CSE 190D PIXELS: COMPUTATION WITH IMAGES

FINAL PROJECT REPORT, 6 JUNE 2013

Developer & Researcher: David N. Ruiz

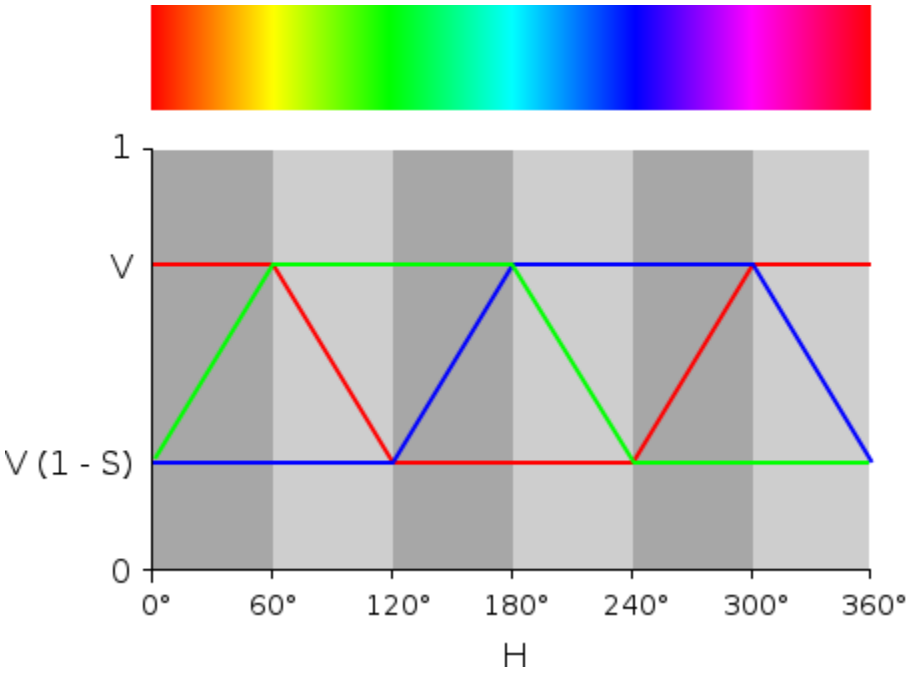
Image Search Data

Creating a search algorithm is a well-defined and complicated matter. It attempts to find similarities between images by looking at image data and creating a vector that compares images based on name, hue, size and many other details. Unfortunately data is not perfect my project is an approach to retrieve more data that can become helpful when trying to find similarities and differences between images.

I was able to retrieve detailed image data by looking at the tags of an image and storing them in a python dict where I can later loop through them for retrieval. Unfortunately not all images contain these tags and so I am left with minimal data to use for comparison. Hue however is a fail-safe approach to comparing images which was the second major piece of data I retrieved from the image. In order to get the hue I used a custom algorithm that retrieved all the rgb values of each pixel in the image. Then I transformed the rgb values into h (ignoring s and v) values.

I used the HSV spectrum and equation to guide me with the final hue selection process.

Spectrum



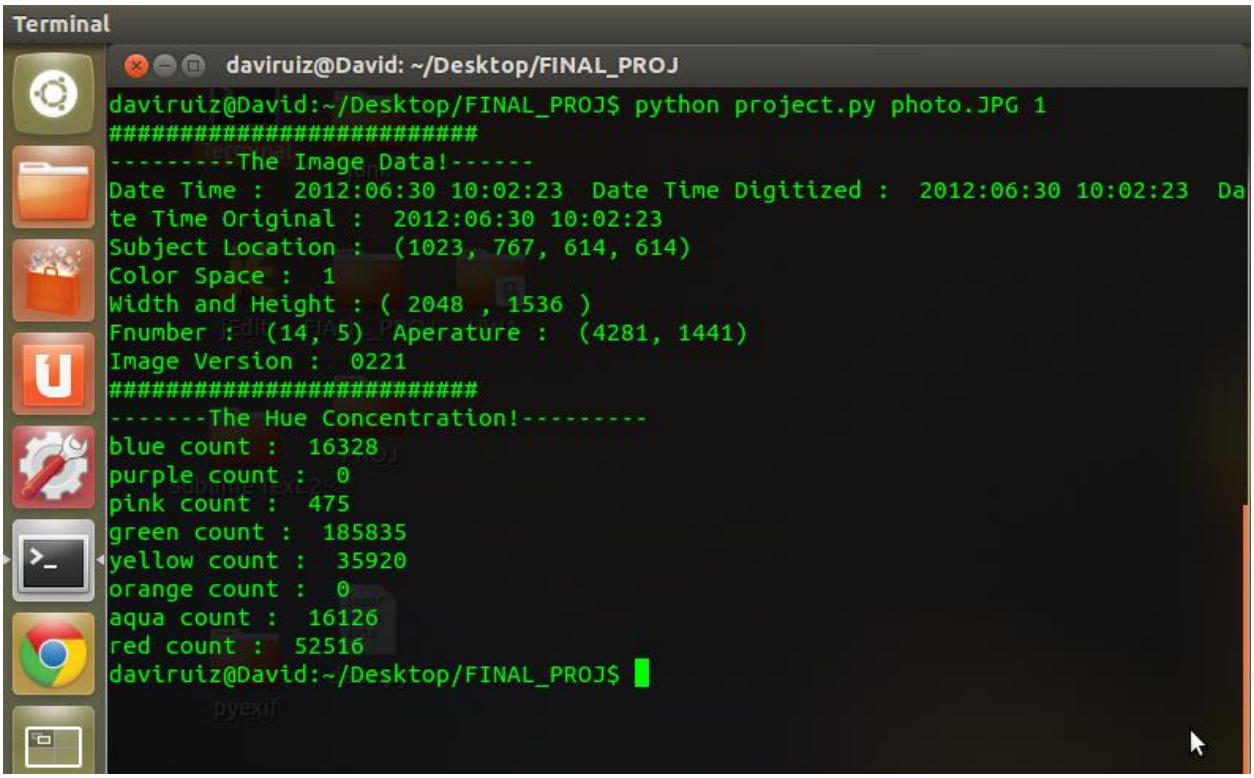
Equation

$$H' = \begin{cases} \text{undefined}, & \text{if } C = 0 \\ \frac{G-B}{C} \bmod 6, & \text{if } M = R \\ \frac{B-R}{C} + 2, & \text{if } M = G \\ \frac{R-G}{C} + 4, & \text{if } M = B \end{cases}$$

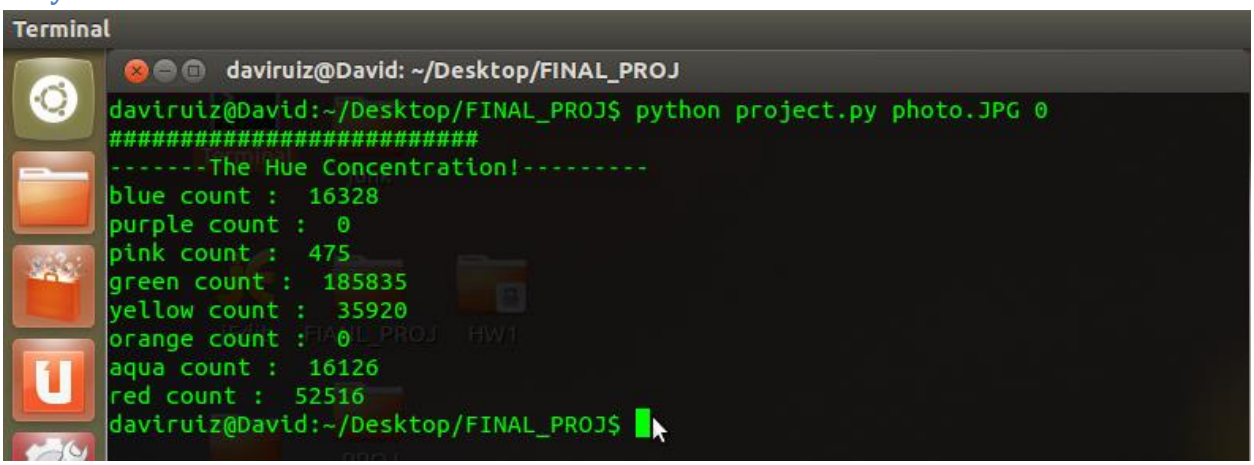
$$H = 60^\circ \times H'$$

Demos and Screenshots

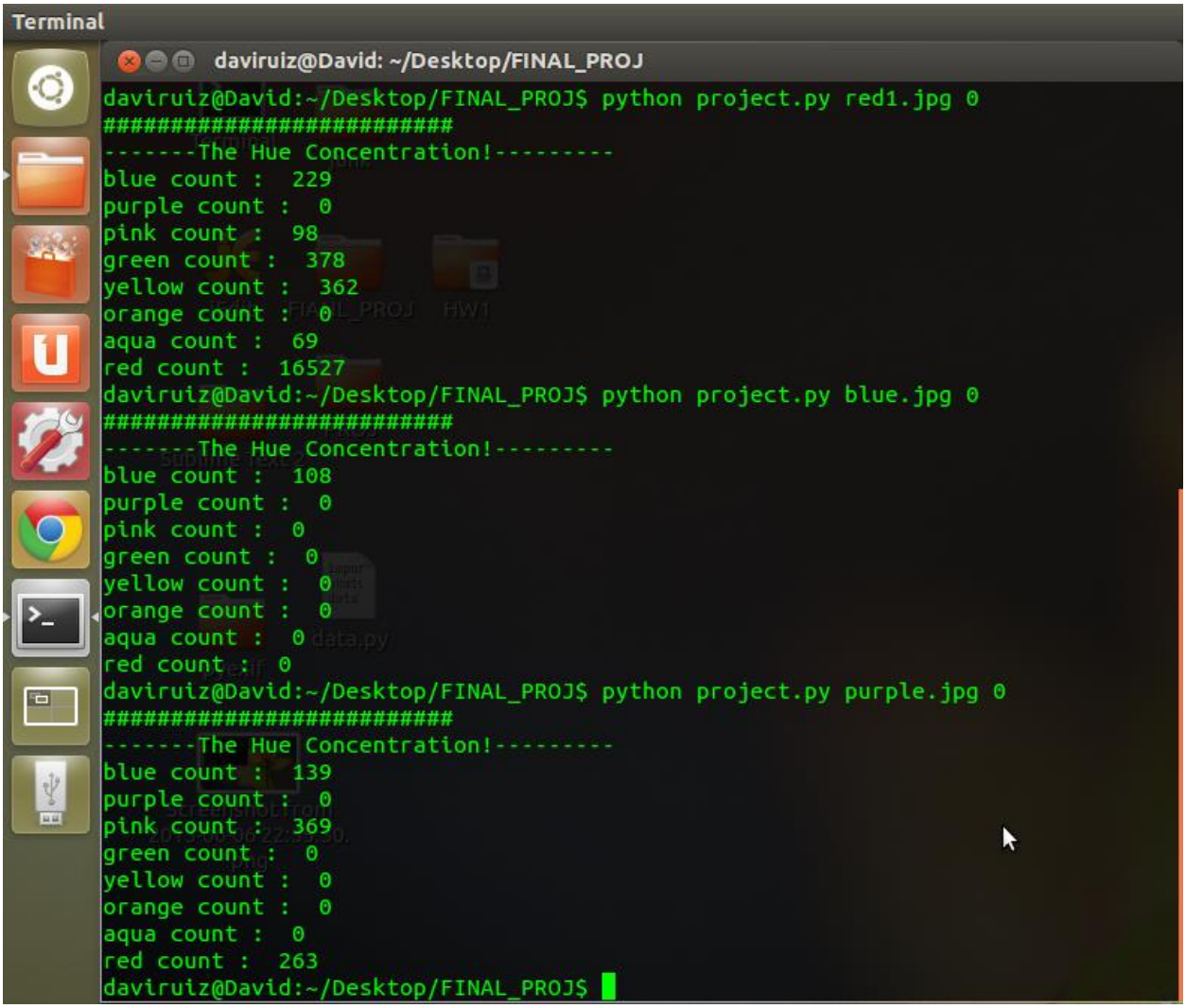
Data and Hue



Only Hue



Color Hues



Brief Demo Instructions

When you open the terminal you type in

python project.py {path/image.file} {image_data either 1 or 0}

If you took the image yourself it is likely to have image data in which case you want to use 1 which indicates true for image data. If downloaded or an image that was obviously created it is likely not to have data in which case you will want to switch image_data to 0 or false.

Command should look like

$ python project.py image.jpg 1

or

$ python project.py image.jpg 0

Code – Hue Formula

In the categorize function I used the following equation to compute the Hue value in the spectrum from 0 degrees to 360 degrees

computedH = 60 *(h - d/(maxRGB - minRGB));

Reflection

I learned to work with sys, PIL, EXIF python packages. I also learned to import packages parse command line arguments. In previous attempts I also worked with sqlite, webpy and django all which proved to be a much more challenging and time consuming aspect.

At a certain point I learned that it was more important to break the process into small steps and quickly eliminate aspects of the goals that seemed unrealistic.

I learned how to convert RGB values to HSV values and find a way to customize an algorithm for my needs.

The Future

If I had more time I would like to make a web application that can use this code and then link with the Google image search API to use the web as my data resource rather than create my own database. I would also like to use request python module and flask or django to create a web application that can make ajax request to the API.

Appendix A

Code

```
from PIL import Image
from PIL.ExifTags import TAGS
import sys

global image_data_color, image_data

hue_dict = {'red':0, 'orange':0, 'yellow':0, 'green':0, 'aqua':0, 'blue':0, 'purple':0, 'pink':0}

def display_hues():
    print "#########################","\n","-------The Hue Concentration!---------"
    for color in hue_dict:
        print color, "count : ", hue_dict[color]

def display_data():
    print "#########################","\n","---------The Image Data!------"
    print 'Date Time : ',image_data['DateTime']
    print 'Date Time Digitized : ',image_data['DateTimeDigitized']
    print 'Date Time Original : ', image_data['DateTimeOriginal']
    print 'Subject Location : ', image_data['SubjectLocation']
    print 'Color Space : ', image_data['ColorSpace']
    print 'Width and Height : (', image_data['ExifImageWidth'],',',image_data['ExifImageHeight'],')'
    print 'Fnumber : ', image_data['FNumber'], ' Aperature : ', image_data['ApertureValue']
    print 'Image Version : ', image_data['ExifVersion']

def main():
    global image_data_color
    sent_file = open(sys.argv[1])
    data_present = sys.argv[2]

    if data_present == '1':
        get_exif(sent_file)
        categorize()
        display_data()
        display_hues()
    else:
        img = Image.open(sent_file)
        image_data_color = list(img.getdata())
        categorize()
        display_hues()
```

```
def get_exif(fn):
    global image_data_color, image_data
    image_data = {}
    i = Image.open(fn)
    image_data_color = list(i.getdata())
    info = i._getexif()
    for tag, value in info.items():
        decoded = TAGS.get(tag, tag)
        image_data[decoded] = value

def categorize():
    for rgb in image_data_color:
        r = rgb[0]/255
        g = rgb[1]/255
        b = rgb[2]/255

        minRGB = min(r, min(g,b))
        maxRGB = max(r, max(g,b))
        d = 0
        h = 0

        if r == minRGB:
            d = g -b
            h = 3
        elif b == minRGB:
            d = r -g
            h = 1
        else:
            d = b -r
            h = 5
```

```
        if r == minRGB:
            d = g -b
            h = 3
        elif b == minRGB:
            d = r -g
            h = 1
        else:
            d = b -r
            h = 5

        if minRGB != maxRGB:
            computedH = 60 *( h - d/(maxRGB - minRGB));
            if computedH <= 5:
                hue_dict['red'] += 1
            elif computedH > 5 and computedH <= 55:
                hue_dict['orange'] += 1
            elif computedH > 55 and computedH <= 65:
                hue_dict['yellow'] += 1
            elif computedH > 65 and computedH <= 145:
                hue_dict['green'] += 1
            elif computedH > 145 and computedH <= 185:
                hue_dict['aqua'] += 1
            elif computedH > 185 and computedH <= 245:
                hue_dict['blue'] += 1
            elif computedH > 245 and computedH <= 280:
                hue_dict['purple'] += 1
            elif computedH > 280 and computedH <= 320:
                hue_dict['pink'] += 1
            else:
                hue_dict['red'] += 1

if __name__ == '__main__':
    main()
```

Packages

The following two modules where used extensively for this project PIL and sys.