# 用 PySimpleGUI 为人工智能应用程序设计图形用户界面

作者：谢作如（温州科技高级中学）

摘要：

在中小学人工智能教育中，训练模型是最核心的学习内容之一。当人工智能模型训练完成之后，无论是结合多模态技术解决真实问题，还是部署模型演示推理能力，往往都需要设计图形用户界面。本文在介绍 PySimpleGUI 这一 Python 界面设计工具的同时，以一个目标检测的 ONNX 模型为例，结合 OpenCV、BaseDeploy、BaseDT 等工具，完整地呈现了一个带图形界面的人工智能应用程序的开发过程。

关键词：模型部署 深度学习 Python 图形用户界面设计



人工智能是一种内敛的技术，需要借助多模态交互技术，才能“具象化”。在 XEdu 的帮助下，学生不需要编写代码也能快速训练出计算机视觉方向的模型，如图像分类和目标检测。那么，下一步的需求就是如何呈现这一学习成果，比如结合摄像头，编写出支持实时画面推理的应用程序。我曾经用 PyWebIO、Gradio 等库介绍过 AI 模型部署方面的案例，但更多的老师希望能编写一个类似 Windows 应用程序一样，有一个中规中矩的图形用户界面，方便学生体验。于是我花了一定的时间，认真研究了 Python 的 GUI 工具选择，毕竟给中小学生使用的工具既要代码简洁，又要功能强大。

## 一、常见的 Python 图形界面设计工具

图形用户界面（Graphical User Interface，简称 GUI，又称图形用户接口）是指采用图形方式显示的计算机操作用户界面。图形用户界面是一种人与计算机通信的界面显示格式，允许用户使用鼠标等输入设备操纵屏幕上的图标或菜单选项，以选择命令、调用文件、启动程序或执行其它一些日常任务。用户界面通常包括许多视觉元素，如图标、按钮、图形、显示文本和多种输入控件，如复选框、文本输入框等。

Python 作为一个容易上手，简单方便的开源编程语言，第三方的开发工具数不胜数，在 GUI 这个方向同样有很多的工具可以选择。比较常用的 GUI 开发工具有 Tkinter、PyQt、wxPython、Gtk+、Kivy、FLTK 和 OpenGL 等，其中最常用是 Tkinter。Tkinter 的优点在于是 Python 内置标准库，无需额外安装，兼容性好，但缺点在于实现效果较为普通，开发体验不好。比较受程序员推崇的是 PyQt 和 wxPython，功能强大，界面优美，相关教程也很多，可惜学习曲线有些陡峭。在比较了多款 GUI

开发工具之后，我最终选择了 PySimpleGUI。

## 二、PySimpleGUI 的安装和典型代码

顾名思义，PySimpleGUI 是一个简单的 GUI 设计工具。PySimpleGUI 的安装和其他库一样，使用 pip 命令即可。整个库很小，很快就能完成。参考命令如下：

pip install  PySimpleGUI

一个典型的 PySimpleGUI 代码，大致包含了导入库、设计窗体布局、创建窗体、窗体事件控制和关闭窗体等五个部分。以一个简单的输入文本并输出结果的程序为例，对核心代码的作用分析如表 1 所示。

表 1 PySimpleGUI 代码分析

| 代码片段 | 作用分析 |
| --- | --- |
| import PySimpleGUI as sg | 导入 PySimpleGUI 库 |
| layout = [<br>[sg.Text('请输入你的名字：')],<br>[sg.Input(key='in')],<br>[sg.Button('确认'), sg.Button('取消')],<br>[sg.Text('输出：'), sg.Text(key='out')]<br>] | 设计窗体布局，用列表来定义每一个元素。 |
| window = sg.Window('PySimpleGUI 范例', layout) | 创建窗体 |
| while True:<br># event 为按钮的名称，values 为一个字典<br>event, values = window.read()<br>if event in (None, '取消'):<br>window['in'].update('')<br>window['out'].update('')<br>else:<br>if values:<br>s = '欢迎你，' + values['in'] | 监视窗体的事件，并响应。 |

| | |
|---|---|
| window['out'].update(s) | |
| window.close() | 关闭窗体 |

这段代码中最核心的部分在于窗体设计和窗体事件控制部分。其中“window.read()”返回的信息中，event 为按钮的名称，values 则为一个字典，键名是控件的名称，如“{'in': 'XEdu'}”。仔细观察 PySimpleGUI 代码，会发现和 Arduino、掌控板之类开源硬件程序的运行逻辑非常类似——用一个无限循环来处理输入和输出窗体事件。该代码的运行效果如下，界面样式中规中矩，看起来并不丑。


## 三、用 PySimpleGUI 编写模型推理程序

借助这样的简单范例，就能写出带输入和输出功能的程序了。考虑到我们需要编写的程序需要结合摄像头，界面中得显示实时画面，那就得借助 Image 类型的对象，然后在窗体事件控制部分中实时更新画面即可。

核心代码介绍：在定义窗口布局中增加 Image 类型对象，用 size 来定义画面的大小。

```
Python
layout = [
  [sg.Image(filename='', key='image',size=(600, 400))],
  [sg.Button('关闭', size=(20, 1))],
  [sg.Text('推理结果：',key='res')]
]
```

核心代码介绍：实时读取摄像头的画面内容，然后用 OpenCV 的 tobytes 功能将图片转化为字节流，并更新 Image 对象。

```
Python
#打开摄像头
cap = cv2.VideoCapture(0)
while True:
```

```
    #实时读取图像，重设画面大小
    ret, frame = cap.read()
    imgSrc = cv2.resize(frame, (600,400))
    res, img = my_inf(frame)
    # 实时更新画面
    imgbytes = cv2.imencode('.png', imgSrc)[1].tobytes()
    window['image'].update(data=imgbytes)
```

核心代码介绍：在事件处理代码中，调用了模型推理的函数“my_inf”，这个函数使用了 BaseDeploy 库。BaseDeploy 库是 XEdu 的一个组件，支持通用的 ONNX 模型。

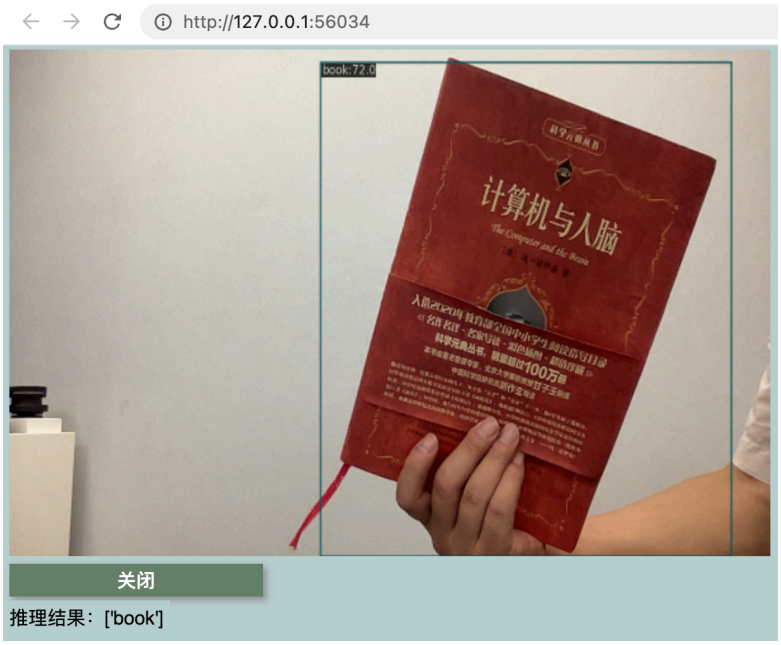
```
Python
model_path = 'det.onnx'
model = bd(model_path)
def my_inf(frame):
    res1, img = model.inference(frame,get_img='cv2')
    # 转换推理结果
    res2 = model.print_result(res1)
    if len(res2) == 0:
        return None,None
    classes = []
    # 提取预测结果中的物品名称
    for res in res2:
        classes.append(res['预测结果'])
    return str(classes),img
```

最终的程序非常简洁，五十行左右，程序运行结果如图所示。

## 四、用 PySimpleGUI 编写 Web 应用程序

我们选择 PySimpleGUI 的最大原因在于其内置了多个 GUI 框架。目前已经集成了 tkinter、PyQt、wxPython 和 Remi 等四种。2021 年，我曾经在一篇《用 Python 写一个基于 Web 的物联网应用程序》文章中介绍过 Remi 库。其支持 Web 界面开发，也就说用 PySimpleGUI 也能开发出 Web 应用。

如果想要将上面的代码效果以 Web 界面呈现，只要将前面的导入库“import PySimpleGUI as sg”一句改为“import PySimpleGUIWeb as sg”，其他代码都不需要改变，体现了“一次编写、处处运行”的理念。运行程序后，浏览器就会自动打开一个页面，显示效果如图所示。




## 五、结语

也许是有 VB 和 Arduino 的编程基础，我在用 PySimpleGUI 编写 GUI 的过程中，感觉非常顺手。可见 PySimpleGUI 很适合有创客教育或者开源硬件编程基础的师生使用。毕竟在几分钟内用几行代码就可以构建自定义 GUI 布局，是很愉快的编程体验。再加上 PySimpleGUI 还内置了 100 多种颜色主题，适合“懒人”拿来就用。当我们能用简洁的代码设计出 GUI，那么将会有更多精彩的人工智能应用程序涌现出来。让我们拭目以待吧。

参考文献：

[1]谢作如.用 Python 写一个基于 Web 的物联网应用程序[J].中国信息技术教育,2021(09):78-81.

[2]谢作如.用 PyWebIO“交互”呈现人工智能学习成果[J].中国信息技术教育,2021(15):82-84.