

Homework1_Report

1. Problem1

Please see the “pictureMe.png”.

2. Problem2

2.1. Read and show the image

The Python code reading and showing the image is as follows:

```
from PIL import Image
im=Image.open("G:wolves.png")
im.show()
```

Therefore, we can see the screenshot from the file named as “pictureMe.png”.

2.2. Find the digit signature of your unity id

2.2.1. Count numbers

The Python code counting number of occurrence of each of the digit number in each color channel of the image is as follows:

```
from PIL import Image
im=Image.open("G:wolves.png")
r,g,b=im.split()
print(r.size)
print(g.size)
print(b.size)
print(im.size)                                     # Know the numbers of row and column
```

```
sequ_r=r.getdata()
sequ_g=g.getdata()
sequ_b=b.getdata()
sequ=im.getdata()
sequ0=list(sequ)
sequ1=list(sequ_r)
sequ2=list(sequ_g)
sequ3=list(sequ_b)                                # Get data of three channels
```

```
n=len(sequ0)
```

Operate the red channel:

```
i,y_num1,f_num1,u_num1,_28_num1=0,0,0,0,0
```

```
while (i<n):
```

```
    if sequ1[i]==121: y_num1+=1      # Count the number of "y" in the red channel
    if sequ1[i]==102: f_num1+=1     # Count the number of "f" in the red channel
    if sequ1[i]==117: u_num1+=1     # Count the number of "u" in the red channel
    if sequ1[i]==28: _28_num1+=1    # Count the number of "28" in the red channel
    i+=1
```

Operate the green channel:

```
i,y_num2,f_num2,u_num2,_28_num2=0,0,0,0,0
```

```
while (i<n):
```

```
    if sequ2[i]==121: y_num2+=1      # Count the number of "y" in the green channel
    if sequ2[i]==102: f_num2+=1      # Count the number of "f" in the green channel
    if sequ2[i]==117: u_num2+=1      # Count the number of "u" in the green channel
    if sequ2[i]==28: _28_num2+=1     # Count the number of "28" in the green channel
    i+=1
```

Operate the blue channel:

```
i,y_num3,f_num3,u_num3,_28_num3=0,0,0,0,0
```

```
while (i<n):
```

```
    if sequ3[i]==121: y_num3+=1      # Count the number of "y" in the blue channel
    if sequ3[i]==102: f_num3+=1      # Count the number of "f" in the blue channel
    if sequ3[i]==117: u_num3+=1      # Count the number of "u" in the blue channel
    if sequ3[i]==28: _28_num3+=1     # Count the number of "28" in the blue channel
    i+=1
```

Show the numbers of occurrence:

```
print('y_num1=',y_num1,',f_num1=',f_num1,',u_num1=',u_num1,',and _28_num1=',_28_num1,':')
print('y_num2=',y_num2,',f_num2=',f_num2,',u_num2=',u_num2,',and _28_num2=',_28_num2,':')
print('y_num3=',y_num3,',f_num3=',f_num3,',u_num3=',u_num3,',and _28_num3=',_28_num3,':')
```

After running the code, I can summarize my results as shown in Table 1:

Table 1. The numbers of occurrence

Character	ASCII	# in Red channel	# in Green channel	# in Blue channel
'y'	121	2336	2942	1919
'f'	102	2785	3378	2478
'u'	117	2474	3115	2015
28	28	5525	5913	7767

2.2.2. Change values

In this section, I am about to change the pixel values of the 5 by 5 sub-image centered at each occurrence into 255 and then show the result image. The python code is as follows:

```
from PIL import Image
from pylab import *
im=Image.open("G:wolves.png")
```

Split the image into three channels and get their matrix data individually:

```
r,g,b=im.split()
arr_r=array(r)
arr_g=array(g)
arr_b=array(b)
```

Define the function to change values to meet requirements:

```
def change_values(arr_value1,value2,value3,value4):
    for i in range(0,538):
        for j in range(0,1499):
            if (i<=1 or j<=1 or i>=537 or j>=1498) and (arr[i,j]== value1 or arr[i,j]== value2 or arr[i,j]==
```

```

value3 or arr_[i,j]== value4):                                # Special cares for locations near image boundary
    if i==0:
        if j==0:
            for m in range(0,2):
                for n in range(0,2):arr_[m,n]=255
        elif j==1:
            for m in range(0,2):
                for n in range(0,3):arr_[m,n]=255
        elif j==1498:
            for m in range(0,2):
                for n in range(1496,1499):arr_[m,n]=255
        elif j==1499:
            for m in range(0,2):
                for n in range(1497,1499):arr_[m,n]=255
    elif i==1:
        if j==0:
            for m in range(0,3):
                for n in range(0,2):arr_[m,n]=255
        elif j==1:
            for m in range(0,3):
                for n in range(0,3):arr_[m,n]=255
        elif j==1498:
            for m in range(0,3):
                for n in range(1496,1499):arr_[m,n]=255
        elif j==1499:
            for m in range(0,3):
                for n in range(1497,1499):arr_[m,n]=255
    elif i==537:
        if j==0:
            for m in range(535,538):
                for n in range(0,2):arr_[m,n]=255
        elif j==1:
            for m in range(535,538):
                for n in range(0,3):arr_[m,n]=255
        elif j==1498:
            for m in range(535,538):
                for n in range(1496,1499):arr_[m,n]=255
        elif j==1499:
            for m in range(535,538):
                for n in range(1497,1499):arr_[m,n]=255
    elif i==538:
        if j==0:
            for m in range(536,538):
                for n in range(0,2):arr_[m,n]=255

```

```

elif j==1:
    for m in range(536,538):
        for n in range(0,3):arr_[m,n]=255
elif j==1498:
    for m in range(536,538):
        for n in range(1496,1499):arr_[m,n]=255
elif j==1499:
    for m in range(536,538):
        for n in range(1497,1499):arr_[m,n]=255

if (i>1 and j>1 and i<537 and j<1498) and (arr_[i,j]== value1 or arr_[i,j]== value2 or arr_[i,j]==
value3 or arr_[i,j]== value4):
    # Normal operations to change values
    for m in range(i-2,i+2):
        for n in range(j-2,j+2):
            arr_[m,n]=255

return arr_

arr_r=change_values(arr_r,121,102,117,28)    # Operate the red-channel image
new_r=Image.fromarray(arr_r)
arr_g=change_values(arr_g,121,102,117,28)    # Operate the green-channel image
new_g=Image.fromarray(arr_g)
arr_b=change_values(arr_b,121,102,117,28)    # Operate the blue-channel image
new_b=Image.fromarray(arr_b)

new_im=[new_r,new_g,new_b]                   # Merge the image
im_merge=Image.merge("RGB",new_im)
im_merge.show()                             # Show the result image
im_merge.save("G:yfu28_signature.png")       # Save the result image

```

After running the code, we could see the image from the file named as “ yfu28_signature.png”.