HandIn-Lab02

Original Image:

Compressed Image:

Code:

```
#Download the image that was used above and convert it to an array
img_url =
"https://upload.wikimedia.org/wikipedia/commons/d/d7/RGB_24bits_palette_sam
ple_image.jpg"
img = download_img(img_url)
img_arr = img2arr(img)

#Reshape the array so that it can be used
img_arr_reshaped = img_arr.reshape((-1, 3))

#Check how many clusters
distortions = []
clusters = [2, 4, 6, 8, 16, 32, 64, 128, 256]
for i in clusters:
  kMeans = KMeans(n_clusters=i, random_state=0).fit(img_arr_reshaped)
  distortions.append(kMeans.inertia_)

plt.plot(clusters, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.show()

#The 5th cluster seems to produce the best value, so we do the kMean again
with the value 16
kMeans = KMeans(n_clusters=16, random_state=0).fit(img_arr_reshaped)

# plot the cluster centers on top of the colors of the imag
rg_chroma_plot(img_arr_reshaped, kMeans.cluster_centers_)

# replace original colors by their nearest neighbors out of the candidate
centers
newImg = replace_nearest_color(img_arr, kMeans.cluster_centers_)
# convert to PIL.Image and visualize
display(arr2img(newImg))
```

Comment on the clustering

I verfyfied the clustering by using the elbow method, here I plotted the graph and saw, that the value 16 was placed best.