Data Structures and Programming Spring 2019

Prog #2

Student: 謝昌諭

ID : b06602052

Implementation

Each part of program

The first part is for input file and check form. Using open() to read input file line by line. Then, we change each line into list type and remove the last term which is the '\n'. Now, we have a list with element are content of this pgm's each line. If 'P2' is not the first element of this list, program print error message and exit(), for the form is wrong.

The second part is for getting the size, max of image and remove comment in file. Using while loop to remove all line with # as start. Then, pop the first two elements in list. Saving first one in new list named size and second one in a integer variable named max for the below operation.

The third part is for reshape the list into matrix. Combine all the elements into a big list and change it into array by numpy. Then, reshape this array to matrix. In there, we need the size get above.

The fourth part is the algorithm itself. Build a new matrix by former size and assign new pixel value determined by algorithm.

- Pseudo code

```
for row from 0 to size[1]-1
    for column from 0 to size[0]-1
        // deal with special case in four corners with assume the outer pixel are all 255
        If x and y equal to 0 or size[0 or 1]-1    ……
        // deal with special ase in four sides with assume the outer pixel are all 255
        eIif x and y equal to size[0 or 1]-1 and x and y not equal to 0    ……
        // remaining part is middle which doesn't need other operation
        else ……
        if x^2 + y^2 > 128^2
            new_pixel = 0
        else
            new_pixel = 255
```

The last part is for output file. Using Image in PIL module to change matrix to image and save it as a pgm file.

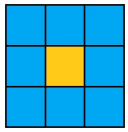
Program running

Run this in cmd need three parameter

➔ python file_name.py X.pgm

Bonus

Method to do edge detection



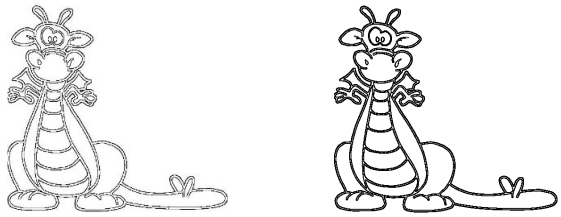
My design is getting the average of each pixel value of around square (blue area in above graph). If the T (target, yellow area) is bigger than this AVG, then T will be black. The most outside pixel are assign white directly. And I add a parameter (activate) to determine how much of the difference will be valid. For example, if activate is 20, then T will be assign black when original T minus AVG is more than 20.

- Pseudo code:

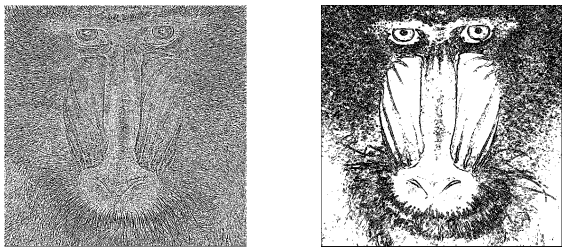
```
for row from 0 to size[1]-1
    for column from 0 to size[0]-1
        If current pixel is in the most outside
            New_pixel = 255
        else                        // in the middle of image
            //    1 2 3
            //    4 T 5
            //    6 7 8
            I~VIII = values of corresponding pixel
            AVG = sum(I~VIII) / 8 //compute the average
        activate = threshold // threshold is an input parameter to determine valid difference
        if pixel of T – AVG > activate
            New_pixel = 0
        else
            New_pixel = 255
```

Idea and concept

Using average is because if we do edge detection on images have simple composition and significant differential color which like American comics, images are always having black frame and very bright color. In this condition, the pixel value change is large in edge. Therefore, performance will be great.



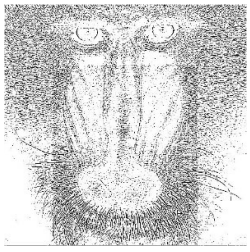
In above picture, left is my algorithm and right is designated one. It is obvious that the edge is more thin.



However, if the image is a complex one, the performance will be extremely bad as above two images. left is my algorithm and right is designated one. So, there is a parameter to determine how much difference can be viewed as valid.

My idea is that when we do edge detection on a picture with very similar but different color (like baboon in the test file), new generated file will be very dirty, for difference is very small, which cause the new image have many black point. If we increase threshold, the influence of small difference will be ignored.

The activate of above four are respectively 5,10,15,20. We can find that some detail disappears as activate increase. But, this algorithm can make activate effect on only specific region. That is user can print many images with distinct activate and then choose use what part of each image.



The middle part use activate equal to 9 and the other use 20. We preserve the

detail in cheek and reduce point outside.

Advantage

1. Easy to implementation and understand
2. Doing on image with boundary is good
3. Can use activate to adjust some specific area

Disadvantage

1. Deal with complex, real photo will cause dirty result
2. The output need manual adjust. Can't reach real automatic.

Program running

Run this in cmd need three parameter

➔ python file_name.py X.pgm activate

Reference

Wikipedia

https://zh.wikipedia.org/wiki/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B

演算法筆記

http://www.csie.ntnu.edu.tw/~u91029/Image.html