2017csb1078_LAB1_CS518

August 28, 2019

[1]:
```
import matplotlib.pyplot as plt
import os
from os.path import join
import numpy as np
from PIL import Image
import matplotlib.image as mpimg
from skimage.color import rgb2gray
from skimage.color import label2rgb
from skimage.filters import gaussian
from sklearn.cluster import KMeans
from skimage import img_as_ubyte
```

[2]:
```
plt.close('all')
clear = lambda: os.system('clear')
clear()
np.random.seed(110)

colors = [[1,0,0],[0,1,0],[0,0,1],[0,0.5,0.5],[0.5,0,0.5]] #List of colours

imgNames = ['water_coins','jump','tiger']#{'balloons', 'mountains', 'nature',␣
 ↪'ocean', 'polarlights'};
segmentCounts = [2,3,4,5]
```

[3]:
```
img_num = np.zeros((len(imgNames)*len(segmentCounts)),dtype='int') #stores the␣
 ↪last iteration number before convergence of EM
itr_cnt = -1 #iteration counter to traverse img_num

for imgName in imgNames:
    for SegCount in segmentCounts:
        itr_cnt+=1 #increment iteration counter
        # Load the imageusing MatPlotLib
        img_mtlb = mpimg.imread("Input/" + imgName+ ".png")
        print('Using Matplotlib Image Library: Image is of datatype ',img_mtlb.
 ↪dtype,'and size ',img_mtlb.shape) # Image is of type float

        # Load the Pillow-- the Python Imaging Library
        img = Image.open("Input/" + imgName+ ".png").convert('RGB')
```

```
        print('Using Pillow (Python Image Library): Image is of datatype ',img.
↪info,'and size ',img.size) # Image is of type uint8


        #%% %Define Parameters
        nSegments = SegCount   # of color clusters in image
        nPixels = img_mtlb.shape[0]*img_mtlb.shape[1];    # Image can be␣
↪represented by a matrix of size nPixels*nColors
        maxIterations = 20; #maximum number of iterations allowed for EM␣
↪algorithm.
        nColors = 3;
        #%% Determine the output path for writing images to files
        outputPath = join(''.join(['Output/',str(SegCount), '_segments/',␣
↪imgName , '/']));
        if not(os.path.exists(outputPath)):
            os.makedirs(outputPath)
        mpimg.imsave(''.join([outputPath,'0.png']),img_mtlb) #save using␣
↪Matplotlib image library
        #%% Vectorizing image for easier loops- done as im(:) in Matlab
        pixels = img
        pixels = np.array(img).reshape(nPixels,nColors,1)

        #%%
        """ Initialize pi (mixture proportion) vector and mu matrix (containing␣
↪means of each distribution)
            Vector of probabilities for segments... 1 value for each segment.
            Best to think of it like this...
            When the image was generated, color was determined for each pixel␣
↪by selecting
            a value from one of "n" normal distributions. Each value in this␣
↪vector
            corresponds to the probability that a given normal distribution was␣
↪chosen."""


        """ Initial guess for pi's is 1/nSegments. Small amount of noise added␣
↪to slightly perturb
           GMM coefficients from the initial guess"""

        pi = 1/nSegments*(np.ones((nSegments, 1),dtype='float'))
        increment = np.random.normal(0,.0001,1)
        for seg_ctr in range(len(pi)):
            if(seg_ctr%2==1):
                pi[seg_ctr] = pi[seg_ctr] + increment
                if pi[seg_ctr] > 1:
                    pi[seg_ctr] = 1
```

```
            else:
                pi[seg_ctr] = pi[seg_ctr] - increment
                if pi[seg_ctr] < 0:
                    pi[seg_ctr] = 0

        #%%
        """Similarly, the initial guess for the segment color means would be a␣
↪perturbed version of [mu_R, mu_G, mu_B],
           where mu_R, mu_G, mu_B respectively denote the means of the R,G,B␣
↪color channels in the image.
           mu is a nSegments X nColors matrrix,(seglabels*255).np.asarray(int)␣
↪where each matrix row denotes mean RGB color for a particcular segment"""

        mu = 1/nSegments*(np.ones((nSegments, nColors),dtype='float'))  #for␣
↪even start
        #add noise to the initialization (but keep it unit)
        for seg_ctr in range(nSegments):
            if(seg_ctr%2==1):
                increment = np.random.normal(0,.0001,1)
            for col_ctr in range(nColors):
                if(seg_ctr%2==1):
                    mu[seg_ctr,col_ctr] = np.mean(pixels[:,col_ctr]) + increment
                else:
                    mu[seg_ctr,col_ctr] = np.mean(pixels[:,col_ctr]) -␣
↪increment;


        #%% EM-iterations begin here. Start with the initial (pi, mu) guesses  ␣
↪

        mu_last_iter = mu;
        pi_last_iter = pi;


        for iteration in range(maxIterations):
            img_num[itr_cnt] = iteration
            """%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
               % ------------------   E-step  -----estimating likelihoods and␣
↪membership weights (Ws)
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"""

            print(''.join(['Image: ',imgName,' nSegments: ',str(nSegments),'␣
↪iteration: ',str(iteration+1), ' E-step']))
            # Weights that describe the likelihood that pixel denoted by␣
↪"pix_import scipy.miscctr" belongs to a color cluster "seg_ctr"
```

```
            Ws = np.ones((nPixels,nSegments),dtype='float')  # temporarily␣
↪reinitialize all weights to 1, before they are recomputed

            """ logarithmic form of the E step."""

            for pix_ctr in range(nPixels):
                # Calculate Ajs
                logAjVec = np.zeros((nSegments,1),dtype='float')
                for seg_ctr in range(nSegments):
                    x_minus_mu_T  = np.transpose(pixels[pix_ctr,:]-(mu[seg_ctr,:
↪])[np.newaxis].T)
                    x_minus_mu    = ((pixels[pix_ctr,:]-(mu[seg_ctr,:])[np.
↪newaxis].T))
                    logAjVec[seg_ctr] = np.log(pi[seg_ctr]) - .5*(np.
↪dot(x_minus_mu_T,x_minus_mu))

                # Note the max
                logAmax = max(logAjVec.tolist())

                # Calculate the third term from the final eqn in the above link
                thirdTerm = 0;
                for seg_ctr in range(nSegments):
                    thirdTerm = thirdTerm + np.exp(logAjVec[seg_ctr]-logAmax)

                # Here Ws are the relative membership weights(p_i/sum(p_i)),␣
↪but computed in a round-about way
                for seg_ctr in range(nSegments):
                    logY = logAjVec[seg_ctr] - logAmax - np.log(thirdTerm)
                    Ws[pix_ctr][seg_ctr] = np.exp(logY)


            """%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                % -----------------   M-step  --------------------
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"""

            print(''.join(['Image: ',imgName,' nSegments: ',str(nSegments),'␣
↪iteration: ',str(iteration+1), ' M-step: Mixture coefficients']))
            #%% temporarily reinitialize mu and pi to 0, before they are␣
↪recomputed
            mu = np.zeros((nSegments,nColors),dtype='float') # mean color for␣
↪each segment
            pi = np.zeros((nSegments,1),dtype='float') #mixture coefficients


            for seg_ctr in range(nSegments):
                '''
```

```
                denominatorSum = 0;
                for pix_ctr in range(nPixels):
                    mu[seg_ctr] = mu[seg_ctr] + pixels[pix_ctr,:
↪,0]*Ws[pix_ctr,seg_ctr]
                    denominatorSum = denominatorSum + Ws[pix_ctr][seg_ctr]
                '''
                denominatorSum = np.sum(Ws[:,seg_ctr])
                mu[seg_ctr] = np.sum(np.multiply(pixels[:,:,0],np.tile(np.
↪reshape(Ws[:,seg_ctr],(Ws[:,seg_ctr].shape[0],1)),(1,3))),axis=0)

                ## Update mu
                mu[seg_ctr,:] =  mu[seg_ctr,:]/ denominatorSum;
                ## Update pi
                pi[seg_ctr] = denominatorSum / nPixels; #sum of weights (each␣
↪weight is a probability) for given segment/total num of pixels


            print(np.transpose(pi))

            muDiffSq = np.sum(np.multiply((mu - mu_last_iter),(mu -␣
↪mu_last_iter)))
            piDiffSq = np.sum(np.multiply((pi - pi_last_iter),(pi -␣
↪pi_last_iter)))

            if (muDiffSq < .0000001 and piDiffSq < .0000001): #sign of␣
↪convergence
                print('Convergence Criteria Met at Iteration: ',iteration, '--␣
↪Exiting code')
                break;


            mu_last_iter = mu;
            pi_last_iter = pi;


            ##Draw the segmented image using the mean of the color cluster as␣
↪the
            ## RGB value for all pixels in that cluster.
            segpixels = np.array(pixels)
            cluster = 0
            for pix_ctr in range(nPixels):
                cluster = np.where(Ws[pix_ctr,:] == max(Ws[pix_ctr,:]))
                vec     = np.squeeze(np.transpose(mu[cluster,:]))
                segpixels[pix_ctr,:] =  vec.reshape(vec.shape[0],1)
```

```
            segpixels = np.reshape(segpixels,(img_mtlb.shape[0],img_mtlb.
↪shape[1],nColors)) ## reshape segpixels to obtain R,G, B image
            segpixels = img_as_ubyte(segpixels)
            segpixels = rgb2gray(segpixels)

            kmeans = KMeans(n_clusters = SegCount).fit(np.
↪reshape(segpixels,(nPixels, 1)))
            seglabels = np.reshape(kmeans.labels_, (img_mtlb.shape[0], img_mtlb.
↪shape[1]))
            seglabels = gaussian(np.clip(label2rgb(seglabels,colors= colors),␣
↪0,1), sigma = 2, multichannel = False)

            mpimg.imsave(''.join([outputPath,str(iteration+1),'.
↪png']),seglabels) #save the segmented output
```

```
Using Matplotlib Image Library: Image is of datatype  float32 and size  (312,
252, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (72, 72)} and
size  (252, 312)
Image: water_coins nSegments: 2 iteration: 1 E-step
Image: water_coins nSegments: 2 iteration: 1 M-step: Mixture coefficients
[[0.49996714 0.50003286]]
Image: water_coins nSegments: 2 iteration: 2 E-step
Image: water_coins nSegments: 2 iteration: 2 M-step: Mixture coefficients
[[0.44622235 0.55377765]]
Image: water_coins nSegments: 2 iteration: 3 E-step
Image: water_coins nSegments: 2 iteration: 3 M-step: Mixture coefficients
[[0.44233313 0.55766687]]
Image: water_coins nSegments: 2 iteration: 4 E-step
Image: water_coins nSegments: 2 iteration: 4 M-step: Mixture coefficients
[[0.4420263 0.5579737]]
Image: water_coins nSegments: 2 iteration: 5 E-step
Image: water_coins nSegments: 2 iteration: 5 M-step: Mixture coefficients
[[0.4419647 0.5580353]]
Image: water_coins nSegments: 2 iteration: 6 E-step
Image: water_coins nSegments: 2 iteration: 6 M-step: Mixture coefficients
[[0.44196429 0.55803571]]
Convergence Criteria Met at Iteration:  5 -- Exiting code
Using Matplotlib Image Library: Image is of datatype  float32 and size  (312,
252, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (72, 72)} and
size  (252, 312)
Image: water_coins nSegments: 3 iteration: 1 E-step
Image: water_coins nSegments: 3 iteration: 1 M-step: Mixture coefficients
[[0.33319247 0.33350716 0.33330038]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
```

```
clusters (2) found smaller than n_clusters (3). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: water_coins nSegments: 3 iteration: 2 E-step
Image: water_coins nSegments: 3 iteration: 2 M-step: Mixture coefficients
[[0.00119731 0.44569995 0.55310274]]
Image: water_coins nSegments: 3 iteration: 3 E-step
Image: water_coins nSegments: 3 iteration: 3 M-step: Mixture coefficients
[[0.04250423 0.42040175 0.53709402]]
Image: water_coins nSegments: 3 iteration: 4 E-step
Image: water_coins nSegments: 3 iteration: 4 M-step: Mixture coefficients
[[0.04630196 0.41498179 0.53871626]]
Image: water_coins nSegments: 3 iteration: 5 E-step
Image: water_coins nSegments: 3 iteration: 5 M-step: Mixture coefficients
[[0.0474237  0.4123865  0.54018979]]
Image: water_coins nSegments: 3 iteration: 6 E-step
Image: water_coins nSegments: 3 iteration: 6 M-step: Mixture coefficients
[[0.04821937 0.41063214 0.54114849]]
Image: water_coins nSegments: 3 iteration: 7 E-step
Image: water_coins nSegments: 3 iteration: 7 M-step: Mixture coefficients
[[0.04906744 0.40940582 0.54152675]]
Image: water_coins nSegments: 3 iteration: 8 E-step
Image: water_coins nSegments: 3 iteration: 8 M-step: Mixture coefficients
[[0.04934349 0.40875594 0.54190057]]
Image: water_coins nSegments: 3 iteration: 9 E-step
Image: water_coins nSegments: 3 iteration: 9 M-step: Mixture coefficients
[[0.04966063 0.40824027 0.5420991 ]]
Image: water_coins nSegments: 3 iteration: 10 E-step
Image: water_coins nSegments: 3 iteration: 10 M-step: Mixture coefficients
[[0.04998498 0.40787776 0.54213726]]
Image: water_coins nSegments: 3 iteration: 11 E-step
Image: water_coins nSegments: 3 iteration: 11 M-step: Mixture coefficients
[[0.05009089 0.40756086 0.54234825]]
Image: water_coins nSegments: 3 iteration: 12 E-step
Image: water_coins nSegments: 3 iteration: 12 M-step: Mixture coefficients
[[0.04998644 0.4074945  0.54251907]]
Image: water_coins nSegments: 3 iteration: 13 E-step
Image: water_coins nSegments: 3 iteration: 13 M-step: Mixture coefficients
[[0.05016    0.40725759 0.54258241]]
Image: water_coins nSegments: 3 iteration: 14 E-step
Image: water_coins nSegments: 3 iteration: 14 M-step: Mixture coefficients
[[0.05026017 0.40712829 0.54261154]]
Image: water_coins nSegments: 3 iteration: 15 E-step
Image: water_coins nSegments: 3 iteration: 15 M-step: Mixture coefficients
[[0.05025178 0.40712758 0.54262064]]
Image: water_coins nSegments: 3 iteration: 16 E-step
Image: water_coins nSegments: 3 iteration: 16 M-step: Mixture coefficients
```

```
[[0.05025164 0.40712756 0.5426208 ]]
Convergence Criteria Met at Iteration:  15 -- Exiting code
Using Matplotlib Image Library: Image is of datatype  float32 and size  (312,
252, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (72, 72)} and
size  (252, 312)
Image: water_coins nSegments: 4 iteration: 1 E-step
Image: water_coins nSegments: 4 iteration: 1 M-step: Mixture coefficients
[[0.25010824 0.24989581 0.25010661 0.24988934]]
Image: water_coins nSegments: 4 iteration: 2 E-step
Image: water_coins nSegments: 4 iteration: 2 M-step: Mixture coefficients
[[0.00085428 0.55280229 0.00098861 0.44535482]]
Image: water_coins nSegments: 4 iteration: 3 E-step
Image: water_coins nSegments: 4 iteration: 3 M-step: Mixture coefficients
[[0.02635153 0.53689707 0.01687922 0.41987217]]
Image: water_coins nSegments: 4 iteration: 4 E-step
Image: water_coins nSegments: 4 iteration: 4 M-step: Mixture coefficients
[[0.03774957 0.5316639  0.02345893 0.4071276 ]]
Image: water_coins nSegments: 4 iteration: 5 E-step
Image: water_coins nSegments: 4 iteration: 5 M-step: Mixture coefficients
[[0.04270925 0.5301344  0.02640085 0.4007555 ]]
Image: water_coins nSegments: 4 iteration: 6 E-step
Image: water_coins nSegments: 4 iteration: 6 M-step: Mixture coefficients
[[0.04511921 0.53006054 0.02766993 0.39715032]]
Image: water_coins nSegments: 4 iteration: 7 E-step
Image: water_coins nSegments: 4 iteration: 7 M-step: Mixture coefficients
[[0.04653641 0.53056311 0.02812219 0.39477829]]
Image: water_coins nSegments: 4 iteration: 8 E-step
Image: water_coins nSegments: 4 iteration: 8 M-step: Mixture coefficients
[[0.04744707 0.53115886 0.02844057 0.39295351]]
Image: water_coins nSegments: 4 iteration: 9 E-step
Image: water_coins nSegments: 4 iteration: 9 M-step: Mixture coefficients
[[0.04815341 0.53165251 0.02875825 0.39143583]]
Image: water_coins nSegments: 4 iteration: 10 E-step
Image: water_coins nSegments: 4 iteration: 10 M-step: Mixture coefficients
[[0.04859172 0.53205417 0.02912314 0.39023097]]
Image: water_coins nSegments: 4 iteration: 11 E-step
Image: water_coins nSegments: 4 iteration: 11 M-step: Mixture coefficients
[[0.04889172 0.53251153 0.0293309  0.38926584]]
Image: water_coins nSegments: 4 iteration: 12 E-step
Image: water_coins nSegments: 4 iteration: 12 M-step: Mixture coefficients
[[0.04899845 0.53281727 0.02967006 0.38851422]]
Image: water_coins nSegments: 4 iteration: 13 E-step
Image: water_coins nSegments: 4 iteration: 13 M-step: Mixture coefficients
[[0.04988263 0.53324208 0.02960123 0.38727406]]
Image: water_coins nSegments: 4 iteration: 14 E-step
Image: water_coins nSegments: 4 iteration: 14 M-step: Mixture coefficients
[[0.04995359 0.53342491 0.02992724 0.38669426]]
```

```
Image: water_coins nSegments: 4 iteration: 15 E-step
Image: water_coins nSegments: 4 iteration: 15 M-step: Mixture coefficients
[[0.05038829 0.53359203 0.03017987 0.38583982]]
Image: water_coins nSegments: 4 iteration: 16 E-step
Image: water_coins nSegments: 4 iteration: 16 M-step: Mixture coefficients
[[0.0508183  0.53382013 0.03029471 0.38506687]]
Image: water_coins nSegments: 4 iteration: 17 E-step
Image: water_coins nSegments: 4 iteration: 17 M-step: Mixture coefficients
[[0.05109757 0.53392631 0.03053223 0.38444388]]
Image: water_coins nSegments: 4 iteration: 18 E-step
Image: water_coins nSegments: 4 iteration: 18 M-step: Mixture coefficients
[[0.05131127 0.53403542 0.03063197 0.38402134]]
Image: water_coins nSegments: 4 iteration: 19 E-step
Image: water_coins nSegments: 4 iteration: 19 M-step: Mixture coefficients
[[0.05115913 0.53409747 0.03087486 0.38386855]]
Image: water_coins nSegments: 4 iteration: 20 E-step
Image: water_coins nSegments: 4 iteration: 20 M-step: Mixture coefficients
[[0.05189987 0.53421008 0.0309609  0.38292915]]
Using Matplotlib Image Library: Image is of datatype  float32 and size  (312,
252, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (72, 72)} and
size  (252, 312)
Image: water_coins nSegments: 5 iteration: 1 E-step
Image: water_coins nSegments: 5 iteration: 1 M-step: Mixture coefficients
[[0.19999077 0.20001427 0.19999342 0.20001051 0.19999103]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (2) found smaller than n_clusters (5). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: water_coins nSegments: 5 iteration: 2 E-step
Image: water_coins nSegments: 5 iteration: 2 M-step: Mixture coefficients
[[5.10832074e-04 4.45624996e-01 5.52927544e-01 4.82488821e-04
  4.54139163e-04]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (4) found smaller than n_clusters (5). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: water_coins nSegments: 5 iteration: 3 E-step
Image: water_coins nSegments: 5 iteration: 3 M-step: Mixture coefficients
[[0.01580788 0.41984636 0.53682151 0.00118878 0.02633547]]
Image: water_coins nSegments: 5 iteration: 4 E-step
Image: water_coins nSegments: 5 iteration: 4 M-step: Mixture coefficients
[[0.01708504 0.40712758 0.53141385 0.0103003  0.03407324]]
```

```
Image: water_coins nSegments: 5 iteration: 5 E-step
Image: water_coins nSegments: 5 iteration: 5 M-step: Mixture coefficients
[[0.02048245 0.39861219 0.52633444 0.01625238 0.03831854]]
Image: water_coins nSegments: 5 iteration: 6 E-step
Image: water_coins nSegments: 5 iteration: 6 M-step: Mixture coefficients
[[0.02532487 0.39247312 0.52087366 0.02014658 0.04118178]]
Image: water_coins nSegments: 5 iteration: 7 E-step
Image: water_coins nSegments: 5 iteration: 7 M-step: Mixture coefficients
[[0.03091045 0.3869266  0.51490175 0.02248045 0.04478075]]
Image: water_coins nSegments: 5 iteration: 8 E-step
Image: water_coins nSegments: 5 iteration: 8 M-step: Mixture coefficients
[[0.03773487 0.38236269 0.50765644 0.02436709 0.04787891]]
Image: water_coins nSegments: 5 iteration: 9 E-step
Image: water_coins nSegments: 5 iteration: 9 M-step: Mixture coefficients
[[0.04639041 0.37906796 0.49852895 0.02596319 0.05004948]]
Image: water_coins nSegments: 5 iteration: 10 E-step
Image: water_coins nSegments: 5 iteration: 10 M-step: Mixture coefficients
[[0.05776478 0.37575946 0.48657264 0.02731675 0.05258637]]
Image: water_coins nSegments: 5 iteration: 11 E-step
Image: water_coins nSegments: 5 iteration: 11 M-step: Mixture coefficients
[[0.0709906  0.37285392 0.47261944 0.02854654 0.0549895 ]]
Image: water_coins nSegments: 5 iteration: 12 E-step
Image: water_coins nSegments: 5 iteration: 12 M-step: Mixture coefficients
[[0.08712091 0.37148566 0.45588751 0.02945789 0.05604804]]
Image: water_coins nSegments: 5 iteration: 13 E-step
Image: water_coins nSegments: 5 iteration: 13 M-step: Mixture coefficients
[[0.10452866 0.36978231 0.43810155 0.02986805 0.05771943]]
Image: water_coins nSegments: 5 iteration: 14 E-step
Image: water_coins nSegments: 5 iteration: 14 M-step: Mixture coefficients
[[0.12047077 0.36860169 0.42159036 0.03049282 0.05884436]]
Image: water_coins nSegments: 5 iteration: 15 E-step
Image: water_coins nSegments: 5 iteration: 15 M-step: Mixture coefficients
[[0.13604451 0.36836913 0.40550216 0.03095002 0.05913419]]
Image: water_coins nSegments: 5 iteration: 16 E-step
Image: water_coins nSegments: 5 iteration: 16 M-step: Mixture coefficients
[[0.14872838 0.36834231 0.39255682 0.03105133 0.05932116]]
Image: water_coins nSegments: 5 iteration: 17 E-step
Image: water_coins nSegments: 5 iteration: 17 M-step: Mixture coefficients
[[0.15898795 0.36837233 0.38201234 0.03113835 0.05948903]]
Image: water_coins nSegments: 5 iteration: 18 E-step
Image: water_coins nSegments: 5 iteration: 18 M-step: Mixture coefficients
[[0.16663081 0.36841114 0.37399401 0.0313494  0.05961465]]
Image: water_coins nSegments: 5 iteration: 19 E-step
Image: water_coins nSegments: 5 iteration: 19 M-step: Mixture coefficients
[[0.17437146 0.36851492 0.36609965 0.03124134 0.05977263]]
Image: water_coins nSegments: 5 iteration: 20 E-step
Image: water_coins nSegments: 5 iteration: 20 M-step: Mixture coefficients
[[0.18073894 0.36858747 0.35947826 0.03136378 0.05983154]]
```

```
Using Matplotlib Image Library: Image is of datatype  float32 and size  (480,
319, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (300, 300)}
and size  (319, 480)
Image: jump nSegments: 2 iteration: 1 E-step
Image: jump nSegments: 2 iteration: 1 M-step: Mixture coefficients
[[0.50009936 0.49990064]]
Image: jump nSegments: 2 iteration: 2 E-step
Image: jump nSegments: 2 iteration: 2 M-step: Mixture coefficients
[[0.37244433 0.62755567]]
Image: jump nSegments: 2 iteration: 3 E-step
Image: jump nSegments: 2 iteration: 3 M-step: Mixture coefficients
[[0.29946799 0.70053201]]
Image: jump nSegments: 2 iteration: 4 E-step
Image: jump nSegments: 2 iteration: 4 M-step: Mixture coefficients
[[0.25350046 0.74649954]]
Image: jump nSegments: 2 iteration: 5 E-step
Image: jump nSegments: 2 iteration: 5 M-step: Mixture coefficients
[[0.22492592 0.77507408]]
Image: jump nSegments: 2 iteration: 6 E-step
Image: jump nSegments: 2 iteration: 6 M-step: Mixture coefficients
[[0.20888811 0.79111189]]
Image: jump nSegments: 2 iteration: 7 E-step
Image: jump nSegments: 2 iteration: 7 M-step: Mixture coefficients
[[0.20058279 0.79941721]]
Image: jump nSegments: 2 iteration: 8 E-step
Image: jump nSegments: 2 iteration: 8 M-step: Mixture coefficients
[[0.19707283 0.80292717]]
Image: jump nSegments: 2 iteration: 9 E-step
Image: jump nSegments: 2 iteration: 9 M-step: Mixture coefficients
[[0.19542281 0.80457719]]
Image: jump nSegments: 2 iteration: 10 E-step
Image: jump nSegments: 2 iteration: 10 M-step: Mixture coefficients
[[0.19469429 0.80530571]]
Image: jump nSegments: 2 iteration: 11 E-step
Image: jump nSegments: 2 iteration: 11 M-step: Mixture coefficients
[[0.19445532 0.80554468]]
Image: jump nSegments: 2 iteration: 12 E-step
Image: jump nSegments: 2 iteration: 12 M-step: Mixture coefficients
[[0.19429358 0.80570642]]
Image: jump nSegments: 2 iteration: 13 E-step
Image: jump nSegments: 2 iteration: 13 M-step: Mixture coefficients
[[0.19421823 0.80578177]]
Image: jump nSegments: 2 iteration: 14 E-step
Image: jump nSegments: 2 iteration: 14 M-step: Mixture coefficients
[[0.19415137 0.80584863]]
Image: jump nSegments: 2 iteration: 15 E-step
Image: jump nSegments: 2 iteration: 15 M-step: Mixture coefficients
```

```
[[0.1940697 0.8059303]]
Image: jump nSegments: 2 iteration: 16 E-step
Image: jump nSegments: 2 iteration: 16 M-step: Mixture coefficients
[[0.19393694 0.80606306]]
Image: jump nSegments: 2 iteration: 17 E-step
Image: jump nSegments: 2 iteration: 17 M-step: Mixture coefficients
[[0.19385218 0.80614782]]
Image: jump nSegments: 2 iteration: 18 E-step
Image: jump nSegments: 2 iteration: 18 M-step: Mixture coefficients
[[0.19380079 0.80619921]]
Image: jump nSegments: 2 iteration: 19 E-step
Image: jump nSegments: 2 iteration: 19 M-step: Mixture coefficients
[[0.19378445 0.80621555]]
Image: jump nSegments: 2 iteration: 20 E-step
Image: jump nSegments: 2 iteration: 20 M-step: Mixture coefficients
[[0.19377988 0.80622012]]
Using Matplotlib Image Library: Image is of datatype  float32 and size  (480,
319, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (300, 300)}
and size  (319, 480)
Image: jump nSegments: 3 iteration: 1 E-step
Image: jump nSegments: 3 iteration: 1 M-step: Mixture coefficients
[[0.33342039 0.33319282 0.33338679]]
Image: jump nSegments: 3 iteration: 2 E-step
Image: jump nSegments: 3 iteration: 2 M-step: Mixture coefficients
[[0.36937679 0.6246944  0.00592881]]
Image: jump nSegments: 3 iteration: 3 E-step
Image: jump nSegments: 3 iteration: 3 M-step: Mixture coefficients
[[0.21917043 0.51896878 0.26186078]]
Image: jump nSegments: 3 iteration: 4 E-step
Image: jump nSegments: 3 iteration: 4 M-step: Mixture coefficients
[[0.18038114 0.50878261 0.31083624]]
Image: jump nSegments: 3 iteration: 5 E-step
Image: jump nSegments: 3 iteration: 5 M-step: Mixture coefficients
[[0.17609555 0.5145243  0.30938015]]
Image: jump nSegments: 3 iteration: 6 E-step
Image: jump nSegments: 3 iteration: 6 M-step: Mixture coefficients
[[0.17539842 0.5193025  0.30529907]]
Image: jump nSegments: 3 iteration: 7 E-step
Image: jump nSegments: 3 iteration: 7 M-step: Mixture coefficients
[[0.17516964 0.52201573 0.30281463]]
Image: jump nSegments: 3 iteration: 8 E-step
Image: jump nSegments: 3 iteration: 8 M-step: Mixture coefficients
[[0.17507358 0.52394265 0.30098377]]
Image: jump nSegments: 3 iteration: 9 E-step
Image: jump nSegments: 3 iteration: 9 M-step: Mixture coefficients
[[0.17502603 0.5244161  0.30055787]]
Image: jump nSegments: 3 iteration: 10 E-step
```

```
Image: jump nSegments: 3 iteration: 10 M-step: Mixture coefficients
[[0.17500653 0.52437038 0.30062309]]
Image: jump nSegments: 3 iteration: 11 E-step
Image: jump nSegments: 3 iteration: 11 M-step: Mixture coefficients
[[0.17500653 0.52457208 0.30042139]]
Image: jump nSegments: 3 iteration: 12 E-step
Image: jump nSegments: 3 iteration: 12 M-step: Mixture coefficients
[[0.17500594 0.52477041 0.30022365]]
Image: jump nSegments: 3 iteration: 13 E-step
Image: jump nSegments: 3 iteration: 13 M-step: Mixture coefficients
[[0.17499638 0.52475405 0.30024957]]
Image: jump nSegments: 3 iteration: 14 E-step
Image: jump nSegments: 3 iteration: 14 M-step: Mixture coefficients
[[0.17498868 0.52473764 0.30027367]]
Image: jump nSegments: 3 iteration: 15 E-step
Image: jump nSegments: 3 iteration: 15 M-step: Mixture coefficients
[[0.17498685 0.52472788 0.30028527]]
Image: jump nSegments: 3 iteration: 16 E-step
Image: jump nSegments: 3 iteration: 16 M-step: Mixture coefficients
[[0.17498651 0.52472477 0.30028872]]
Image: jump nSegments: 3 iteration: 17 E-step
Image: jump nSegments: 3 iteration: 17 M-step: Mixture coefficients
[[0.17498643 0.52472408 0.30028949]]
Convergence Criteria Met at Iteration:  16 -- Exiting code
Using Matplotlib Image Library: Image is of datatype  float32 and size  (480,
319, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (300, 300)}
and size  (319, 480)
Image: jump nSegments: 4 iteration: 1 E-step
Image: jump nSegments: 4 iteration: 1 M-step: Mixture coefficients
[[0.25000149 0.24999077 0.25004981 0.24995794]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (3) found smaller than n_clusters (4). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: jump nSegments: 4 iteration: 2 E-step
Image: jump nSegments: 4 iteration: 2 M-step: Mixture coefficients
[[0.00452529 0.36762143 0.62332456 0.00452873]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (3) found smaller than n_clusters (4). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: jump nSegments: 4 iteration: 3 E-step
Image: jump nSegments: 4 iteration: 3 M-step: Mixture coefficients
```

```
[[0.08627288 0.21859387 0.51872145 0.1764118 ]]
Image: jump nSegments: 4 iteration: 4 E-step
Image: jump nSegments: 4 iteration: 4 M-step: Mixture coefficients
[[0.13828477 0.17484623 0.46744522 0.21942377]]
Image: jump nSegments: 4 iteration: 5 E-step
Image: jump nSegments: 4 iteration: 5 M-step: Mixture coefficients
[[0.14106004 0.17220434 0.42770704 0.25902858]]
Image: jump nSegments: 4 iteration: 6 E-step
Image: jump nSegments: 4 iteration: 6 M-step: Mixture coefficients
[[0.14625832 0.17202514 0.39722027 0.28449627]]
Image: jump nSegments: 4 iteration: 7 E-step
Image: jump nSegments: 4 iteration: 7 M-step: Mixture coefficients
[[0.1532088  0.17206113 0.38306623 0.29166384]]
Image: jump nSegments: 4 iteration: 8 E-step
Image: jump nSegments: 4 iteration: 8 M-step: Mixture coefficients
[[0.15963534 0.17213295 0.37559449 0.29263722]]
Image: jump nSegments: 4 iteration: 9 E-step
Image: jump nSegments: 4 iteration: 9 M-step: Mixture coefficients
[[0.16546926 0.17225706 0.36726762 0.29500605]]
Image: jump nSegments: 4 iteration: 10 E-step
Image: jump nSegments: 4 iteration: 10 M-step: Mixture coefficients
[[0.17145972 0.17239503 0.36302254 0.29312271]]
Image: jump nSegments: 4 iteration: 11 E-step
Image: jump nSegments: 4 iteration: 11 M-step: Mixture coefficients
[[0.1779211  0.17256402 0.35758141 0.29193347]]
Image: jump nSegments: 4 iteration: 12 E-step
Image: jump nSegments: 4 iteration: 12 M-step: Mixture coefficients
[[0.18316239 0.17269462 0.34724521 0.29689779]]
Image: jump nSegments: 4 iteration: 13 E-step
Image: jump nSegments: 4 iteration: 13 M-step: Mixture coefficients
[[0.18904295 0.17279224 0.3306045  0.30756031]]
Image: jump nSegments: 4 iteration: 14 E-step
Image: jump nSegments: 4 iteration: 14 M-step: Mixture coefficients
[[0.1957961  0.17295587 0.31199733 0.3192507 ]]
Image: jump nSegments: 4 iteration: 15 E-step
Image: jump nSegments: 4 iteration: 15 M-step: Mixture coefficients
[[0.20236073 0.17307944 0.29613681 0.32842302]]
Image: jump nSegments: 4 iteration: 16 E-step
Image: jump nSegments: 4 iteration: 16 M-step: Mixture coefficients
[[0.20650061 0.17328261 0.28223182 0.33798495]]
Image: jump nSegments: 4 iteration: 17 E-step
Image: jump nSegments: 4 iteration: 17 M-step: Mixture coefficients
[[0.21112228 0.17345079 0.26561074 0.34981619]]
Image: jump nSegments: 4 iteration: 18 E-step
Image: jump nSegments: 4 iteration: 18 M-step: Mixture coefficients
[[0.21519137 0.17356904 0.24166805 0.36957154]]
Image: jump nSegments: 4 iteration: 19 E-step
Image: jump nSegments: 4 iteration: 19 M-step: Mixture coefficients
```

```
[[0.21939828 0.17366125 0.20338163 0.40355885]]
Image: jump nSegments: 4 iteration: 20 E-step
Image: jump nSegments: 4 iteration: 20 M-step: Mixture coefficients
[[0.22606876 0.17376567 0.15239378 0.44777178]]
Using Matplotlib Image Library: Image is of datatype  float32 and size  (480,
319, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (300, 300)}
and size  (319, 480)
Image: jump nSegments: 5 iteration: 1 E-step
Image: jump nSegments: 5 iteration: 1 M-step: Mixture coefficients
[[0.19998078 0.20001583 0.20012365 0.1998954  0.19998434]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (3) found smaller than n_clusters (5). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: jump nSegments: 5 iteration: 2 E-step
Image: jump nSegments: 5 iteration: 2 M-step: Mixture coefficients
[[0.00513388 0.36417622 0.61917981 0.00499579 0.0065143 ]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (4) found smaller than n_clusters (5). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: jump nSegments: 5 iteration: 3 E-step
Image: jump nSegments: 5 iteration: 3 M-step: Mixture coefficients
[[0.10530097 0.21724356 0.51467063 0.01447716 0.14830768]]
Image: jump nSegments: 5 iteration: 4 E-step
Image: jump nSegments: 5 iteration: 4 M-step: Mixture coefficients
[[0.14386418 0.17367449 0.41808199 0.05756961 0.20680974]]
Image: jump nSegments: 5 iteration: 5 E-step
Image: jump nSegments: 5 iteration: 5 M-step: Mixture coefficients
[[0.15038573 0.17260319 0.37183976 0.06014603 0.24502529]]
Image: jump nSegments: 5 iteration: 6 E-step
Image: jump nSegments: 5 iteration: 6 M-step: Mixture coefficients
[[0.16025388 0.17262278 0.34788938 0.06075524 0.25847872]]
Image: jump nSegments: 5 iteration: 7 E-step
Image: jump nSegments: 5 iteration: 7 M-step: Mixture coefficients
[[0.16952062 0.1727403  0.33588538 0.06082843 0.26102527]]
Image: jump nSegments: 5 iteration: 8 E-step
Image: jump nSegments: 5 iteration: 8 M-step: Mixture coefficients
[[0.17674902 0.17277952 0.3255047  0.06067198 0.26429478]]
Image: jump nSegments: 5 iteration: 9 E-step
Image: jump nSegments: 5 iteration: 9 M-step: Mixture coefficients
[[0.18242103 0.17279258 0.31828132 0.06049715 0.26600791]]
```

```
Image: jump nSegments: 5 iteration: 10 E-step
Image: jump nSegments: 5 iteration: 10 M-step: Mixture coefficients
[[0.18604    0.17282733 0.31003091 0.0603484  0.27075336]]
Image: jump nSegments: 5 iteration: 11 E-step
Image: jump nSegments: 5 iteration: 11 M-step: Mixture coefficients
[[0.18847341 0.17286137 0.29988237 0.06019356 0.27858929]]
Image: jump nSegments: 5 iteration: 12 E-step
Image: jump nSegments: 5 iteration: 12 M-step: Mixture coefficients
[[0.19099695 0.17286442 0.28858709 0.05994132 0.28761022]]
Image: jump nSegments: 5 iteration: 13 E-step
Image: jump nSegments: 5 iteration: 13 M-step: Mixture coefficients
[[0.19405851 0.1728839  0.27742154 0.05977603 0.29586002]]
Image: jump nSegments: 5 iteration: 14 E-step
Image: jump nSegments: 5 iteration: 14 M-step: Mixture coefficients
[[0.19694108 0.1729187  0.26832279 0.05957758 0.30223985]]
Image: jump nSegments: 5 iteration: 15 E-step
Image: jump nSegments: 5 iteration: 15 M-step: Mixture coefficients
[[0.19881236 0.17292973 0.26076946 0.05940524 0.30808321]]
Image: jump nSegments: 5 iteration: 16 E-step
Image: jump nSegments: 5 iteration: 16 M-step: Mixture coefficients
[[0.20085182 0.17293063 0.25613283 0.0593005  0.31078422]]
Image: jump nSegments: 5 iteration: 17 E-step
Image: jump nSegments: 5 iteration: 17 M-step: Mixture coefficients
[[0.20202665 0.17295591 0.25305894 0.0592386  0.3127199 ]]
Image: jump nSegments: 5 iteration: 18 E-step
Image: jump nSegments: 5 iteration: 18 M-step: Mixture coefficients
[[0.20249298 0.17297524 0.25095946 0.05916238 0.31440993]]
Image: jump nSegments: 5 iteration: 19 E-step
Image: jump nSegments: 5 iteration: 19 M-step: Mixture coefficients
[[0.20419972 0.17297544 0.24697994 0.05906966 0.31677525]]
Image: jump nSegments: 5 iteration: 20 E-step
Image: jump nSegments: 5 iteration: 20 M-step: Mixture coefficients
[[0.20526637 0.17298031 0.2464119  0.05903016 0.31631126]]
Using Matplotlib Image Library: Image is of datatype  float32 and size  (492,
654, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (192, 192)}
and size  (654, 492)
Image: tiger nSegments: 2 iteration: 1 E-step
Image: tiger nSegments: 2 iteration: 1 M-step: Mixture coefficients
[[0.4998657 0.5001343]]
Image: tiger nSegments: 2 iteration: 2 E-step
Image: tiger nSegments: 2 iteration: 2 M-step: Mixture coefficients
[[0.62346924 0.37653076]]
Image: tiger nSegments: 2 iteration: 3 E-step
Image: tiger nSegments: 2 iteration: 3 M-step: Mixture coefficients
[[0.69180906 0.30819094]]
Image: tiger nSegments: 2 iteration: 4 E-step
Image: tiger nSegments: 2 iteration: 4 M-step: Mixture coefficients
```

```
[[0.7243346 0.2756654]]
Image: tiger nSegments: 2 iteration: 5 E-step
Image: tiger nSegments: 2 iteration: 5 M-step: Mixture coefficients
[[0.74006908 0.25993092]]
Image: tiger nSegments: 2 iteration: 6 E-step
Image: tiger nSegments: 2 iteration: 6 M-step: Mixture coefficients
[[0.74757434 0.25242566]]
Image: tiger nSegments: 2 iteration: 7 E-step
Image: tiger nSegments: 2 iteration: 7 M-step: Mixture coefficients
[[0.75112256 0.24887744]]
Image: tiger nSegments: 2 iteration: 8 E-step
Image: tiger nSegments: 2 iteration: 8 M-step: Mixture coefficients
[[0.75268148 0.24731852]]
Image: tiger nSegments: 2 iteration: 9 E-step
Image: tiger nSegments: 2 iteration: 9 M-step: Mixture coefficients
[[0.75347321 0.24652679]]
Image: tiger nSegments: 2 iteration: 10 E-step
Image: tiger nSegments: 2 iteration: 10 M-step: Mixture coefficients
[[0.75380328 0.24619672]]
Image: tiger nSegments: 2 iteration: 11 E-step
Image: tiger nSegments: 2 iteration: 11 M-step: Mixture coefficients
[[0.75394844 0.24605156]]
Image: tiger nSegments: 2 iteration: 12 E-step
Image: tiger nSegments: 2 iteration: 12 M-step: Mixture coefficients
[[0.75401301 0.24598699]]
Image: tiger nSegments: 2 iteration: 13 E-step
Image: tiger nSegments: 2 iteration: 13 M-step: Mixture coefficients
[[0.75404796 0.24595204]]
Image: tiger nSegments: 2 iteration: 14 E-step
Image: tiger nSegments: 2 iteration: 14 M-step: Mixture coefficients
[[0.75406906 0.24593094]]
Image: tiger nSegments: 2 iteration: 15 E-step
Image: tiger nSegments: 2 iteration: 15 M-step: Mixture coefficients
[[0.75408226 0.24591774]]
Image: tiger nSegments: 2 iteration: 16 E-step
Image: tiger nSegments: 2 iteration: 16 M-step: Mixture coefficients
[[0.75409061 0.24590939]]
Image: tiger nSegments: 2 iteration: 17 E-step
Image: tiger nSegments: 2 iteration: 17 M-step: Mixture coefficients
[[0.7540959 0.2459041]]
Image: tiger nSegments: 2 iteration: 18 E-step
Image: tiger nSegments: 2 iteration: 18 M-step: Mixture coefficients
[[0.75409926 0.24590074]]
Image: tiger nSegments: 2 iteration: 19 E-step
Image: tiger nSegments: 2 iteration: 19 M-step: Mixture coefficients
[[0.75410138 0.24589862]]
Image: tiger nSegments: 2 iteration: 20 E-step
Image: tiger nSegments: 2 iteration: 20 M-step: Mixture coefficients
```

```
[[0.75410272 0.24589728]]
Using Matplotlib Image Library: Image is of datatype  float32 and size  (492,
654, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (192, 192)}
and size  (654, 492)
Image: tiger nSegments: 3 iteration: 1 E-step
Image: tiger nSegments: 3 iteration: 1 M-step: Mixture coefficients
[[0.33328293 0.33336093 0.33335614]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (2) found smaller than n_clusters (3). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: tiger nSegments: 3 iteration: 2 E-step
Image: tiger nSegments: 3 iteration: 2 M-step: Mixture coefficients
[[0.0103453  0.61820347 0.37145124]]
Image: tiger nSegments: 3 iteration: 3 E-step
Image: tiger nSegments: 3 iteration: 3 M-step: Mixture coefficients
[[0.28702984 0.47137664 0.24159352]]
Image: tiger nSegments: 3 iteration: 4 E-step
Image: tiger nSegments: 3 iteration: 4 M-step: Mixture coefficients
[[0.35968113 0.43534746 0.20497141]]
Image: tiger nSegments: 3 iteration: 5 E-step
Image: tiger nSegments: 3 iteration: 5 M-step: Mixture coefficients
[[0.37442039 0.43270599 0.19287362]]
Image: tiger nSegments: 3 iteration: 6 E-step
Image: tiger nSegments: 3 iteration: 6 M-step: Mixture coefficients
[[0.37466689 0.43730468 0.18802843]]
Image: tiger nSegments: 3 iteration: 7 E-step
Image: tiger nSegments: 3 iteration: 7 M-step: Mixture coefficients
[[0.37192632 0.44249774 0.18557593]]
Image: tiger nSegments: 3 iteration: 8 E-step
Image: tiger nSegments: 3 iteration: 8 M-step: Mixture coefficients
[[0.36777825 0.44806644 0.18415531]]
Image: tiger nSegments: 3 iteration: 9 E-step
Image: tiger nSegments: 3 iteration: 9 M-step: Mixture coefficients
[[0.36427126 0.4525639  0.18316484]]
Image: tiger nSegments: 3 iteration: 10 E-step
Image: tiger nSegments: 3 iteration: 10 M-step: Mixture coefficients
[[0.36137567 0.45617922 0.18244511]]
Image: tiger nSegments: 3 iteration: 11 E-step
Image: tiger nSegments: 3 iteration: 11 M-step: Mixture coefficients
[[0.35958028 0.45860012 0.18181961]]
Image: tiger nSegments: 3 iteration: 12 E-step
Image: tiger nSegments: 3 iteration: 12 M-step: Mixture coefficients
[[0.3580888 0.4605373 0.1813739]]
```

```
Image: tiger nSegments: 3 iteration: 13 E-step
Image: tiger nSegments: 3 iteration: 13 M-step: Mixture coefficients
[[0.35712165 0.46185491 0.18102344]]
Image: tiger nSegments: 3 iteration: 14 E-step
Image: tiger nSegments: 3 iteration: 14 M-step: Mixture coefficients
[[0.35607291 0.46315361 0.18077348]]
Image: tiger nSegments: 3 iteration: 15 E-step
Image: tiger nSegments: 3 iteration: 15 M-step: Mixture coefficients
[[0.35509091 0.46433761 0.18057148]]
Image: tiger nSegments: 3 iteration: 16 E-step
Image: tiger nSegments: 3 iteration: 16 M-step: Mixture coefficients
[[0.35433653 0.46527654 0.18038693]]
Image: tiger nSegments: 3 iteration: 17 E-step
Image: tiger nSegments: 3 iteration: 17 M-step: Mixture coefficients
[[0.3538852 0.4659033 0.1802115]]
Image: tiger nSegments: 3 iteration: 18 E-step
Image: tiger nSegments: 3 iteration: 18 M-step: Mixture coefficients
[[0.3534054  0.46647659 0.18011801]]
Image: tiger nSegments: 3 iteration: 19 E-step
Image: tiger nSegments: 3 iteration: 19 M-step: Mixture coefficients
[[0.35285683 0.46709133 0.18005184]]
Image: tiger nSegments: 3 iteration: 20 E-step
Image: tiger nSegments: 3 iteration: 20 M-step: Mixture coefficients
[[0.35253885 0.46749094 0.17997021]]
Using Matplotlib Image Library: Image is of datatype  float32 and size  (492,
654, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (192, 192)}
and size  (654, 492)
Image: tiger nSegments: 4 iteration: 1 E-step
Image: tiger nSegments: 4 iteration: 1 M-step: Mixture coefficients
[[0.24995051 0.25005399 0.24995922 0.25003628]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (3) found smaller than n_clusters (4). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: tiger nSegments: 4 iteration: 2 E-step
Image: tiger nSegments: 4 iteration: 2 M-step: Mixture coefficients
[[0.01440464 0.36791719 0.60670049 0.01097768]]
Image: tiger nSegments: 4 iteration: 3 E-step
Image: tiger nSegments: 4 iteration: 3 M-step: Mixture coefficients
[[0.15462114 0.24167011 0.45693894 0.1467698 ]]
Image: tiger nSegments: 4 iteration: 4 E-step
Image: tiger nSegments: 4 iteration: 4 M-step: Mixture coefficients
[[0.2484054  0.1923689  0.37434101 0.18488469]]
Image: tiger nSegments: 4 iteration: 5 E-step
```

```
Image: tiger nSegments: 4 iteration: 5 M-step: Mixture coefficients
[[0.30962265 0.17044593 0.32742292 0.1925085 ]]
Image: tiger nSegments: 4 iteration: 6 E-step
Image: tiger nSegments: 4 iteration: 6 M-step: Mixture coefficients
[[0.35003303 0.15902621 0.29874121 0.19219955]]
Image: tiger nSegments: 4 iteration: 7 E-step
Image: tiger nSegments: 4 iteration: 7 M-step: Mixture coefficients
[[0.37540109 0.15263852 0.28200264 0.18995775]]
Image: tiger nSegments: 4 iteration: 8 E-step
Image: tiger nSegments: 4 iteration: 8 M-step: Mixture coefficients
[[0.39272043 0.14874817 0.27100671 0.18752469]]
Image: tiger nSegments: 4 iteration: 9 E-step
Image: tiger nSegments: 4 iteration: 9 M-step: Mixture coefficients
[[0.40271199 0.14603511 0.26534375 0.18590915]]
Image: tiger nSegments: 4 iteration: 10 E-step
Image: tiger nSegments: 4 iteration: 10 M-step: Mixture coefficients
[[0.40929843 0.14362731 0.26211235 0.18496191]]
Image: tiger nSegments: 4 iteration: 11 E-step
Image: tiger nSegments: 4 iteration: 11 M-step: Mixture coefficients
[[0.41303484 0.14209927 0.26118249 0.18368339]]
Image: tiger nSegments: 4 iteration: 12 E-step
Image: tiger nSegments: 4 iteration: 12 M-step: Mixture coefficients
[[0.41503265 0.14103812 0.26136235 0.18256688]]
Image: tiger nSegments: 4 iteration: 13 E-step
Image: tiger nSegments: 4 iteration: 13 M-step: Mixture coefficients
[[0.41599065 0.14032499 0.2621583  0.18152606]]
Image: tiger nSegments: 4 iteration: 14 E-step
Image: tiger nSegments: 4 iteration: 14 M-step: Mixture coefficients
[[0.41618079 0.1396825  0.26376474 0.18037197]]
Image: tiger nSegments: 4 iteration: 15 E-step
Image: tiger nSegments: 4 iteration: 15 M-step: Mixture coefficients
[[0.41544442 0.13913981 0.26603332 0.17938245]]
Image: tiger nSegments: 4 iteration: 16 E-step
Image: tiger nSegments: 4 iteration: 16 M-step: Mixture coefficients
[[0.41446094 0.13875729 0.26834699 0.17843478]]
Image: tiger nSegments: 4 iteration: 17 E-step
Image: tiger nSegments: 4 iteration: 17 M-step: Mixture coefficients
[[0.41294643 0.13847171 0.27092398 0.17765788]]
Image: tiger nSegments: 4 iteration: 18 E-step
Image: tiger nSegments: 4 iteration: 18 M-step: Mixture coefficients
[[0.41119527 0.13821931 0.27373145 0.17685397]]
Image: tiger nSegments: 4 iteration: 19 E-step
Image: tiger nSegments: 4 iteration: 19 M-step: Mixture coefficients
[[0.40965452 0.13790759 0.27641701 0.17602088]]
Image: tiger nSegments: 4 iteration: 20 E-step
Image: tiger nSegments: 4 iteration: 20 M-step: Mixture coefficients
[[0.40643272 0.13762895 0.28079202 0.17514631]]
Using Matplotlib Image Library: Image is of datatype  float32 and size  (492,
```

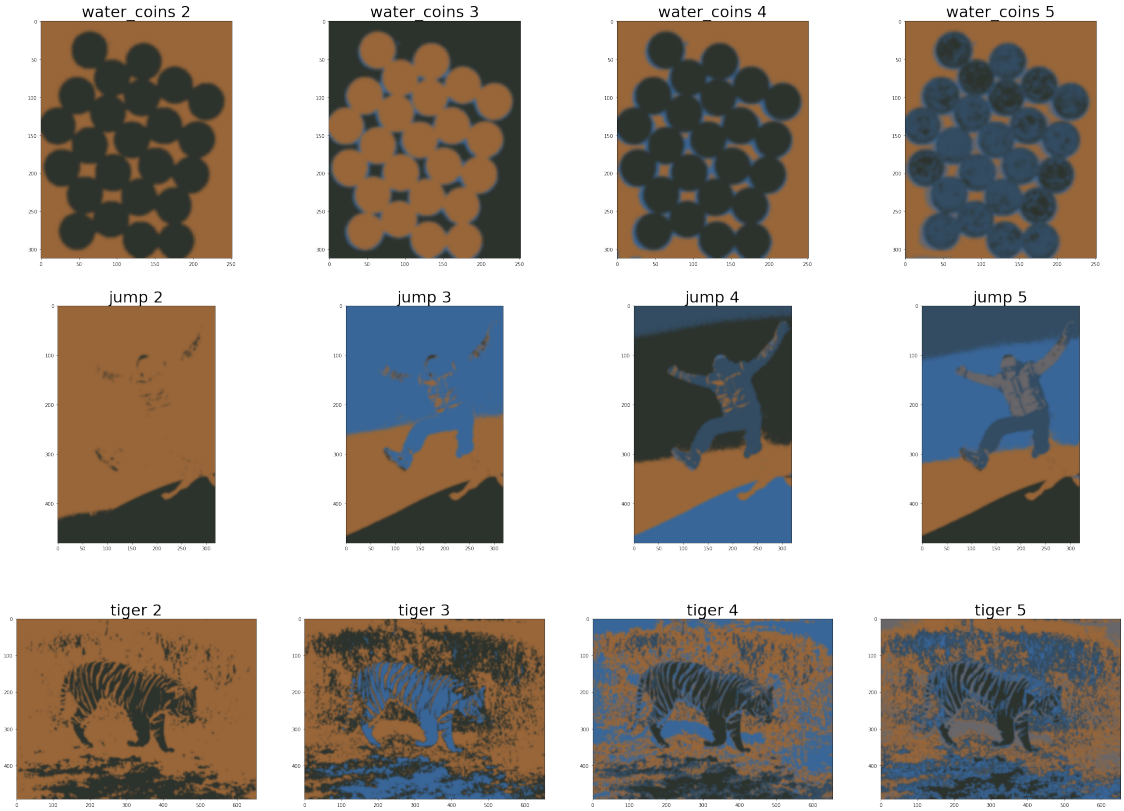
```
654, 3)
Using Pillow (Python Image Library): Image is of datatype  {'dpi': (192, 192)}
and size  (654, 492)
Image: tiger nSegments: 5 iteration: 1 E-step
Image: tiger nSegments: 5 iteration: 1 M-step: Mixture coefficients
[[0.19993678 0.20006673 0.19993671 0.20009792 0.19996186]]

/home/harshit/anaconda3/lib/python3.7/site-
packages/sklearn/cluster/k_means_.py:969: ConvergenceWarning: Number of distinct
clusters (3) found smaller than n_clusters (5). Possibly due to duplicate points
in X.
  return_n_iter=True)

Image: tiger nSegments: 5 iteration: 2 E-step
Image: tiger nSegments: 5 iteration: 2 M-step: Mixture coefficients
[[0.00785695 0.00773622 0.00782183 0.36612204 0.61046297]]
Image: tiger nSegments: 5 iteration: 3 E-step
Image: tiger nSegments: 5 iteration: 3 M-step: Mixture coefficients
[[0.08818695 0.13901936 0.07178315 0.23802683 0.4629837 ]]
Image: tiger nSegments: 5 iteration: 4 E-step
Image: tiger nSegments: 5 iteration: 4 M-step: Mixture coefficients
[[0.14471863 0.17761673 0.1080569  0.18984149 0.37976625]]
Image: tiger nSegments: 5 iteration: 5 E-step
Image: tiger nSegments: 5 iteration: 5 M-step: Mixture coefficients
[[0.18984479 0.17085344 0.16418719 0.16810586 0.30700873]]
Image: tiger nSegments: 5 iteration: 6 E-step
Image: tiger nSegments: 5 iteration: 6 M-step: Mixture coefficients
[[0.19873718 0.16526869 0.21562311 0.15523648 0.26513454]]
Image: tiger nSegments: 5 iteration: 7 E-step
Image: tiger nSegments: 5 iteration: 7 M-step: Mixture coefficients
[[0.20101939 0.16099761 0.25589863 0.14685274 0.23523162]]
Image: tiger nSegments: 5 iteration: 8 E-step
Image: tiger nSegments: 5 iteration: 8 M-step: Mixture coefficients
[[0.21357396 0.15769589 0.28368104 0.14084366 0.20420545]]
Image: tiger nSegments: 5 iteration: 9 E-step
Image: tiger nSegments: 5 iteration: 9 M-step: Mixture coefficients
[[0.2313819  0.15553904 0.29586531 0.13676384 0.18044991]]
Image: tiger nSegments: 5 iteration: 10 E-step
Image: tiger nSegments: 5 iteration: 10 M-step: Mixture coefficients
[[0.24875339 0.15388885 0.30050741 0.13397662 0.16287374]]
Image: tiger nSegments: 5 iteration: 11 E-step
Image: tiger nSegments: 5 iteration: 11 M-step: Mixture coefficients
[[0.26562437 0.15205252 0.3002231  0.13194973 0.15015028]]
Image: tiger nSegments: 5 iteration: 12 E-step
Image: tiger nSegments: 5 iteration: 12 M-step: Mixture coefficients
[[0.28046816 0.15057856 0.29783095 0.13020798 0.14091436]]
Image: tiger nSegments: 5 iteration: 13 E-step
Image: tiger nSegments: 5 iteration: 13 M-step: Mixture coefficients
```

```
[[0.29330485 0.14910021 0.29448116 0.12869393 0.13441985]]
Image: tiger nSegments: 5 iteration: 14 E-step
Image: tiger nSegments: 5 iteration: 14 M-step: Mixture coefficients
[[0.30267907 0.14719266 0.29213337 0.12758988 0.13040503]]
Image: tiger nSegments: 5 iteration: 15 E-step
Image: tiger nSegments: 5 iteration: 15 M-step: Mixture coefficients
[[0.30948883 0.14573342 0.29004814 0.12656725 0.12816235]]
Image: tiger nSegments: 5 iteration: 16 E-step
Image: tiger nSegments: 5 iteration: 16 M-step: Mixture coefficients
[[0.31527245 0.14451806 0.28808421 0.12565009 0.12647519]]
Image: tiger nSegments: 5 iteration: 17 E-step
Image: tiger nSegments: 5 iteration: 17 M-step: Mixture coefficients
[[0.31883649 0.14315119 0.28718571 0.12508061 0.125746  ]]
Image: tiger nSegments: 5 iteration: 18 E-step
Image: tiger nSegments: 5 iteration: 18 M-step: Mixture coefficients
[[0.32197261 0.14223679 0.28575855 0.12453073 0.12550132]]
Image: tiger nSegments: 5 iteration: 19 E-step
Image: tiger nSegments: 5 iteration: 19 M-step: Mixture coefficients
[[0.32425641 0.14145137 0.28471606 0.12403049 0.12554567]]
Image: tiger nSegments: 5 iteration: 20 E-step
Image: tiger nSegments: 5 iteration: 20 M-step: Mixture coefficients
[[0.32617763 0.14075988 0.28380584 0.12357968 0.12567697]]
```

[4]:
```
# Displaying final segmented outputs
itr_cnt = 0
fig = plt.figure(figsize = (40,30))
for imgName in imgNames:
    for SegCount in segmentCounts:
        outputPath = join(''.join(['Output/',str(SegCount), '_segments/',␣
 ↪imgName , '/']))
        img = mpimg.imread(outputPath + str(img_num[itr_cnt]) + ".png")
        itr_cnt+=1
        a = fig.add_subplot(3,4,itr_cnt)
        a.set_title(imgName + " " + str(SegCount),fontsize=32)
        plt.imshow(img)
```


water_coins 2
water_coins 3
water_coins 4
water_coins 5
jump 2
jump 3
jump 4
jump 5
tiger 2
tiger 3
tiger 4
tiger 5

[ ]: