Project: Image manipulation in Numpy

Overview

In this exercise, you should look for an image that you will use to do some manipulations with. It is highly important that you do all your image manipulations using the raw image data in Numpy (and do not use the PIL object). You can find in the following location an example of a notebook where we used Numpy for image manipulation:

`Dropbox/Syntra/slides/chapter09 Numpy/motiverend voorbeeld`

Detailed assignment

Assignment: Setting Up Project and Developing a Production Simulation Program

In this assignment, you will follow a series of steps to set up a project structure, manage version control using Git, create a virtual environment with Anaconda, and develop a Python program for image manipulation

Step 1: Initialize Git Repository

Initialize a git repository on your personal GitHub account to track changes and collaborate effectively. You can add the data into this git repo.

Step 2: Create Virtual Environment with Anaconda

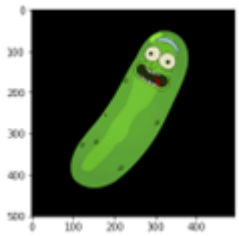
Utilize Anaconda to establish a virtual environment for your project. Ensure that you install only the necessary packages required for the project.

Step 3: Choose & load image

Choose an image that you want to do the manipulations for. Use the following code to read in this image and get the underlying numpy object.

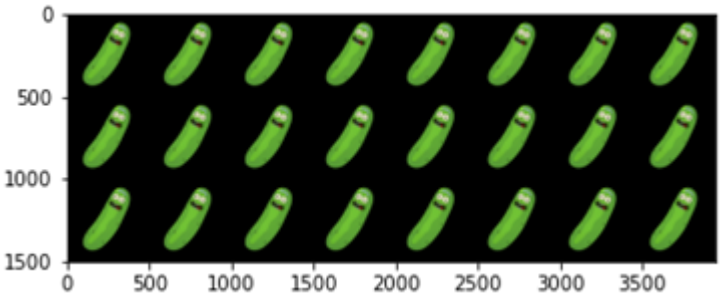
```
from PIL import Image
from matplotlib.pyplot as plt
import numpy as np
import os

loc_input_img = os.path.join('..', 'data', 'input', 'pickle_rick.png')
np_image = np.array(image)
plt.imshow(np_image)
plt.show()
```

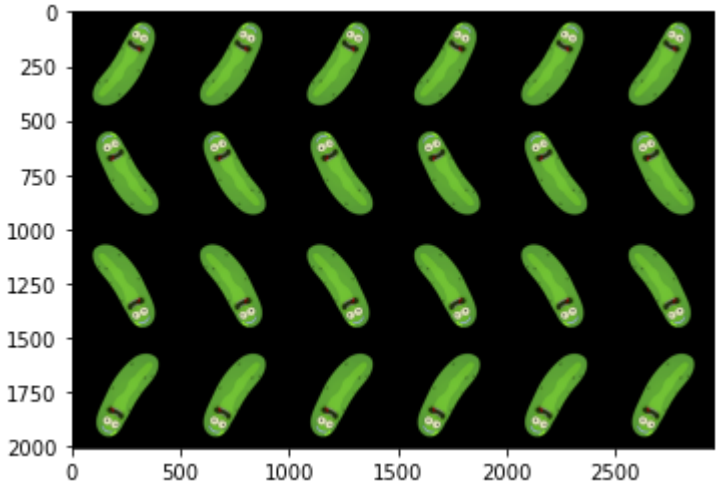
I chose the following image:

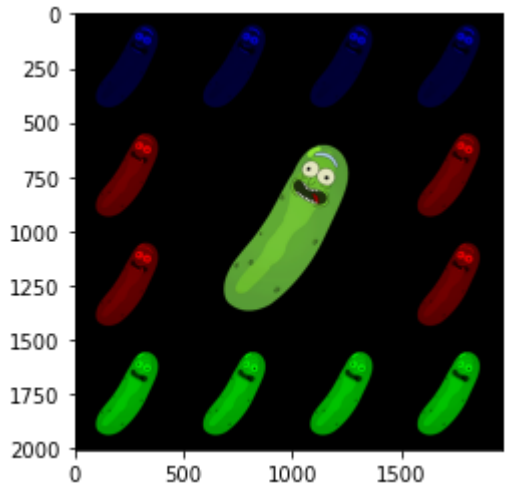


Step 4: Do my image manipulations

Use Numpy to create all of the following manipulated versions of the original image:







Step 5: Generalize these manipulations Generalize the image manipulations that you created in **Step 4**, generalize to the following (or even more general if you want):

- Define a function `grid_with_flips(image, matrix)`, where `matrix` is a matrix containing the type of flips that you do with your image. You could say `0` for you image not flipped, `1` for flipping your image left right, `2` for flipping it upside down and `3` for flipping it both left right and upside down. For

example the two images you created in **Step 4** would be the result of using the matrix: `[[1 for i in range(7)] for j in range(3)]` and `[[j for i in range(7)] for i in range(3)]`.

- Define a function `create_colorful_big_one(colors)` where `colors` is a list of colors (starting left top and rotating clockwise). The image from `Step 4` is the result of calling the function `create_colorful_big_one(['b', 'b', 'b', 'b', 'r', 'r', 'g', 'g', 'g', 'g', 'r', 'r'])`.

Replace the code your wrote in **Step 4** by 2 simple function calls from the functions you defined in this step.

Step 6: Your own manipulations Choose 2 more ways in which you manipulate your image and implement them.

Step 7: Share your solution

Compose a *README.md* file to document and guide users through your project. Include clear instructions on setting up the project, and executing the simulation. Provide concise explanations of the project structure and include an environment *YAML* file that I can use to setup a virtual environment.

By completing these steps, you will establish a well-organized project structure, manage your code using version control, create a virtual environment for efficient development, and develop a simulation program to model the production process effectively.