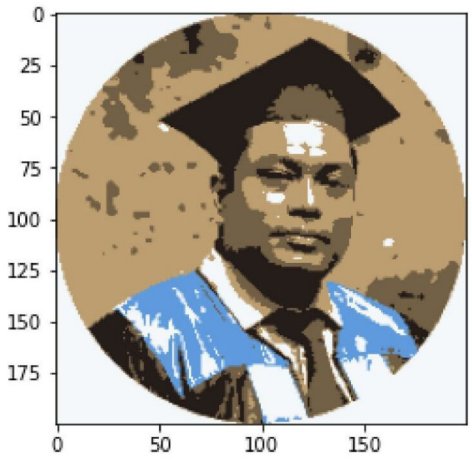


====Fuzzy C-Means Clustering with Color Image====#

In [14]:
```
'''Load the Required Packages'''
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from skimage import io
from ipywidgets import interact, widgets
from IPython.display import display
#!pip install opencv-python
import cv2 as cv
import skfuzzy as fuzz
from skimage.color import rgb2gray
```

In [15]:
```
# Load the image
image=cv.imread('E:/Village of Study/STAT - 811 IMAGE CLASSIFICATION/HABIB JU PROFILE IMAGE.jpg')
# Convert to RGB (if needed)
image = cv.cvtColor(image, cv.COLOR_BGR2RGB)
# Check the image shape of Color Image
print(image.shape)
c_pixels = image.reshape(-1, 3)
c_pixels
```

```
(200, 200, 3)
```

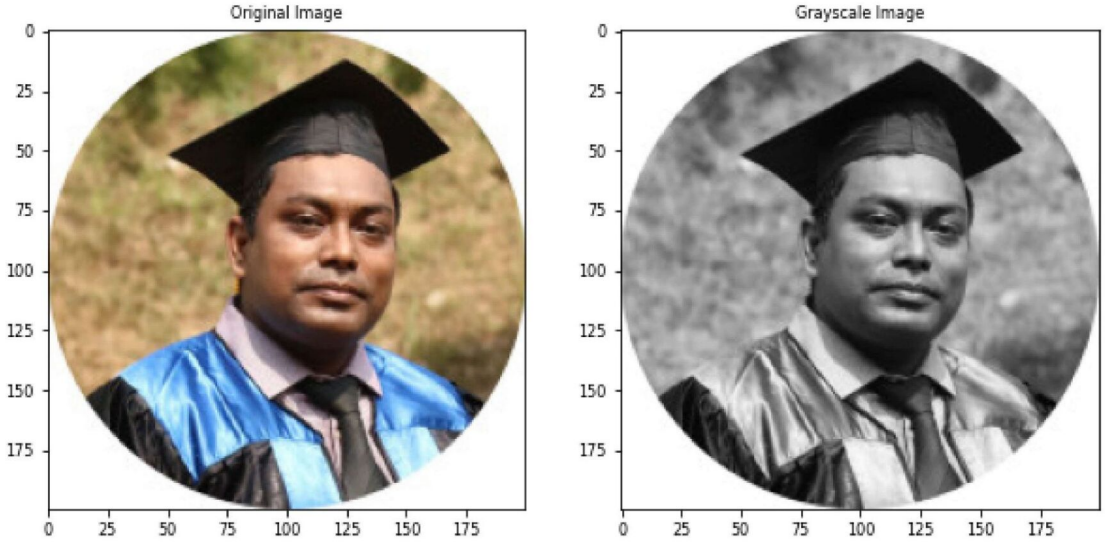
Out[15]:
```
array([[255, 255, 255],
       [255, 255, 255],
       [255, 255, 255],
       ...,
       [255, 255, 255],
       [255, 255, 255],
       [255, 255, 255]], dtype=uint8)
```

In [16]:
```
# Convert to grayscale
gray_image = rgb2gray(image)
print(gray_image.shape)
# Reshape the image into a 1D array
pixels = gray_image.reshape(-1, 1)
pixels
```

```
(200, 200)
```

Out[16]:
```
array([[1.],
       [1.],
       [1.],
       ...,
       [1.],
       [1.],
       [1.]])
```

In [17]:
```
# Display the original and grayscale images
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title('Original Image',fontsize=8)
plt.imshow(image)
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.subplot(1, 2, 2)
plt.title('Grayscale Image',fontsize=8)
plt.imshow(gray_image, cmap='gray')
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.savefig('D:/Fuzzy C-Means_Gray Image.jpg',dpi=700)
plt.show()
```



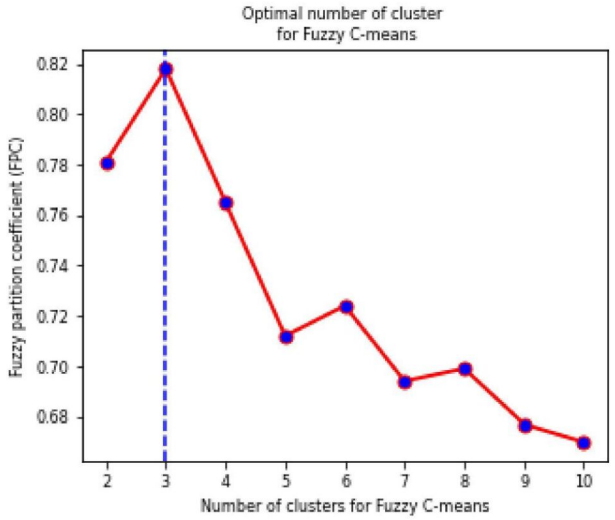
In [18]:
```
np.random.seed(104729)
FPC = []
K=[]
for k in range(2, 11):
    # Perform fuzzy c-means clustering
    cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(
        c_pixels.T, k, m=2, error=0.0005, maxiter=1000, init=None)
    FPC.append(fpc)
    K.append(k)

K
FPC
```

Out[18]:
```
[0.781473149934885,
 0.8181218851340527,
 0.7654658171267089,
 0.7116250731509263,
 0.7237632394997168,
 0.6943915720701362,
 0.6994479744924967,
 0.6770309621987213,
 0.6700146314558745]
```

In [19]:
```
#==Optimal Value of k for Fuzzy C-means Clustering===#
k=pd.DataFrame(FPC).idxmax()+2
k
k[0]
```

Out[19]:
```
3
```

In [20]:
```
plt.figure(figsize = (5, 4))
plt.plot(range(2, 11), np.round(FPC,3), ls='-',
         linewidth=2, color='red',marker='o',
         markerfacecolor='blue', markersize=7)
plt.axvline(x=k[0], color='b',ls='--',linewidth=1.5)
#plt.axhline(y=78, color='b',ls='--',linewidth=2)
plt.title('Optimal number of cluster \n for Fuzzy C-means',fontsize=8)
plt.xlabel('Number of clusters for Fuzzy C-means',fontsize=8)
plt.ylabel('Fuzzy partition coefficient (FPC)',fontsize=8)
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.savefig('D:/Fuzzy C-Means_Optimal number of Cluster for Image.jpg',dpi=700)
plt.show()
```

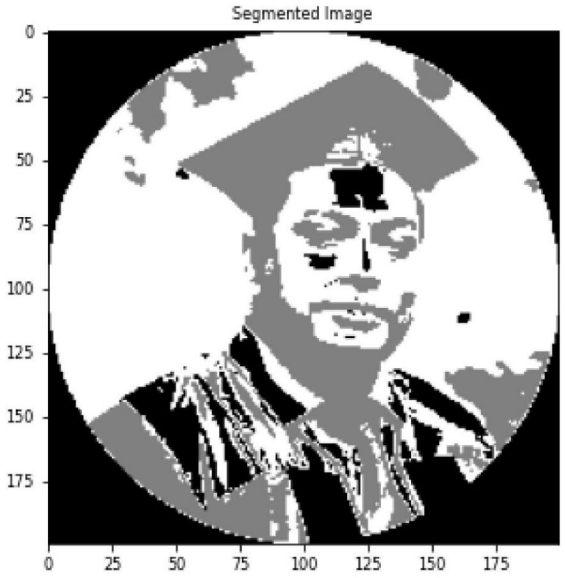


In [21]:
```
# Define the number of clusters
n_clusters = k[0]

# Apply Fuzzy C-Means clustering


cntr, u, u0, d, jm, p, fpc= fuzz.cluster.cmeans(c_pixels.T, n_clusters, 2,
                                 error=0.005, maxiter=1000, init=None)
```

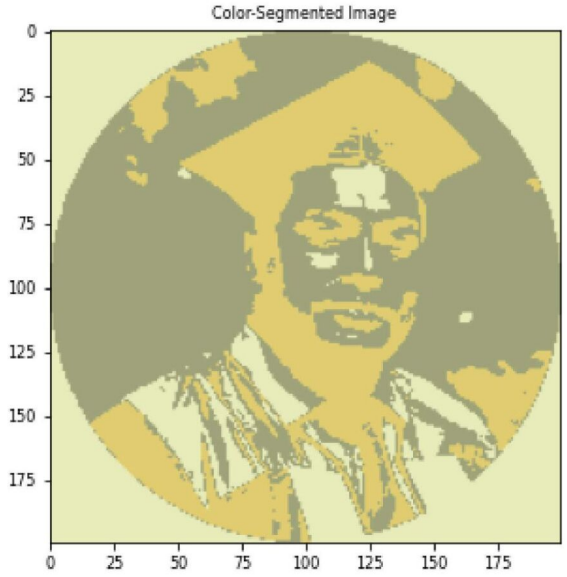
In [22]:
```
cntr
u
u0
d
jm
p
fpc
```

Out[22]:
```
0.8181219809132241
```

In [23]:
```
# Get cluster membership for each pixel
cluster_membership = np.argmax(u, axis=0)

# Reshape the clustered data back to the original image shape
segmented_image = cluster_membership.reshape(gray_image.shape)

plt.figure(figsize=(5, 5))
plt.imshow(segmented_image, cmap='gray')
plt.title('Segmented Image',fontsize=8)
plt.axis()
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
#plt.axis('off')
plt.savefig('D:/Fuzzy C-Means_Segmented Gray Image.jpg',dpi=700)
plt.show()
```



In [24]:
```
# Create an empty image with the same shape as the original
colored_segmented_image = np.zeros((gray_image.shape[0],
                                    gray_image.shape[1], 3))

np.random.seed(12357)
# Assign colors to each cluster
for i in range(n_clusters):
    colored_segmented_image[segmented_image == i] = np.random.rand(3)

# Display the color-segmented image
plt.figure(figsize=(5, 5))
plt.title('Color-Segmented Image',fontsize=8)
plt.imshow(colored_segmented_image)
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.savefig('D:/Fuzzy C-Means_Segmented Color Image.jpg',dpi=700)
plt.show()
```



In [25]:
```
# Display the original and Clustered Image images
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title('Original Image',fontsize=8)
plt.imshow(image)
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.subplot(1, 2, 2)
plt.title('Color-Segmented Image',fontsize=8)
plt.imshow(colored_segmented_image)
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.savefig('D:/Fuzzy C-Means_Segmented Color and Gray Image.jpg',dpi=700)
plt.show()
```