K-means

May 26, 2020

1 K-means Clustering

1.0.1 Import resources and display image

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2

%matplotlib inline

# Read in the image
## TODO: Check out the images directory to see other images you can work with
# And select one!
image = cv2.imread('images/monarch.jpg')

# Change color to RGB (from BGR)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.imshow(image)
```

Out[1]:
```
<matplotlib.image.AxesImage at 0x7f6d65c0d128>
```

1.0.2 Prepare data for k-means

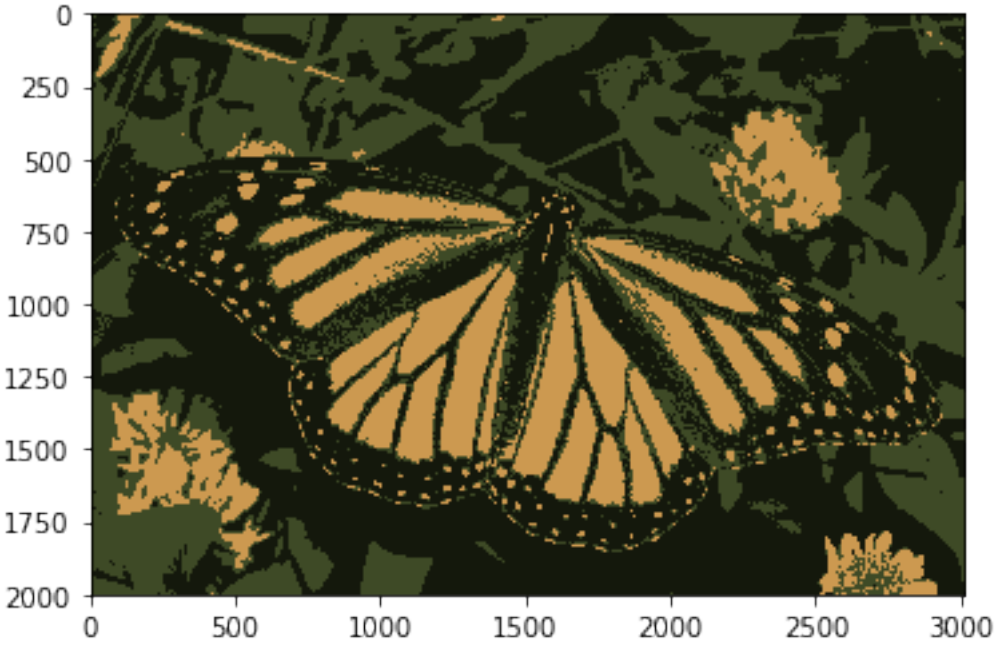
```
In [2]: # Reshape image into a 2D array of pixels and 3 color values (RGB)
        pixel_vals = image.reshape((-1,3))

        # Convert to float type
        pixel_vals = np.float32(pixel_vals)
```

1.0.3 Implement k-means clustering

```
In [3]: # define stopping criteria
        # you can change the number of max iterations for faster convergence!
        criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.2)

        ## TODO: Select a value for k
        # then perform k-means clustering
        k = 3
        retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10, cv2.KMEANS_RANDO

        # convert data into 8-bit values
        centers = np.uint8(centers)
        segmented_data = centers[labels.flatten()]

        # reshape data into the original image dimensions
        segmented_image = segmented_data.reshape((image.shape))
        labels_reshape = labels.reshape(image.shape[0], image.shape[1])
```

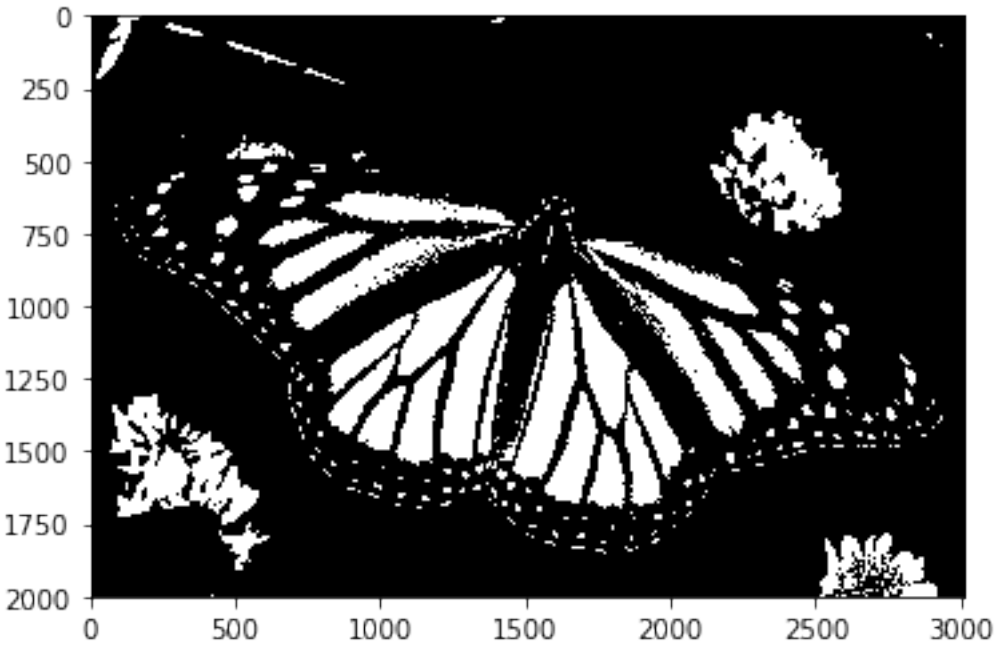
```
plt.imshow(segmented_image)
```

Out[3]: <matplotlib.image.AxesImage at 0x7f6d65bb7c50>



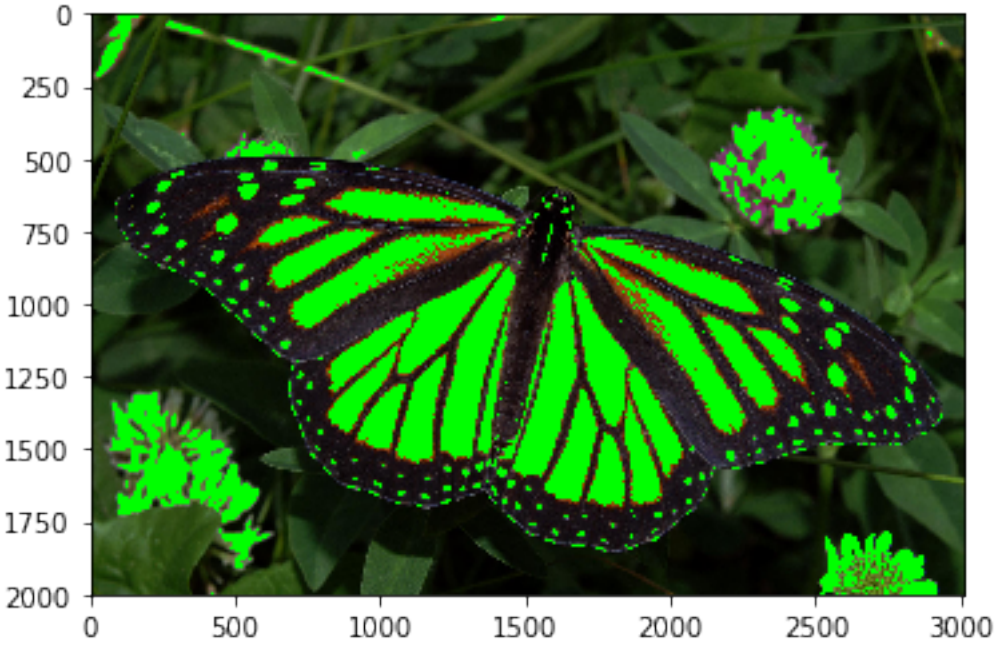
In [4]:
```
## TODO: Visualize one segment, try to find which is the leaves, background, etc!
plt.imshow(labels_reshape==0, cmap='gray')
```

Out[4]: <matplotlib.image.AxesImage at 0x7f6d4404e128>



In [5]:
```
# mask an image segment by cluster

cluster = 0 # the first cluster

masked_image = np.copy(image)
# turn the mask green!
masked_image[labels_reshape == cluster] = [0, 255, 0]

plt.imshow(masked_image)
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x7f6d2ccb2e48>
```


0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000
2500
3000

In [ ]: