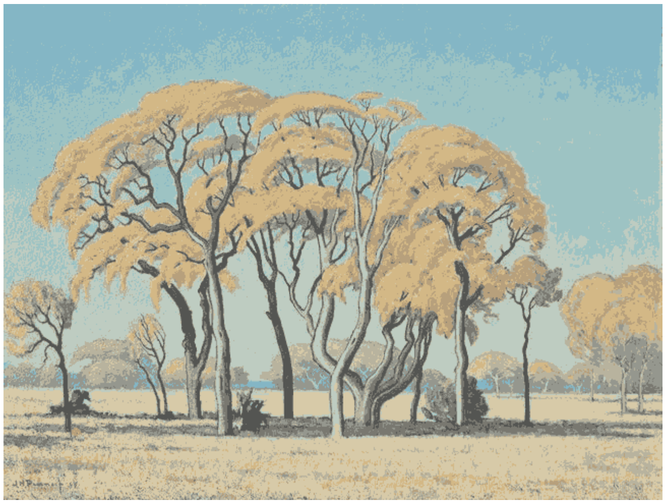
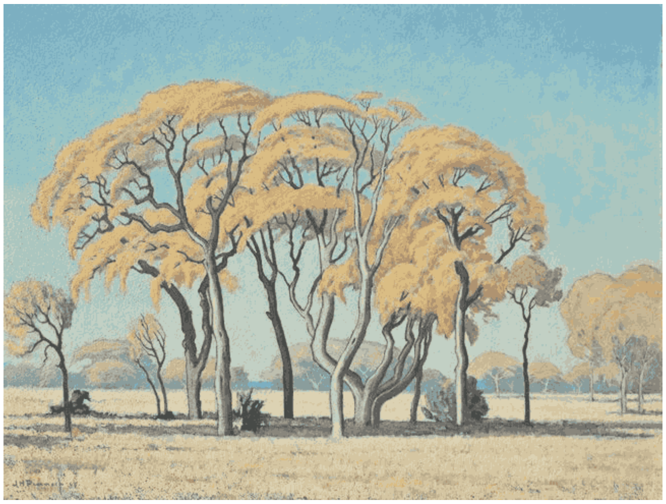
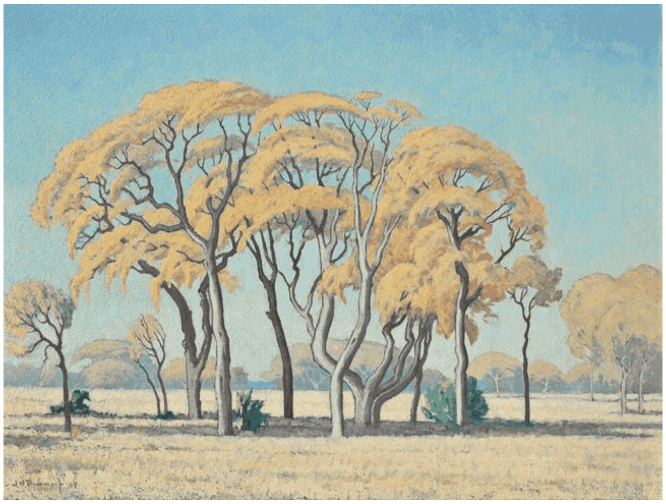
CS498AML_HW8_panz2



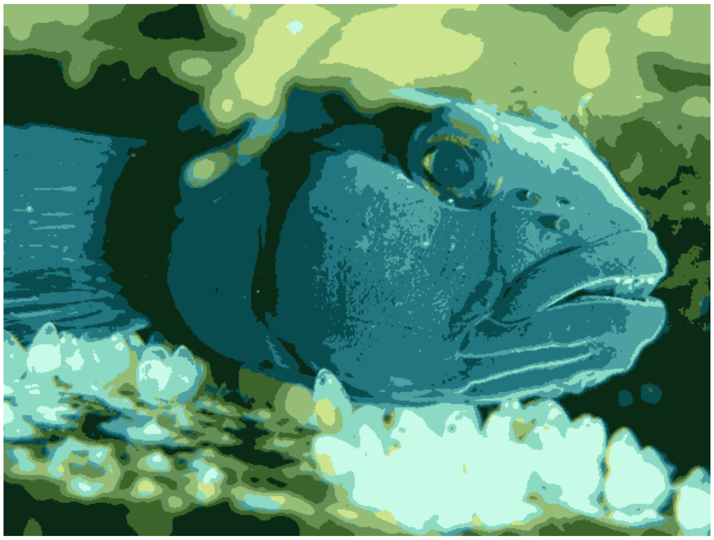
10 segment



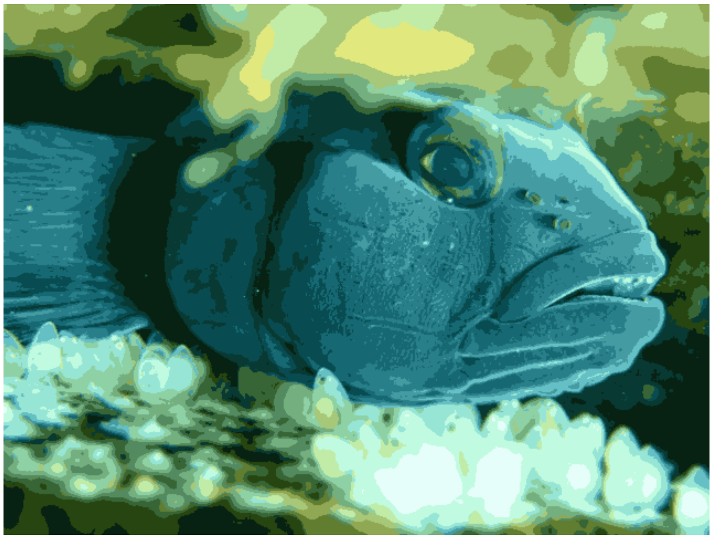
20 segment



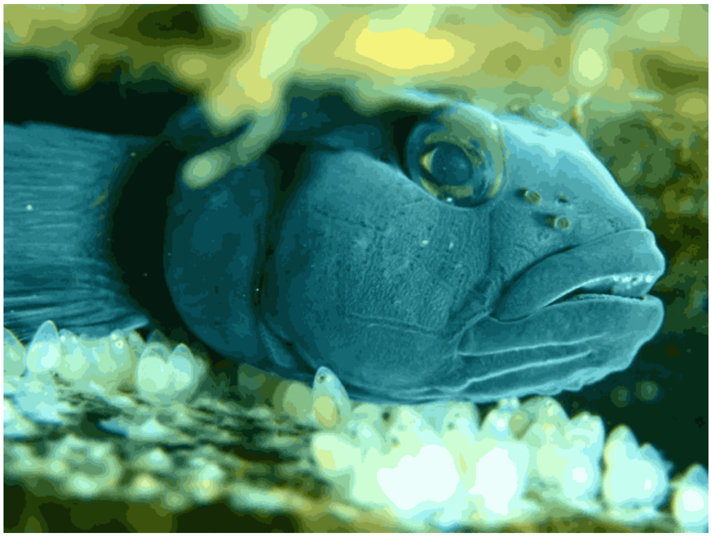
50 segment



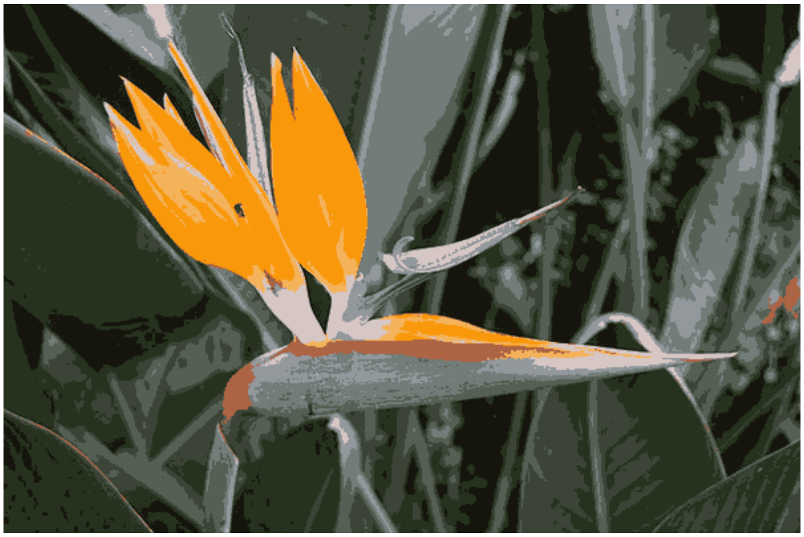
10 segment



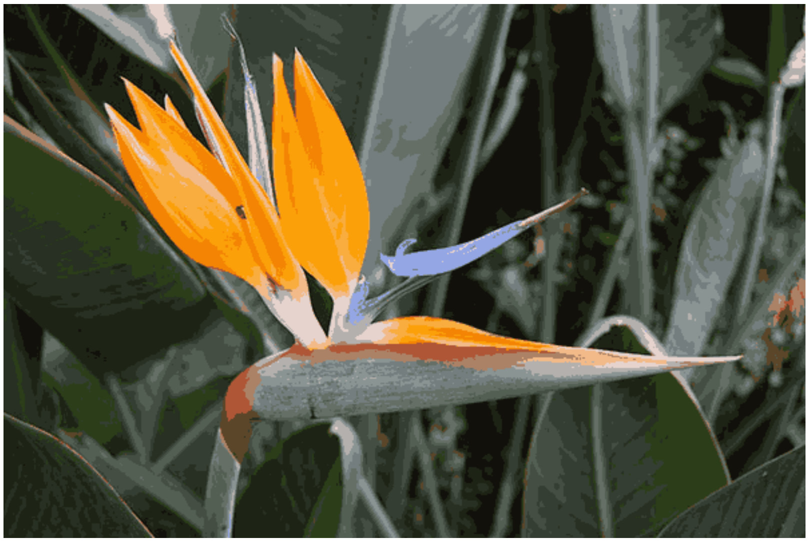
20 segment



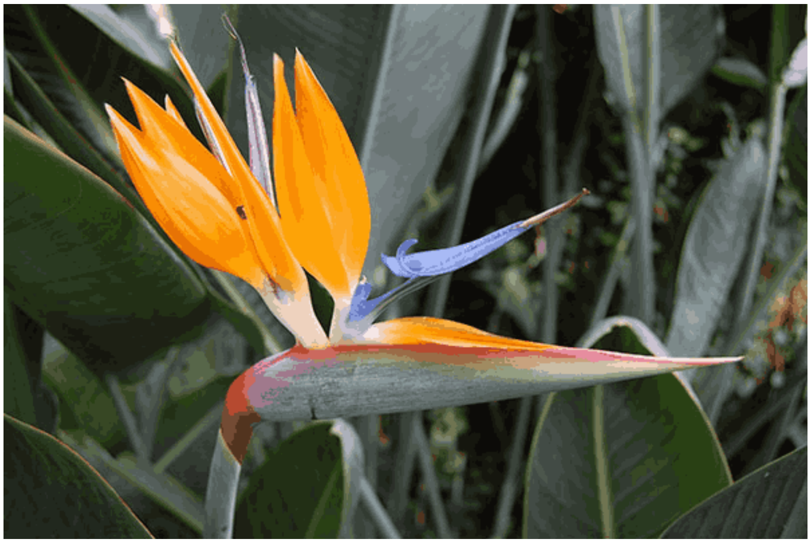
50 segment



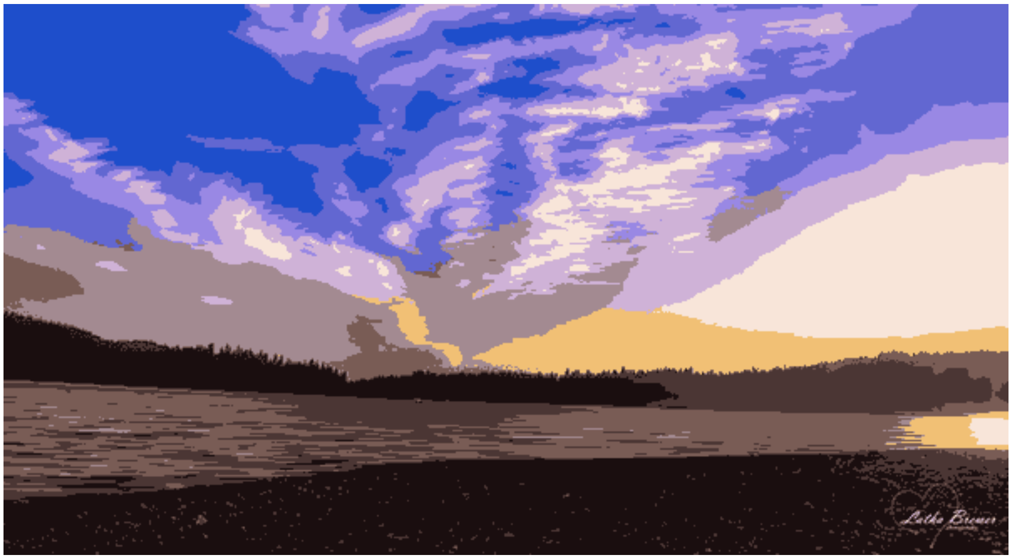
10 segment



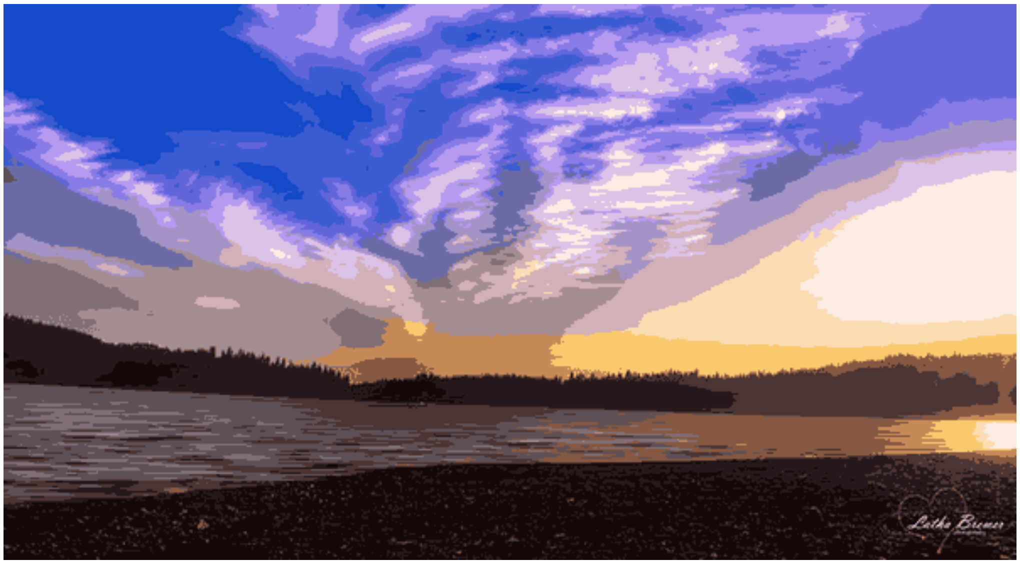
20 segment



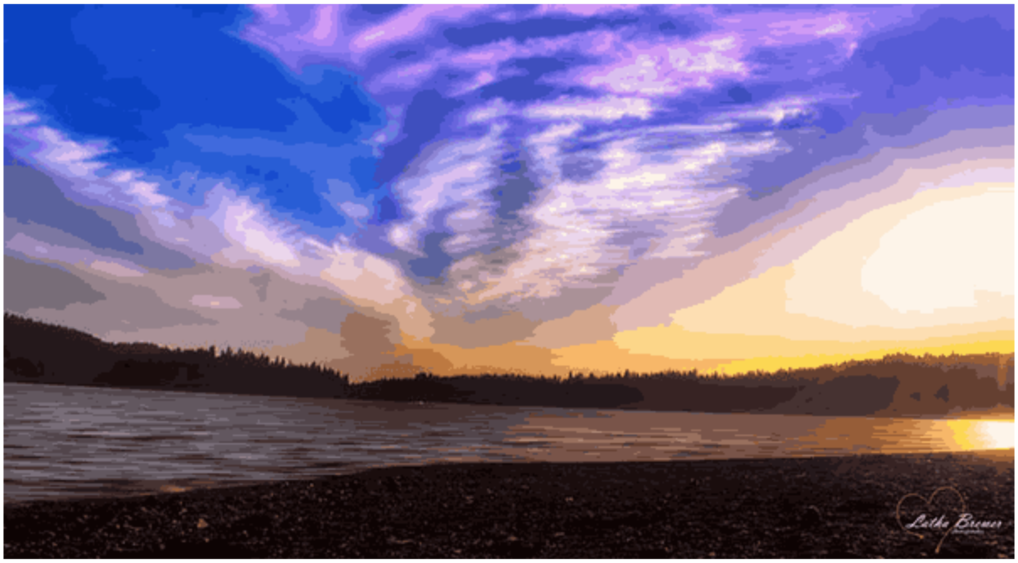
50 segment



10 segment



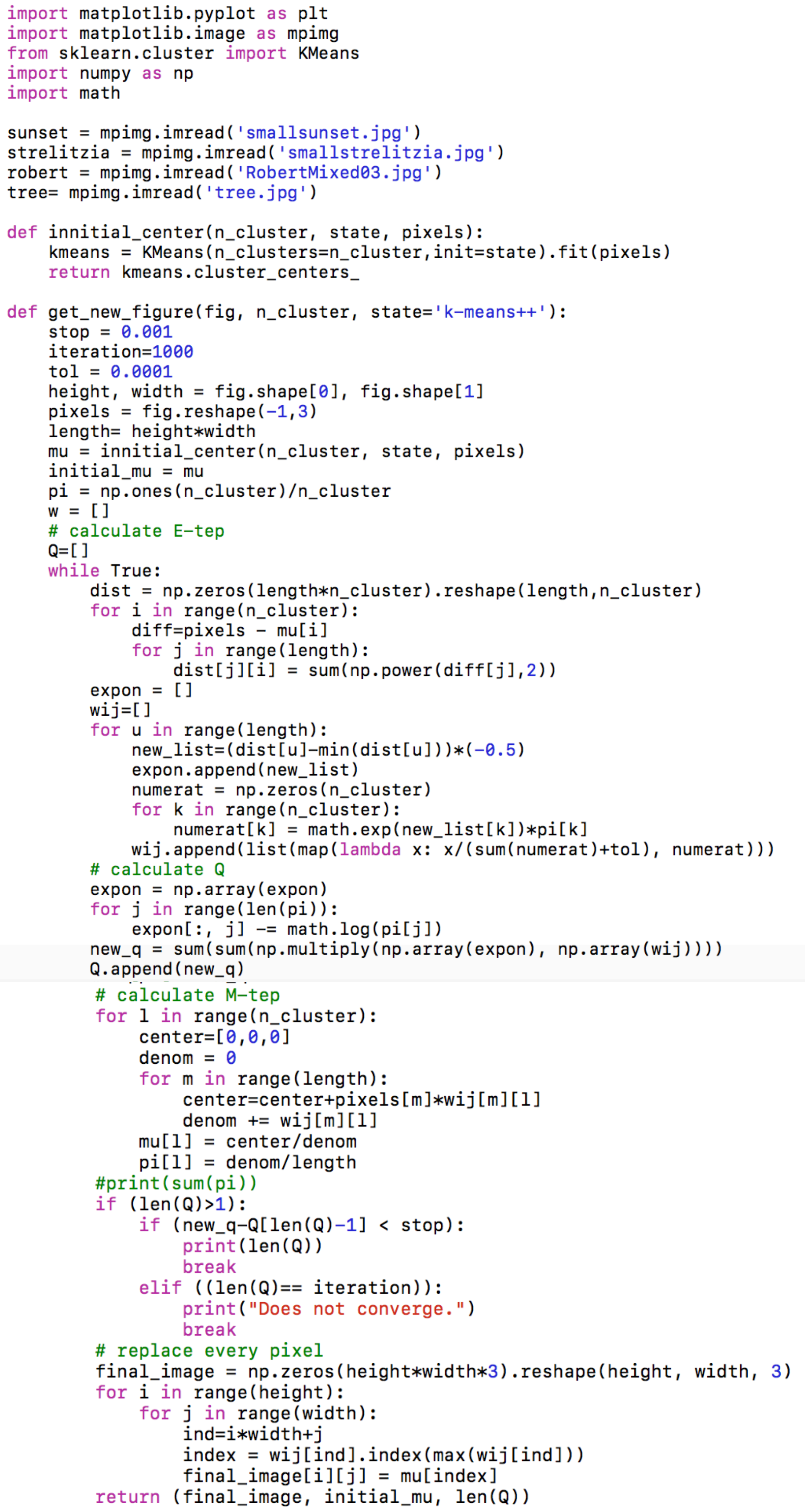
20 segment



50 segment

Display of tree image with 20 segments with 5 different initial points:

Five different start points don’t result in obvious difference in the pictures.

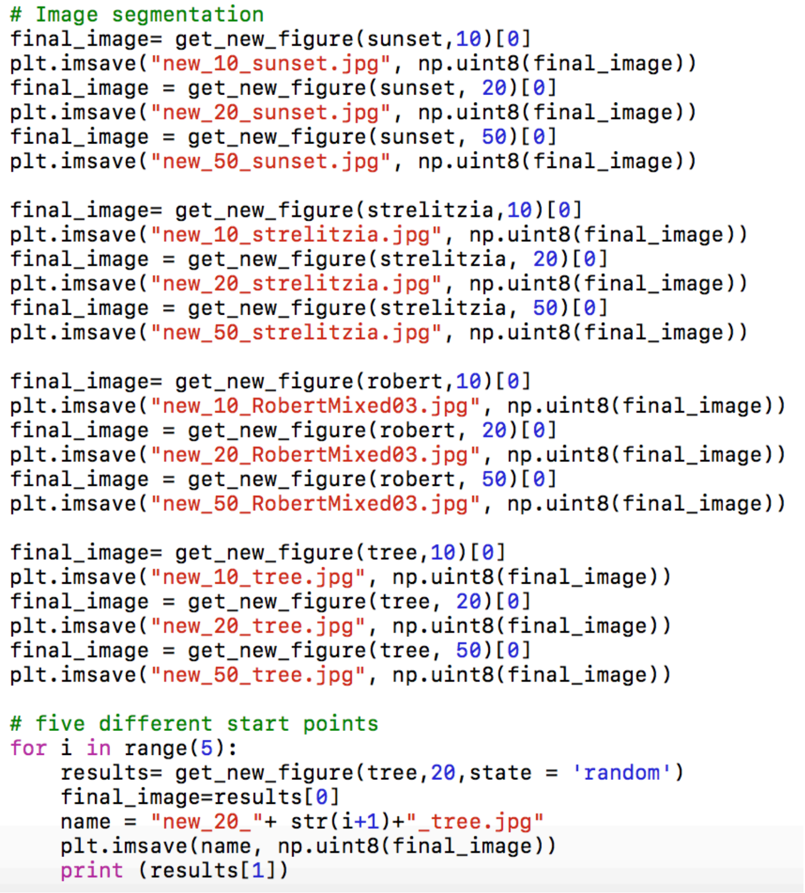


```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from sklearn.cluster import KMeans
import numpy as np
import math

sunset = mpimg.imread('smallsunset.jpg')
strelitzia = mpimg.imread('smallstrelitzia.jpg')
robert = mpimg.imread('RobertMixed03.jpg')
tree= mpimg.imread('tree.jpg')

def innitial_center(n_cluster, state, pixels):
    kmeans = KMeans(n_clusters=n_cluster,init=state).fit(pixels)
    return kmeans.cluster_centers_

def get_new_figure(fig, n_cluster, state='k-means++'):
    stop = 0.001
    iteration=1000
    tol = 0.0001
    height, width = fig.shape[0], fig.shape[1]
    pixels = fig.reshape(-1,3)
    length= height*width
    mu = innitial_center(n_cluster, state, pixels)
    initial_mu = mu
    pi = np.ones(n_cluster)/n_cluster
    w = []
    # calculate E-tep
    Q=[]
    while True:
        dist = np.zeros(length*n_cluster).reshape(length,n_cluster)
        for i in range(n_cluster):
            diff=pixels - mu[i]
            for j in range(length):
                dist[j][i] = sum(np.power(diff[j],2))
        expon = []
        wij=[]
        for u in range(length):
            new_list=(dist[u]-min(dist[u]))*(-0.5)
            expon.append(new_list)
            numerat = np.zeros(n_cluster)
            for k in range(n_cluster):
                numerat[k] = math.exp(new_list[k])*pi[k]
            wij.append(list(map(lambda x: x/(sum(numerat)+tol), numerat)))
        # calculate Q
        expon = np.array(expon)
        for j in range(len(pi)):
            expon[:, j] -= math.log(pi[j])
        new_q = sum(sum(np.multiply(np.array(expon), np.array(wij))))
        Q.append(new_q)
        # calculate M-tep
        for l in range(n_cluster):
            center=[0,0,0]
            denom = 0
            for m in range(length):
                center=center+pixels[m]*wij[m][l]
                denom += wij[m][l]
            mu[l] = center/denom
            pi[l] = denom/length
        #print(sum(pi))
        if (len(Q)>1):
            if (new_q-Q[len(Q)-1] < stop):
                print(len(Q))
                break
            elif ((len(Q)== iteration)):
                print("Does not converge.")
                break
        # replace every pixel
        final_image = np.zeros(height*width*3).reshape(height, width, 3)
        for i in range(height):
            for j in range(width):
                ind=i*width+j
                index = wij[ind].index(max(wij[ind]))
                final_image[i][j] = mu[index]
        return (final_image, initial_mu, len(Q))
```



```
# Image segmentation
final_image= get_new_figure(sunset,10)[0]
plt.imsave("new_10_sunset.jpg", np.uint8(final_image))
final_image = get_new_figure(sunset, 20)[0]
plt.imsave("new_20_sunset.jpg", np.uint8(final_image))
final_image = get_new_figure(sunset, 50)[0]
plt.imsave("new_50_sunset.jpg", np.uint8(final_image))

final_image= get_new_figure(strelitzia,10)[0]
plt.imsave("new_10_strelitzia.jpg", np.uint8(final_image))
final_image = get_new_figure(strelitzia, 20)[0]
plt.imsave("new_20_strelitzia.jpg", np.uint8(final_image))
final_image = get_new_figure(strelitzia, 50)[0]
plt.imsave("new_50_strelitzia.jpg", np.uint8(final_image))

final_image= get_new_figure(robert,10)[0]
plt.imsave("new_10_RobertMixed03.jpg", np.uint8(final_image))
final_image = get_new_figure(robert, 20)[0]
plt.imsave("new_20_RobertMixed03.jpg", np.uint8(final_image))
final_image = get_new_figure(robert, 50)[0]
plt.imsave("new_50_RobertMixed03.jpg", np.uint8(final_image))

final_image= get_new_figure(tree,10)[0]
plt.imsave("new_10_tree.jpg", np.uint8(final_image))
final_image = get_new_figure(tree, 20)[0]
plt.imsave("new_20_tree.jpg", np.uint8(final_image))
final_image = get_new_figure(tree, 50)[0]
plt.imsave("new_50_tree.jpg", np.uint8(final_image))

# five different start points
for i in range(5):
    results= get_new_figure(tree,20,state = 'random')
    final_image=results[0]
    name = "new_20_"+ str(i+1)+"_tree.jpg"
    plt.imsave(name, np.uint8(final_image))
    print (results[1])
```