Assignment 1: Color Quantization

Color quantization is the process of reducing number of colors used in an image while trying to maintain the visual appearance of the original image. In general, it is a form of cluster analysis, if each RGB color value is considered as a coordinate triple in the 3D colorspace. The following example shows an rgb image (~16 Million possible colors) quantized using 2,4,8,16,32 colors.

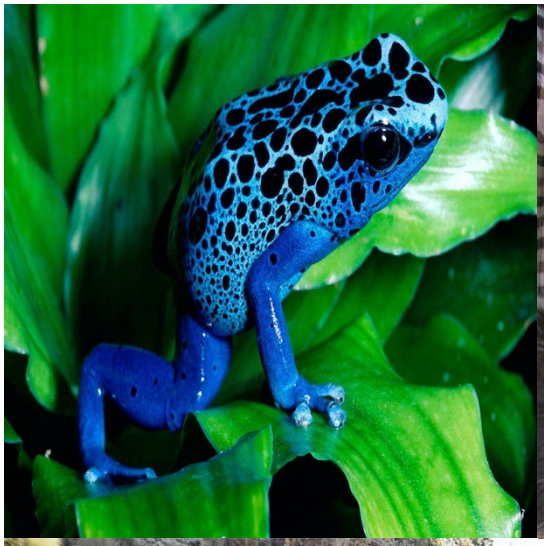
In your first assignment, you will be working on a **Python** implementation of basic image processing operations and color quantization techniques.

You are going to write the function "quantize(img, K)" that takes two inputs and returns a single output. The first input is an rgb formatted Image and the second one is the number of colors (K) in an image we aim to quantize. The function is supposed to calculate the quantized version of the image using the following method:

1. Convert input image to a matrix of pixels. Depending on your image size your matrix should be of size [height*width,3]
2. Pick K initial colors to begin quantization. Use the following methods to initialize your algorithm

    1. Choose initial color centers manually, by clicking on the image: Using matplotlib librarys pyplot.ginput function, choose k points. Example code snippet to interactively select points from an image is provided at the end of this document.
    2. Choose initial color centers randomly. Pick color centers randomly using numpys `numpy.random.uniform` function.
3. Using your own implementation of the k-means algorithm, cluster your color matrix and find K clusters and cluster centers. You may limit the maximum number of iterations to 10 or a suitable number depending on the size of your images.
4. Generate an output image. In the output image, each pixel should have the color of the cluster-center that pixelo has been assigned to.

Once you are done with your image processing pipeline, you are going to experiment on the following images and present 2,4,8,16 and 32 color quantized versions of these images with both initialization methods in your report.



Bonus - 1: Try using K-means using an uniform color coordinate system (Lab colorspace)
Bonus - 2: Choose initial color centers by calculating an 8*8*8 3d color histogram and choosing K histogram peaks with the most samples.

Development Environment

Installing Python

Throughout the semester, we will be using **Python 3** as our programming language. We recommend that you use **Conda** and **PyCharm Professional Edition** to setup your development environment and install the necessary libraries.

You can obtain from **Miniconda**. Follow the installation instructions for your operating system. Once the installation is complete, navigate to your installation directory on a command line / terminal and use the following commands to install the necessary libraries as follows:

- **cd** /path/to/your/miniconda/installation**/bin/**
- **conda install opencv matplotlib numpy**

If you have added the miniconda/bin directory to your system path, you can test your installation by typing the following:

- python
- >> import cv2

If there are no messages like '*module not found error*' no module named '*cv2*', your installation was successful. If you get the error message, please checkout OpenCV for Python.

Installing an IDE

We recommend that you use PyCharm as your IDE for ease of debugging. You are free to use any IDE of your choice.

Obtain PyCharm Professional Edition for free by first signing up as a student at https://www.jetbrains.com/shop/eform/students with your university email and then downloading the professional version of the software.

Login to your new PyCharm account to obtain your username and password, and use them to register your IDE.

Run PyCharm, open a new project. While creating the project, select the python environment you created like follows:

Deliverables

- **Project report (pdf & tex):** A maximum two page **pdf** report file which includes small versions of all the requested figures, images and descriptions of the methods that you have implemented. Comment on the results.
- **Source codes (py files):** Submit all of the functions that you have implemented for the assignment. **Do not print any source code in your report**.

WARNING! Submit all files as one zip file. Please send files in correct format. Use zip for packaging, do not use rar, 7z etc. If you are not able to upload your assignment files to **Moodle**, submit a cloud storage (Dropbox / google drive etc) link of the compressed package.

Email address: alp.kindiroglu@boun.edu.tr

Code Snippet For Interactively Selecting Points

```
from PIL import Image
from matplotlib import pyplot as plt


plt.imshow(im)
points = plt.ginput(3,show_clicks=True)
print("clicked", points)
im = Image.open("frog2.jpg")
```

Note: If using pycharm you may need to do the following if ginput does not work. Under Settings => Tools => Python Scientific Uncheck the (only) box "Show plots in toolwindow". Future plots should appear "normally" and not in **SciView**.