Python Course

Exercises #3

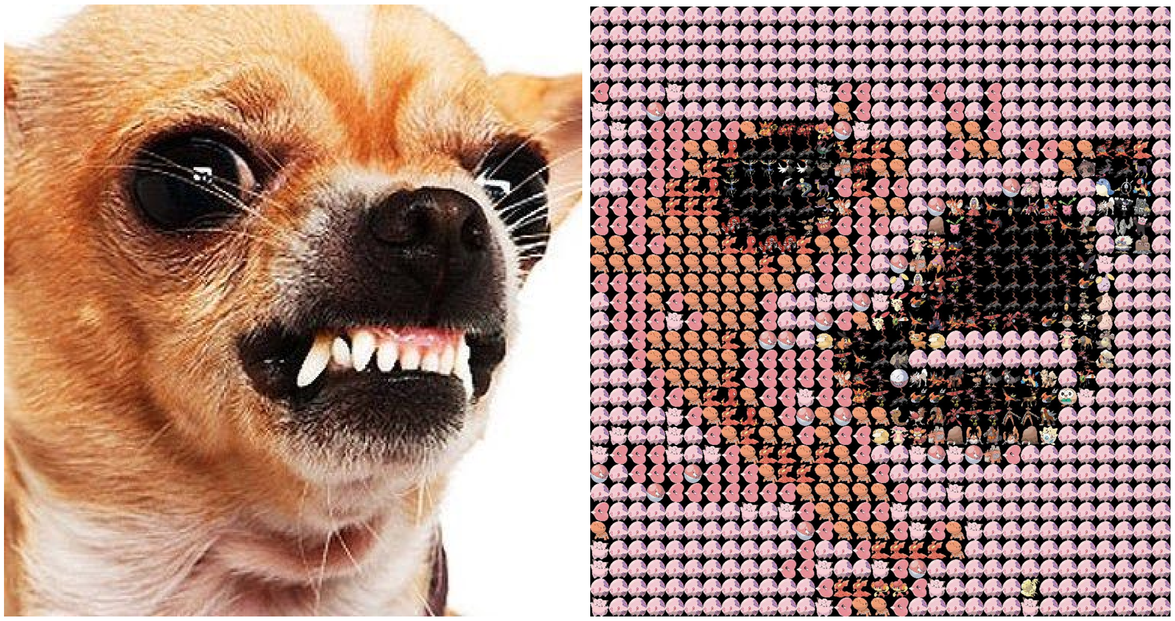
Purpose

In this unit, we will start implementing the final project.

Project

I have a running version of the project on my site. You can upload images and turn them into pokemon filled ones. You can find the code for the project on my Github.

We'll be developing an application that provides the same functionality but with live video through the remainder of this course. The following exercises contain snippets of functionality required to build the final program.



Original image and pokemon version

Reading JSON Files

The JSON library that ships Python 3 provides functions that let us read and write to JSON files. A JSON file can contain primitive types, lists, and dictionaries. When loaded from a file, the data either has a type of list or dictionary.

In the **“loadJSON.py”** script, we initially load a list and iterate through it to get the height, width, and the name of images. As done with Python dictionaries, we can index a field with a string literal to get the value we want.

Exercise 301

Recalling exercise 202, of **‘ex_202.py’.** In this exercise, you have to generate a list of metadata for images. This time the data should contain the name of the image and the average color of that image. Use the images inside the **‘common/tiles’** folder.

You have to come up with an algorithm for the average color of images. In simple terms, you have to **average each color channel** and store the results as float numbers.

Store the resulting JSON into the out folder.

An example of a metadata object is as follows:

```
{
  "name": "0001.png",
  "average_color": [85.8, 114.5, 97.1]
}
```

The average color field represents red, green, and blue respectively. You can change this in your implementation, for example to:

“average_color”: {“red”: 85.8, “green”: 114.5, “blue”: 97.1}

Bonus challenge: Ignore the black color that is present in every Pokemon tile when calculating the average color. This step is optional.

Exercise 302

Building upon the previous exercise, you’re expected to display a Pokemon tile with its respective average color next to it as a tile with the same size as the Pokemon tile.

Examine the **“loadFile.py”** file to get an insight into how to read JSON files and process them.

Keep in mind that OpenCV uses the BGR color space. If you’ve saved the average color as RGB, then you have to convert it to BGR.

Use this Numpy function to create a color tile image with the same size as the Pokemon tile:

avgColorTile = np.full(shape=(img_height, img_width, 3), fill_value=*{your_average_color}*, dtype=np.uint8)

The **fill_value** has to be a list that represents the average color of the Pokemon tile. For example, for the color green in RGB color space this would be **[0, 255, 0].**

You can then either save the resulting images or display them in an OpenCV window.

Some example outputs: