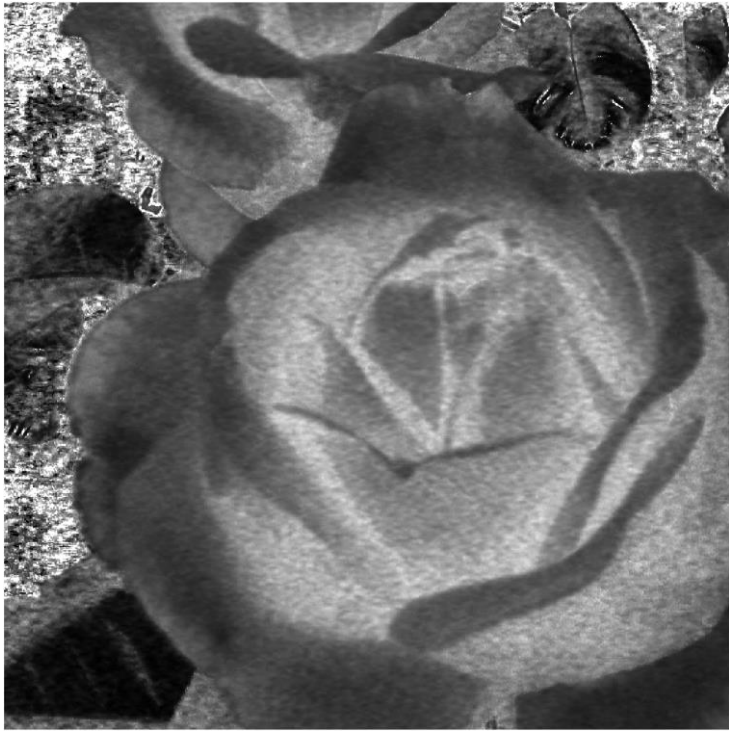
figure(11);
imshow(rose_i_sharpened, []);
fig= gcf;
exportgraphics(fig, 'rose_i_sharpened.png', 'Resolution', 200);

figure(12);
imshow(rose_hsi_sharpened, []);
fig= gcf;
exportgraphics(fig, 'rose_hsi_sharpened.png', 'Resolution', 200);

figure(13);
imshow(rose_diff, []);
fig= gcf;
exportgraphics(fig, 'rose_diff.png', 'Resolution', 200);
```

(b) Images of R, G, B, H, S and I component images:

R	G
B	H
S	I



(c) Output images enhanced by RGB-sharpening and HSI-sharpening scheme:

RGB-sharpening



HSI-sharpening



(d) Difference image of two images obtained in (c):

