CMPSC 102
Discrete Structures
Fall 2018

Practical 6: Creating an Analysis of Snow From Images, Determining Colour in images and Hacking Python Code

Refer to your notes, slides and sample Python code from this week and other weeks.

Summary

Image analysis using algorithms has become an important area of computer science for its ability to scan countless pieces of graphical data (files) to search for specific types of information (parsing). In environmental research, for example, image analysis can be done by automated means and is used to determine erosion, changes in the landscape and other types of environmental change over time. For instance, by studying a series of satellite images of mountain ranges covered in various amounts of snow, taken at the same time of year spanning across several years, image analysis is able to determine the differences in snow fall. This data analysis helps to drive further research learn of environmental change and motivates to create policies of environmental protection.

GitHub Starter Link

`https://classroom.github.com/a/otqFouaH`

To use this link, please follow the steps below.

- Click on the link and accept the assignment.
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the practical locally.
- As you are working on your practical, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):
  - `git commit <nameOfFile> -m ''Your notes about commit here''`
  - `git push`

  Alternatively, you can use the following commands to add multiple files from your repository:
  - `git add -A`
  - `git commit -m ''Your notes about commit here''`
  - `git push`

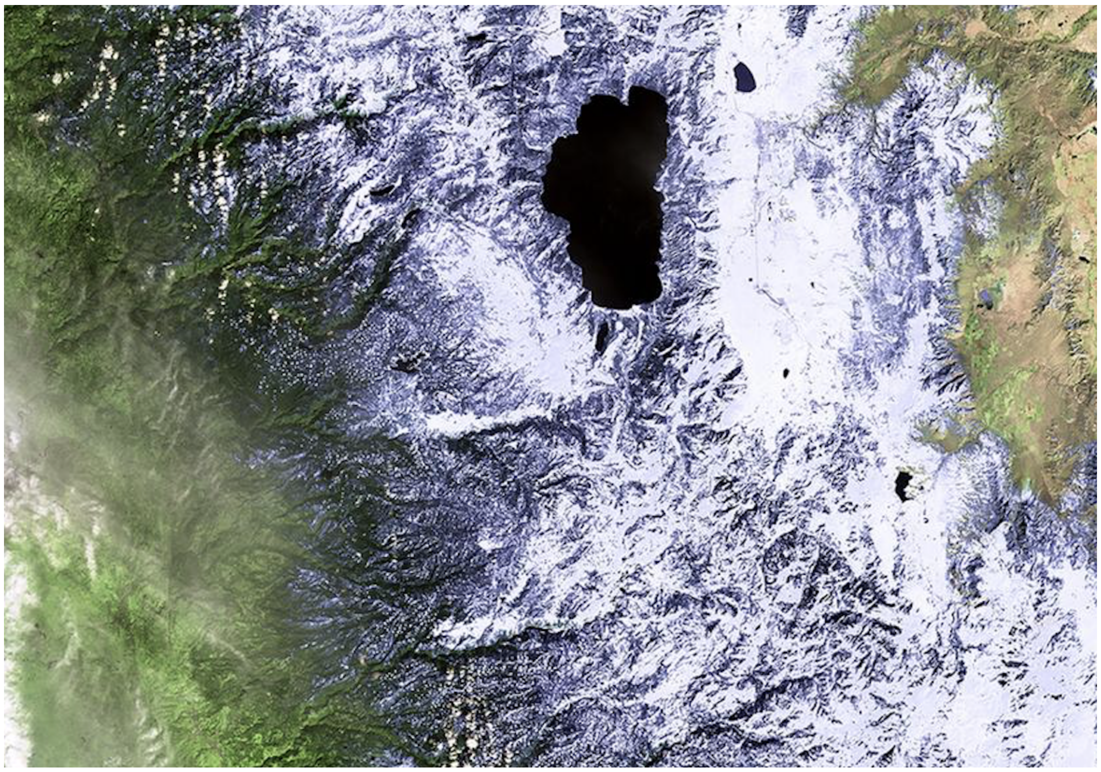

Figure 1: A satellite image of a California mountain range showing lots of white snow on the range. If we look at more recent photos, is there less white snow to see?

What to do for this practical

True and False statements can be used in a wide variety of applications. You will note that they have been used for parsing image files. In this practical, you will be checking images for snow coverage using a supplied Python program. You will then *hack* the program to connect another given function `getAvgRGB()` which works to determine the average thresholds of red, green and blues which are found in image files. Finally, you will create a new function to answer a question of your own choosing about image colour composition.

Study and experiment with the given program

Locate the program source code (file: `src/imageScanner.py`) with which you are to work. Also, note that there are some graphics files in the `graphics/` directory, such as the one of Figure 1. The files, `Feb2011.png`, `Feb2013.png` and `Feb2014.png` are satellite images of a mountain range in California from the years; 2011, 2013 and 2014. Launch your image scanner program to check that it works and that it is able to find an amount of snow coverage from each image. The program works by looking through each pixel to count those which are close to white in colour.

To run the program, type the following commands.

```
chmod +x imageScanner.py #used to make the file executable in the terminal
# then,
./imageScanner.py ../graphics/Feb2011.png ../graphics/Feb2014.png
```

Determine magnitudes of red, green and blue colour values

When you have connected the `getAvgRGB()` function to your `main()` function by un-commenting its code in the `main()` function, your output now includes new data. Run `imageScanner.py` using the file `green.png` as a parameter. Type the following command.

```
./imageScanner.py ../graphics/green.png
```

Your additional output is the following (from this part of the program).

```
 * graphics/green.png    The average RGB colours are the following :
                           [0.0, 0.65, 0.36]
```

This output represents the average magnitudes of red, green and blue pixels, respectively, that were found in the `green.png` file. There are other image files with which you may experiment. Since each of these colour files contains only one colour, the *average* of the magnitudes may be understood to be the measurement of the only hue (i.e., red, green or blue) in the entire image.

Write a new function, based on `computeSnow()`, to determine different colour variations

Now create your own function that parses pixels to determine other colours or colour combinations. You will first need to come up with the question your function will try to answer. For example, can you write a function that determines greenery in an image or red rock formations?

Your new function can be similar to the `computeSnow()` method and it should scan through the satellite images (i.e., `Feb2011.png`, `Feb2013.png` and `Feb2014.png`) to answer the question you decide to tackle. You can also use the average thresholds to look for combinations of colours in images to detect changes between the satellite imagers. For example, can you determine if the changes in the images from year to year is due to differing foliage or water coverage? What about the any change (from erosion, for instance) from year to year concerning the red hues from the rocks? You can also use the other images (plain green, blue and red) included in this practical to test your new function. **Please be sure to leave the original functions intact when you add your new function to the code.**

Be sure to have fun with this open-ended portion of the practical and to explore your code. See what else you can hack from this code. Please remember to submit your updated code in `src/imageScanner.py` and `writing/miniReflection.md` document to reflect on the snow coverage program and your extension of it.

Deliverables

1. Your completed python code (`src/imageScanner.py`)

2. Your mini-reflection Markdown document `writing/miniReflection.md` to answer the questions posed in this document (one or two sentences for each question).