

# **CMP\_SC 8001 Biomedical Image Processing**

## **Assignment 2 Part 1**

### **Understanding Kmeans clustering**

**Dewi Kharismawati**

**14231619 / dek8v5**

In this assignment, we are going to observe the use of kmeans in biomedical image processing by segmenting of individual cells and nuclei. So far, segmenting individual cell or nuclei is important, yet so challenging. Segmentation cell is useful to analyze all cell attributes, such as cell behavior, mortality, distribution, and morphology, which include shape, structure, color, and texture. Therefore, by this assignment we will understand and explore different steps used in cell and nuclei segmentation: preprocessing and segmentation using kmeans.

### Task 1 preprocessing

Smoothing the image and remove the noise using linear or non-linear filter. I am using Gaussian Linear Filter.

```
%read the image
%Img = imread('lungCT.jpg');
Img = imread('3TS0007.tif');

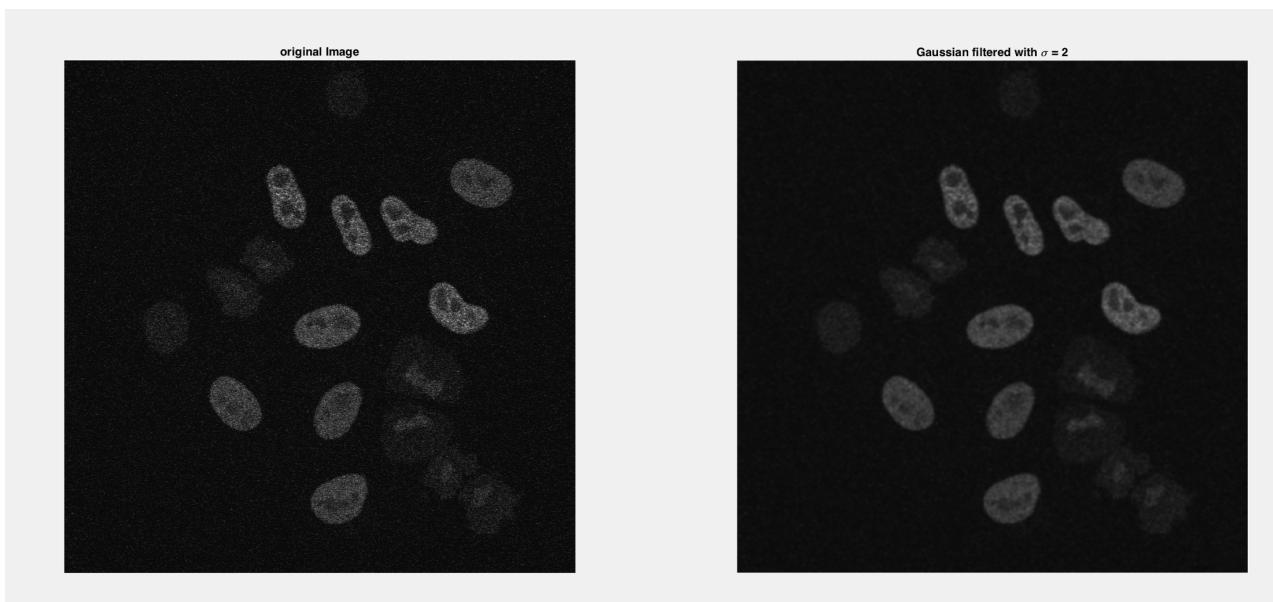
%display the original image
figure,
subplot(1,2,1);
imshow(Img);
title('original Image');

%apply gaussian filter with standard derivation = 3
sigma = 2;
Img = double(Img);

Img = imgaussfilt(Img, sigma);

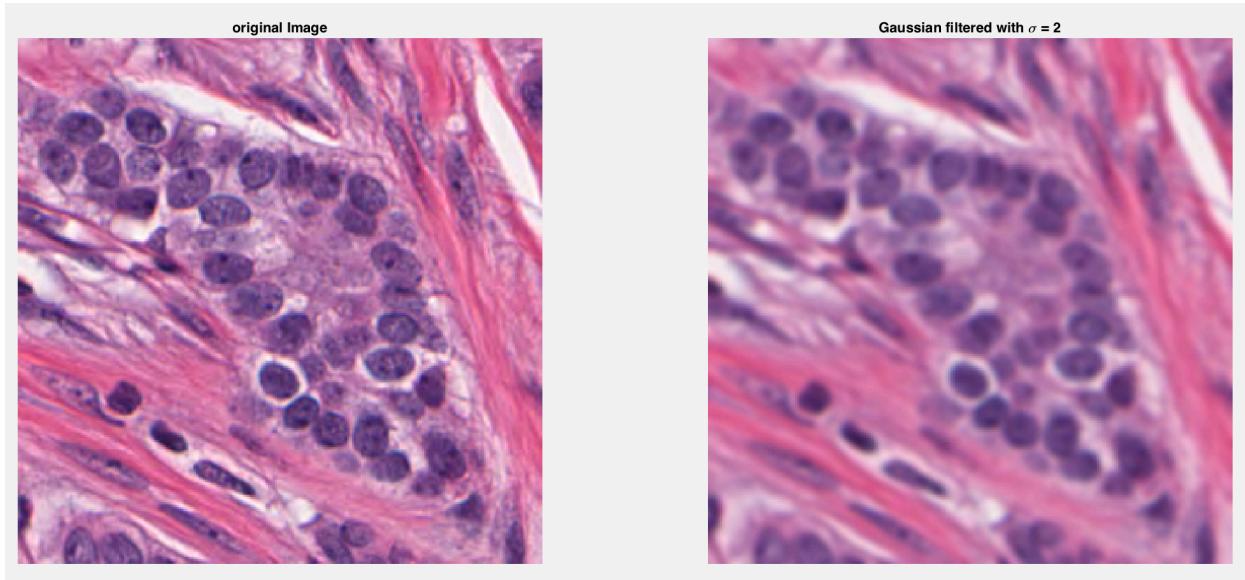
subplot(1,2,2);
imshow(uint8(Img));
title(['Gaussian filtered with \sigma = ' num2str(sigma)]);
```

The result is shown as follow:



With  $\sigma = 2$ , the image smoothed nicely. Great amount of noise is disappeared.

For RGB image, here is the result we get



with sigma = 2, the image is smoothed well.

### Task 2 cell/nuclei segmentation using Kmeans.

Kmeans is one technique to cluster and segment images. Kmeans clustering is dividing data into some classes cluster, where each data belongs to the cluster with the nearest mean. In this part, we are not going to write implementation of kmeans from scratch, whereas implementing the build in kmeans function from Matlab.

Matlab kmeans required parameter of Image matrix in  $n \times p$  size and number of clusters  $k$  that we want. The kmeans function will take the input  $n \times p$  (where  $n$  is number of data points and  $p$  is feature vector), and will partitioned them into  $k$  cluster. In order to use this matlab function, we need to alter our matrix in a way it fulfilled the requirement. We can use reshape function to do so. In color image case, we need to reshape our matrix of  $[row \times col \times channels]$  into  $[n \times p]$ , which  $[(row * col) \text{-by-} channel]$ . With this format of input, we will get  $[n \times 1]$  output. To display the result, we need to reshape back the size into original  $[row \times col]$  size.

Here is an example line to implement kmeans function. Where  $\text{Img}$  is already been reshaped into  $[n \times p]$ , and I want to clutered them into 4 classes ( $k=4$ ).

```
clustered = kmeans(Img(:, 4));
```

## Dewi Kharismawati / Assignment 2 part I

CASE 1: Use intensity as the feature. So, the feature = 1 here. If the image is RGB, we need to convert them into grayscale image first.

```
%task 1, use gaussian linear filtering to smooth and reduce noise

%read the image
Img = imread('lungCT.jpg');
%Img = imread('3TS0007.tif');
%Img = imread('BreastCacer_img_1127.tif');
%Img = imread('malaria_cropped2.tif');

[row, col, channel] = size(Img);
if channel > 1
    Img = rgb2gray(Img);
end

%display the original image
figure,
subplot(2,3,1);
imshow(Img);
title('original Image');

%apply gaussian filter with standard derivation = 3
sigma = 2;
Img = double(Img);

Img = imgaussfilt(Img, sigma);

subplot(2,3,2);
imshow(uint8(Img));
title(['Gaussian filtered with \sigma = ' num2str(sigma)]);

%task2 use kmeans clustering method to segment the nuclei from background
%reshape the matrix
Img = reshape(Img, [row*col, 1]);

clustered = kmeans(Img(:, 1), 4);

%reshape the result to row by col
clustered = reshape(clustered, [row, col]);

%plot the result
subplot(2,3,3)
imshow(clustered==1);
title('Img cluster = 1');

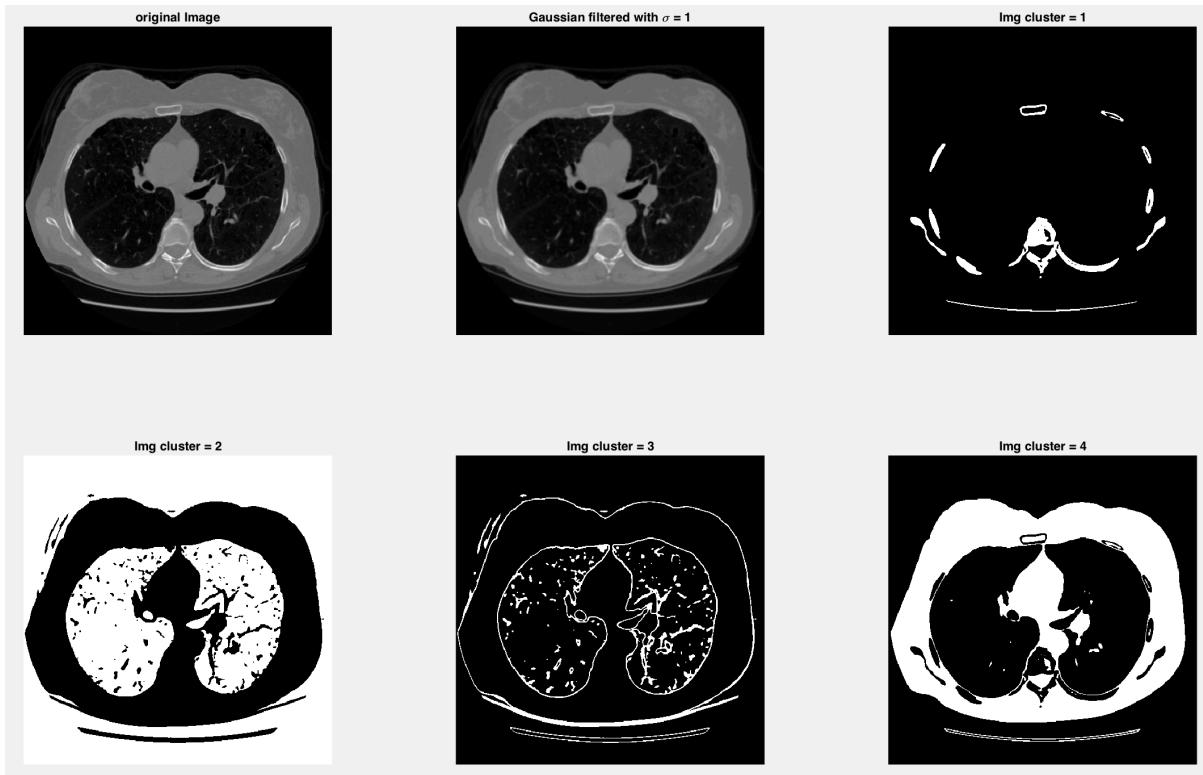
subplot(2,3,4)
imshow(clustered==2);
title('Img cluster = 2');

subplot(2,3,5)
imshow(clustered==3);
title('Img cluster = 3');

subplot(2,3,6)
imshow(clustered==4);
title('Img cluster = 4');
```

## Result and Discussion

### 1. Result for lung CT

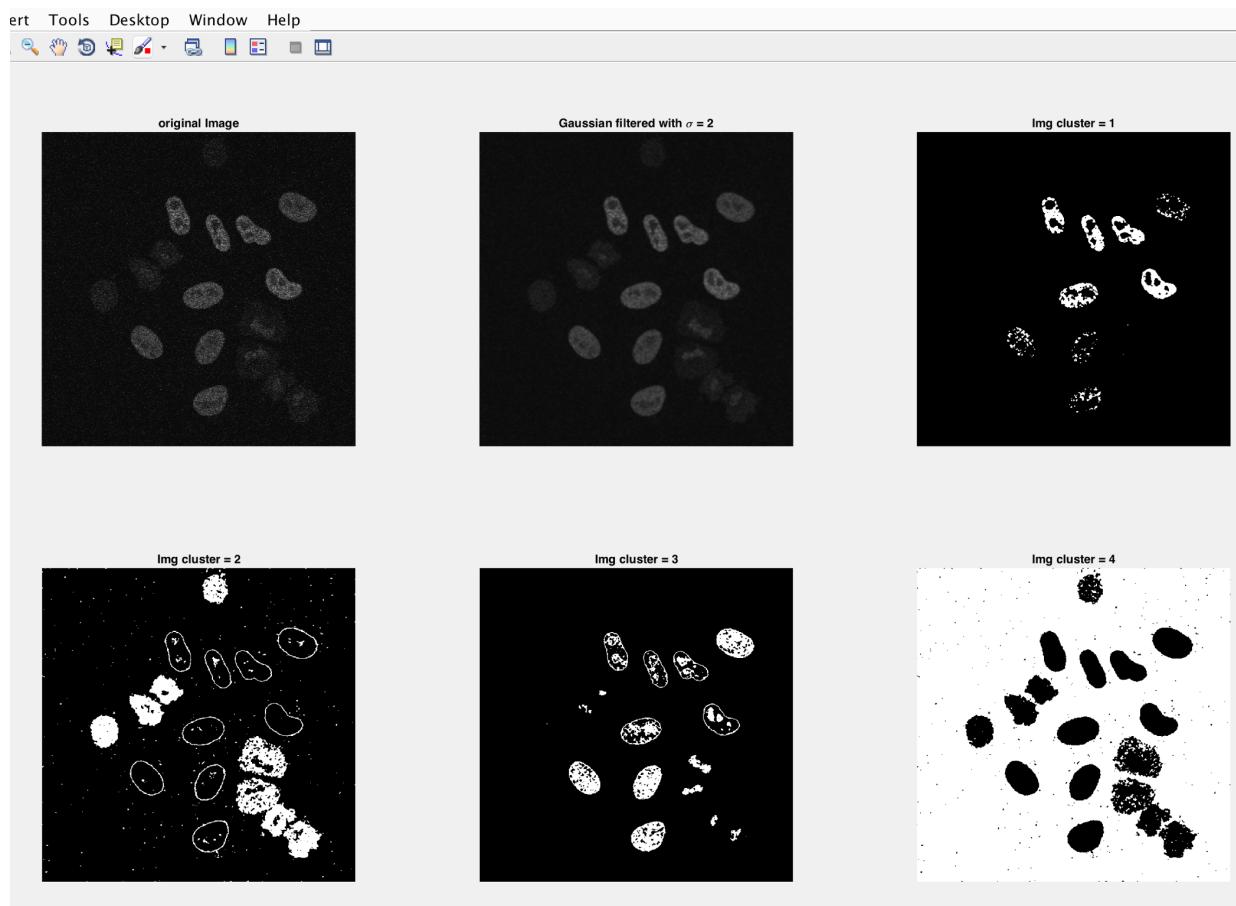


After sometimes reviewing the result, I figured that in each cluster, white area represents point of interest, while black represent the background. From 4 clusters that I tried to implement, each clustered is partitioned based on 4 intensity group, where each data belongs to the cluster with the nearest mean.

In this case, our point of interest is the lungs. We want to segment the lungs out from everything else. By the cluster image, there are two possibility that we can use, which are the cluster = 2 and cluster = 3.

In cluster = 2, the object of interest is with low intensity or near to black. As we can see, the lung area in that CT image is black or darkest. Therefore, this area is able to segment both lungs area and the actual image background. In cluster = 3, the object of interest is grayish. It provides us with some edges that divides between lungs and outer layer. So, we can observe lungs' shape. Also, it pointed the lungs' nerves as the object of interest as well. So, if we need to observe particles, cell, cancer, or nerve inside the lungs, cluster 3 is the best option to segment.

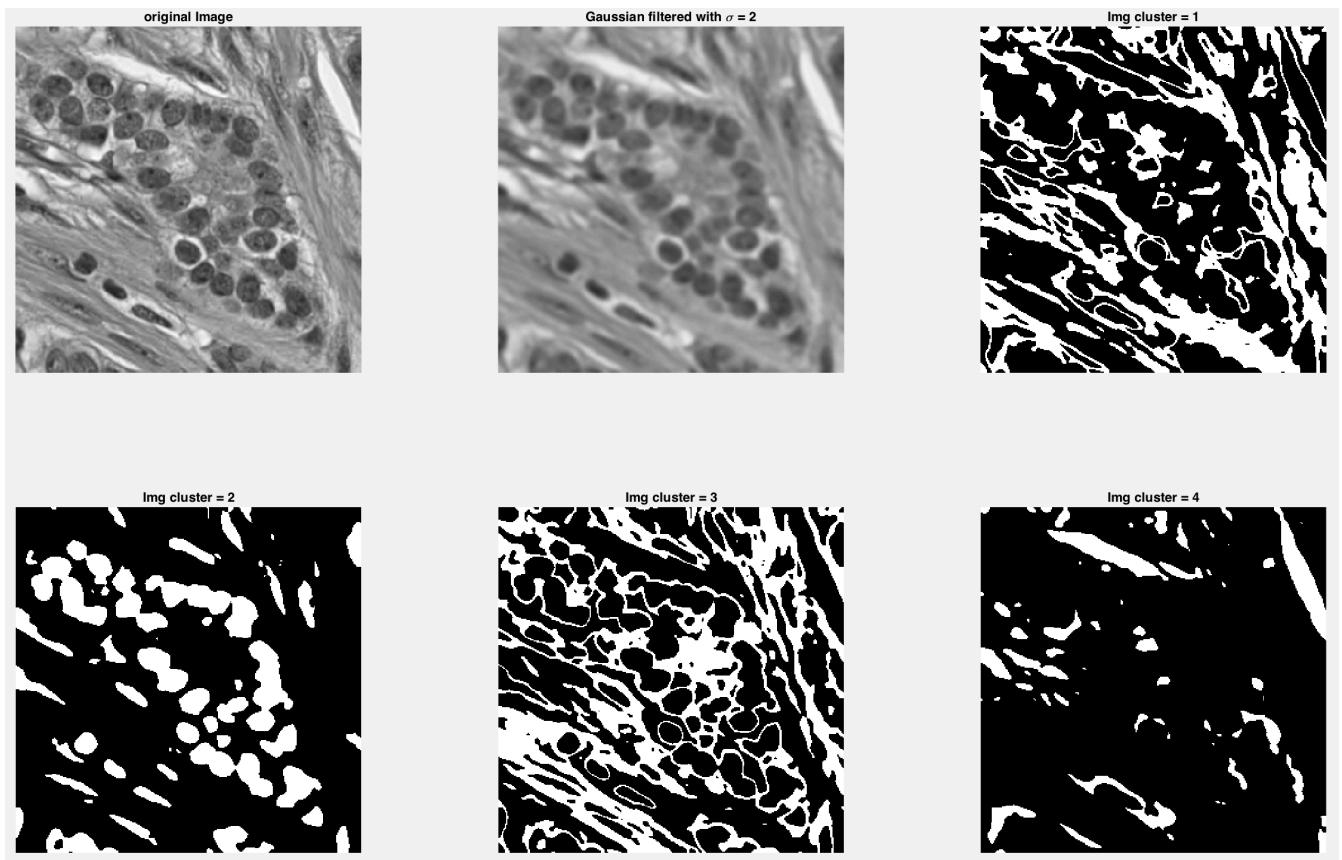
## 2. Result for Cell



After observing this image, 4 clusters are applied. I do not know what cell are this. But, based on the result above, we can see some of the cell attributes such as their shape, structure, nuclei, and distribution.

With cluster 2, we can see the shape of the cell pretty clear. As in some cells, it gave us edges. In cluster 2 and 3, we can see significant nuclei in some cells, especially in cells that have higher intensity, and the nuclei is clearly has a different level of intensity. In cluster 4, it shown that it separates between background and the cells, even though noises are still be detected. If we use information in cluster = 4, we can count the exact number of cells, and maybe the distribution of them in the image.

### 3. Breast Cancer Cells Image



In this part, we tried to run RGB image into kmeans with intensity value. So, we pushed the RGB image to be converted into grayscale. But, the result is not as expected.

The nearest breast cancer cell is segmented in cluster 2. Cluster 2 is segmenting the cancer cells. It is pretty good, but there are some really close to each other cells and some overlapped cells are not segmented correctly. They are treated as one cell.

By this observation, I conclude that we cannot implement RGB images into kmeans with intensity feature. If we convert the RGB image into grayscale, we will lose a lot of information brought by each channel of the image.

Therefore, the next section will discuss about treating RGB image with kmeans using channels Red, Green, Blue in the image. This will give a better result for color image.

## Dewi Kharismawati / Assignment 2 part I

CASE 2: use RGB color channel as the features. In color image, we will have 3 features, which are Red channel, Green channel, and Blue channel.

In RGB images, we will use its channels as the kmeans feature. In this case, I am going to use 3 features, which came from image color channel, Red, Green, and Blue.

The process of reshaping the image is similar to what we did in grayscale image. In color image case, we need to reshape our matrix of [row×col×channels] into [n×p], which [(row\*col)-by-channel]. With this format of input, we will get [n×1] output. To display the result, we need to reshape back the size into original [row×col×channels] size.

Code bellow is the implementation in Matlab.

```
%read the image
Img = imread('BreastCacer_img_1127.tif');
%Img = imread('malaria_cropped2.tif');

%display the original image
figure,
subplot(1,2,1);
imshow(Img);
title('original Image');

%apply gaussian filter with standard derivation = 3
sigma = 1;
Img = double(Img);
Img = imgaussfilt(Img, sigma);

subplot(1,2,2);
imshow(uint8(Img));
title(['Gaussian filtered with \sigma = ' num2str(sigma)]);

%reshape the image matrix to be n×p
%task2 use kmeans clustering method to segment the nuclei from background
[row, col, channel] = size(Img);

%reshape the matrix
Img = reshape(Img, [row*col, channel]);

clustered = kmeans(Img(:, ), 4);

%reshape the result to row by col
clustered = reshape(clustered, [row, col, channel]);

%plot the result of channel 1
figure,
subplot(2,2,1)
imshow(clustered(:,:1)==1);
title('Img channel 1 (Red); cluster=1');

subplot(2,2,2)
imshow(clustered(:,:1)==2);
title('Img channel 1 (Red); cluster=2');

subplot(2,2,3)
imshow(clustered(:,:1)==3);
title('Img channel 1 (Red); cluster=3');

subplot(2,2,4)
imshow(clustered(:,:1)==4);
title('Img channel 1 (Red); cluster=4');

%plot the result of channel 2
figure,
subplot(2,2,1)
imshow(clustered(:,:2)==1);
title('Img channel 2 (Green); cluster=1');

subplot(2,2,2)
imshow(clustered(:,:2)==2);
title('Img channel 2 (Green); cluster=2');

subplot(2,2,3)
imshow(clustered(:,:2)==3);
```

# Dewi Kharismawati / Assignment 2 part I

```
title('Img channel 2 (Green); cluster=3');

subplot(2,2,4)
imshow(clustered(:,:2)==4);
title('Img channel 2 (Green); cluster=4');

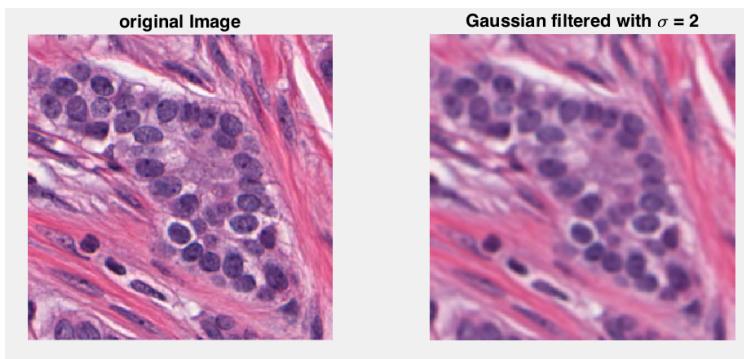
%plot the result of channel 3
figure,
subplot(2,2,1)
imshow(clustered(:,:3)==1);
title('Img channel 3 (Blue); cluster=1');

subplot(2,2,2)
imshow(clustered(:,:3)==2);
title('Img channel 3 (Blue); cluster=2');

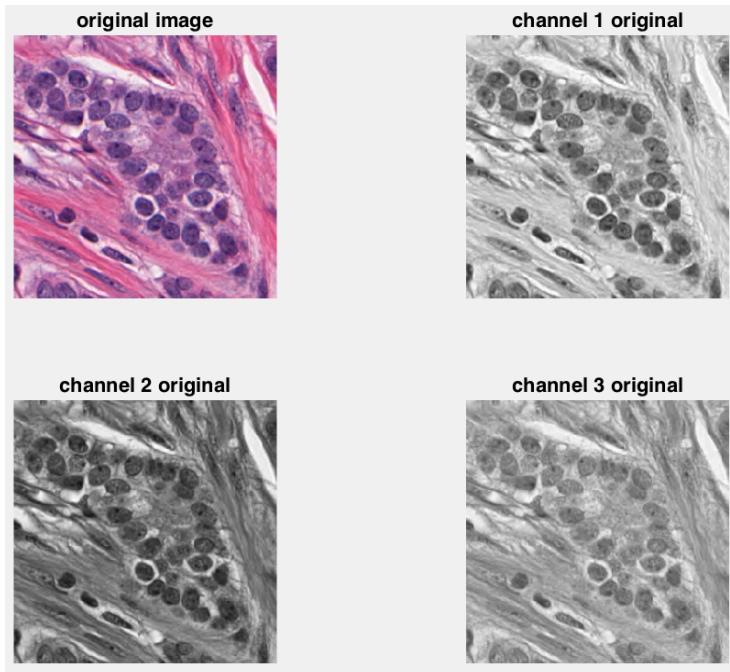
subplot(2,2,3)
imshow(clustered(:,:3)==3);
title('Img channel 3 (Blue); cluster=3');

subplot(2,2,4)
imshow(clustered(:,:3)==4);
title('Img channel 3 (Blue); cluster=4');
```

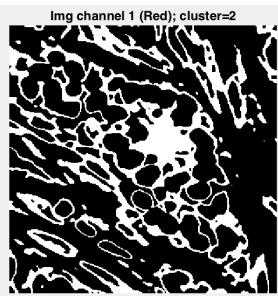
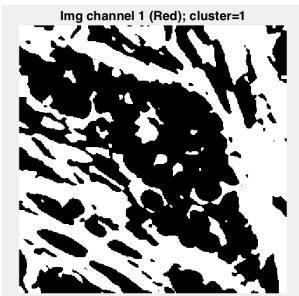
## Result 1:



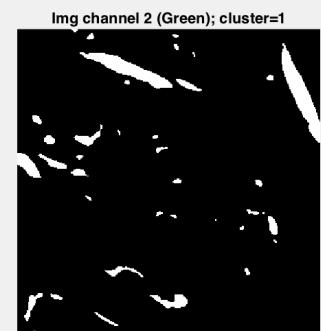
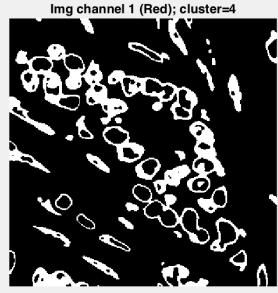
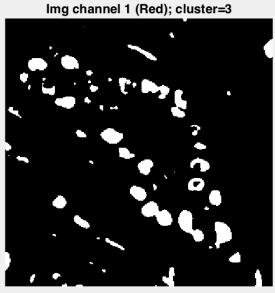
This image is the original image and the smoothed image with Gaussian filter with standard deviation = 2.



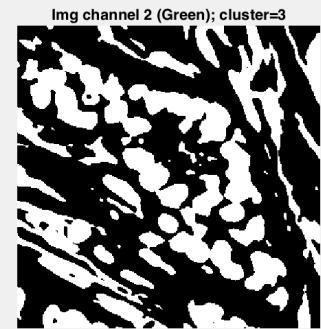
This is the original image and also original image in each channel.

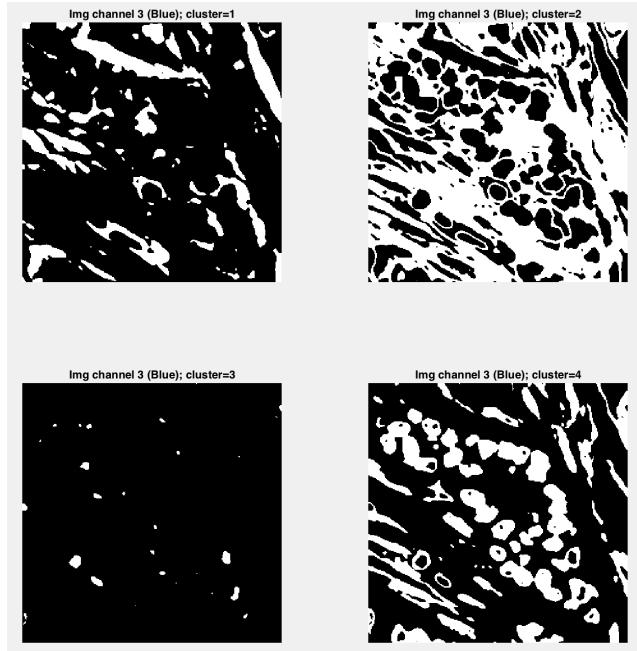


This is the result for channel 1 Red,  
and displayed all clusters.



This is the result for channel 2  
Green, and displayed all clusters.





This is the result for channel 3 Blue, and displayed all clusters.

The idea of getting the segmentation on breast cancer cell is hard to do with RGB channel feature. The best result we can get is from channel Red (1), with cluster 3 and 4. In cluster 3, the cells are separated well. Here, we can count round cells pretty well. In cluster 4, it segments some edges of the cells. It will help to define the shape of the cells.

The challenge from this case is that, clusters were not able to segment overlapped or attached to each other cells. Red channel cluster 3 almost able to solve this problem, but the segmented cells are smaller compared to the original. Here, the consequence is that we lose some information. But, we can retrieve from other channels and clusters.

Result 2. Malaria blood infected image

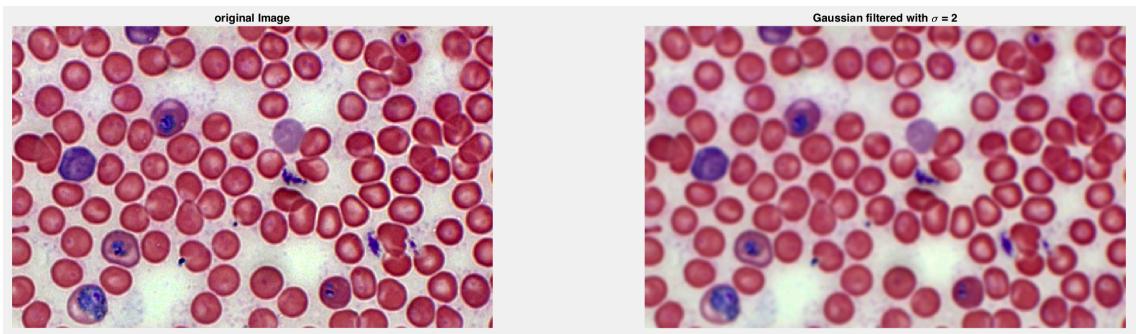
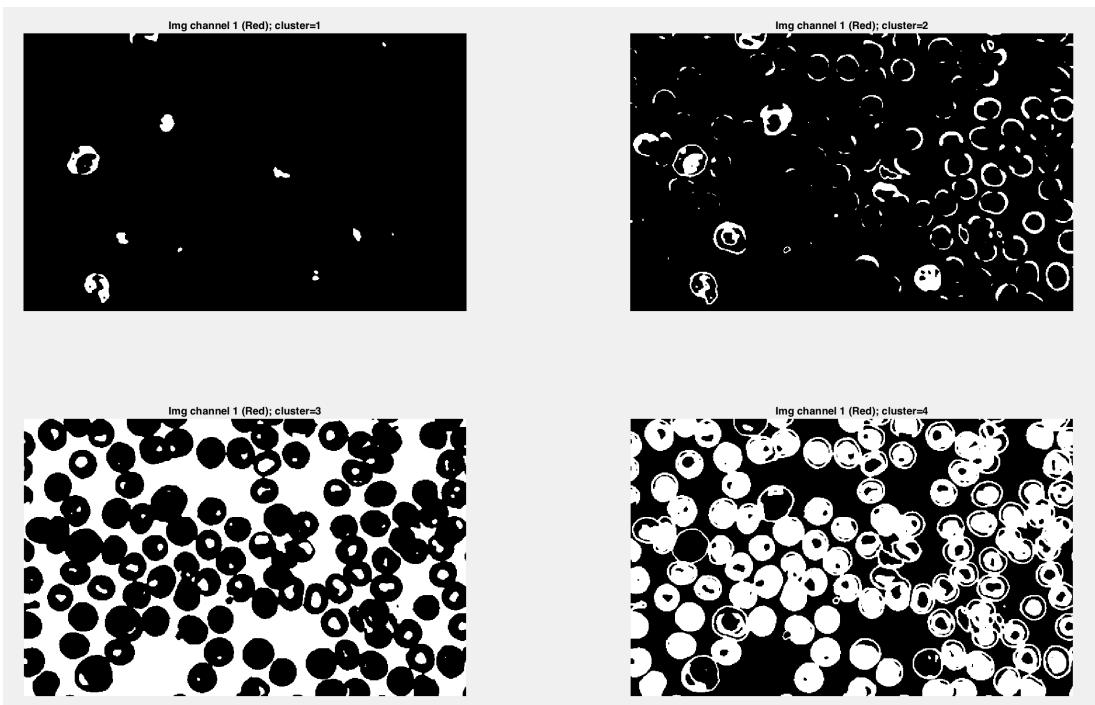


Figure 1 Original image and smoothed image for Malaria blood cells

In this malaria case, we want to segment cells that being infected by malaria virus. From the original microscopy image above, cells with blue is infected by malaria. Let us focus on segmenting those cells in this part.



Result of the red channel, and all clusters.

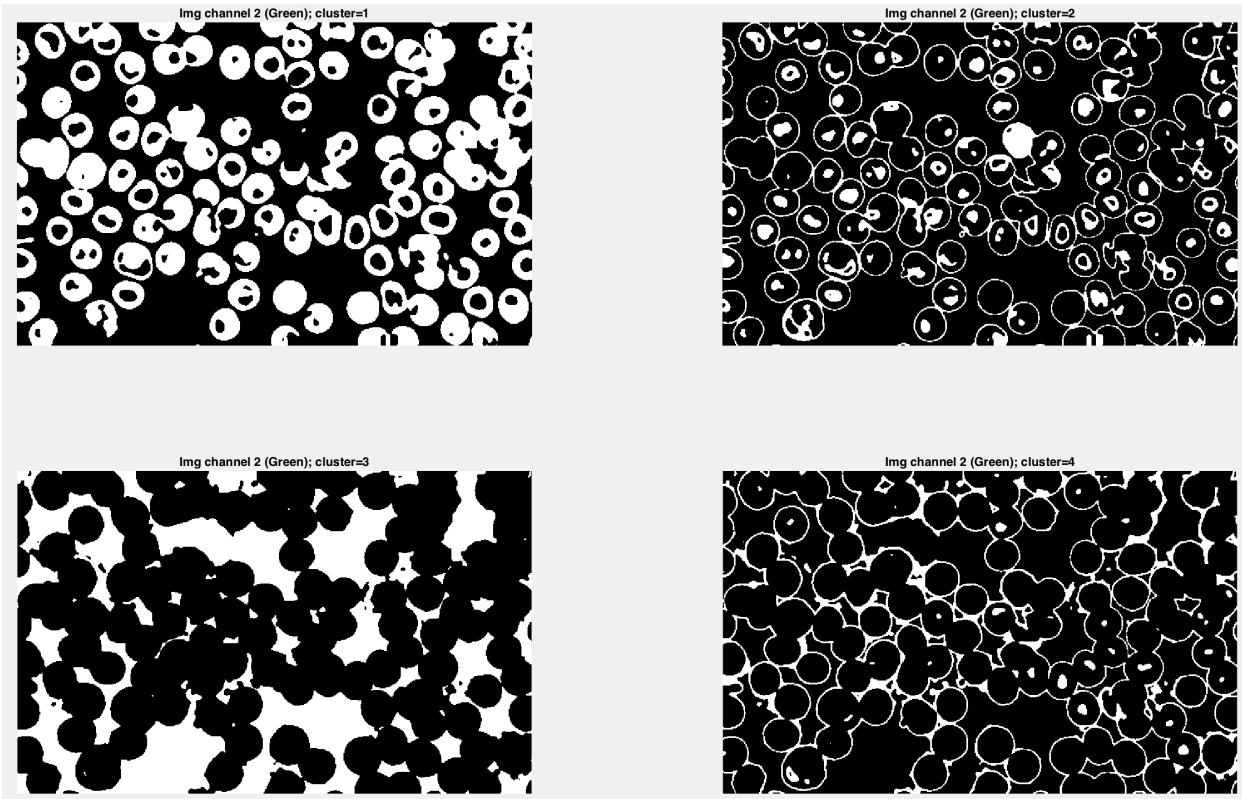


Figure 2 result in channel 2 and all the clusters

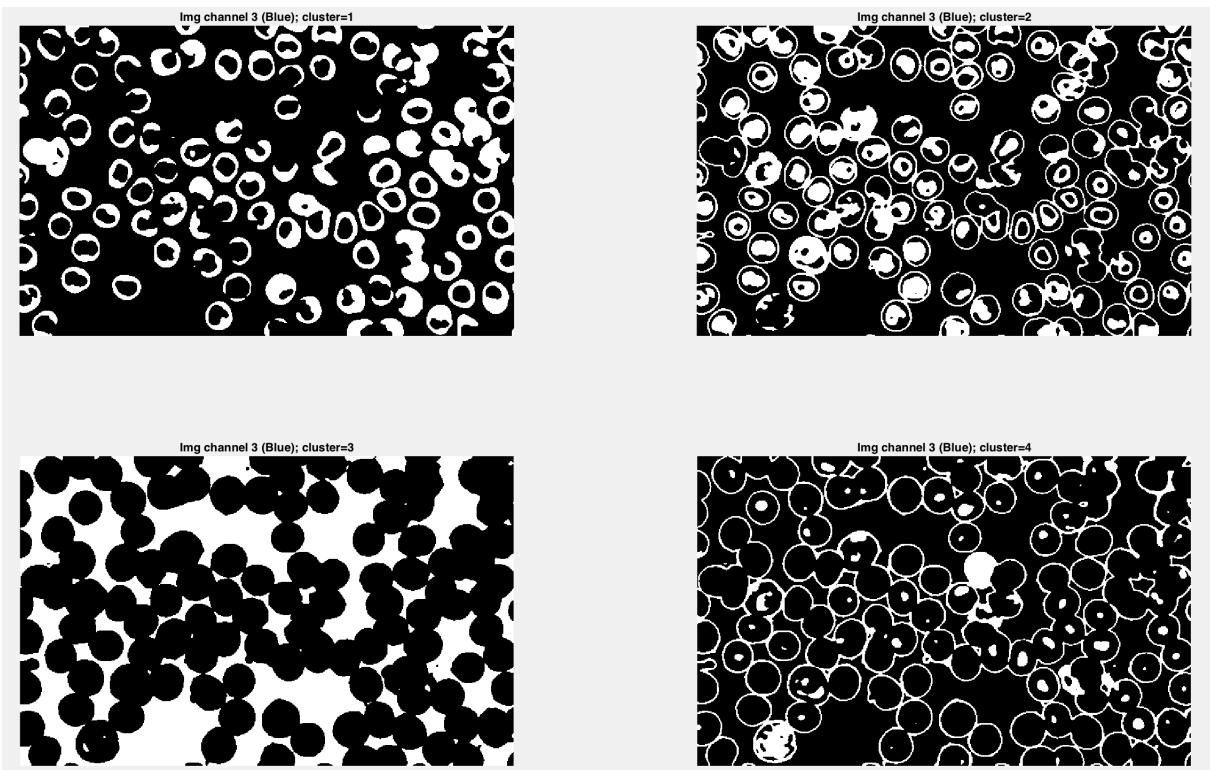


Figure 3 Result in channel 3 with all the clusters

## Malaria discussion

This malaria case is so much interesting. Our main goal is to find which cells is infected with malaria virus. From all the channels and clusters. Channel 1 and cluster 1 is specifically gave us information about which cell is infected by the malaria virus. From here, we can confirm that the blood is actually infected. Channel 1 cluster 1, provide segment for cell with blue color.

There are a lot more information can be retrieved from this result. For example, when we want to segment the nuclei of all cells, we can use several channels and clusters, such as channel 3 cluster 2, channel 2 cluster 2, and channel 1 cluster 3. To get the shape, we can use channel 1 cluster 4, channel 2 cluster 2 and 4, and channel 3 cluster 2 and 4. However, there is still some difficulties to segment separately between overlapped cells or attached each other.

## Segmentation using L\*a\*b based on Matlab tutorial

The segmentation is based on L is for lightness, a and b is for color component, a is green-red and b is blue-yellow.

Algorithm implemented:

```
clc;
clear all;
close all;

%read the image
Img = imread('BreastCacer_img_1127.tif');
figure,
subplot(2,3,1);
imshow(uint8(Img));
title('Original Image');

%apply cform
cform = makecform('srgb2lab');
lab_img = applycform(Img,cform);

%get the ab color
Img_ab = double(lab_img(:,:,2:3));

[row, col, channel] = size(Img_ab);

%reshape based on ab feature
Img_ab = reshape(Img_ab,row*col,2);

nColors = 3;
% repeat the clustering 3 times to avoid local minima
clustered = kmeans(Img_ab,nColors);

%reshape the clustered
clustered = reshape(clustered,row,col);
subplot(2,3,2);
imshow(clustered,[]), title('image clustered by index');

%segmenting the
img_segmented = cell(1,3);
rgb_label = repmat(clustered,[1 1 3]);
```

# Dewi Kharismawati / Assignment 2 part I

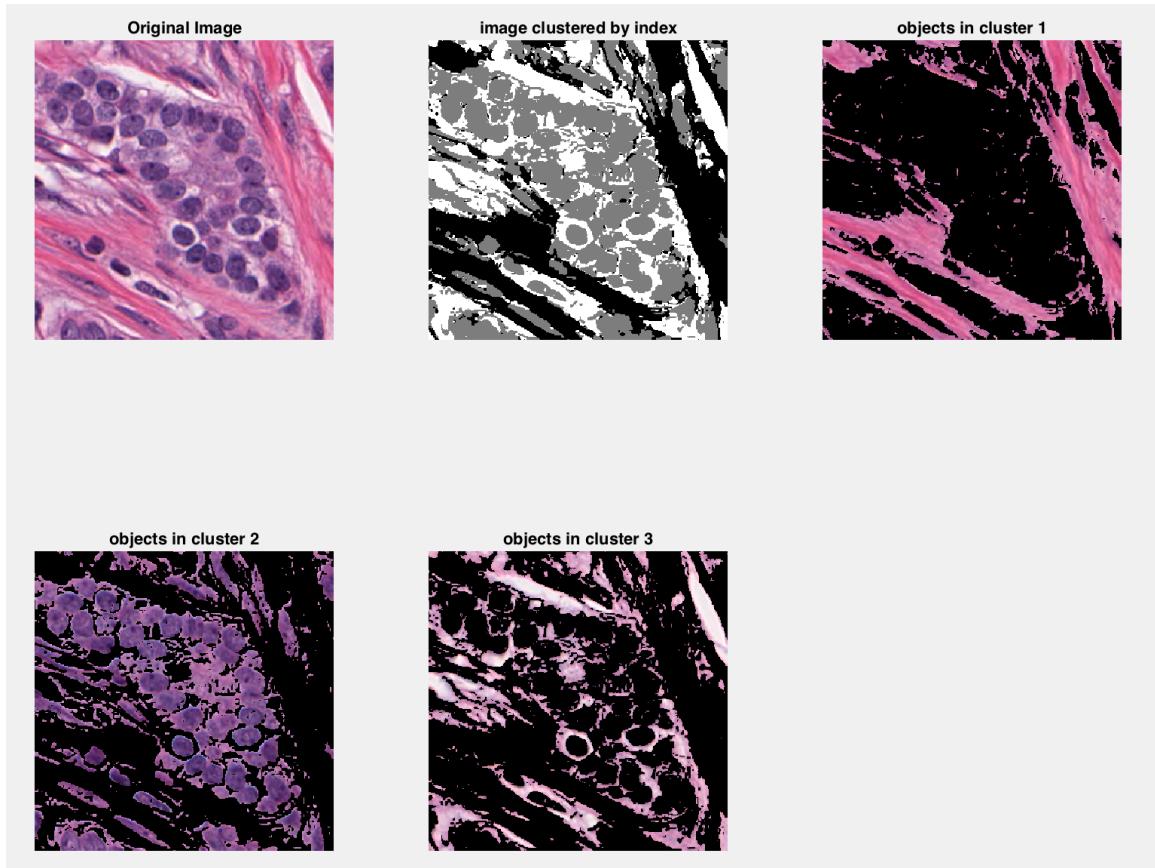
```
for k = 1:nColors
    color = Img;
    color(rgb_label ~= k) = 0;
    img_segmented{k} = color;
end

%plot the result
subplot(2,3,3);
imshow(img_segmented{1});
title('objects in cluster 1');

subplot(2,3,4);
imshow(img_segmented{2});
title('objects in cluster 2');

subplot(2,3,5);
imshow(img_segmented{3})
title('objects in cluster 3');
```

## Result



the result shows the segmentation based on color. The segmentation is similar to the previous cases, but in here we got colors. We can see that cluster 2 gave us the best segmentation where our object of interest is the breast cancer cell. The cluster 1 provides us with image in pink color, and this is representing the breast tissues. For cluster 3, it gave us info about the brighter area, and I assumed they are halos (honestly, I have no idea about this area)

## Discussion

We have observed the kmeans clustering with 2 different features: grayscale intensity, and RGB color channel. This segmentation by clustering method can give us many information and segmented point of interest. And it is up to us to decide which one to use based on our main goal of each case. Using kmeans clustering is easy and simple. We only use basic operation that includes finding means and group each data to each cluster means, based on the closeness relation. Kmeans cluster also can process multi-dimensional images. However, there are some drawbacks for this segmentation as well. We need to pick  $k$ , which force the user to do brute-force and try to find the best  $k$  to segment the image. Kmeans also have difficulties in individually segmenting overlapped and attached cells. Also, kmeans is sensitive to outlines like noise. Therefore, we need smoothing to do remove noise, which reduce the accuracy of the segmentation.

All in all, kmeans clustering is not a perfect technique to segment an image. There are some drawbacks that can reduce the accuracy of the result. Therefore, we still need to combine kmeans result with other technique to improve the result accuracy. We cannot trust kmeans alone to segment an image.