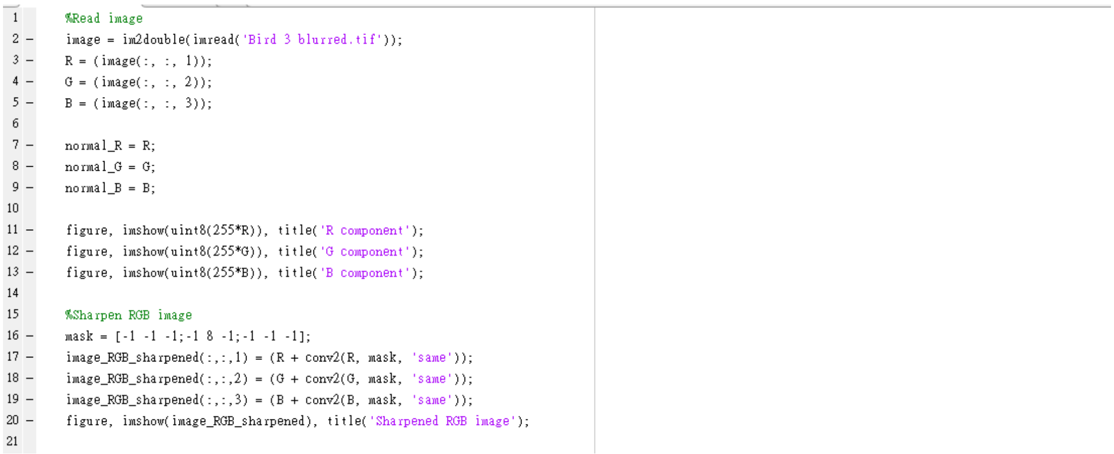
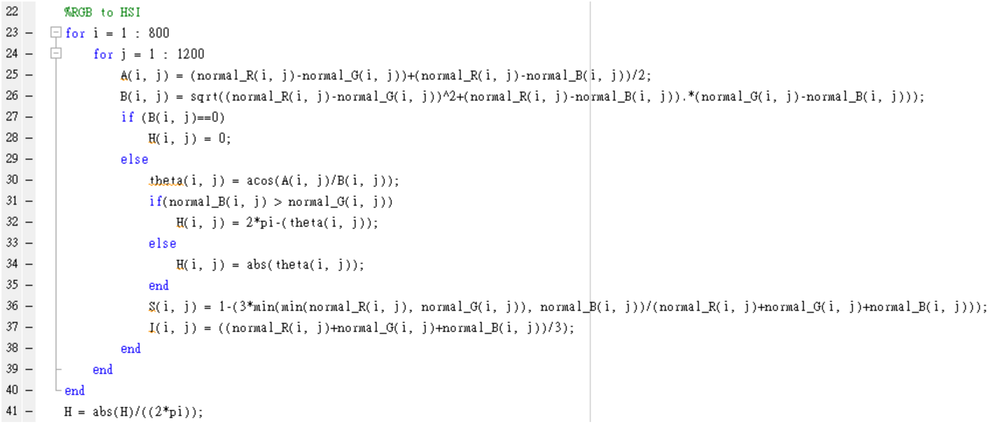
DIP Project 4

電信碩 沈衍薰 309513047

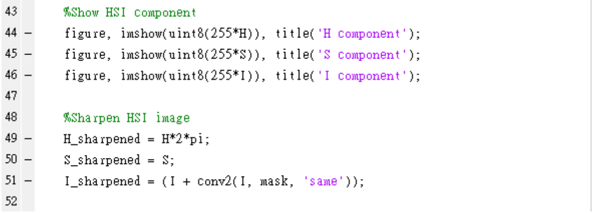
1. Source code



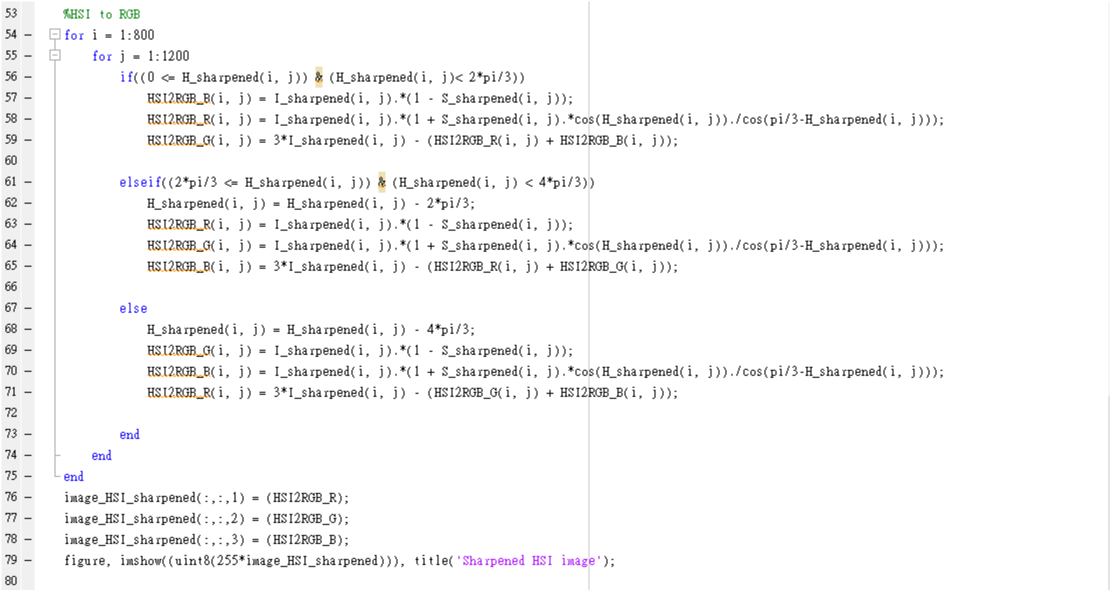
```
%Read image
image = im2double(imread('Bird 3 blurred.tif'));
R = (image(:, :, 1));
G = (image(:, :, 2));
B = (image(:, :, 3));

normal_R = R;
normal_G = G;
normal_B = B;

figure, imshow(uint8(255*R)), title('R component');
figure, imshow(uint8(255*G)), title('G component');
figure, imshow(uint8(255*B)), title('B component');

%Sharpen RGB image
mask = [-1 -1 -1;-1 8 -1;-1 -1 -1];
image_RGB_sharpened(:,:,1) = (R + conv2(R, mask, 'same'));
image_RGB_sharpened(:,:,2) = (G + conv2(G, mask, 'same'));
image_RGB_sharpened(:,:,3) = (B + conv2(B, mask, 'same'));
figure, imshow(image_RGB_sharpened), title('Sharpened RGB image');

```



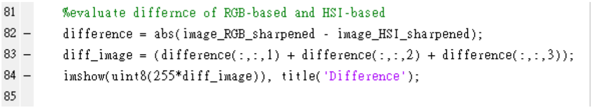
```
%RGB to HSI
for i = 1 : 800
    for j = 1 : 1200
        A(i, j) = (normal_R(i, j)-normal_G(i, j))+(normal_R(i, j)-normal_B(i, j))/2;
        B(i, j) = sqrt((normal_R(i, j)-normal_G(i, j))^2+(normal_R(i, j)-normal_B(i, j)).*(normal_G(i, j)-normal_B(i, j)));
        if (B(i, j)==0)
            H(i, j) = 0;
        else
            theta(i, j) = acos(A(i, j)/B(i, j));
            if(normal_B(i, j) > normal_G(i, j))
                H(i, j) = 2*pi-(theta(i, j));
            else
                H(i, j) = abs(theta(i, j));
            end
            S(i, j) = 1-(3*min(min(normal_R(i, j), normal_G(i, j)), normal_B(i, j))/(normal_R(i, j)+normal_G(i, j)+normal_B(i, j)));
            I(i, j) = ((normal_R(i, j)+normal_G(i, j)+normal_B(i, j))/3);
        end
    end
end
H = abs(H)/((2*pi));
```



```
%Show HSI component
figure, imshow(uint8(255*H)), title('H component');
figure, imshow(uint8(255*S)), title('S component');
figure, imshow(uint8(255*I)), title('I component');

%Sharpen HSI image
H_sharpened = H*2*pi;
S_sharpened = S;
I_sharpened = (I + conv2(I, mask, 'same'));

```



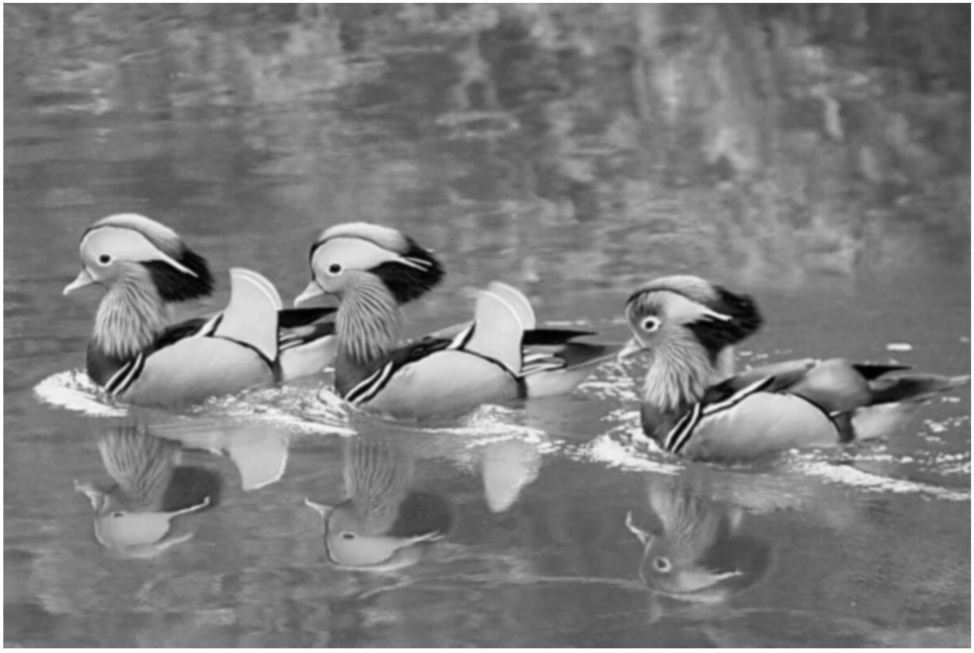
```
%HSI to RGB
for i = 1:800
    for j = 1:1200
        if((0 <= H_sharpened(i, j)) & (H_sharpened(i, j)< 2*pi/3))
            HSI2RGB_B(i, j) = I_sharpened(i, j).*(1 - S_sharpened(i, j));
            HSI2RGB_R(i, j) = I_sharpened(i, j).*(1 + S_sharpened(i, j).*cos(H_sharpened(i, j))./cos(pi/3-H_sharpened(i, j)));
            HSI2RGB_G(i, j) = 3*I_sharpened(i, j) - (HSI2RGB_R(i, j) + HSI2RGB_B(i, j));

        elseif((2*pi/3 <= H_sharpened(i, j)) & (H_sharpened(i, j) < 4*pi/3))
            H_sharpened(i, j) = H_sharpened(i, j) - 2*pi/3;
            HSI2RGB_R(i, j) = I_sharpened(i, j).*(1 - S_sharpened(i, j));
            HSI2RGB_G(i, j) = I_sharpened(i, j).*(1 + S_sharpened(i, j).*cos(H_sharpened(i, j))./cos(pi/3-H_sharpened(i, j)));
            HSI2RGB_B(i, j) = 3*I_sharpened(i, j) - (HSI2RGB_R(i, j) + HSI2RGB_G(i, j));

        else
            H_sharpened(i, j) = H_sharpened(i, j) - 4*pi/3;
            HSI2RGB_G(i, j) = I_sharpened(i, j).*(1 - S_sharpened(i, j));
            HSI2RGB_B(i, j) = I_sharpened(i, j).*(1 + S_sharpened(i, j).*cos(H_sharpened(i, j))./cos(pi/3-H_sharpened(i, j)));
            HSI2RGB_R(i, j) = 3*I_sharpened(i, j) - (HSI2RGB_G(i, j) + HSI2RGB_B(i, j));

        end
    end
end
image_HSI_sharpened(:,:,1) = (HSI2RGB_R);
image_HSI_sharpened(:,:,2) = (HSI2RGB_G);
image_HSI_sharpened(:,:,3) = (HSI2RGB_B);
figure, imshow((uint8(255*image_HSI_sharpened))), title('Sharpened HSI image');

```



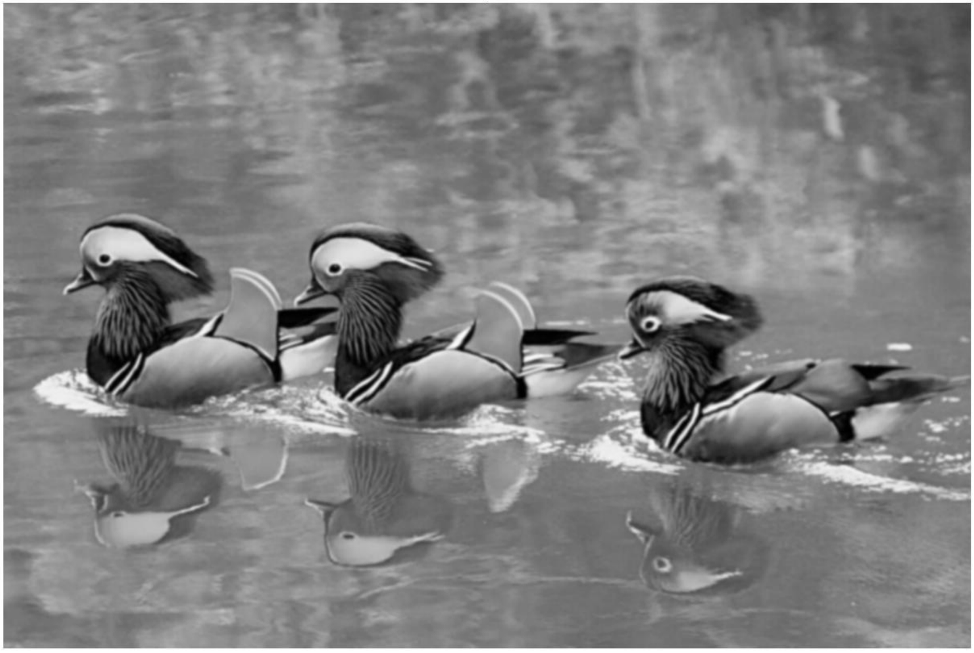
```
%evaluate differnce of RGB-based and HSI-based
difference = abs(image_RGB_sharpened - image_HSI_sharpened);
diff_image = (difference(:,:,1) + difference(:,:,2) + difference(:,:,3));
imshow(uint8(255*diff_image)), title('Difference');

```

2. Figures of R, G, B, H, S, I component images

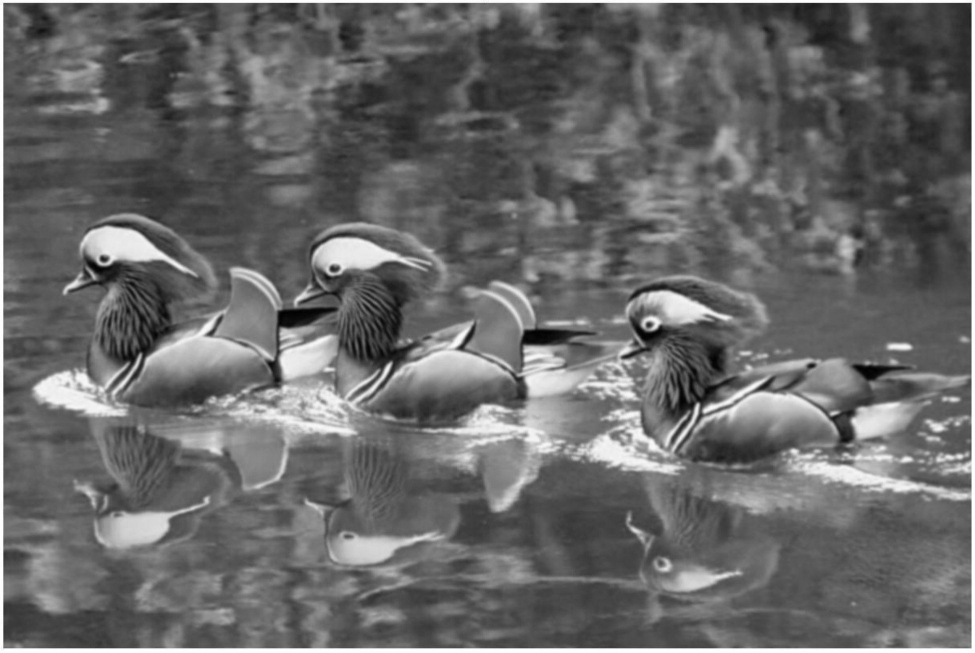
R component



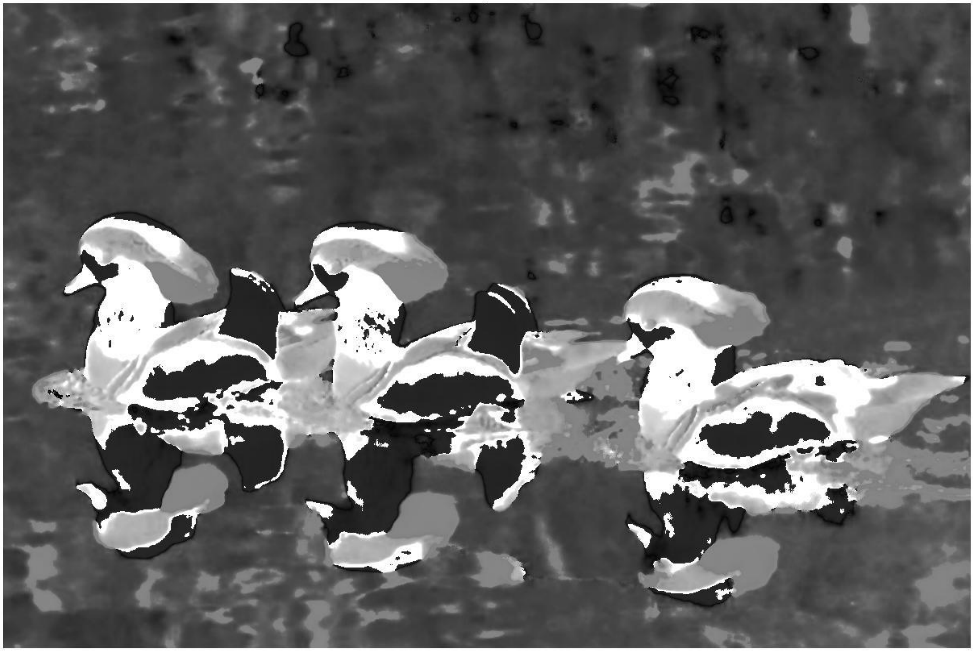
G component



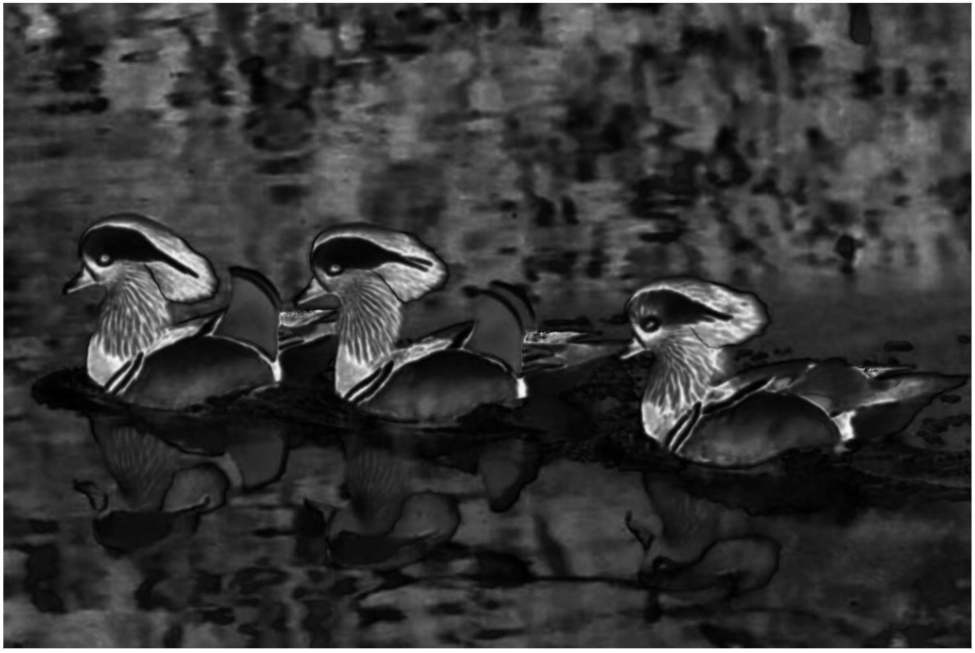
B component



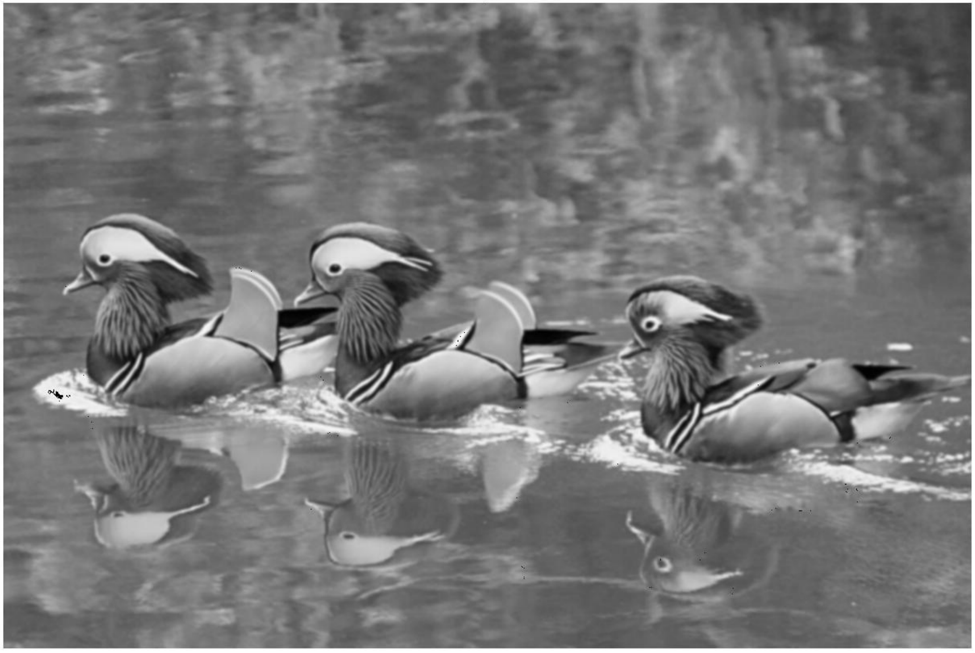
H component



S component

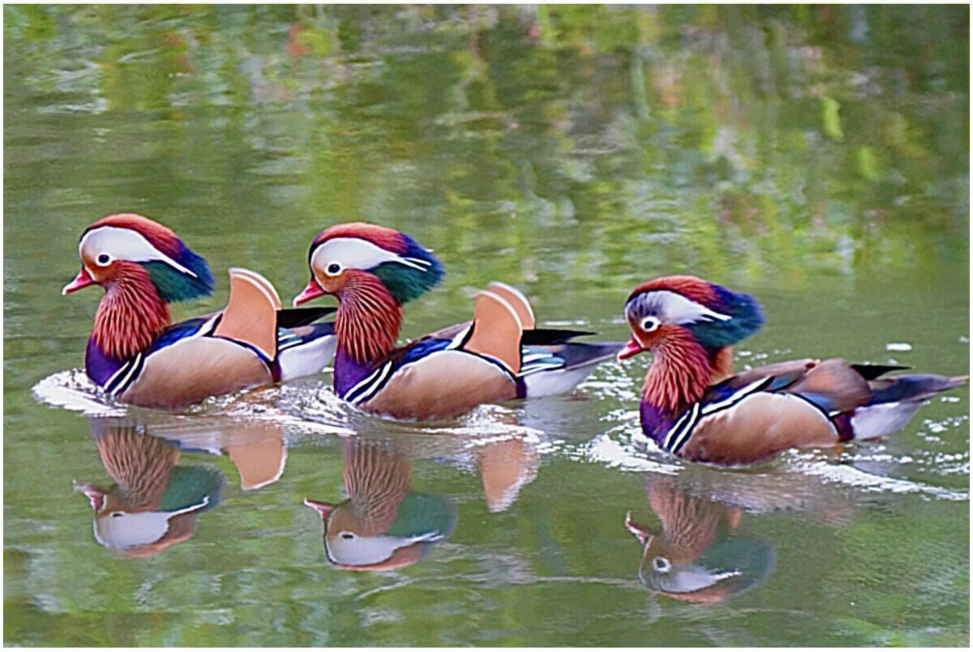


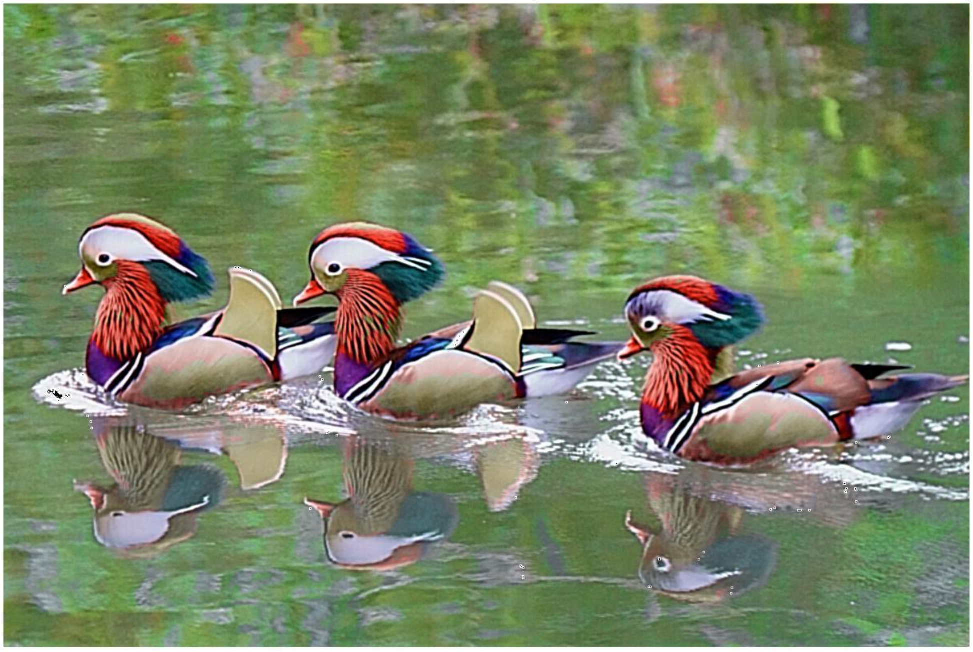
I component

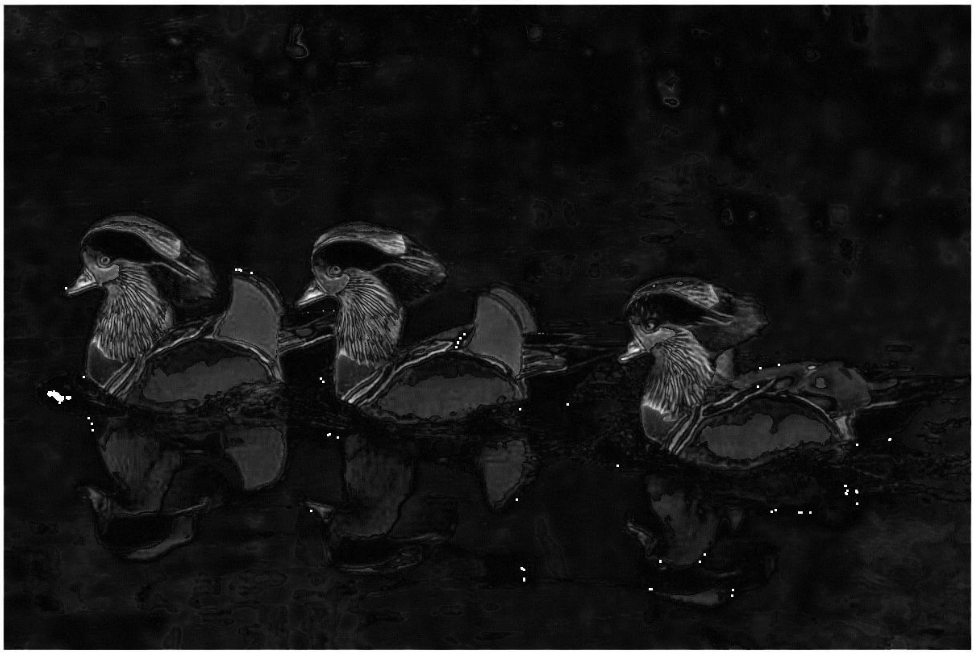


3. Figures of RGB-Based and HIS-based sharpened images and their difference image

Sharpened RGB image




Sharpened HSI image


Difference