# DIP Project 6

電信碩 沈衍薰 309513047

## 1. Source code

```
image = imread('fruit on tree.tif');

R = (image(:, :, 1));
G = (image(:, :, 2));
B = (image(:, :, 3));

x = size(image,1);
y = size(image,2);
```

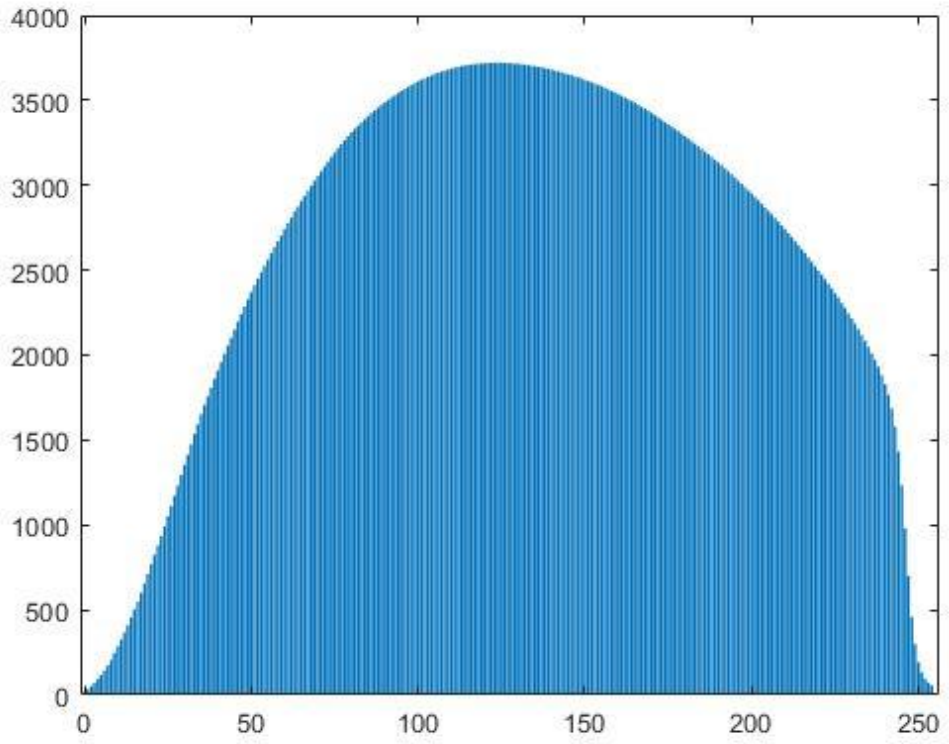
```
%% Otsu's method
intensity(1:256) = 0;
sigma_g_square(1:256) = 0;
for i = 0:255
    for j = 1:x
        for k = 1:y
            if (R(j, k) == i)
                intensity(i+1) =  intensity(i+1) + 1;
            end
        end
    end
end
```

```
for T = 0:255
    p = intensity(1:256)/(x*y);
    P1 = 0;
    P2 = 0;
    for i = 0:T
        P1 = P1 + p(i+1);
    end
    P2 = 1-P1;

    m1 = 0;
    m2 = 0;
    for i = 0:T
        m1 = m1 + i*p(i+1);
    end
    m1 = m1./P1;
    for i = T+1:255
        m2 = m2 + i*p(i+1);
    end
    m2 = m2./P2;
    mg = P1*m1 + P2*m2;

    for i = 0:255
        sigma_g_square(T+1) = sigma_g_square(T+1) + ((i-mg)^2)*p(i+1);
    end

    sigma_b_square(T+1) = P1*((m1-mg)^2) + P2*((m2-mg)^2);
end
figure, plot([0:255], sigma_b_square);
[M, k] = max(sigma_b_square);
A = sigma_b_square(k)./sigma_g_square(k);
```

```
image_after_Otsu=zeros(size(image));
for i = 1:x
    for j = 1:y
        if (R(i, j) > k)
            image_after_Otsu(i, j, 1) = image(i, j, 1);
            image_after_Otsu(i, j, 2) = image(i, j, 2);
            image_after_Otsu(i, j, 3) = image(i, j, 3);
        else
            image_after_Otsu(i, j, 1) = 0.5*255;
            image_after_Otsu(i, j, 2) = 0.5*255;
            image_after_Otsu(i, j, 3) = 0.5*255;
        end
    end
end

figure, imshow(uint8(image_after_Otsu)), title("Image After Otsu's Method");
```

```
%% K-maans
for threshold=[1,5,10]
    L = zeros(size(image));
    [L(:, :, 1),C1] = imsegkmeans(R,2, 'Threshold', threshold);
    [L(:, :, 2),C2] = imsegkmeans(G,2, 'Threshold', threshold);
    [L(:, :, 3),C3] = imsegkmeans(B,2, 'Threshold', threshold);


    for i = 1:x
        for j = 1:y
            if (L(i, j, 1) == 2)
                L(i, j, 1) = R(i, j);
            else
                L(i, j, 1) = 0.5*255;
            end

            if (L(i, j, 2) == 2)
                L(i, j, 2) = G(i, j);
            else
                L(i, j, 2) = 0.5*255;
            end

            if(L(i, j, 3) == 2)
                L(i, j, 3) = B(i, j);
            else
                L(i, j, 3) = 0.5*255;
            end
        end
    end
    s = sprintf("k-means Clustering with Threshold = %d",threshold);
    figure, imshow(uint8(L)), title(s);
end

```

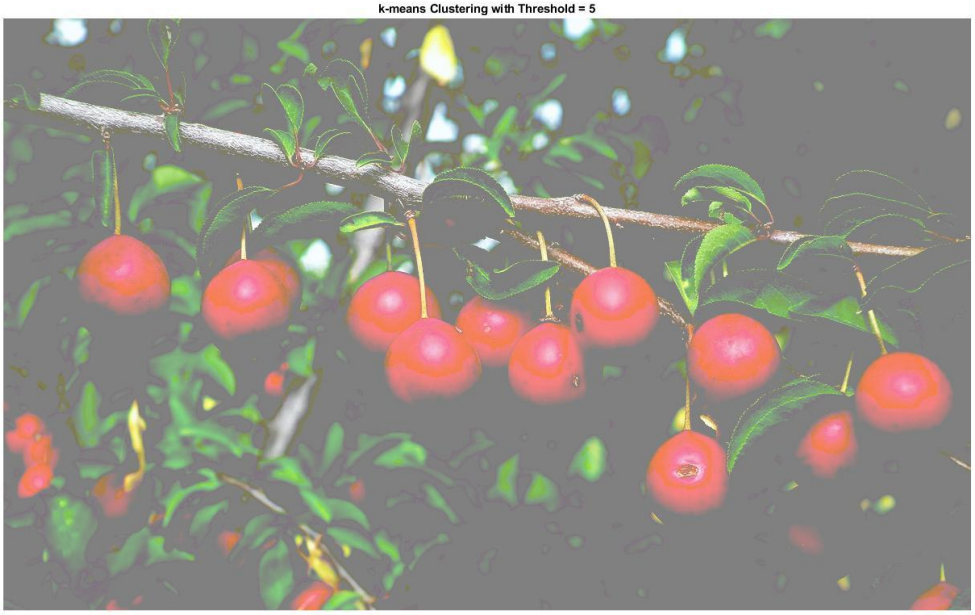
## 2. Plot of the curve of between-class variance depending on all possible threshold values



3. **Image of patterns extracted by Otsu's algorithm**


4. **Images of patterns extracted by K-means clustering with different threshold values**

k-means Clustering with Threshold = 5



k-means Clustering with Threshold = 10