

**“数字图像处理基础”实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题目 | 实验三：直方图绘制 | | |
| 姓名 | 钟军凯 | 学号 | 22211374 |
| 指导教师 | 彭亚辉 | | |
| 日期 | 2025年3月8日 | | |

**电子信息工程学院**

目 录

[1. 实验目的分析 3](#_Toc191473896)

[2. 实验方案设计 3](#_Toc191473897)

[3. 实验结果 4](#_Toc191473898)

[4. 分析与讨论 6](#_Toc191473899)

[5. 问题发现与探究 7](#_Toc191473900)

[6. 实验总结 8](#_Toc191473901)

[指导老师评价 1](#_Toc191473902)2

1. 实验目的分析

本实验旨在掌握灰度图像与彩色图像直方图的统计方法，并通过使用Python及其相关库进行直方图绘制来提升编程实践能力。

1. 实验方案设计
   1. **实验内容一**

对于gray\_histogram(image)函数，形参设置为PIL的Image对象。用一个长度为256的数组gray\_distribution来记录每一个灰度值出现的频率。通过image.getdata()获取Image对象中的图像数据，然后通过numpy.array()转化为数组。通过numpy.count\_nonzero()来得到一个数组中某个特定数值的个数，然后就可以计算出频率了。接着使用matplotlib来绘制直方图。

* 1. **实验内容二**

对于color\_histogram(image)函数，形参设置为PIL的Image对象。用一个大小为3\*256的数组color\_distribution来记录RGB三个通道的像素值出现的频率。通过image.getdata()获取Image对象中的图像数据，然后通过numpy.array()转化为数组。通过numpy.count\_nonzero()来得到一个数组中某个特定数值的个数，然后就可以计算出频率了。接着使用matplotlib来绘制直方图。

* 1. **实验内容三**

首先使用ttk.Button创建一个按键，这个按键的command属性定义到open\_file函数，也就是说，按下这个按键的时候，将会执行open\_file函数。open\_file函数的大致内容为：使用tkinter的filedialog中的askopenfilename()函数来打开文件选择窗口，该函数只能选择一个文件，最终返回文件的路径。获取到图片文件的路径之后，使用PIL中的Image.open()函数打开图片，得到一个Image对象。然后通过Image.mode判断是什么类型的图像，然后选择gray\_distribution(image)或者color\_distribution(image)。

* 1. **实验内容四**

首先创建一个Figure对象figure，然后创建FigureCanvasTkAgg对象canvas与figure和tkinter界面关联，然后再创建一个NavigationToolbar2Tk对象toolbar与canvas和tkinter界面关联。canvas是图表显示的地方。toolbar是工具栏，可以对图表进行移动、缩放等操作。此时需要注意的是，需要使用canvas.draw()函数，确保canvas中的图像可以更新，需要使用toolbar.update()函数，确保工具栏可以对图像进行操作。若要在canvas中添加图像，可通过graph = figure.add\_subplot()函数来添加。可以使用graph.set\_title()来设置图表标题，使用graph.set\_xlabel()和graph.set\_ylabel()来设置横轴和竖轴的标签。因为直方图是一个柱状图，可以使用graph.bar()来绘制。

1. 实验结果



图 1：初始界面

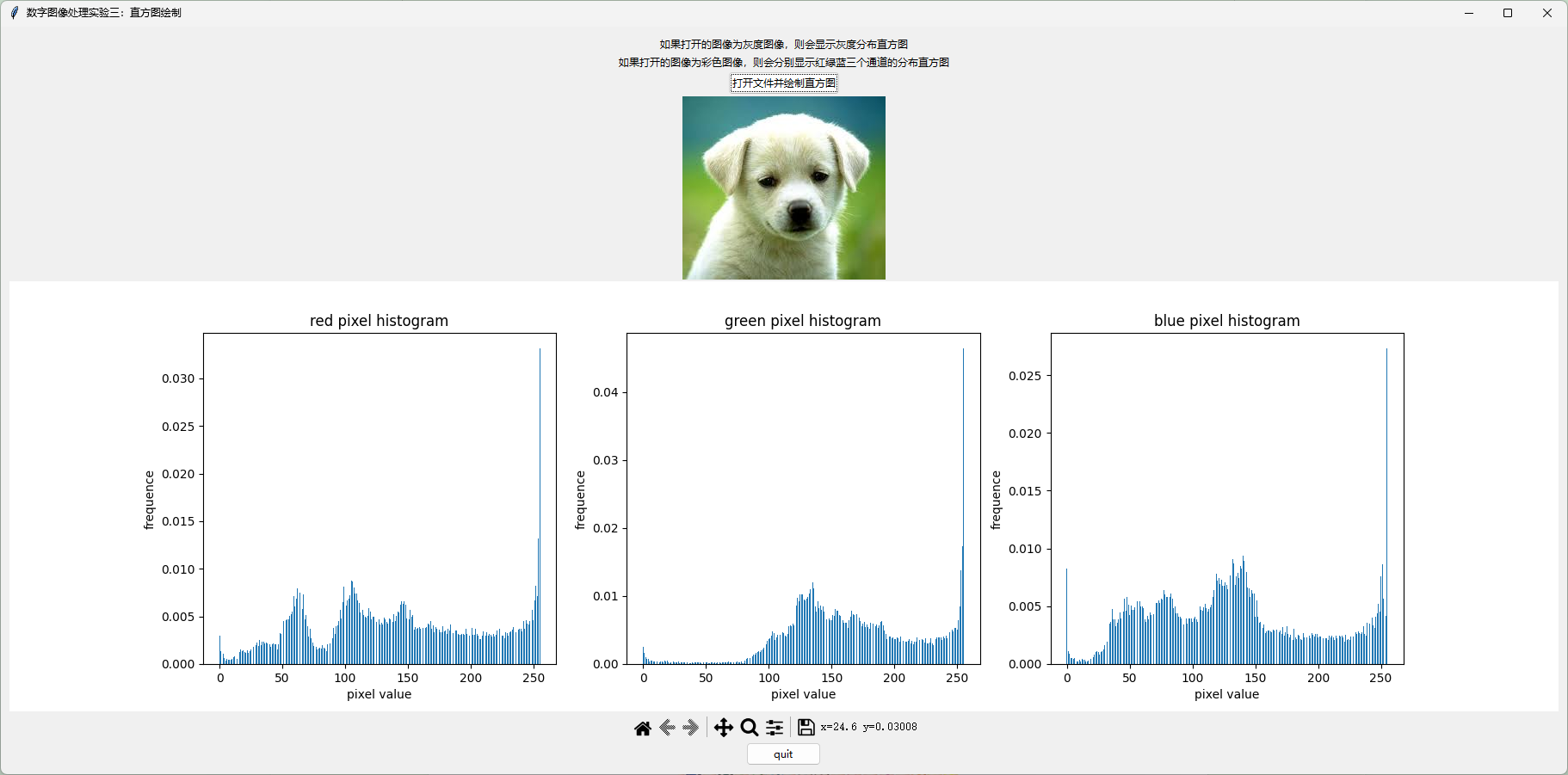


图 2：彩色图像直方图绘制示例1

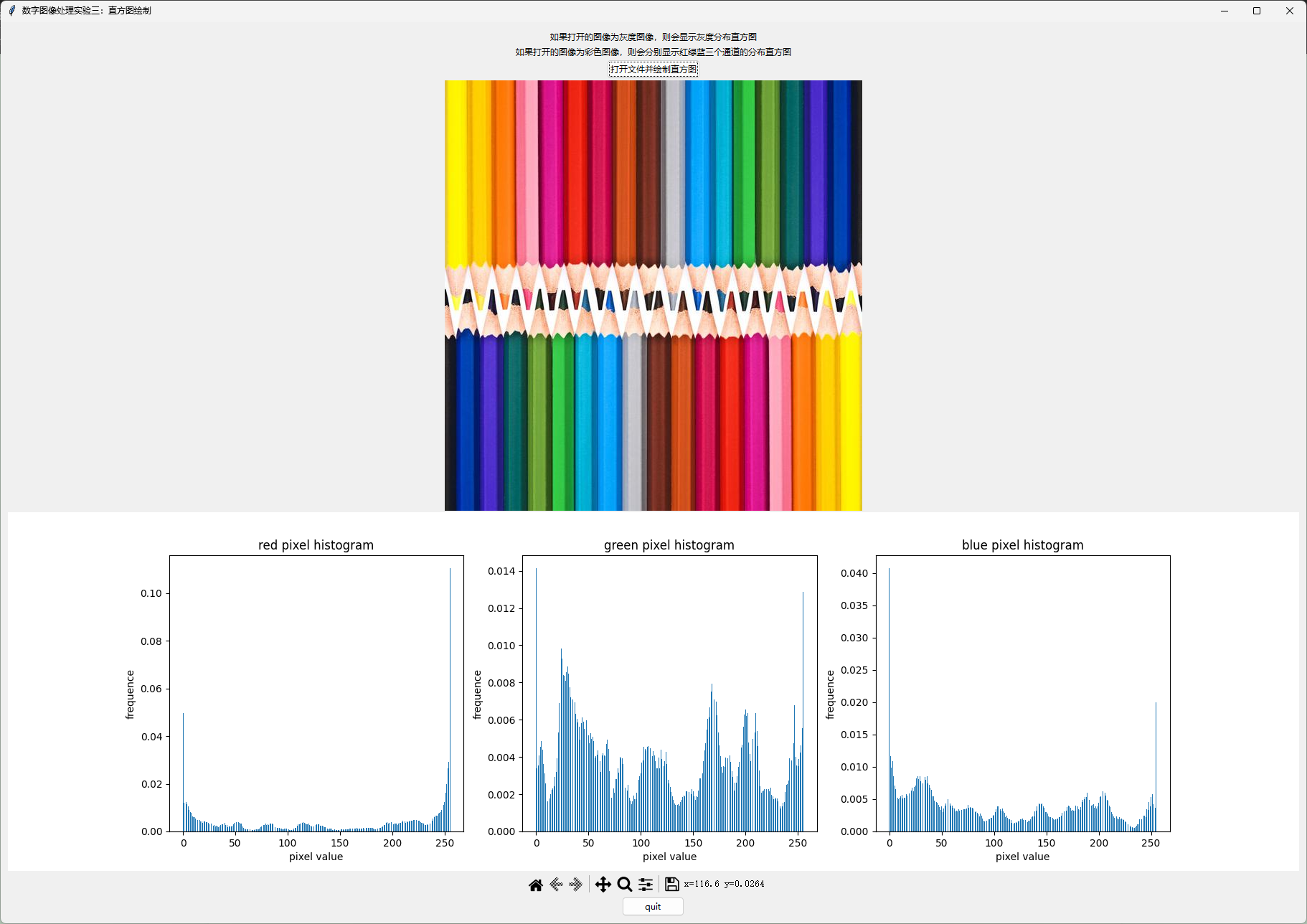


图 3：彩色图像直方图绘制示例2

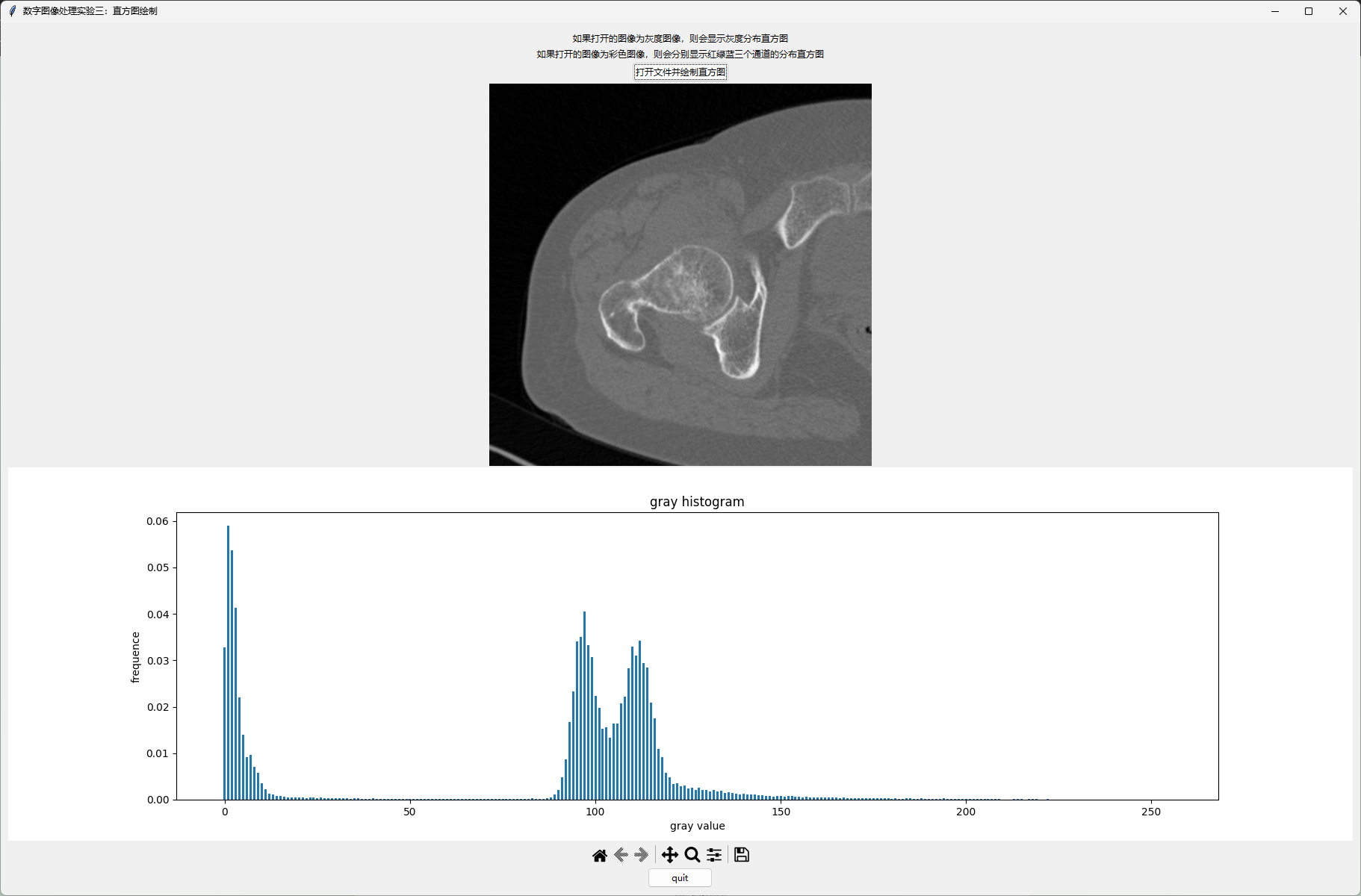


图 4：灰度图像直方图绘制示例1

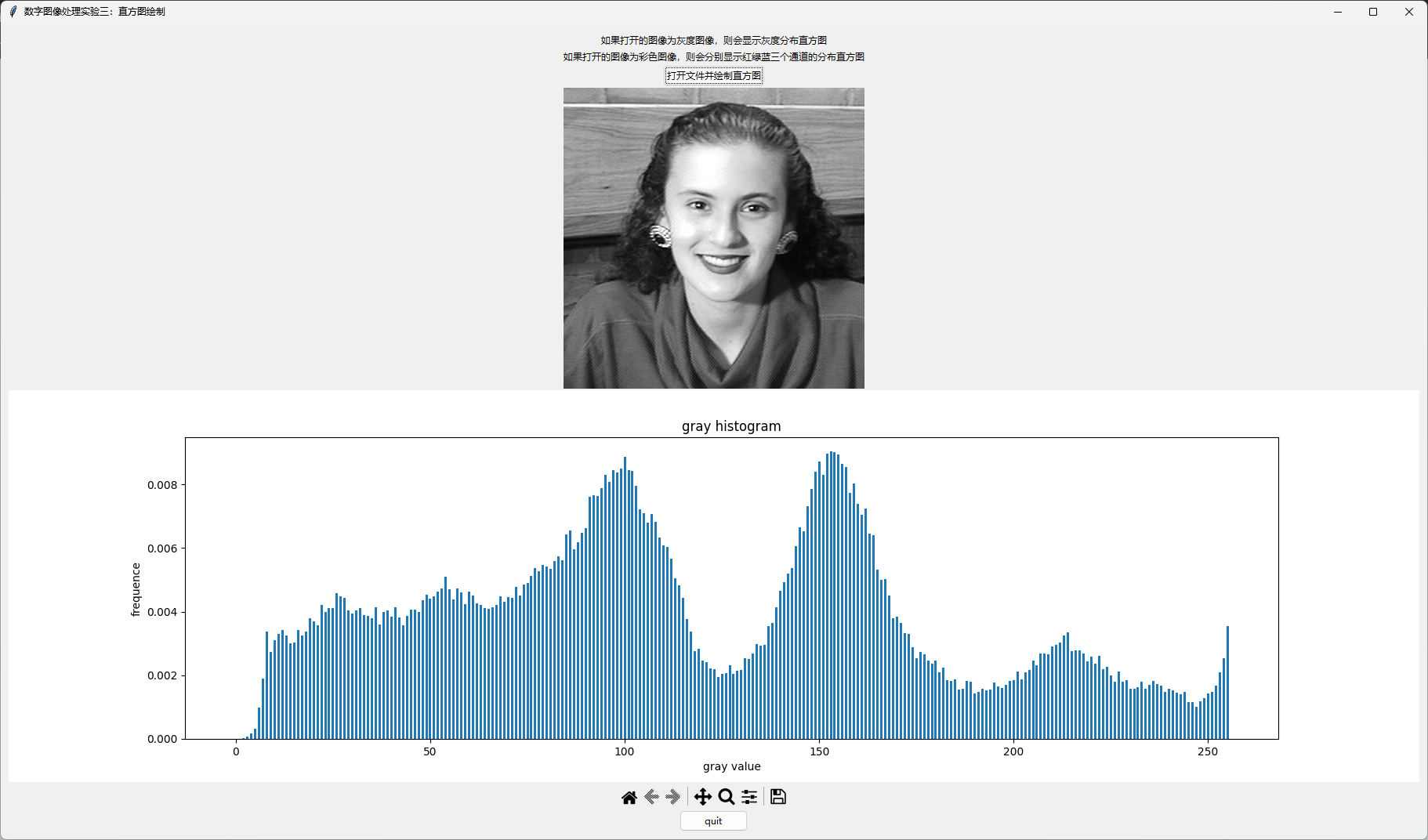


图 5：灰度图像直方图绘制示例2

1. 分析与讨论

对于如图2所示的结果，由于图片背景的偏绿的颜色，又因为图片中的狗是白色的，所以绿色像素的像素值分布会总体位于像素值较高的位置，也就是说，绿色像素具有较多的高像素值。还是因为图中的狗是白色的，所以红色像素和蓝色像素在高像素值处也会有一定数量的分布。因为小狗的眼睛和鼻子处是偏黑的颜色，所以RGB在像素值0处附近也会有相应数量的分布。

对于如图3所示的结果，由于图像在左上和右下两个区域都是偏红色的色调，所以红色像素会在高像素值处有着相应数量的分布，此时蓝色像素的像素值会偏低，所以在低像素值处会有相应数量的分布。由于图像在左下和右上两个区域都是偏蓝色的色调，所以蓝色像素会在高像素值处有着相应数量的分布，此时红色像素的像素值会偏低，所以在低像素值处有着相应数量的分布。而绿色像素的分布相对红色和蓝色来说，会比较均匀。

对于如图4所示的结果，由于图像的颜色较少，所以直方图呈现出了分段的形状。因为图像黑色和灰色所占比例较大，所以在灰度值为0的位置附近和直方图中间的位置，会有灰度像素的分布。图像中也有偏白的颜色，但是所占比例太少，所以在直方图中的呈现效果不明显。

对于如图5所示的结果，因为该灰度图像的颜色较为均匀，所以它的灰度直方图所呈现出来的灰度分布也会比较均匀。

1. 问题发现与探究
   1. **问题一**

【**问题描述**】

histogram.title("红色通道直方图")报错：TypeError: 'Text' object is not callable。'Text'是不可调用的。

【**问题解决**】

histogram.title是一个属性，也可以理解为一个变量，不是函数，所以不可调用（not callable）。应该使用histogram.set\_title()来设置标题。

* 1. **问题二**

【**问题描述**】

如图6所示，图表更新时，旧图表与新图标重叠。

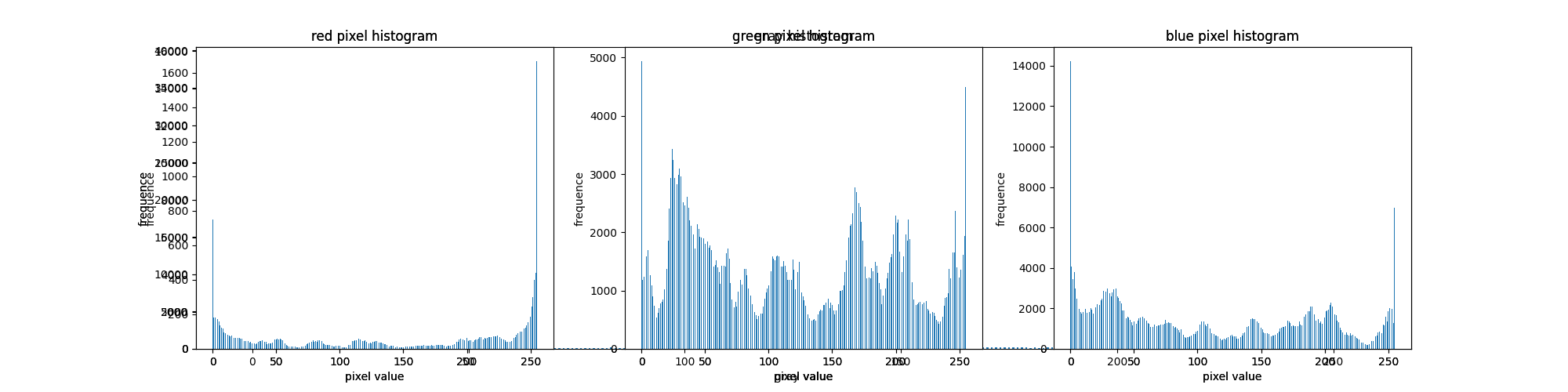


图 6：新旧图表重叠

【**问题解决**】

尝试了多种方式都没有解决：在绘图部分的代码之后加上canvas.draw()函数来使图表更新，但是没有作用；在绘图部分的代码之前加上histogram.clear()来清空，但是也是不行。

最后发现，自己在每次绘图之前，都会使用histogram = figure.add\_subplot()函数，所以导致每次绘图都会添加新的plot区域，新旧图表重叠就是因为这是不同的plot区域，这些plot区域是相互独立的，所以上一段中的两种方式会没有效果。所以，在每次绘图之前，使用figure.clear()清空所有plot区域，然后在使用histogram = figure.add\_subplot()来创建新的plot区域。

1. 实验总结

在本次实验中，我学会了如何在tkinter界面中嵌入matplotlib的图表，也对matplotlib的使用有了一定程度的了解。与此同时也对tkinter的使用进行了巩固，对相关操作比之前更加熟悉了，如文件交互、显示图片等。此外，我还了解到了numpy中可以提高效率的函数，比如numpy.count\_nonzero()、numpy.arange()等。

附录：

【仿真程序】

|  |
| --- |
| import tkinter as tk  from tkinter import ttk  from tkinter import filedialog as fd  from PIL import Image  from PIL import ImageTk  import numpy as np  from matplotlib import pyplot as plt  from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg, NavigationToolbar2Tk  from matplotlib.figure import Figure  # 创建基本的界面  root = tk.Tk()  root.title("数字图像处理实验三：直方图绘制") # 设置界面标题  frame = ttk.Frame(root, padding = 10)  frame.grid()  '''  @brief 获取灰度直方图分布  @param image: Image对象  '''  def gray\_histogram (image):  all\_gray = np.arange(256) # 所有灰度值  image\_array = np.array(image.getdata()) # 获取图像数据并转化为数组  gray\_distribution = np.zeros(256) # 用于统计每个灰度值出现的频率  # 遍历所有灰度值  for i in range(256):  gray\_distribution[i] = np.count\_nonzero(image\_array == i) / (image.size[0] \* image.size[1]) # 获取图像中每个灰度值出现的频率  # 绘制直方图  figure.clear() # 清除图窗中的图像  histogram = figure.add\_subplot()  histogram.bar(all\_gray, gray\_distribution, width = 0.6)  histogram.set\_title("gray histogram")  histogram.set\_xlabel("gray value")  histogram.set\_ylabel("frequence")  canvas.draw()    '''  @brief 获取RGB三个通道的直方图分布  @param image: Image对象  '''  def color\_histogram (image):  all\_pixel\_value = np.arange(256) # 所有像素值  image\_array = np.array(image.getdata()) # 获取图像数据并转化为数组  color\_distribution = np.zeros((3, 256)) # 用于分别统计每个RGB值出现的频率  # 遍历所有RGB值  for i in range(3):  for j in range(256):  color\_distribution[i, j] = np.count\_nonzero(image\_array[:, i] == j) / (image.size[0] \* image.size[1]) # 分别获取图像中每个RGB值出现的频率  # 绘制直方图  figure.clear() # 清除图窗中的图像  histogram = figure.add\_subplot(1, 3, 1)  histogram.bar(all\_pixel\_value, color\_distribution[0, :], width = 0.6)  histogram.set\_title("red pixel histogram")  histogram.set\_xlabel("pixel value")  histogram.set\_ylabel("frequence")  histogram = figure.add\_subplot(1, 3, 2)  histogram.bar(all\_pixel\_value, color\_distribution[1, :], width = 0.6)  histogram.set\_title("green pixel histogram")  histogram.set\_xlabel("pixel value")  histogram.set\_ylabel("frequence")  histogram = figure.add\_subplot(1, 3, 3)  histogram.bar(all\_pixel\_value, color\_distribution[2, :], width = 0.6)  histogram.set\_title("blue pixel histogram")  histogram.set\_xlabel("pixel value")  histogram.set\_ylabel("frequence")  canvas.draw()  '''  @brief 打开文件  @param none  '''  def open\_file ():  global image\_tk  file\_path = fd.askopenfilename() # 获取文件路径  image = Image.open(file\_path) # 通过图片路径打开图片  image\_tk = ImageTk.PhotoImage(image) # 将图像转化为tkinter可用的PhotoImage对象  # 显示图片  image\_label = ttk.Label(frame, image = image\_tk)  image\_label.grid(row = 3, column = 0)  # 判断显示什么类型图像的直方图  if image.mode == "L":  gray\_histogram(image)  elif image.mode == "RGB":  color\_histogram(image)  # 提示语  tip\_1 = ttk.Label(frame, text = "如果打开的图像为灰度图像，则会显示灰度分布直方图")  tip\_1.grid(row = 0, column = 0)  tip\_2 = ttk.Label(frame, text = "如果打开的图像为彩色图像，则会分别显示红绿蓝三个通道的分布直方图")  tip\_2.grid(row = 1, column = 0)  # 创建“打开文件”按键  open\_file\_button = ttk.Button(frame, text = "打开文件并绘制直方图", command = open\_file) # 将按键的回调定位到open\_file函数  open\_file\_button.grid(row = 2, column = 0)  # 创建“quit”按键  quit\_button = ttk.Button(frame, text = "quit", command = root.destroy)  quit\_button.grid(row = 6, column = 0)  # 创建图窗  figure = Figure(figsize = (18, 5), dpi = 100)  # 创建嵌入tkinter界面的图窗部件  canvas = FigureCanvasTkAgg(figure, frame)  canvas.draw() # 确保图窗中的图像可以更新  canvas.get\_tk\_widget().grid(row = 4, column = 0)  # 创建图窗工具栏  toolbar = NavigationToolbar2Tk(canvas, frame, pack\_toolbar = False)  toolbar.update() # 确保工具栏能够更新图像  toolbar.grid(row = 5, column = 0)  root.mainloop() |

指导老师评价

|  |  |  |
| --- | --- | --- |
|  | | |
| 仿真实验报告成绩 | 交流展示成绩 | 总评成绩 |
|  |  |  |

指导教师：

年 月 日

**实验报告提交说明:**

文件名 **姓名\_学号\_实验x.docx**

文件格式 Microsoft Word