源代码

项目：MAGIC-SCAN 图片编辑系统

代码开发负责人：16020510041 陈桂海

申请分数：92

开发环境：Python3.5

```
# -*- coding: utf-8 -*-
# Created by: PyQt5 UI code generator 5.6

#3.17 更新的功能：
#   修复了打开，保存时候如果没有正确选择文件而导致的异常关闭，增加了退出时候的提醒。
#   新建了一个默认保存软件图片的文件夹。在 C 盘下面的 Magpic pictures。
#   全面美化了界面，给背景、按钮都添加了背景图片。
"""
注意的地方：
1、图片选择错误的话会自动异常退出
2、程序报出 log4cplus ： ERROR xxx 等等，可以不用理会，不影响运行
"""

from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import *
from PyQt5.QtCore import QCoreApplication
# 打开/保存文件的对话框，需要这个包：
from PyQt5.QtWidgets import QFileDialog
import PIL
from PIL import Image,ImageFilter,ImageFont,ImageDraw,ImageEnhance
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtGui import *
import sys
import os
import argparse
import cv2

# 创建默认存放文件的文件夹：
cur_dir="C:"
folder_name='Magpic pictures'
if os.path.isdir("C:/Magpic pictures"):
    print("Already exist!")
else: os.mkdir(os.path.join(cur_dir, folder_name))

# 为了字符画而创建的字符集，共 70 个字符。
ascii_char = list("$@B%8&WM#*oahkbdpqwmZOOQLCJUYXzcvunxrjft/\|()1{}[]?-_+~<>i!lI;:,\"^`'.
")


class Ui_MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.retranslateUi(self)
```

```
        self.setWindowTitle("MagicPicPS")
        self.setWindowIcon(QIcon("icon/logo.ico"))
        # 定义了一些图片，filename 存放的是打开图片的文件名，newPic 存放 P 图后的文件，
        # origPic 存放一个初始化图片，这个先忽略吧。
        self.filename="pic\\bg.png"
        self.newPic="pic\\bg.png"
        self.origPic=Image.open("E:\\python_project_space\\APP\\GUI\\pic\\bg.png")
        self.origPic.save("E:\\python_project_space\\APP\\GUI\\pic\\tempPic.png")
        self.sliderPic=""
        self.if_sliderPic=False  #是否保存 sliderPic

#----------------------窗口初始化函数：-----------------------------------------------------
---------
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1280, 700)     # 本来是 1440*900  这里换成 1280*800
        MainWindow.setStyleSheet("background")
        self.centralWidget = QtWidgets.QWidget(MainWindow)
        self.centralWidget.setObjectName("centralWidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.centralWidget)
        self.verticalLayout.setContentsMargins(11, 11, 11, 11)
        self.verticalLayout.setSpacing(0)
        self.verticalLayout.setObjectName("verticalLayout")
        self.widget = QtWidgets.QWidget(self.centralWidget)
        self.widget.setMinimumSize(QtCore.QSize(500, 500))  # 500,500
        self.widget.setStyleSheet("background-color:rgb(60,63,65);")
        self.widget.setObjectName("widget")
        self.horizontalLayout = QtWidgets.QHBoxLayout(self.widget)
        self.horizontalLayout.setContentsMargins(11, 11, 11, 11)
        self.horizontalLayout.setSpacing(6)
        self.horizontalLayout.setObjectName("horizontalLayout")
# 图片区域：
        self.picArea = QtWidgets.QWidget(self.widget)
        self.picArea.setEnabled(True)
        self.picArea.setMinimumSize(QtCore.QSize(300, 0))
        self.picArea.setStyleSheet("background-color:rgb(43,43,43);background-
image:url(pic/bg.png);")     ##@@@@这里加了：
        self.picArea.setObjectName("picArea")
    # 放置图片的标签：
        self.picLabel = QtWidgets.QLabel(self.picArea)
        self.picLabel.setGeometry(QtCore.QRect(0, 0, 1005, 622))          # 这个就是图片的最
大像素。
        self.picLabel.setObjectName("picLabel")
        self.horizontalLayout.addWidget(self.picArea)
```

```
        self.picLabel.setStyleSheet("background-color:red;color:red;vertical-align:super;")
# 工具区域：
        self.toolArea = QtWidgets.QWidget(self.widget)
        self.toolArea.setMinimumSize(QtCore.QSize(250, 600))
        self.toolArea.setMaximumSize(QtCore.QSize(250, 800))
        self.toolArea.setStyleSheet("background-color:rgb(185,148,106);background-
image:url(images/wood4.jpg);")
        self.toolArea.setObjectName("toolArea")
    # 工具栏：
        self.toolBox = QtWidgets.QToolBox(self.toolArea)
        self.toolBox.setGeometry(QtCore.QRect(0, 0, 251, 600))      #  0,0,251,600
        #！！！！！！！！ 这里本来是 741，因为有的屏幕分辨率比较低，这样下面的一些工具栏就显
示不出来了。
        self.toolBox.setMinimumSize(QtCore.QSize(0, 0))
        self.toolBox.setObjectName("toolBox")
        self.toolBox.setStyleSheet("font:bold;font-family:微软雅黑;font-size:16px;")
      # 工具栏第一页（菜鸟页）：
        self.page1 = QtWidgets.QWidget()
        self.page1.setGeometry(QtCore.QRect(0, 0, 251, 581))
        self.page1.setObjectName("page1")
        # 放大按钮：
        self.bigger_Button = QtWidgets.QPushButton(self.page1)
        self.bigger_Button.setGeometry(QtCore.QRect(40, 15, 52, 52))
        self.bigger_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/plus1.png);")
        self.bigger_Button.setObjectName("bigger_Button")
        self.bigger_Button.clicked.connect(self.bigger)          #  bigger 的信息槽
        # 缩小按钮：
        self.smaller_Button = QtWidgets.QPushButton(self.page1)
        self.smaller_Button.setGeometry(QtCore.QRect(160, 15, 52, 52))
        self.smaller_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/minus1.png);")
        self.smaller_Button.setObjectName("smaller_Button")
        self.smaller_Button.clicked.connect(self.smaller)      # samller 的信息槽
        # 旋转按钮：
        self.rotate_Button = QtWidgets.QPushButton(self.page1)
        self.rotate_Button.setGeometry(QtCore.QRect(60, 95, 141, 51))            # QRect（左
上角，宽，高）
        self.rotate_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/button_01.png);",)          # 添加了背景图片！
```

```
        self.rotate_Button.setObjectName("rotate_Button")
        self.rotate_Button.clicked.connect(self.rotatePic)                ## 信息槽
        # 上下翻转：top-bottom
        self.TB_Button = QtWidgets.QPushButton(self.page1)
        self.TB_Button.setGeometry(QtCore.QRect(60, 150, 141, 51))
        self.TB_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/button_02.png)")
        self.TB_Button.setObjectName("TB_Button")
        self.TB_Button.clicked.connect(self.TBPic)                ## 信息槽
        # 左右翻转：left-fight
        self.LR_Button = QtWidgets.QPushButton(self.page1)
        self.LR_Button.setGeometry(QtCore.QRect(60, 205, 141, 51))
        self.LR_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/button_07.png)")
        self.LR_Button.setObjectName("LR_Button")
        self.LR_Button.clicked.connect(self.LRPic)                ## 信息槽
        # 拼接按钮：
        self.together_Button = QtWidgets.QPushButton(self.page1)
        self.together_Button.setGeometry(QtCore.QRect(60, 270, 141, 51))
        self.together_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/button_04.png)")
        self.together_Button.setObjectName("together_Button")
        self.together_Button.clicked.connect(self.together)                ## 拼接的信息-槽
        # 剪切按钮：
        self.cut_Button = QtWidgets.QPushButton(self.page1)
        self.cut_Button.setGeometry(QtCore.QRect(60, 325, 141, 51))
        self.cut_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/button_05.png)")
        self.cut_Button.setObjectName("cut_Button")

         # 局部消除（去水印）按钮：
        self.addSig_Button = QtWidgets.QPushButton(self.page1)
        self.addSig_Button.setGeometry(QtCore.QRect(60, 380, 141, 51))
        self.addSig_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/button_06.png)")
        self.addSig_Button.setObjectName("addSig_Button")
        self.addSig_Button.clicked.connect(self.add)                ## 信息槽
        self.toolBox.addItem(self.page1, "")
      # 工具栏第二页（进阶页）：
```

```
        self.page2 = QtWidgets.QWidget()
        self.page2.setGeometry(QtCore.QRect(0, 0, 251, 581))
        self.page2.setObjectName("page2")
        # 模糊化滑动条:
        self.blur_Slider = QtWidgets.QSlider(self.page2)
        self.blur_Slider.setGeometry(QtCore.QRect(20, 20, 211, 22))
        self.blur_Slider.setOrientation(QtCore.Qt.Horizontal)
        self.blur_Slider.setObjectName("blur_Slider")
        self.blur_Slider.setMinimum(0)
        self.blur_Slider.setMaximum(50)
        self.blur_Slider.valueChanged.connect(self.MagicBarPic)  # 模糊化的信号-槽

        # 锐化滑动条:
        self.sharpen_Slider = QtWidgets.QSlider(self.page2)
        self.sharpen_Slider.setGeometry(QtCore.QRect(20, 100, 211, 22))
        self.sharpen_Slider.setOrientation(QtCore.Qt.Horizontal)
        self.sharpen_Slider.setObjectName("sharpen_Slider")
        self.sharpen_Slider.valueChanged.connect(self.MagicBarPic)  # 锐化的信号-槽
        # 油画滑动条:
        self.oil_Slider = QtWidgets.QSlider(self.page2)
        self.oil_Slider.setGeometry(QtCore.QRect(20, 180, 211, 22))
        self.oil_Slider.setOrientation(QtCore.Qt.Horizontal)
        self.oil_Slider.setObjectName("oil_Slider")
        self.oil_Slider.setMinimum(0)
        self.oil_Slider.setMaximum(30)
        self.oil_Slider.valueChanged.connect(self.MagicBarPic)  # 油画的信号-槽
        # 七彩 滑动条:
        self.colorful_Slider = QtWidgets.QSlider(self.page2)
        self.colorful_Slider.setGeometry(QtCore.QRect(20, 260, 211, 22))
        self.colorful_Slider.setOrientation(QtCore.Qt.Horizontal)
        self.colorful_Slider.setObjectName("colorful_Slider")
        self.colorful_Slider.setMinimum(0)
        self.colorful_Slider.setMaximum(44)
        self.colorful_Slider.valueChanged.connect(self.MagicBarPic)  # 油画的信号-槽
        # 下面这些分别是这些滑动条的标签:
        self.blur_label = QtWidgets.QLabel(self.page2)
        self.blur_label.setGeometry(QtCore.QRect(50, 35, 151, 31))
        self.blur_label.setStyleSheet("margin:0 auto;\n"
"font:bold;\n"
"font-family:微软雅黑;")
        self.blur_label.setObjectName("blur_label")
        self.sharpen_label = QtWidgets.QLabel(self.page2)
        self.sharpen_label.setGeometry(QtCore.QRect(60, 115, 131, 31))
        self.sharpen_label.setStyleSheet("margin:0 auto;\n"
```

```
"font:bold;\n"
"font-family:微软雅黑;")
        self.sharpen_label.setObjectName("sharpen_label")
        self.oil_label = QtWidgets.QLabel(self.page2)
        self.oil_label.setGeometry(QtCore.QRect(60, 195, 121, 31))
        self.oil_label.setStyleSheet("margin:0 auto;\n"
"font:bold;\n"
"font-family:微软雅黑;")
        self.oil_label.setObjectName("oil_label")
        self.colorful_label = QtWidgets.QLabel(self.page2)
        self.colorful_label.setGeometry(QtCore.QRect(60, 275, 121, 31))
        self.colorful_label.setStyleSheet("margin:0 auto;\n"
                                          "font:bold;\n"
                                          "font-family:微软雅黑;")
        self.colorful_label.setObjectName("colorful_label")
        # 保存模糊、锐化、油画按钮：!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        self.saveSlider_Button = QtWidgets.QPushButton(self.page2)
        self.saveSlider_Button.setGeometry(QtCore.QRect(60, 310, 141, 51))
        self.saveSlider_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:red;")
        self.saveSlider_Button.setObjectName("saveSlider_Button")
        self.saveSlider_Button.clicked.connect(self.saveSlider)    #保存模糊、锐化、油画按
钮：!!!!!!!!

        self.toolBox.addItem(self.page2, "")
      # 工具栏第三页（我是逗逼页）：
        self.page3 = QtWidgets.QWidget()
        self.page3.setObjectName("page3")



        # 融合按钮:
        self.blend_Button = QtWidgets.QPushButton(self.page3)
        self.blend_Button.setGeometry(QtCore.QRect(30, 15, 191, 51))
        self.blend_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/bigger_button_7.png)")
        self.blend_Button.setObjectName("blend_Button")
        self.blend_Button.clicked.connect(self.blend)
        # 生成字符画按钮:
        self.charPic_Button = QtWidgets.QPushButton(self.page3)
        self.charPic_Button.setGeometry(QtCore.QRect(30, 70, 191, 51))
        self.charPic_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/bigger_button_8.png)")
```

```
        self.charPic_Button.setObjectName("charPic_Button")
        self.charPic_Button.clicked.connect(self.createCodePic)    # 生成字符画的信息-槽
        # 生成表情包按钮：
        self.emoji_Button = QtWidgets.QPushButton(self.page3)
        self.emoji_Button.setGeometry(QtCore.QRect(30, 125, 191, 51))
        self.emoji_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/bigger_button_9.png)")
        self.emoji_Button.setObjectName("emoji_Button")
        self.emoji_Button.clicked.connect(self.emoji)                ## 信息槽
        # 计算脸缘（相似度）按钮：
        self.similar_Button = QtWidgets.QPushButton(self.page3)
        self.similar_Button.setGeometry(QtCore.QRect(30, 180, 191, 51))     # 191,51
        self.similar_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/bigger_button_1.png)")
        self.similar_Button.setObjectName("similar_Button")
        self.similar_Button.clicked.connect(self.compare)                ## 信息槽
        # 变形按钮：
        self.shape_Button = QtWidgets.QPushButton(self.page3)
        self.shape_Button.setGeometry(QtCore.QRect(30, 235, 191, 51))
        self.shape_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/bigger_button_2.png)")
        self.shape_Button.setObjectName("shape_Button")
        # self.shape_Button.clicked.connect(self.add)              ## 信息槽
        self.toolBox.addItem(self.page3, "")
      # 工具栏第四页（滤镜页）：
        self.page4 = QtWidgets.QWidget()
        self.page4.setObjectName("page4")
        # 添加一个说明标签：
        self.notice_label = QtWidgets.QLabel(self.page4)
        self.notice_label.setGeometry(QtCore.QRect(0, 5, 250, 100))
        self.notice_label.setStyleSheet("margin:0 auto;\n"
                                        "font:bold;\n"
                                        "font-family:微软雅黑;\n"
                                         "color:red;\n"
                                          "font-size:14px\n"
                                         )
        self.notice_label.setObjectName("blur_label")

        # 黑白空间按钮：
        self.bnw_Button = QtWidgets.QPushButton(self.page4)
        self.bnw_Button.setGeometry(QtCore.QRect(30, 115, 191, 51))
```

```
        self.bnw_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/pen.png);border:4px solid black;border-radius:5%")
        self.bnw_Button.setObjectName("bnw_Button")
        self.bnw_Button.clicked.connect(self.bnwPic)              ##黑白照片 的信息-槽
        # 显示轮廓
        self.contour_Button = QtWidgets.QPushButton(self.page4)
        self.contour_Button.setGeometry(QtCore.QRect(30, 170, 191, 51))
        self.contour_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/lunkuo.png);border:4px solid rgb(147,130,114);border-radius:5%")
        self.contour_Button.setObjectName("contour_Button")
        self.contour_Button.clicked.connect(self.contourPic)                ##轮廓 的信息-槽
        # 浮雕:
        self.emboss_Button = QtWidgets.QPushButton(self.page4)
        self.emboss_Button.setGeometry(QtCore.QRect(30, 225, 191, 51))
        self.emboss_Button.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/emboss.png);border:4px solid rgb(32,32,32);border-radius:5%;")
        self.emboss_Button.setObjectName("emboss_Button")
        self.emboss_Button.clicked.connect(self.embossPic)          ## 浮雕 的信息-槽
        # 熔岩魔鬼 按钮:
        self.fireGoast_pushButton = QtWidgets.QPushButton(self.page4)
        self.fireGoast_pushButton.setGeometry(QtCore.QRect(30, 280, 191, 51))
        self.fireGoast_pushButton.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;background-color:transparent;background-
image:url(images/rongyan.png);border:4px solid rgb(146,14,14);border-
radius:5%;color:orange")
        self.fireGoast_pushButton.setObjectName("fireGoast_pushButton")
        self.fireGoast_pushButton.clicked.connect(self.fireGoastPic)    ## 熔岩魔鬼 的信息-槽
        self.picArea.raise_()
        self.bnw_Button.raise_()
        self.contour_Button.raise_()
        self.emboss_Button.raise_()
        self.fireGoast_pushButton.raise_()
        self.toolBox.addItem(self.page4, "")
        self.horizontalLayout.addWidget(self.toolArea)
        self.verticalLayout.addWidget(self.widget)
    #窗口底部 footer:
        self.footer = QtWidgets.QWidget(self.centralWidget)
        self.footer.setMaximumSize(QtCore.QSize(1280, 100))
        self.footer.setMinimumSize(QtCore.QSize(200, 32))
        self.footer.setStyleSheet("background-color:transparent;background-
image:url(images/wood4.jpg)") # 底部的样式
```

```
        self.footer.setObjectName("footer")
        # 底部有5个标签：footerLabel1--footerLabel5,
        # 还有三个显示时、分、秒的数字屏 lcdH、lcdM、lcdS：
        self.footerLabel1 = QtWidgets.QLabel(self.footer)
        self.footerLabel1.setGeometry(QtCore.QRect(100, 10, 251, 31))
        self.footerLabel1.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;\n"
"color:darkblue;")
        self.footerLabel1.setObjectName("footerLabel1")
        self.lcdH = QtWidgets.QLCDNumber(self.footer)
        self.lcdH.setGeometry(QtCore.QRect(340, 0, 71, 51))     # 340, 2, 71,51
        self.lcdH.setObjectName("lcdH")
        self.footerLabel2 = QtWidgets.QLabel(self.footer)
        self.footerLabel2.setGeometry(QtCore.QRect(430, 11, 31, 31))
        self.footerLabel2.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;\n"
"color:darkblue;")
        self.footerLabel2.setObjectName("footerLabel2")
        self.lcdM = QtWidgets.QLCDNumber(self.footer)
        self.lcdM.setGeometry(QtCore.QRect(460, 1, 71, 51))
        self.lcdM.setObjectName("lcdM")
        self.footerLabel3 = QtWidgets.QLabel(self.footer)
        self.footerLabel3.setGeometry(QtCore.QRect(550, 10, 31, 31))
        self.footerLabel3.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;\n"
"color:darkblue;")
        self.footerLabel3.setObjectName("footerLabel3")
        self.lcdS = QtWidgets.QLCDNumber(self.footer)
        self.lcdS.setGeometry(QtCore.QRect(580, 1, 71, 51))
        self.lcdS.setObjectName("lcdS")
        self.footerLabel4 = QtWidgets.QLabel(self.footer)
        self.footerLabel4.setGeometry(QtCore.QRect(670, 10, 31, 31))
        self.footerLabel4.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;\n"
"color:darkblue;")
        self.footerLabel4.setObjectName("footerLabel4")
        self.footerLabel5 = QtWidgets.QLabel(self.footer)
        self.footerLabel5.setGeometry(QtCore.QRect(700, 10, 251, 31))
        self.footerLabel5.setStyleSheet("font:bold;\n"
"font-family:微软雅黑;\n"
"color:darkblue;")
        self.footerLabel5.setObjectName("footerLabel5")
        self.verticalLayout.addWidget(self.footer)
        MainWindow.setCentralWidget(self.centralWidget)
```

```
    # 菜单栏：
        self.menuBar = QtWidgets.QMenuBar(MainWindow)
        self.menuBar.setGeometry(QtCore.QRect(0, 0, 1280, 36))
        self.menuBar.setStyleSheet("background-color:rgb(185,148,106);background-
image:url(images/wood4.jpg);font-size:16px;font-family:微软雅黑;")
        self.menuBar.setObjectName("menuBar")
      # 文件菜单：
        self.menuFile = QtWidgets.QMenu(self.menuBar)
        self.menuFile.setObjectName("menuFile")
        self.menuFile.setStyleSheet("")
      # 帮助菜单：
        self.menuHelp = QtWidgets.QMenu(self.menuBar)
        self.menuHelp.setObjectName("menuHelp")
      # 分享菜单：
        self.menuShare = QtWidgets.QMenu(self.menuBar)
        self.menuShare.setObjectName("menuShare")
        MainWindow.setMenuBar(self.menuBar)
        # 说明书动作：
        self.actionGuide = QtWidgets.QAction(MainWindow)
        self.actionGuide.setObjectName("actionGuide")
        # 联系我们动作：
        self.actionContact = QtWidgets.QAction(MainWindow)
        self.actionContact.setObjectName("actionContact")
        # 分享到微信和 QQ 的动作：
        self.actionWeChat = QtWidgets.QAction(MainWindow)
        self.actionWeChat.setObjectName("actionWeChat")
        self.actionQQ = QtWidgets.QAction(MainWindow)
        self.actionQQ.setObjectName("actionQQ")
        # 打开文件动作：
        self.actionOpen = QtWidgets.QAction(MainWindow)
        self.actionOpen.setObjectName("actionOpen")
        self.actionOpen.triggered.connect(self.openPic) ## 打开图片的信息-槽
        # 保存文件动作：
        self.actionSave = QtWidgets.QAction(MainWindow)
        self.actionSave.setObjectName("actionSave")
        self.actionSave.triggered.connect(self.savePic) ## 保存图片的信息-槽
        # 退出动作：
        self.actionQuit = QtWidgets.QAction(MainWindow)
        self.actionQuit.setObjectName("actionQuit")
        self.actionQuit.triggered.connect(QCoreApplication.quit)



        self.menuFile.addAction(self.actionOpen)
```

```
        self.menuFile.addAction(self.actionSave)
        self.menuFile.addSeparator()
        self.menuFile.addAction(self.actionQuit)
        self.menuHelp.addAction(self.actionGuide)
        self.menuHelp.addAction(self.actionContact)
        self.menuShare.addAction(self.actionWeChat)
        self.menuShare.addAction(self.actionQQ)
        self.menuBar.addAction(self.menuFile.menuAction())
        self.menuBar.addAction(self.menuHelp.menuAction())
        self.menuBar.addAction(self.menuShare.menuAction())

        self.retranslateUi(MainWindow)
        self.toolBox.setCurrentIndex(0)  # 设置最开始显示的工具栏
        QtCore.QMetaObject.connectSlotsByName(MainWindow)
#————————————————————组件翻译函数：————————————————————————————————
————————
    def retranslateUi(self, MainWindow): # 这个函数好像是专门给组件命名的。
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.picLabel.setText(_translate("MainWindow", " "))
        # 在这里设置初始的背景图片：
        # 这个如果放在前面（定义标签的时候），就显示不出来了。
        self.pic=QtGui.QPixmap("pic\\bg.png")
        self.picLabel.setPixmap(self.pic)

        self.bigger_Button.setText(_translate("MainWindow", " "))   #大
        self.smaller_Button.setText(_translate("MainWindow", " "))   #小
        self.rotate_Button.setText(_translate("MainWindow", "旋转吧♋"))
        self.TB_Button.setText(_translate("MainWindow", "上下倒立♓"))
        self.LR_Button.setText(_translate("MainWindow", "照镜子♌"))
        self.together_Button.setText(_translate("MainWindow", "图图拼接"))
        self.cut_Button.setText(_translate("MainWindow", "剪切✂"))
        self.toolBox.setItemText(self.toolBox.indexOf(self.page1), _translate("MainWindow",
"基本工具"))
        self.blur_label.setText(_translate("MainWindow", "模糊化"))
        self.sharpen_label.setText(_translate("MainWindow", "锐化"))
        self.oil_label.setText(_translate("MainWindow", "油画"))
        self.colorful_label.setText(_translate("MainWindow", "~七彩~"))
        self.saveSlider_Button.setText(_translate("MainWindow", "保存修改"))
        self.addSig_Button.setText(_translate("MainWindow", "个性签名※"))
        self.toolBox.setItemText(self.toolBox.indexOf(self.page2), _translate("MainWindow",
"超级魔法棒"))

        self.blend_Button.setText(_translate("MainWindow", "@_@图片融合术"))
```

```
        self.charPic_Button.setText(_translate("MainWindow", "->#生成字符画"))
        self.emoji_Button.setText(_translate("MainWindow", "⊙o⊙制作表情包"))
        self.similar_Button.setText(_translate("MainWindow", "❤o❤我爱算脸缘"))
        self.shape_Button.setText(_translate("MainWindow", ">_<形变。。待定？？"))
        self.toolBox.setItemText(self.toolBox.indexOf(self.page3), _translate("MainWindow",
"我是逗逼"))

        self.notice_label.setText(_translate("MainWindow", "提醒：\n魔力太强大，不能恢复！
\n建议每个效果只使用 1 次"))
        self.bnw_Button.setText(_translate("MainWindow", "黑白空间■"))
        self.contour_Button.setText(_translate("MainWindow", "轮廓印象▒"))
        self.emboss_Button.setText(_translate("MainWindow", "浮雕变化▣"))
        self.fireGoast_pushButton.setText(_translate("MainWindow", "熔岩魔鬼☹"))
        self.toolBox.setItemText(self.toolBox.indexOf(self.page4), _translate("MainWindow",
"潘多拉魔盒"))
        self.footerLabel1.setText(_translate("MainWindow", "您已经在 MagicPicPS 花费了"))
        self.footerLabel2.setText(_translate("MainWindow", "时"))
        self.footerLabel3.setText(_translate("MainWindow", "分"))
        self.footerLabel4.setText(_translate("MainWindow", "秒"))
        self.footerLabel5.setText(_translate("MainWindow", "的美好时光~0(∩_∩)0~"))
        self.menuFile.setTitle(_translate("MainWindow", "文件"))
        self.menuHelp.setTitle(_translate("MainWindow", "帮助"))
        self.menuShare.setTitle(_translate("MainWindow", "分享"))
        self.actionGuide.setText(_translate("MainWindow", "说明书"))
        self.actionContact.setText(_translate("MainWindow", "撩我们"))
        self.actionWeChat.setText(_translate("MainWindow", "分享到微信"))
        self.actionQQ.setText(_translate("MainWindow", "分享到 QQ"))
        self.actionOpen.setText(_translate("MainWindow", "打开"))
        self.actionSave.setText(_translate("MainWindow", "保存"))
        self.actionQuit.setText(_translate("MainWindow", "强行退出"))


#————————————————功能函数区：————————————————————————————————————
##打开和保存：
    def openPic(self):
        #初始化 slider 的值：
        self.blur_Slider.setValue(0)
        self.sharpen_Slider.setValue(0)
        self.colorful_Slider.setValue(0)
        self.oil_Slider.setValue(0)

        address=QFileDialog.getOpenFileName(self,"亲，选取一张图片哦！","F:\视频·图片\图
片！！！\上财照片","Image files(*.png *.jpg *.jpeg *.gif);;all files(*.*)")
        self.filename=address[0]      # 这个 address 是一个列表，所以需要取第一项
```

```
        if self.filename:
            print(self.filename)
            # 把这个图片显示在画面上：
            self.pic=QtGui.QPixmap(self.filename)
            self.picLabel.setPixmap(self.pic)
            self.newPic=Image.open(self.filename)
            # # 要把新打开的图片，和所有修改过的图片，都存进 tempPic.jpg 里面，作为一个中间变量。
            self.newPic.save("pic\\tempPic.png")
            self.newPic="pic\\tempPic.png"



    def savePic(self):
        print("1")
        saveAddress=QFileDialog.getSaveFileName(self,"保存文件","C:/Magpic pictures","Image files(*.png *.jpg *.jpeg *.gif);;all files(*.*)")
        print("2")
        if saveAddress[0]:
            self.newPic=Image.open("pic\\tempPic.png")   # 注意，每一次 open，都把 newPic 重新赋值了。
            self.newPic.save(saveAddress[0])  # 保存新图片 newPic
            self.newPic="pic\\tempPic.png"
            print("3")
            print(saveAddress[0],"已经保存成功！")
# 退出警告：
    def closeEvent(self, event):  # 这个 closeEvent 是父类的方法，这里把它重写了一下！
        reply = QMessageBox.question(self, "Waring", "<b><font color=red>嘿！你真的就这么走了吗？</font></b>",
                                     QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
        if reply == QMessageBox.Yes:
            event.accept()
        if reply == QMessageBox.No:
            event.ignore()



## 模糊化：
    def MagicBarPic(self):
        # 首先获取原始图片：
        # getPic=Image.open(self.filename)
        getPic=Image.open(self.newPic)
        blur_radiusValue=self.blur_Slider.value()  # 参数的值
```

```
        sharpen_radiusValue=self.sharpen_Slider.value()
        oil_sizeValue=self.oil_Slider.value()
        alpha=self.colorful_Slider.value()

        # 对图片进行处理，形成新图片：
        self.newPic=getPic.filter(ImageFilter.GaussianBlur(radius=blur_radiusValue))   #模糊
化
        self.newPic.save("pic\\sliderPic.png") #如果没有这一步，上面的效果会被下面的直接覆盖
掉。

self.newPic=self.newPic.filter(ImageFilter.UnsharpMask(radius=sharpen_radiusValue,percent=30
0,threshold=3))   # 锐化
        self.newPic.save("pic\\sliderPic.png")
        self.newPic=self.newPic.filter(ImageFilter.ModeFilter(size=oil_sizeValue))   # 油画
        # 处理完之后，保存为 tempPic.jpg：
        self.newPic.save("pic\\sliderPic.png")

        # 七彩处理————
        size = self.newPic.size
        colorfulPic = Image.new('RGB', size)
        p_pic = self.newPic.load()
        p_colorfulPic = colorfulPic.load()
        # ——————————————

        # 七彩处理：————
#         alpha = 0  # 0~44
        for i in range(size[0]):
            for j in range(size[1]):
                if alpha == 0:
                    p_colorfulPic[i, j] = p_pic[i, j]
                elif p_pic[i, j][0] < alpha and p_pic[i, j][1] < alpha and p_pic[i, j][2] <
alpha:
                    p_colorfulPic[i, j] = (155, 7, 129)
                elif p_pic[i, j][0] < alpha * 2 and p_pic[i, j][1] < alpha * 2 and p_pic[i,
j][2] < alpha * 2:
                    p_colorfulPic[i, j] = (29, 32, 137)
                elif p_pic[i, j][0] < alpha * 3 and p_pic[i, j][1] < alpha * 3 and p_pic[i,
j][2] < alpha * 3:
                    p_colorfulPic[i, j] = (0, 142, 216)
                elif p_pic[i, j][0] < alpha * 4 and p_pic[i, j][1] < alpha * 4 and p_pic[i,
j][2] < alpha * 4:
                    p_colorfulPic[i, j] = (12, 165, 62)
                elif p_pic[i, j][0] < alpha * 5 and p_pic[i, j][1] < alpha * 5 and p_pic[i,
j][2] < alpha * 5:
```

```
                    p_colorfulPic[i, j] = (255, 228, 1)
                elif p_pic[i, j][0] < 225 and p_pic[i, j][1] < (255 - alpha * 5) - 25 and
p_pic[i, j][2] < (
                    255 - alpha * 5) - 25:
                    p_colorfulPic[i, j] = (242, 146, 0)
                else:
                    p_colorfulPic[i, j] = (230, 0, 19)
        colorfulPic.save("pic\\sliderPic.png")
# ————————————————
        self.newPic="pic\\tempPic.png"  #!!!!!!!!!!不能写 sliderPic，因为每一次点击，都会打开一个新的 newPic，如果把 newPic 存放成了 sliderPic.png，那么下次打开的时候效果还是会累加无法复原。
        # 在画面上展示一下效果：
        self.pic=QtGui.QPixmap("pic\\sliderPic.png")
        self.picLabel.setPixmap(self.pic)

    def saveSlider(self):
        sliderPic=Image.open("pic\\sliderPic.png")
        sliderPic.save("pic\\tempPic.png")
        self.newPic="pic\\tempPic.png"


# 各种旋转：
    def rotatePic(self):
        print(self.newPic)
        getPic=Image.open(self.newPic)
        self.newPic=getPic.transpose(PIL.Image.ROTATE_90)
        self.newPic.save("pic\\tempPic.png")   # !!!!注意，这里虽然 newPic 保存成 tempPic，但是 newPic 的值却不等于这个地址。在上一步的时候，newPic 的值被替换成了一个中间格式。
        # print(self.newPic) # 这里可以看出，此时 newPic 的值并不是 tempPic.png，需要下一步：
        self.newPic="pic\\tempPic.png"
        self.pic=QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)

    def TBPic(self):
        print(self.newPic)
        getPic=Image.open(self.newPic)
        self.newPic=getPic.transpose(PIL.Image.FLIP_TOP_BOTTOM)
        self.newPic.save("pic\\tempPic.png")   # 注意，这里虽然 newPic 保存成 tempPic，但是 newPic 的值却不等于这个地址。在上一步的时候，newPic 的值被替换成了一个中间格式。
        # print(self.newPic) # 这里可以看出，此时 newPic 的值并不是 tempPic.png，需要下一步：
        self.newPic="pic\\tempPic.png"
        self.pic=QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)
```

```
    def LRPic(self):
        print(self.newPic)
        getPic=Image.open(self.newPic)
        self.newPic=getPic.transpose(PIL.Image.FLIP_LEFT_RIGHT)
        self.newPic.save("pic\\tempPic.png")    # 注意，这里虽然 newPic 保存成 tempPic，但是 newPic 的值却不等于这个地址。在上一步的时候，newPic 的值被替换成了一个中间格式。
        # print(self.newPic) # 这里可以看出，此时 newPic 的值并不是 tempPic.png，需要下一步：
        self.newPic="pic\\tempPic.png"
        self.pic=QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)

## 拼接：
    def together(self):
        address=QFileDialog.getOpenFileNames(self,"请选择要拼接的图片（2 张）(*^__^*)","C:/")
        if address[0]:
            num=len(address[0])  #因为这个 address 的第一项才存放的地址。
            print("有",num,"个文件被选出来了：")

            width=350
            height=350
            self.newPic=Image.new('RGB',(width*num,height))
            print("hhhhhhhhhhhh")
            for i in range(num):
                print(address[0][i])
                oriPic=Image.open(address[0][i])        # 导入每一张图片
                resizePic=oriPic.resize((width,height))

                self.newPic.paste(resizePic,(i*width,0))
            print("1111111")
            self.newPic.save("pic\\tempPic.png")
            print("2222222")
            self.pic=QtGui.QPixmap("pic\\tempPic.png")
            print("3333333")
            self.picLabel.setPixmap(self.pic)
            print("44444444")
            self.newPic.resize((width*num,height)).save("pic\\tempPic.png")
            self.newPic="pic\\tempPic.png"


## 图片融合术：
    def blend(self):
        address=QFileDialog.getOpenFileNames(self,"请选择要拼接的图片（两张哦！）
```

```
(*^__^*)",",C:/")
        if address[0]:
            print("第一个文件是：",address[0][0])
            print("第二个文件是：",address[0][1])
            pic1=Image.open(address[0][0])
            pic2=Image.open(address[0][1])
            width=600
            height=600
            pic1=pic1.resize((width,height))

            pic2=pic2.resize((width, height))
            self.newPic=Image.blend(pic1.convert("RGBA"),pic2.convert("RGBA"),alpha=0.5)

            self.newPic.save("pic\\tempPic.png")
            print("2222222")
            self.pic=QtGui.QPixmap("pic\\tempPic.png")
            print("3333333")
            self.picLabel.setPixmap(self.pic)
            print("44444444")
            self.newPic="pic\\tempPic.png"


## 轮廓:
    def contourPic(self):
        print(self.newPic)
        getPic = Image.open(self.newPic)
        self.newPic = getPic.filter(ImageFilter.CONTOUR)
        self.newPic.save("pic\\tempPic.png")  # 注意，这里虽然 newPic 保存成 tempPic，但是
newPic 的值却不等于这个地址。在上一步的时候，newPic 的值被替换成了一个中间格式。
        # print(self.newPic) # 这里可以看出，此时 newPic 的值并不是 tempPic.png，需要下一步:
        self.newPic = "pic\\tempPic.png"
        self.pic = QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)


## 浮雕:
    def embossPic(self):
        print(self.newPic)
        getPic = Image.open(self.newPic)
        self.newPic = getPic.filter(ImageFilter.EMBOSS)
        self.newPic.save("pic\\tempPic.png")  # 注意，这里虽然 newPic 保存成 tempPic，但是
newPic 的值却不等于这个地址。在上一步的时候，newPic 的值被替换成了一个中间格式。
        # print(self.newPic) # 这里可以看出，此时 newPic 的值并不是 tempPic.png，需要下一步:
        self.newPic = "pic\\tempPic.png"
```

```
        self.pic = QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)

## 黑白:
    def bnwPic(self):

        self.newPic=Image.open("pic\\tempPic.png")
        size = self.newPic.size
        bnwPic = Image.new('RGB', size)
        p_pic = self.newPic.load()
        p_bnwPic = bnwPic.load()

        for i in range(size[0]):
            for j in range(size[1]):
                # if p_pic[i, j][0] < 80 and p_pic[i, j][1] < 80 and p_pic[i, j][2] < 80:
                #     p_bnwPic[i, j] = (0, 0, 0)
                # elif p_pic[i, j][0] < 160 and p_pic[i, j][1] < 160 and p_pic[i, j][2] <
160:
                #     p_bnwPic[i, j] = (220, 220, 220)
                # else: p_bnwPic[i, j] = (255, 255, 255)  # 这种其实也可以，但是下面这个更加
细致一些。
                alpha=30
                if p_pic[i, j][0] < alpha and p_pic[i, j][1] < alpha and p_pic[i, j][2] <
alpha:
                    p_bnwPic[i, j] = (0,0,0)
                elif p_pic[i, j][0] < alpha * 2 and p_pic[i, j][1] < alpha * 2 and p_pic[i,
j][2] < alpha * 2:
                    p_bnwPic[i, j] = (10,10,10)
                elif p_pic[i, j][0] < alpha * 3 and p_pic[i, j][1] < alpha * 3 and p_pic[i,
j][2] < alpha * 3:
                    p_bnwPic[i, j] = (20,20,20)
                elif p_pic[i, j][0] < alpha * 4 and p_pic[i, j][1] < alpha * 4 and p_pic[i,
j][2] < alpha * 4:
                    p_bnwPic[i, j] = (180,180,180)
                elif p_pic[i, j][0] < alpha * 5 and p_pic[i, j][1] < alpha * 5 and p_pic[i,
j][2] < alpha * 5:
                    p_bnwPic[i, j] = (200,200,200)
                elif p_pic[i, j][0] < 225 and p_pic[i, j][1] < (255 - alpha * 5) - 25 and
p_pic[i, j][2] < (
                    255 - alpha * 5) - 25:
                    p_bnwPic[i, j] = (220,220,220)
                else:
                    p_bnwPic[i, j] = (255,255,255)
        bnwPic.save("pic\\tempPic.png")
```

```
        self.newPic = "pic\\tempPic.png"
        self.pic = QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)

# 熔岩魔鬼 :
    def fireGoastPic(self):
        self.newPic = Image.open("pic\\tempPic.png")
        size = self.newPic.size
        fireGoastPic = Image.new('RGB', size)
        p_pic = self.newPic.load()
        p_fireGoastPic = fireGoastPic.load()

        for i in range(size[0]):
            for j in range(size[1]):
                p_fireGoastPic[i, j] = p_pic[i, j][2] * 9
        fireGoastPic.save("pic\\tempPic.png")

        self.newPic = "pic\\tempPic.png"
        self.pic = QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)


# 生成字符画:
    def rgb_to_char(self,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)]

    def createCodePic(self):
        img = Image.open("pic/tempPic.png")
        img = img.resize((120, 120))
        size = img.size
        width = size[0]
        height = size[1]
        pimg = img.load()

        txt = ""
        for i in range(height):
```

```
            for j in range(width):
                r = pimg[j, i][0]
                g = pimg[j, i][1]
                b = pimg[j, i][2]
                print("1")
                txt = txt + self.rgb_to_char(r, g, b, alpha=256)
                print("1111")
            txt = txt + "\n"
        print("2")
        txtAddress = QFileDialog.getSaveFileName(self, "选择一个路径把你的字符画收好！（以txt格式存储）", "C:/Magpic pictures",
                                                 "Image files(*.txt);;all files(*.*)")
        if txtAddress[0]:
            print("3")
            txtpic = open(txtAddress[0], "w")
            txtpic.write(txt)
            txtpic.close()

    def bigger(self):
        img = cv2.imread(self.newPic)
        res = cv2.resize(img, None, fx=1.5, fy=1.5, interpolation=cv2.INTER_CUBIC)
        cv2.imwrite("pic\\tempPic.png", res)
        # 在画面上展示一下效果：
        self.pic = QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)
        self.newPic = "pic\\tempPic.png"

    def smaller(self):
        img = cv2.imread(self.newPic)
        res = cv2.resize(img, None, fx=2/3, fy=2/3, interpolation=cv2.INTER_CUBIC)
        cv2.imwrite("pic\\tempPic.png", res)
        # 在画面上展示一下效果：
        self.pic = QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)
        self.newPic = "pic\\tempPic.png"

    def add(self):
        input=QInputDialog.getText(self, "请添加个性化水印！","水印:",QLineEdit.Normal,
"Magpic")
        print(input[0])
        font = ImageFont.truetype("C:\\Windows\\Fonts\\Arial.ttf", 50)

        def add_text_to_image(image, text, font=font):
            rgba_image = image.convert('RGBA')
```

```
            text_overlay = Image.new('RGBA', rgba_image.size, (255, 255, 255, 0))
            image_draw = ImageDraw.Draw(text_overlay)

            text_size_x, text_size_y = image_draw.textsize(text, font=font)
            # 设置文本文字位置
            print(rgba_image)
            text_xy = (rgba_image.size[0] - text_size_x, rgba_image.size[1] - text_size_y)
            # 设置文本颜色和透明度
            image_draw.text(text_xy, text, font=font, fill=(185,148,106, 800))

            image_with_text = Image.alpha_composite(rgba_image, text_overlay)

            return image_with_text

        im_before = Image.open(self.newPic)
        im_after = add_text_to_image(im_before, input[0])
        im_after.save("pic\\tempPic.png")
        # 在画面上展示一下效果:
        self.pic = QtGui.QPixmap("pic\\tempPic.png")
        self.picLabel.setPixmap(self.pic)
        self.newPic = "pic\\tempPic.png"

    def emoji(self):
        print("asdfdsfdfafdsfsadfasdfasdfsf")
        hhaddress = QFileDialog.getOpenFileName(self, "请选择一个头像(☆▽☆)", "C:/")
        print("13123213231")
        if hhaddress[0]:
            getPic = Image.open(hhaddress[0])
            print("hahahahah")

            getPic2 = Image.open("pic\\k3.png")
            box = (200, 150, 420, 450)
            # mycrop = getPic.crop(box)
            # mycrop = mycrop.transpose(Image.ROTATE_180)
            # getPic.paste(mycrop, box)
            print("11")
            imgenhancer_Brightness = ImageEnhance.Brightness(getPic)
            newPic1 = imgenhancer_Brightness.enhance(2)
            print("22")
            imgenhancer_Contrast = ImageEnhance.Contrast(newPic1)
            newPic1 = imgenhancer_Contrast.enhance(6)
            print("33")
            newPic2 = newPic1.convert("L")
            region = newPic2
```

```
            print("44")
            region = region.resize((box[2] - box[0], box[3] - box[1]))
            getPic2.paste(region, box)
            # 处理完之后，保存为 tempPic.png:
            getPic2.save("pic\\tempPic.png")
            # 在画面上展示一下效果:
            self.pic = QtGui.QPixmap("pic\\tempPic.png")
            self.picLabel.setPixmap(self.pic)
            self.newPic = "pic\\tempPic.png"

    def compare(self):
        address = QFileDialog.getOpenFileNames(self, "请选择要比较的图片(*^__^*)（2 张）",
"C:/")
        if address[0]:
            num = len(address[0])  # 因为这个 address 的第一项才存放的地址。
            print("有", num, "个文件被选出来了：")
            pic1=address[0][0]
            pic2=address[0][1]
            print(pic1,pic2)

    # #————————————————————————————————————————————————————————————————
————————————
            getPic = Image.open(pic1)
            getPic2 = Image.open(pic2)

            def make_regalur_image(img, size=(256, 256)):
                return img.resize(size).convert('RGB')

            def split_image(img, part_size=(64, 64)):
                w, h = img.size
                pw, ph = part_size

                assert w % pw == h % ph == 0
                return [img.crop((i, j, i + pw, j + ph)).copy() \
                        for i in range(0, w, pw) \
                        for j in range(0, h, ph)]

            def hist_similar(lh, rh):
                assert len(lh) == len(rh)
                return sum(1 - (0 if l == r else float(abs(l - r)) / max(l, r)) for l, r in
zip(lh, rh)) / len(lh)

            def calc_similar(li, ri):
                return sum(
```

```
                    hist_similar(l.histogram(), r.histogram()) for l, r in
zip(split_image(li), split_image(ri))) / 16.0

            def calc_similar_by_path(lf, rf):
                li, ri = lf, rf
                return calc_similar(li, ri)

            self.com = calc_similar_by_path(make_regalur_image(getPic),
make_regalur_image(getPic2))
            # print(self.com)
            print("2343321")
    #_____________________________________
            width = 350
            height = 350
            self.newPic = Image.new('RGB', (width * num, height))
            print("hhhhhhhhhhhhh")
            for i in range(num):
                print(address[0][i])
                oriPic = Image.open(address[0][i])  # 导入每一张图片
                resizePic = oriPic.resize((width, height))

                self.newPic.paste(resizePic, (i * width, 0))
            print("1111111")
            self.newPic.resize((1000,600)).save("pic\\tempPic.png")
            print("2222222")
            self.pic = QtGui.QPixmap("pic\\tempPic.png")
            print("3333333")
            self.picLabel.setPixmap(self.pic)
            print("44444444")
            self.newPic.save("pic\\tempPic.png")
            self.newPic = "pic\\tempPic.png"
            reply = QMessageBox.information(self, "相似度：", "<b><font color=red>两张图片的
相似度是</font></b>"+str('%.2f'%(self.com*100))+"%",)



# 显示相似度：
    def show_similarity(self, event):
        print("1")
        score=self.com
        print("2")
        reply = QMessageBox.question(self, "相似度：", "<b><font color=red>两张图片的相似度
是</font></b>"+str(score),
                                     QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
```

```
        if reply == QMessageBox.Yes:
            event.accept()
        if reply == QMessageBox.No:
            event.ignore()


    #————————————创建实例，展示窗口：————————————————————————————————————
————————
qapp=QApplication(sys.argv)
app=Ui_MainWindow()
app.setStyleSheet('''
    QPushButton:hover{
    background-color:transparent;
    background-image:url(images/hover.png);}
    QMenu{
        font: bold 16px;
        }
    QPushButton{
    background-color:transparent;
    background-image:url(images/button_11.png);}
    font: bold 16px;

       ''')

app.show()
sys.exit(qapp.exec_())


# self.rotate_Button.setStyleSheet("QPushButton:hover{background-
color:transparent;background-image:url(images/button_02.png)}")
```