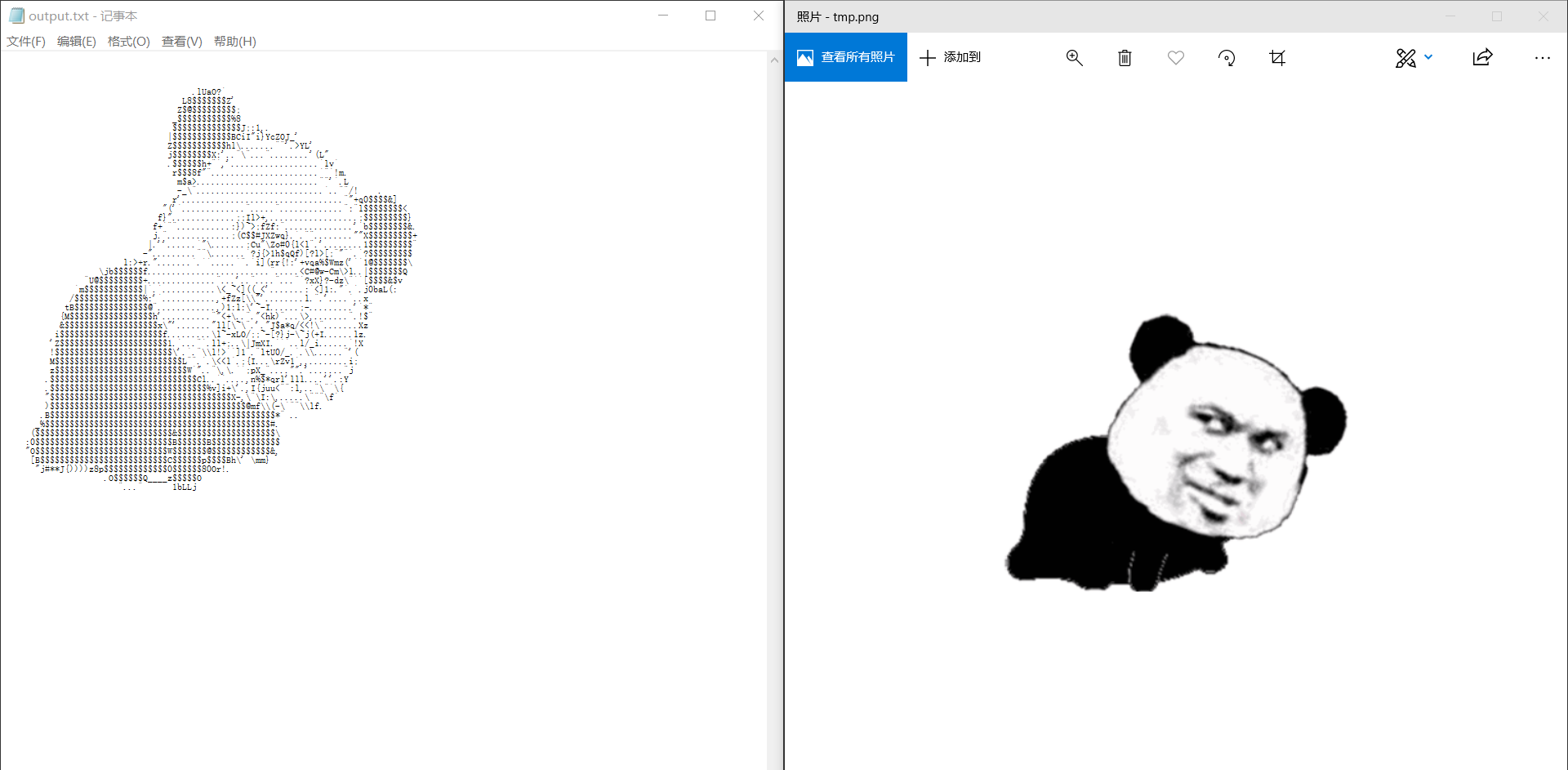
## 设计理念

作品名称：表情包转字符画

效果展示

创意设计说明：

随着互联网时代的深入发展，表情包渐渐深入人们的生活。人们借助丰富多样的表情来展示自己的情感，但部分社交平台因为图片上传方式复杂，公司带宽有限，对安全策略的考虑等等原因，对于图片上传功能有着较大的限制，这对用户非常不友好。人们渴求一种新的表情包展示方式——字符画。基于此，我设计了该程序，该程序可以将图片转化为相应的字符画，帮助用户更好的在社交平台上使用表情包。

## 程序算法部分

图片是一系列形象的组合。我们可以把字符看作是像素块。一个字符可以代表一种颜色。最后显示的是黑白字符串图片，那么如何将彩色图片转换成字符画呢?调整灰度值！

首先需要了解两个概念

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。不同的字符用于表示不同的灰度值。字符的类型和数量可以根据需要的效果进行调整，最好能看到明显的前后变化。

另外一个概念是sRGB与 RGB 色彩：

RGB色彩模式是工业界的一种颜色标准，是通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色的，RGB即是代表红、绿、蓝三个通道的颜色，这个标准几乎包括了人类视力所能感知的所有颜色，是目前运用最广的颜色系统之一。“sRGB”意味着“标准RGB颜色空间”,标准的应用范围很广,很多很多的其他硬件和软件开发商采用SRGB色彩空间的产品标准颜色空间,逐渐成为许多扫描仪和低级的打印机和软件默认的颜色空间,也使用SRGB色彩空间之间的设备,可以实现颜色仿真。同时，sRGB是一个专为网页设计师设计的色彩空间。

RGB向灰度转换时，是以LAB模式为中介的（这和PS中关于LAB模式的一贯解释是完全吻合的）。其基本的流程是：首先将RGB转换为LAB，然后在LAB中去色，使A、B均为零，并依此生成一个RGB等值的灰阶，最后再根据该灰阶向灰度空间转换，考虑使用sRGB IEC61966-2.1公式将像素的RGB值转换为灰度值：

该公式过于复杂，不适合在代码中实现，尝试将其简化得到：

这样就很容易创建一个不重复的字符列表，从小(暗)灰度值开始，以大(亮)灰度值结束。

设置每一个灰度相应的字符，在已经转换好的灰度值图片上，我们只需要将256灰度值映射到相应的字符上即可得到最终的效果。

## 代码部分

# -\*- coding:utf-8 -\*-

#使用PIL库

from PIL import Image

#文件路径，默认在当前文件夹下，更换名字即可

InputImage = r'tmp.png'

#定义输出画面的宽度

ImageWidth = 90

#定义输出画面高度

ImageHeight = 50

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

# 将256灰度映射到70个字符上

def CharReflection(r, g, b, alpha=256):

# 设置alpha透明度

if alpha == 0:

return ' '

length = len(Ascii\_Char)

# 计算灰度

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

unit = (256.0 + 1) / length

# 不同的灰度对应着不同的字符

return Ascii\_Char[int(gray / unit)]

# 通过灰度来区分色块

def main():

#图文转换

Temp = Image.open(InputImage)

Temp = Temp.resize((ImageWidth, ImageHeight), Image.NEAREST)

Text = ""

for i in range(ImageHeight):

for j in range(ImageWidth):

Text += CharReflection(\*Temp.getpixel((j, i)))

Text += '\n'\

#输出图像

print(Text)

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

f.write(Text)

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

main()