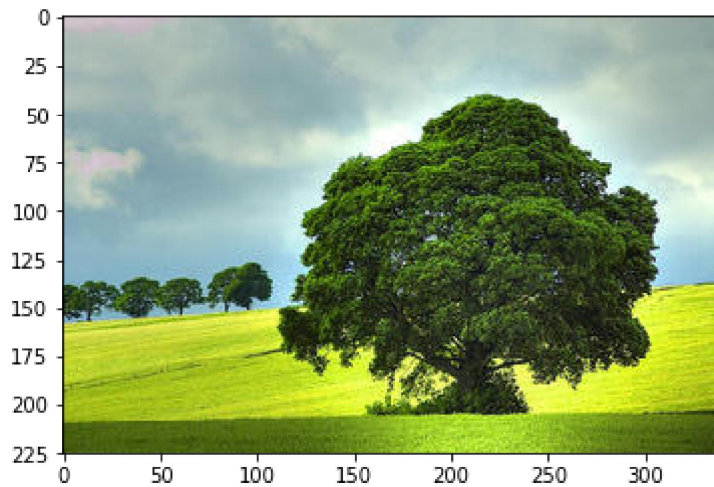


```
In [1]: # https://github.com/JustGlowing/minisom/blob/master/examples/examples.ipynb
import matplotlib.pyplot as plt
```

```
In [2]: # read the image
img = plt.imread('tree.jpg')
```

```
In [3]: import matplotlib.image as mpimg
imgplot = plt.imshow(img)
plt.show()
```



```
In [4]: import numpy as np
# reshaping the pixels matrix
pixels = np.reshape(img, (img.shape[0]*img.shape[1], 3))
```

```
In [5]: pixels
```

```
Out[5]: array([[209, 196, 205],
               [209, 196, 205],
               [209, 196, 205],
               ...,
               [ 55,  70,  11],
               [ 52,  67,   8],
               [ 52,  67,   8]], dtype=uint8)
```

```
In [6]: pixels.shape
```

```
Out[6]: (76500, 3)
```

```
In [7]: from minisom_new import MiniSom
# SOM initialization and training
print('training...')
som = MiniSom(3, 3, 3, sigma=1., learning_rate=0.2, neighborhood_function='bubble')
som.random_weights_init(pixels)
starting_weights = som.get_weights().copy() # saving the starting weights
som.train_random(pixels, 1000)
```

```
training...
```

```
In [8]: print('quantization...')
qnt = som.quantization(pixels) # quantize each pixels of the image
print('building new image...')
clustered = np.zeros(img.shape)
for i, q in enumerate(qnt): # place the quantized values into a new image
    clustered[np.unravel_index(i, dims=(img.shape[0], img.shape[1]))] = q
print('done.')
```

```
quantization...
building new image...
done.
```

```

In [9]: # show the result
plt.figure(figsize=(7, 7))
plt.figure(1)
plt.subplot(221)
plt.title('original')
plt.imshow(img)
plt.subplot(222)
plt.title('result')
plt.imshow(clustered)

plt.subplot(223)
plt.title('initial colors')
plt.imshow(starting_weights, interpolation='none')
plt.subplot(224)
plt.title('learned colors')
plt.imshow(som.get_weights(), interpolation='none')

plt.tight_layout()
plt.savefig('som_color_quantization.png')
plt.show()

```

