

Python 图片转字符画

一、实验介绍

1.1 实验简介

本实验用 50 行 Python 代码完成图片转字符画小工具。通过本实验将学习到 Linux 命令行操作，Python 基础，pillow 库的使用，argparse 库的使用。

1.2 实验知识点

本节实验中我们将实践以下知识：

1. Linux 命令行操作
2. Python 基础
3. pillow 库的使用
4. argparse 库的使用 ([参考教程](#))

1.3 实验环境

- python2.7
- Xfce终端

1.4 适合人群

本课程难度简单，属于 Python 基础课程。

1.5 代码获取

你可以通过下面命令将代码下载到实验楼环境中，作为参照对比进行学习。

```
$ wget http://labfile.oss.aliyuncs.com/courses/370/ascii.py
```

二、实验原理

字符画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。

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，要怎么对应到单色的字符画上去？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

我们可以使用灰度值公式将像素的 RGB 值映射到灰度值：

```
gray = 0.2126 * r + 0.7152 * g + 0.0722 * b
```

这样就好办了，我们可以创建一个不重复的字符列表，灰度值小（暗）的用列表开头的符号，灰度值大（亮）的用列表末尾的符号。

三、开发准备

PIL 是一个 Python 图像处理库，是本课程使用的重要工具，安装 pillow (PIL) 库：

```
$ sudo apt-get update
$ 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
```

首先获取实验用图片

```
$ wget http://labfile.oss.aliyuncs.com/courses/370/ascii_dora.png
```



四、项目文件结构

```
shyanlou:ascii/ $ tree
```

```
.
├── ascii_dora.png
├── ascii.py
├── output.txt
└── test.png
```



五、实验步骤

创建 ascii.py 文件进行编辑

```
$ vi ascii.py
```

首先导入必要的库，argparse 库是用来管理命令行参数输入的

```
from PIL import Image
import argparse
```

下面是我们的字符画所使用的字符集，一共有 70 个字符，字符的种类与数量可以自己根据字符画的效果反复调试

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

下面是RGB值转字符的函数：

```
def get_char(r,g,b,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,g,b,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)))
```

输入以下命令运行脚本查看实验效果

The image shows a desktop environment with three main windows:

- Web Browser (Left):** Displays a page titled "Python 图片转字符画" (Python Image to ASCII Art). It includes a "开始代码" (Start Code) button and a section titled "首先获取实验用图片" (First get the experimental image). Below this is a code block showing a wget command to download an image from a URL. A cartoon character (Doraemon) is shown as the result of the image conversion.
- Terminal (Right):** Shows the execution of the wget command and the output of the image conversion process. The output is a large block of ASCII art code that represents the Doraemon character using various characters and symbols.
- File Explorer (Bottom):** Shows the file system structure, including a folder named "实验报告" (Experiment Report) and a file named "Doraemon.png".

终端显示的字体是不是等宽字体，终端显示的行高和行宽，输入输出的图像宽高等等，这些都会影响显示效果

我们通过这个简单的实验巩固了 Python 基础知识，希望大家在学习的过程中碰到不太熟悉的函数也要去尝试搜索理解，这样才能在实验中获得提升。

七、课后习题

尝试着把这张小羊变成一幅字符画



获取小羊图片的方法

```
$ wget http://labfile.oss.aliyuncs.com/courses/370/test.png  
来源: https://www.shiyanlou.com/courses/370/labs/1191/document
```