

Calculating relative brightness in Python for Schizophrenia research

Alistair Walsh
Swinburne University of Technology
Phil Sumner
Swinburne University of Technology

August 24, 2014

Often the visual and semantic systems of Schizophrenia patients are affected by their illness [1]. Due to this, researchers often need to quantify and modify the brightness of images used in experiments. While MATLAB is commonly used in academia, Python is an easy to learn and versatile program for this work. This program was written because of a specific needs in one experiment and is offered here to help others with similar requirements.

Brightness and Semantics

The brightness of an image may draw attention for being bright rather than any semantic meaning of the image content. In semantic research this may cause a confound where eye tracking is being used to determine aspects of attention and semantic processing. Comparing two images and quantifying their brightness and then being able to modify the brightness of the images so they are of equal brightness is a common task. When there are tens or even hundreds of images, a programatic solution is desirable. Not just for the sake of saving wear and tear on research assistants, but the risk of human error in a boring and repetitive task is high.

Computer Images

A computer image is a grid of pixels, with each pixel being three colours at individual brightnesses. This could be visualised as a matrix of tuples (56,255,4). A simple brightness value can be derived from the average of the three colour values and then an average of all the pixels in the image. This doesn't account for local bright areas within the image but is a reasonable starting point. Another approach would be to visualise each pixel colour in its own matrix and average each matrix before averaging the three colour brightness values.

Python Code to calculate brightness

The Python Imaging Library (PIL) has functions for many of these common tasks.

```
from PIL import Image
import csv
#create a fake brightness value for now
brightness = 255/2
#setup a .csv file to write to
with open('brightness.csv', 'wb') as csvfile:
    # create a file name variable so later when there is a list of
    # files, it can iterate over the list
    for item in file_list:
        fname = item
        # Load the image into imag
        imag = Image.open(fname)
        ## Show me the image
        imag.show()
        #write the file_name to .csv and add brightness value
        brightness_writer = csv.writer(csvfile,
            delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
        brightness_writer.writerow([item] + [brightness])
```

References

- [1] Anthony S. David and John C. Cutting. Visual imagery and visual semantics in the cerebral hemispheres in schizophrenia. *Schizophrenia Research*, 8(3):263 – 271, 1993.