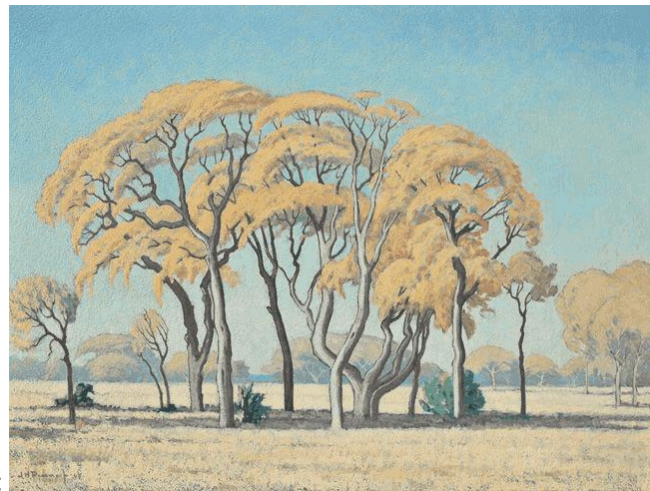




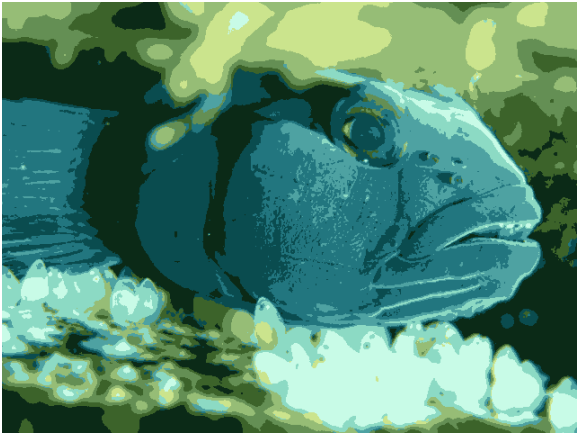
10:



20:



30:



10:



20:



50:

10:



20:

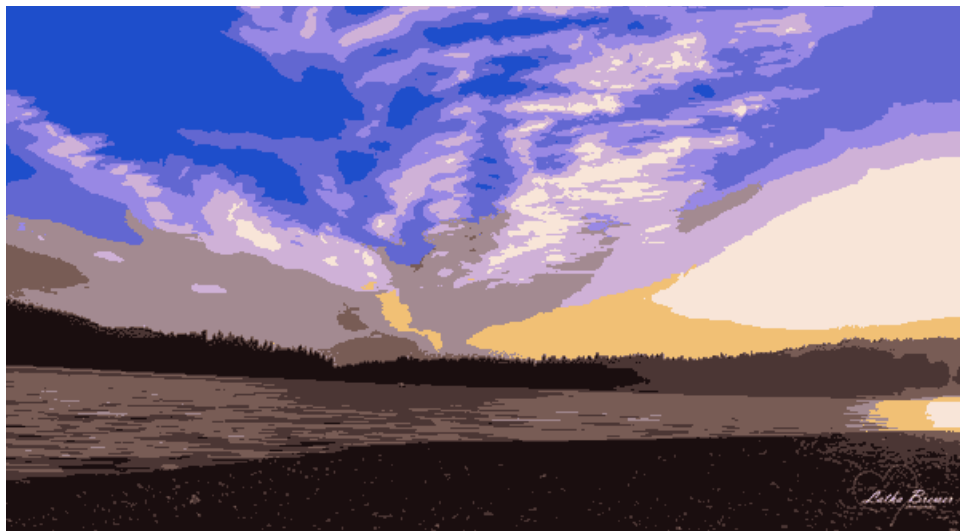


50:

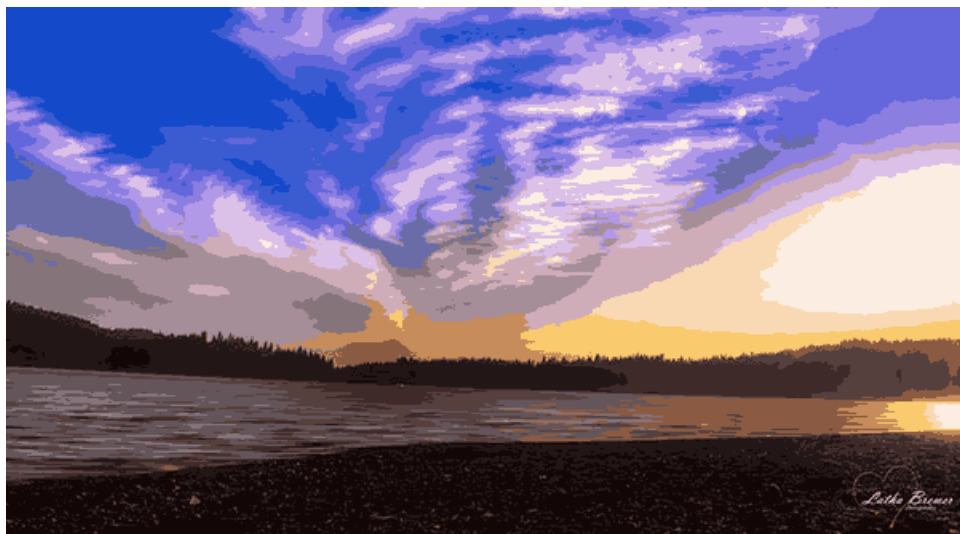




10:



20:



50:





There is very little difference, even though they all have different starting points.

```
def initialize_parameters(pixels, num_seg):

    kmeans = KMeans(n_clusters = num_seg, init = "random", random_state = 0).fit(pixels)
    mu = kmeans.cluster_centers_
    pi = np.ones((num_seg, 1))/num_seg

    return mu, pi
```

```
def estimation(pixels, mu, pi):

    w = np.zeros((pixels.shape[0], mu.shape[0]))
    for x in range(pixels.shape[0]):
        pixel = np.array([pixels[x]]*mu.shape[0])
        diff = np.subtract(pixel, mu)
        square = np.multiply(diff, diff)
        min_idx = np.argmin(np.sum(square, axis = 1))
        dmin = square[min_idx]
        square = np.subtract(square, dmin)
        power = -1/2 * np.sum(square, axis=1)
        weight = np.multiply(np.exp(power), pi.T)
        w[x] = np.divide(weight, np.sum(weight))

    return w
```

```
def maximization(pixels, w):

    mu = np.zeros((w.shape[1], pixels.shape[1]))
    pi = np.zeros((w.shape[1], 1))
    for c in range(w.shape[1]):

        w_c = w.T[c]
        mu_num = np.matmul(pixels.T, w_c)
        mu_denom = np.sum(w_c)
        mu[c] = np.divide(mu_num, mu_denom)
        pi[c] = np.divide(mu_denom, pixels.shape[0])

    return mu, pi
```

```
def EM(iteration, pixels, mu, pi):
    for i in range(iteration):
        w = estimation(pixels, mu, pi)
        mu, pi = maximization(pixels, w)
    return mu, pi
```

```
def implot(im, mu):

    rows = im.shape[0]
    cols = im.shape[1]
    im_out = np.zeros((rows, cols, 3))
    for r in range(rows):
        for c in range(cols):
            idx = np.argmin(np.sum(np.abs(np.subtract(im[r,c:], mu)), axis = 1))
            im_out[r,c,:] = mu[idx]

    return im_out

iteration = 10 # number of iterations
num_seg = 50 # number of segments

mu0, pi0 = initialize_parameters(pixels01, num_seg)
mu1, pi1 = EM(iteration, pixels01, mu0, pi0)
im01_out = implot(im01, mu1)
plt.figure()
plt.imshow(im01_out/255)
plt.imsave("im01_cluster50.png", im01_out/255)
```