

你的颜值有几分(一)

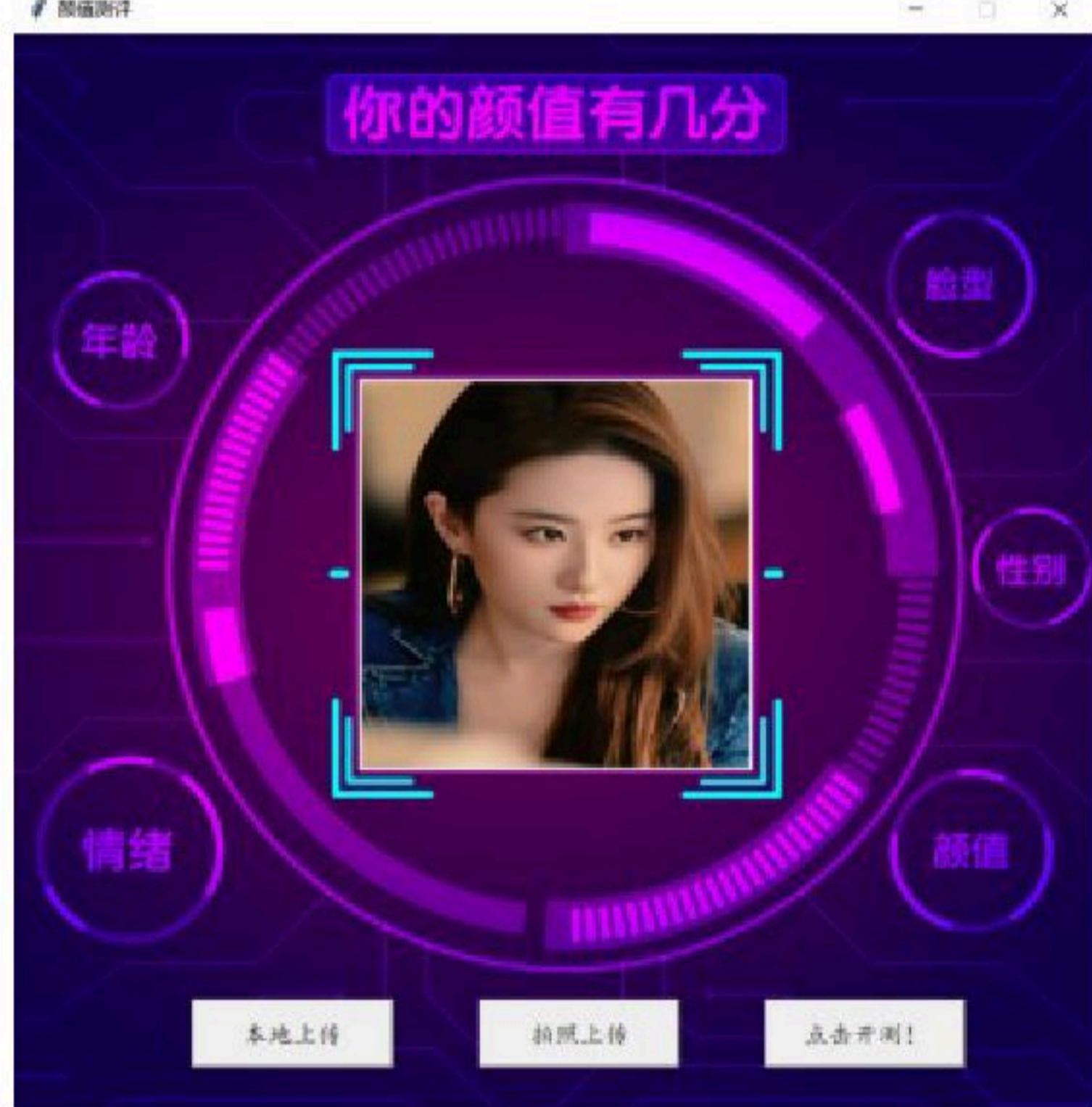




1. 探索新知

1.1

窗口创建与布局



思考：如何搭建左侧的窗口效果？

【标题】

颜值测评

【背景】

创建标签，标签上放置背景图片

【缩略图】

创建标签，标签上设置一张默认图片，后续实时更新要测颜值图片

【按钮】

本地上传、拍照上传、点击开测！



窗口搭建的工作前面已经做过很多次，大家自己先试着独立完成吧



700*700

1.1

窗口创建与布局

...

【一、导入库】

...

```
import tkinter as tk
```

...

【二、初始化】

...

```
root = tk.Tk() # 创建窗口root  
root.geometry('700x700') # 设置窗口的尺寸  
root.title('颜值测评') # 设置窗口的标题  
root.resizable(0, 0) # 设置窗口大小不可调整
```

提供的背景图
尺寸为 700*700

...

【五、启动事件循环】

...

```
root.mainloop()
```

接下来，我们进行【组件配置】，需要配置的有背景图、缩略图、以及三个按钮

...

【四、组件配置】

...

```
# 设置背景图  
# 设置缩略图  
# 设置按钮
```

【设置背景图】

我们需要将【测颜值.png】这个文件设置为窗口的背景图。该怎么实现呢？



1.1

窗口创建与布局

```
bg_image = tk.PhotoImage(file = '测颜值.png')  
bg_label = tk.Label(root, image = bg_image)  
bg_label.place(x=0, y=0, relwidth=1, relheight=1)
```

- 第1行：创建 PhotoImage 对象，PhotoImage 是 Tkinter 用来加载和显示图像的类。将加载的图像存到变量 bg_image 中
- 第2行：创建标签，标签的显示内容为上面创建的 PhotoImage 对象 bg_image
- 第3行：放置标签，标签左上角放在窗口左上角，标签宽高与窗口一致，填充整个窗口

【设置缩略图】



这里创建一个标签，标签中负责展示缩略图
包括默认缩略图、上传后的缩略图、以及拍照上传时的摄像头画面

```
pic0 = Image.open('缩略图.png')  
pic0 = pic0.resize((250, 250))  
pic0_image = ImageTk.PhotoImage(pic0)  
lab_photo = tk.Label(root, image=pic0_image)  
lab_photo.place(x=223, y=223)
```

- 第1-2行：打开图像文件，将图像大小调整为 250*250 像素
- 第3行：转换为 Tkinter 可显示的 PhotoImage 对象
- 第4-5行：创建标签，并放置在恰当的位置

1.1

窗口创建与布局

Image.open(fp, mode='r')函数

- 定义

用于打开图像文件并创建一个 Image 对象。这个对象可以用来进行各种图像处理操作

- 参数

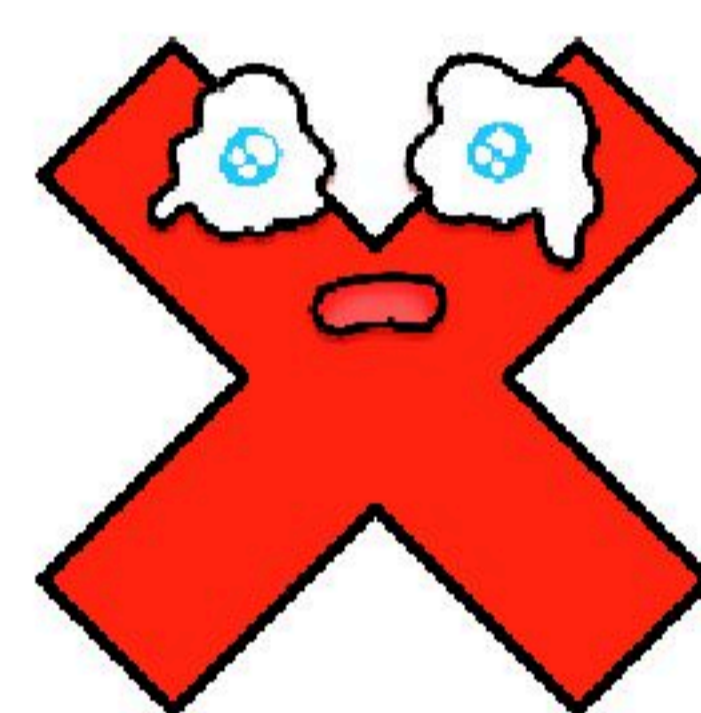
fp -- 文件名（字符串）

mode -- 打开文件的模式。默认为 'r' 只读模式，只能是 'r' 只读模式

- 返回类型

图像对象

`pic0 = pic0.resize((250, 250))` 改为
`pic0 = pic0.resize(250, 250)` 可以吗？
试试看！



报错了

Image对象.resize(size, resample=0)函数

- 定义

用于调整图像的尺寸。返回此图像的调整大小后的副本

- 参数

size -- 一个元组 (width, height) 表示新图像的宽度和高度

resample -- 可选参数，指定在调整大小时使用的采样过滤器

- 返回类型

图像对象

1.1

窗口创建与布局

【设置按钮】

```
# 本地上传按钮
btn_load = tk.Button(root, text="本地上传", command=load,
                    width=15,height=2, font=("楷体", 12))
btn_load.place(x=115, y=625)
```

点击【本地上传】的按钮，会加载选中的图片，这里我们自定义一个load函数，用来实现加载图片的功能

```
# 拍照上传按钮
btn_cam = tk.Button(root, text="拍照上传", command=open_camera,
                   width=15, height=2, font=("楷体", 12))
btn_cam.place(x=300, y=625)
```

点击【拍照上传】的按钮，会打开摄像头，这里我们使用预置的open_camera函数，打开摄像头，并在缩略图区域实时更新摄像头画面

```
# 开始测评按钮
btn_enter = tk.Button(root, text="点击开测！", command=ok,
                    width=15, height=2, font=("楷体", 12))
btn_enter.place(x=485, y=625)
```

点击【点击开测！】的按钮，会对此时显示的缩略图进行人脸检测，返回包含年龄、颜值等的各种信息，这里我们自定义一个ok函数，返回人脸检测结果

```
...
```

【三、函数】

```
...
```

```
...
```

```
# 加载本地图片
def load():
    pass
# 颜值检测开始
def ok():
    pass
```

为了避免程序报错，我们先建立空函数，后面的任务中再一一补充

1.1

窗口创建与布局

任务一新增代码:

```
''' 【一、导入库】 '''
import tkinter as tk
from PIL import Image, ImageTk

''' 【二、初始化】 (变量初始化、窗口初始化、API配置信息) '''
# 初始化Tkinter
root = tk.Tk()
root.geometry('700x700')
root.title('颜值测评')
root.resizable(0, 0)

''' 【三、函数】 '''
...
def load(): # 加载本地图片
    pass
def ok(): # 颜值检测开始
    pass

''' 【四、组件配置】 '''
# 设置背景图
bg_image = tk.PhotoImage(file = '测颜值.png')
bg_label = tk.Label(root, image = bg_image)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)
# 设置缩略图
pic0 = Image.open('缩略图.png')
pic0 = pic0.resize((250, 250))
pic0_image = ImageTk.PhotoImage(pic0)
lab_photo = tk.Label(root, image=pic0_image)
lab_photo.place(x=223, y=223)
# 本地上传按钮
btn_load = tk.Button(root, text="本地上传", command=load, width=15,
                    height=2, font=("楷体", 12))
btn_load.place(x=115, y=625)
# 拍照上传按钮
btn_cam = tk.Button(root, text="拍照上传", command=open_camera,
                    width=15, height=2, font=("楷体", 12))
btn_cam.place(x=300, y=625)
```

1.1

窗口创建与布局

任务一新增代码:

```
# 开始测评按钮
btn_enter = tk.Button(root, text="点击开测!", command=ok, width=15,
                      height=2, font=("楷体", 12))
btn_enter.place(x=485, y=625)

''' 【五、启动事件循环】 '''
root.mainloop()
```

1.2

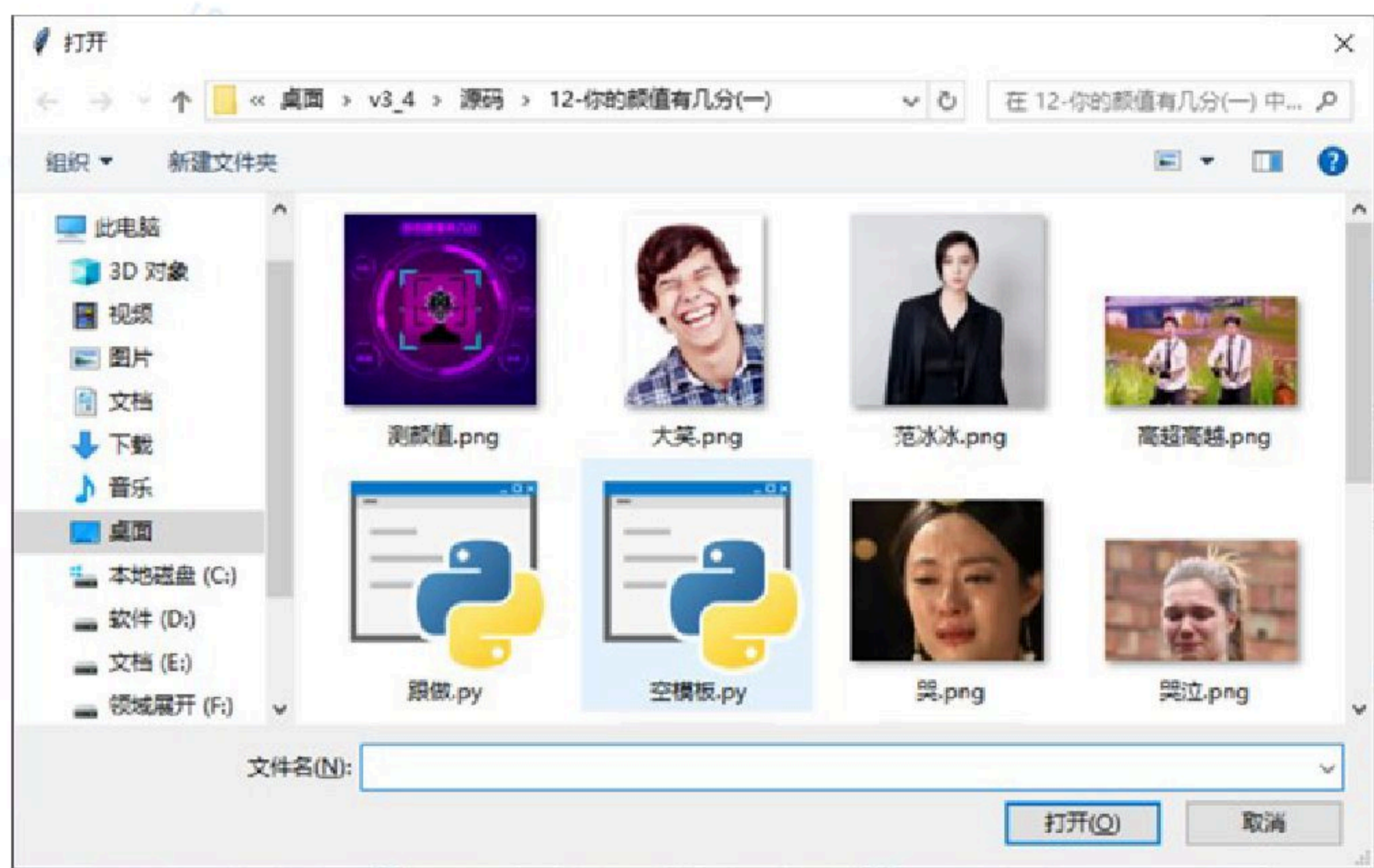
加载本地图片

实现手动选择文件，要用到 tkinter 库中 filedialog 提供的函数

filedialog.askopenfilename()

打开文件对话框，用于选择一个文件，并返回文件路径。

注意：别忘了导入库
from tkinter import filedialog



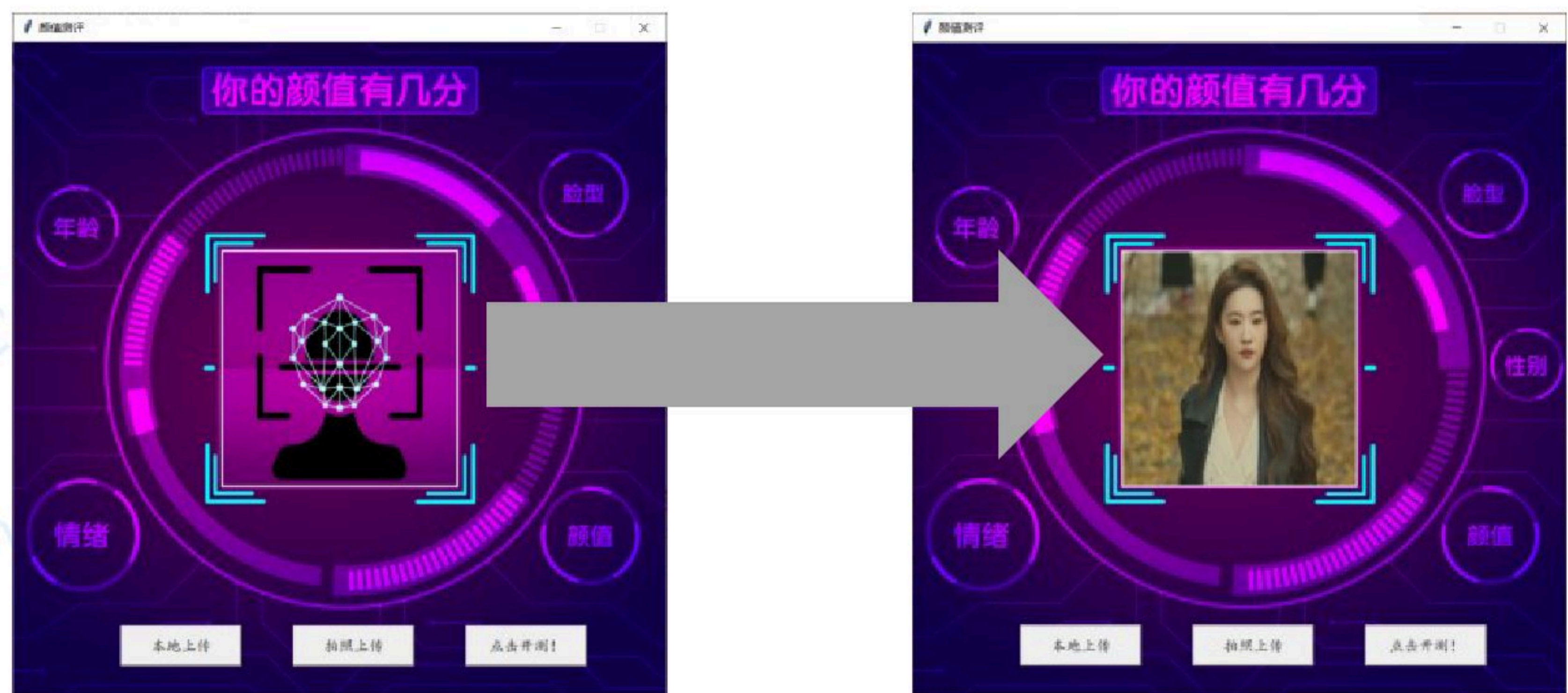
```
from tkinter import filedialog
...
# 加载本地图片
def load():
    file_path = filedialog.askopenfilename()
    ...
# 本地上传按钮
btn_load = tk.Button(root, text="本地上传", command=load, width=15,
                    height=2, font=("楷体", 12))
btn_load.place(x=115, y=625)
...
```



1.2

加载本地图片

选择要加载的文件后，需要将缩略图更新为所选的文件，如何实现？



```
file_path = filedialog.askopenfilename()
if file_path:    # 确保用户已经选择文件
    img = Image.open(file_path)    # 打开该图片文件
    img = img.resize((250, 250))  # 将该文件重设为缩略图大小
    imgtk = ImageTk.PhotoImage(img)
    lab_photo.imgtk = imgtk
    lab_photo.config(image=imgtk)
```

下面详细讲解
这三行代码

imgtk = ImageTk.PhotoImage(img)

问：这行代码我们上个任务也编写过，还记得是什么作用吗？

答：转换为 Tkinter 可显示的 PhotoImage 对象

lab_photo.imgtk = imgtk

lab_photo.config(image=imgtk)



config()函数可以更新标签的属性，我们可以用它更新 lab_photo 的image属性，那么第一行代码的作用是什么呢？

1.2

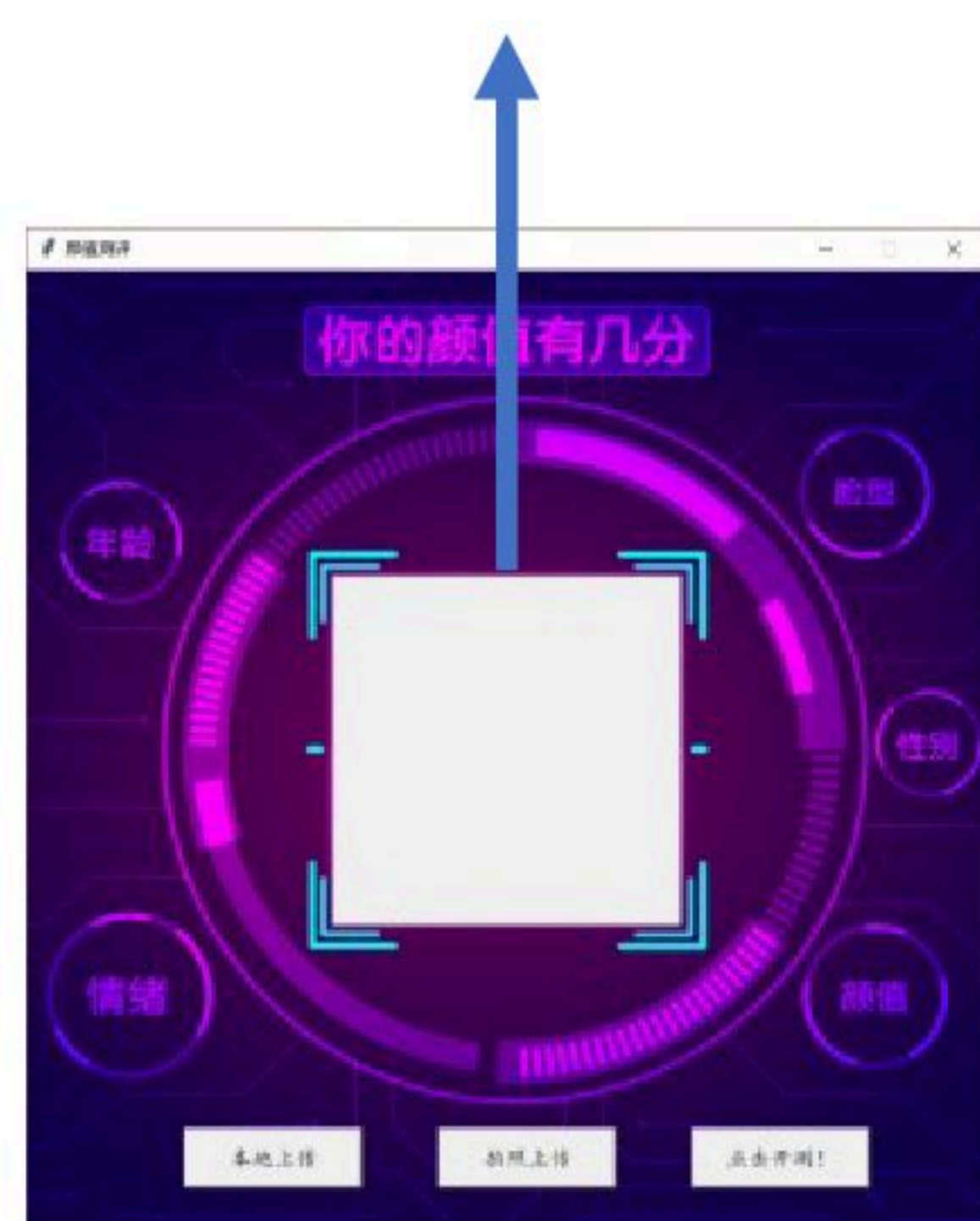
加载本地图片



我们先试试，将 `lab_photo.imgtk = imgtk` 代码注释掉，看效果是否可以实现。

```
file_path = filedialog.askopenfilename()
if file_path:
    img = Image.open(file_path)
    img = img.resize((250, 250))
    imgtk = ImageTk.PhotoImage(img)
    # Lab_photo.imgtk = imgtk
    lab_photo.config(image=imgtk)
```

思考：
图片为什么不见了？



垃圾回收机制

垃圾回收机制（简称GC）是自动内存管理的一部分。主要目的是回收那些不再使用的对象，以释放内存空间。它专注于两件事：找到内存中的无用垃圾资源；清除这些垃圾腾出内存。

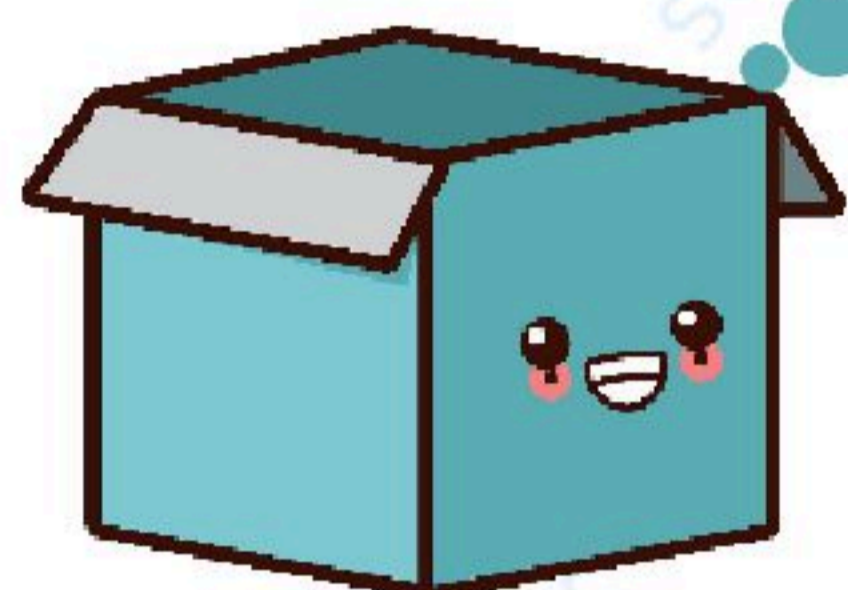


垃圾回收机制跟我的图片消失有什么关系？我的 `imgtk` 变量还有用，为什么就被回收了呢？

这是因为 `tkinter` 小部件对图像对象的引用处理方式。`tkinter` 不是直接通过变量本身，而是通过引用来管理和显示图像。`Tk` 不会保留对图像对象的引用。所以一旦 `load` 函数执行完毕，`Python` 垃圾回收机制可能会认为这个对象 `imgtk` 不再被引用，从而将其回收。

又发现一个无用的变量，回收回收！





一会儿我要用，
赶紧存到我的
私人空间

所以为了确保 `imgtk` 不被垃圾回收，我们通常会使用：`lab_photo.imgtk = imgtk`，将图像对象的一个引用存储在 `lab_photo` 小部件中，确保使用这个小部件的整个生命周期中，该图像对象不被回收。

任务二新增代码：

```
'''
【一、导入库】
'''
from tkinter import filedialog

'''
【三、函数】
'''
# 加载本地图片
def load():
    file_path = filedialog.askopenfilename() # 打开一个文件选择对话框
    if file_path:
        img = Image.open(file_path)
        img = img.resize((250, 250))
        imgtk = ImageTk.PhotoImage(img)
        lab_photo.imgtk = imgtk
        lab_photo.config(image=imgtk)
```



点击【拍照上传】后，若想要改为【本地上传】，发现无法上传图片。该如何优化作品，在上传方式之间丝滑切换呢？

关闭摄像头的代码已经预置好了，更新图片前调用 `close_camera()` 函数即可

任务三新增代码：

```
# 加载本地图片
def load():
    file_path = filedialog.askopenfilename() # 打开一个文件选择对话框
    if file_path:
        close_camera()
        img = Image.open(file_path)
        img = img.resize((250, 250))
        imgtk = ImageTk.PhotoImage(img)
        lab_photo.imgtk = imgtk
        lab_photo.config(image=imgtk)
```

完整代码

```
''' 【一、导入库】 '''
# 预置内容, 请勿改动
import io
import base64
import cv2
import tkinter as tk
from PIL import Image, ImageTk
from tkinter import filedialog

''' 【二、初始化】 (变量初始化、窗口初始化、API配置信息) '''
# 预置内容, 请勿改动
cap = None
imgtk = None
update_id = None
# 初始化Tkinter
root = tk.Tk()
root.geometry('700x700')
root.title('颜值测评')
root.resizable(0, 0)

''' 【三、函数】 '''
# 预置内容, 请勿改动 图像 --> 字节数据 --> Base64 编码格式
def img_to_bytes(photoimg):
    ...
# 预置内容, 请勿改动 打开摄像头
def open_camera():
    ...
# 预置内容, 请勿改动 关闭摄像头
def close_camera():
    ...
# 加载本地图片
def load():
    file_path = filedialog.askopenfilename() # 打开一个文件选择对话框
    if file_path:
        close_camera()
        img = Image.open(file_path)
        img = img.resize((250, 250))
        imgtk = ImageTk.PhotoImage(img)
        lab_photo.imgtk = imgtk
        lab_photo.config(image=imgtk)
```

完整代码

```
# 颜值检测开始
def ok():
    pass

''' 【四、组件配置】 '''
# 设置背景图
bg_image = tk.PhotoImage(file = '测颜值.png')
bg_label = tk.Label(root, image = bg_image)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)
# 设置缩略图
pic0 = Image.open('缩略图.png')
pic0 = pic0.resize((250, 250))
pic0_image = ImageTk.PhotoImage(pic0)
lab_photo = tk.Label(root, image=pic0_image)
lab_photo.place(x=223, y=223)
# 本地上传按钮
btn_load = tk.Button(root, text="本地上传", command=load, width=15,
                      height=2, font=("楷体", 12))
btn_load.place(x=115, y=625)
# 拍照上传按钮
btn_cam = tk.Button(root, text="拍照上传", command=open_camera,
                    width=15, height=2, font=("楷体", 12))
btn_cam.place(x=300, y=625)
# 开始测评按钮
btn_enter = tk.Button(root, text="点击开测!", command=ok,
                      width=15, height=2, font=("楷体", 12))
btn_enter.place(x=485, y=625)

''' 【五、启动事件循环】 '''
root.mainloop()
```



2. 强化练习

1. PIL库的主要用途是什么? ()

- A. 处理文本数据
- B. 处理音频文件
- C. 处理图像文件
- D. 处理视频文件

2. 在Tkinter中, 常用于显示图像的组件是? ()

- A. Entry
- B. Label
- C. Button
- D. Text

3. 如何改变 Tkinter 窗口中组件的大小? ()

- A. 使用width和height属性
- B. 使用size属性
- C. 使用resize方法
- D. 组件大小无法改变

4. 在Python中, 下面哪个函数可以打开文件选择对话框? ()

- A. `filedialog.open()`
- B. `filedialog.show()`
- C. `filedialog.pickfile()`
- D. `filedialog.askopenfilename()`

5. 以下哪个选项正确地表示了如何使用resize方法来改变图像的大小?
()

- A. `image.resize(100, 200)`
- B. `image.resize((100, 200))`
- C. `image.resize([100, 200])`
- D. `image.resize({100, 200})`

3. 术语箱

filedialog	文件对话框
resize	调整大小
load	加载

4. 课后挑战

本地图片浏览器

任务描述：

创建一个简单的图片浏览器。用户可以从计算机中选择一张图片，并在窗口中指定位置显示。

作业要求：

1. 界面：包括【选择】按钮和一个图片显示区域（300*300）
2. 加载本地图片：使用文件对话框让用户选择一张本地图片
3. 图片缩放：选中的图片进行缩放来匹配显示区域
4. 更新显示：选择新图片后，界面应更新所选图片

