Python 图片转字符画

一、实验说明

1. 环境登录

无需密码自动登录，系统用户名shiyanlou，密码shiyanlou

2. 环境介绍

本实验环境采用带桌面的Ubuntu Linux环境，实验中会用到桌面上的程序：

1. LX终端（LXTerminal）: Linux命令行终端，打开后会进入Bash环境，可以使用Linux命令
2. GVim：非常好用的编辑器，最简单的用法可以参考课程[Vim编辑器](http://www.shiyanlou.com/courses/2)

3. 环境使用

使用GVim编辑器输入实验所需的代码及文件，使用LX终端（LXTerminal）运行所需命令进行操作。

完成实验后可以点击桌面上方的“实验截图”保存并分享实验结果到微博，向好友展示自己的学习进度。实验楼提供后台系统截图，可以真实有效证明您已经完成了实验。

实验记录页面可以在“我的主页”中查看，其中含有每次实验的截图及笔记，以及每次实验的有效学习时间（指的是在实验桌面内操作的时间，如果没有操作，系统会记录为发呆时间）。这些都是您学习的真实性证明。

二、实验内容

安装 pillow（PIL）库：

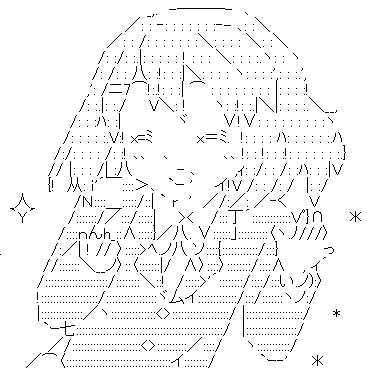
$ sudo apt-get install python-dev

$ sudo apt-get install libtiff5-dev libjpeg8-dev zlib1g-dev \

libfreetype6-dev liblcms2-dev libwebp-dev tcl8.6-dev tk8.6-dev python-tk

$ sudo pip install pillow

今天我们要做的是图片转字符画!



别慌！这是手工做的，我们这个做不了那么好。



我们做的是这一类字符画！



才几行代码，哪可能做成这种效果。



别急别急，这里只负责讲解原理与生成字符，至于显示效果，那还要看你显示用的媒介怎么样了，像是字符是不是等宽啦，行高和行宽是不是相等啦，屏幕够不够大，够不够亮，够不够有型都可能是影响观赏效果的因素呢。

开始码代码

首先获取实验用图片

wget http://labfile.oss.aliyuncs.com/courses/370/ascii\_dora.png



创建 ascii.py 文件进行编辑

$ vi ascii.py

导入必要的库

from PIL import Image

import argparse

一张图片从图像到字符不是一蹴而就的，需要经过很多步骤，光看着可能没什么头绪，先从成品上着手，成品是一系列字符的组合，我们可以把字符看作是比较大块的像素，一个字符能表现一种颜色（暂且这么理解吧），字符的种类越多，可以表现的颜色也越多，图片也会更有层次感，下面是我们选择的字符集。

ascii\_char = list("$@B%8&WM#\*oahkbdpqwmZO0QLCJUYXzcvunxrjft/\|()1{}[]?-\_+~<>i!lI;:,\"^`'. ")

没有自己用手去数吧？告诉你，一共有 70 个字符，当然也不是字符越多越好，具体还得看选择的图片，自己反复尝试效果。

问题来了，我们是要转换一张彩色的图片，这么这么多的颜色，要怎么对应到字符上去？这里就要介绍灰度值的概念了。

灰度值：指黑白图像中点的颜色深度，范围一般从0到255，白色为255，黑色为0，故黑白图片也称灰度图像

这样就好办了，灰度值大的用列表开头的符号，灰度值小的用列表末尾的符号。

灰度值公式

灰度值 ＝ 0.2126 \* r + 0.7152 \* g + 0.0722 \* b

先完成RGB值转字符的函数：

ascii\_char = list("$@B%8&WM#\*oahkbdpqwmZO0QLCJUYXzcvunxrjft/\|()1{}[]?-\_+~<>i!lI;:,\"^`'. ")

def get\_char(r,b,g,alpha = 256):

if alpha == 0:

return ' '

length = len(ascii\_char)

gray = int(0.2126 \* r + 0.7152 \* g + 0.0722 \* b)

unit = (256.0 + 1)/length

return ascii\_char[int(gray/unit)]

完整参考代码：

from PIL import Image

import argparse

#命令行输入参数处理

parser = argparse.ArgumentParser()

parser.add\_argument('file') #输入文件

parser.add\_argument('-o', '--output') #输出文件

parser.add\_argument('--width', type = int, default = 80) #输出字符画宽

parser.add\_argument('--height', type = int, default = 80) #输出字符画高

#获取参数

args = parser.parse\_args()

IMG = args.file

WIDTH = args.width

HEIGHT = args.height

OUTPUT = args.output

ascii\_char = list("$@B%8&WM#\*oahkbdpqwmZO0QLCJUYXzcvunxrjft/\|()1{}[]?-\_+~<>i!lI;:,\"^`'. ")

# 将256灰度映射到70个字符上

def get\_char(r,b,g,alpha = 256):

if alpha == 0:

return ' '

length = len(ascii\_char)

gray = int(0.2126 \* r + 0.7152 \* g + 0.0722 \* b)

unit = (256.0 + 1)/length

return ascii\_char[int(gray/unit)]

if \_\_name\_\_ == '\_\_main\_\_':

im = Image.open(IMG)

im = im.resize((WIDTH,HEIGHT), Image.NEAREST)

txt = ""

for i in range(HEIGHT):

for j in range(WIDTH):

txt += get\_char(\*im.getpixel((j,i)))

txt += '\n'

print txt

#字符画输出到文件

if OUTPUT:

with open(OUTPUT,'w') as f:

f.write(txt)

else:

with open("output.txt",'w') as f:

f.write(txt)



关键步骤除了那一步就没有了（摊手）

argparse是一个管理命令行参数输入的小工具，它的使用可以参考这篇教程：<http://blog.ixxoo.me/argparse.html>

运行代码查看效果吧

$ python ascii.py ascii\_dora.png

