

《多媒体技术》实验报告二

黄勛 22920212204392

1. 运行程序截图和简要说明

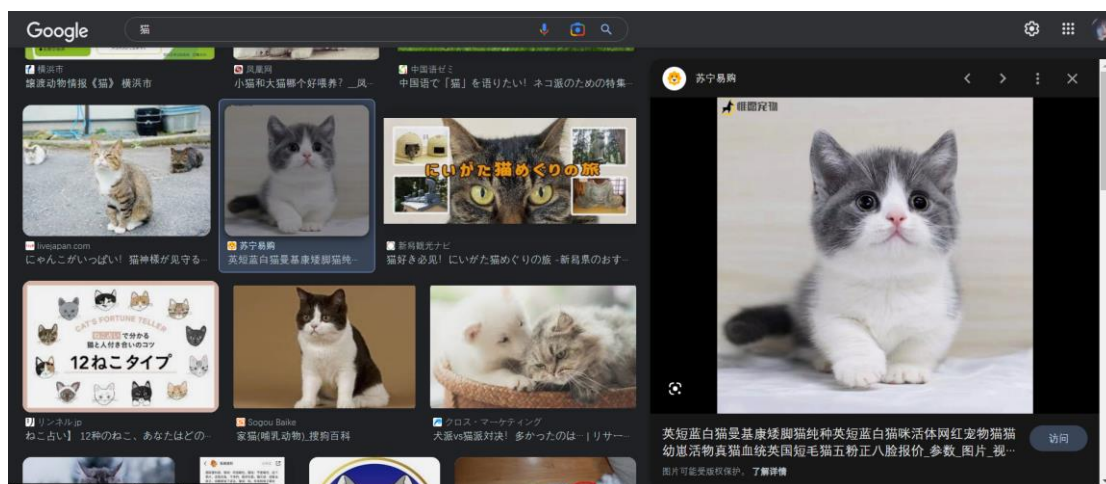
1) Python 环境依赖库安装

```
Successfully installed pip 20.2
PS C:\Windows\system32> pip install numpy
Collecting numpy
  Downloading numpy-1.24.2-cp39-cp39-win_amd64.whl (14.9 MB)
    |#####| 14.9 MB 133 kB/s
Installing collected packages: numpy
Successfully installed numpy-1.24.2
WARNING: You are using pip version 20.2; however, version 23.0.1 is available.
You should consider upgrading via the 'D:\Program Files\python.exe -m pip install --upgrade pip' command.
PS C:\Windows\system32>
```

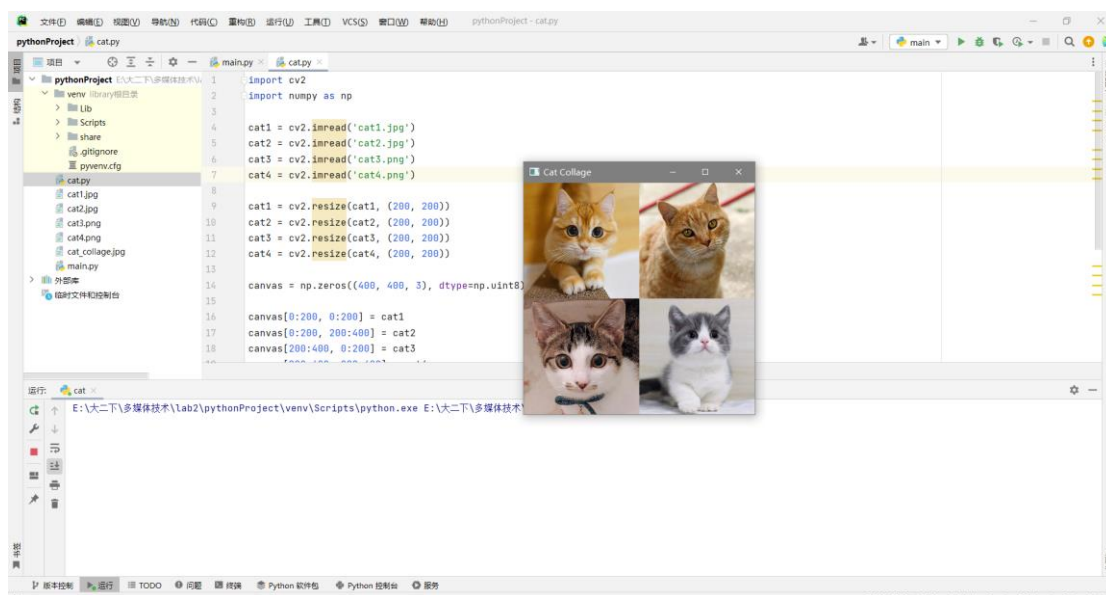
(剩余安装省略)

注：安装 opencv 使用的命令是 `pip install opencv-python`

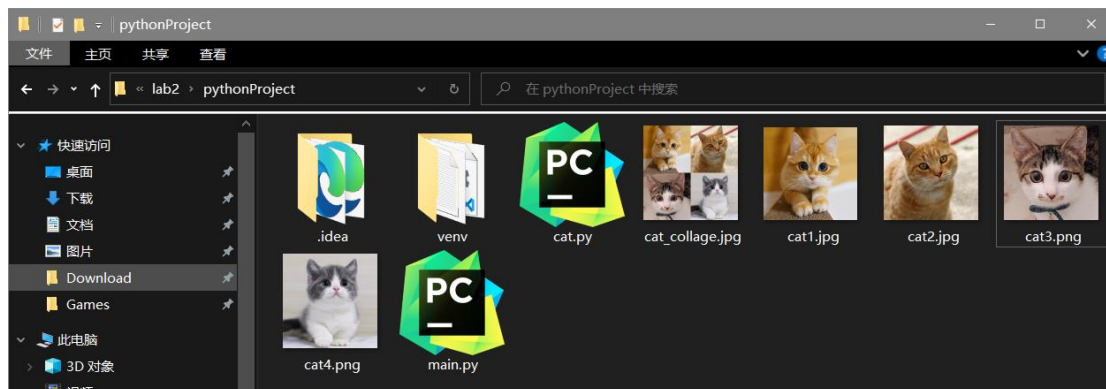
2) 搜索并下载猫的图片：



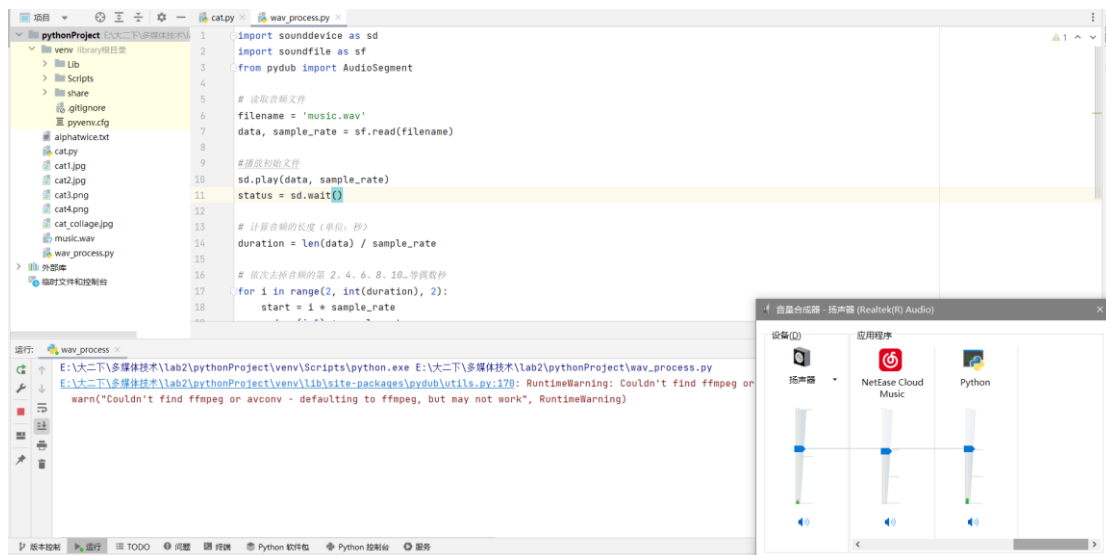
3) 编写代码运行与展示：



4) 保存生成的文件:



5) 利用 sounddevice, SoundFile, pydub 库, 读取一段音乐的音频文件 (music.wav) 并播放

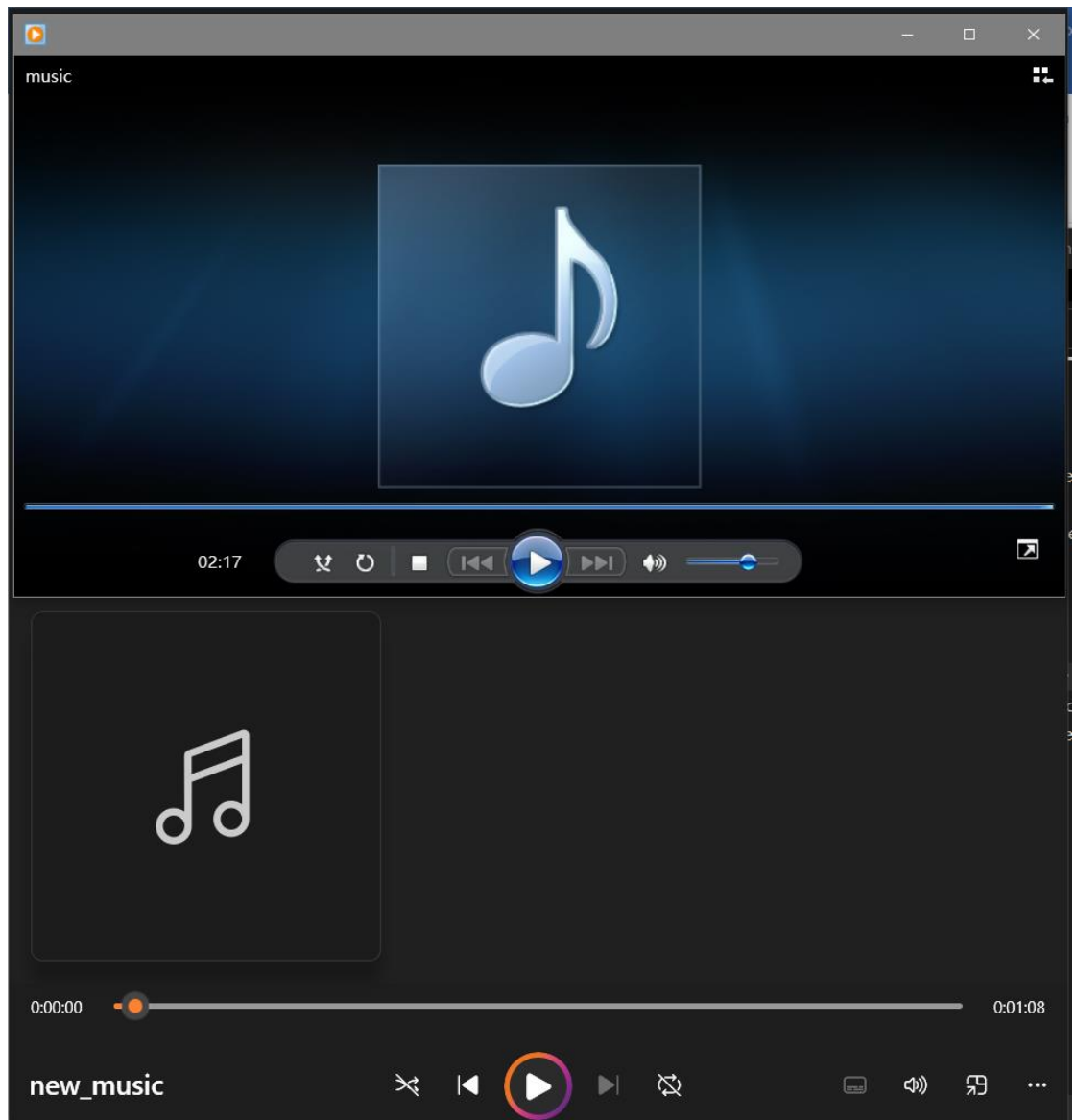


(如上图, 运行的程序正在播放原来的 wav)

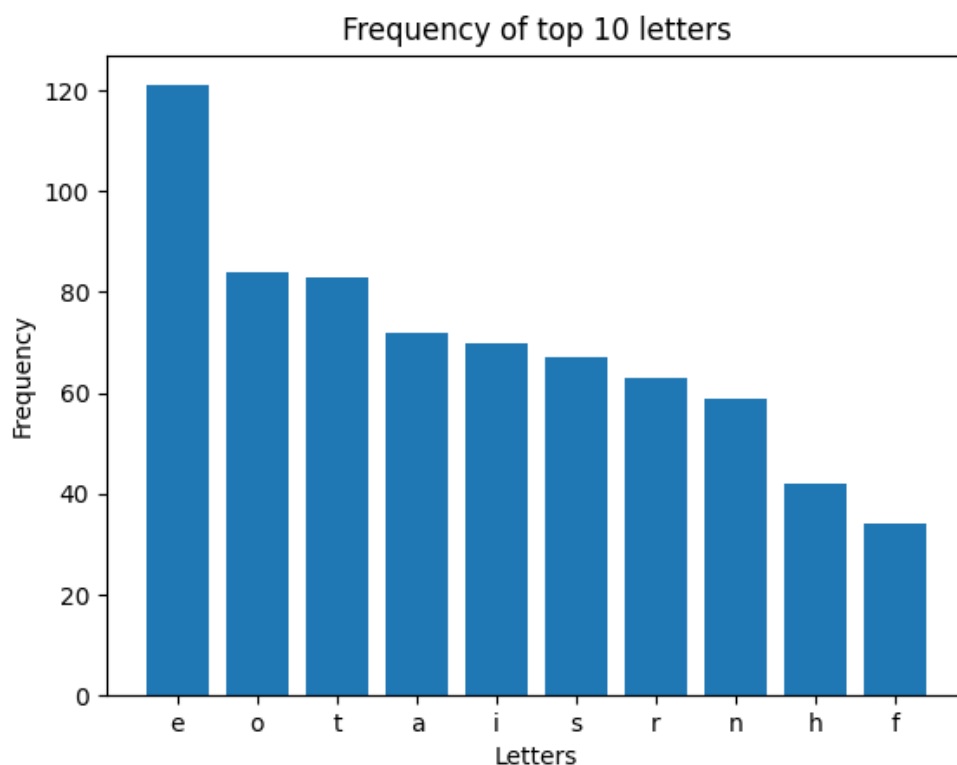
6) 依次去掉该音乐的第 2、4、6、8、10...等偶数秒内的音频, 然后保存为一个新的音频文件。

music.wav	2015/3/12 20:56	WAV 文件	23,727 KB
new music.wav	2023/3/21 17:49	WAV 文件	11,841 KB

通过打开文件, 可以看出长度是原来的一半, 播放效果也是一秒一秒有明显断裂感的。



- 7) 读取一个文本文档 (alphanatwice.txt)，统计该文档内英文字母（区分大小写）的出现次数，选择出现次数最高的 10 个英文字母，利用 matplotlib 库绘制“字母-次数”直方图，最后将直方图保存为一个图像文档。



2.主要代码展示和分析

1) 问题 1

```
1 import cv2
2 import numpy as np
3
4 cat1 = cv2.imread('cat1.jpg')
5 cat2 = cv2.imread('cat2.jpg')
6 cat3 = cv2.imread('cat3.png')
7 cat4 = cv2.imread('cat4.png')
8
9 cat1 = cv2.resize(cat1, (200, 200))
10 cat2 = cv2.resize(cat2, (200, 200))
11 cat3 = cv2.resize(cat3, (200, 200))
12 cat4 = cv2.resize(cat4, (200, 200))
13
14 canvas = np.zeros((400, 400, 3), dtype=np.uint8)
15
16 canvas[0:200, 0:200] = cat1
17 canvas[0:200, 200:400] = cat2
18 canvas[200:400, 0:200] = cat3
19 canvas[200:400, 200:400] = cat4
20
21 cv2.imwrite('cat_collage.jpg', canvas)
22 cv2.imshow('Cat Collage', canvas)
23 cv2.waitKey(0)
24 cv2.destroyAllWindows()
```

我的代码使用了 OpenCV 来读取和处理图像。它的主要功能是将四张猫的图像缩放成 200x200 大小，然后将它们合并成一个 400x400 大小的画布，最后保存为新的图像和在屏幕上显示。

- 首先，导入了必要的库 `cv2` 和 `numpy`，分别用于图像处理和数组操作。
 - 然后，读取四张猫的图片并存储在变量 `cat1`, `cat2`, `cat3` 和 `cat4` 中。`cv2.imread()` 函数读取图片，返回一个包含像素值的 Numpy 数组，信道顺序为 BGR（蓝、绿、红）。
 - 接下来，使用 `cv2.resize()` 函数将每个图像缩放为 200x200 大小，这将使它们适合在画布上显示。
 - 创建一个大小为 400x400x3 的零数组，其中 `dtype` 参数指定数组数据类型为 8 位无符号整数 (`np.uint8`)。
 - 然后，通过赋值操作将四个缩放后的图像放置在画布的适当位置上。切片符号 “:” 用于指定数组的行和列的范围。例如，`canvas [0: 200, 0: 200]` 表示将 `cat1` 放在画布左上角，`canvas [200: 400, 200: 400]` 表示将 `cat4` 放在画布右下角。
 - 最后，将合成的画布保存为 “`cat_collage.jpg`”，并在屏幕上显示。`cv2.imwrite()` 函数将画布保存为 JPG 图像文档，`cv2.imshow()` 函数显示图像，`cv2.waitKey()` 等待用户按下任意键关闭图像窗口，`cv2.destroyAllWindows()` 释放所有与窗口有关的资源。
- (注释掉的部分是一个简单的绘图代码示例，创建一个大小为 23x23 的黑色图像，使用 `cv2.circle()` 函数绘制一个中心坐标为 (100,100)、半径为 22 的圆，并将图像保存为 “`myphoto.jpg`”，然后在屏幕上显示)

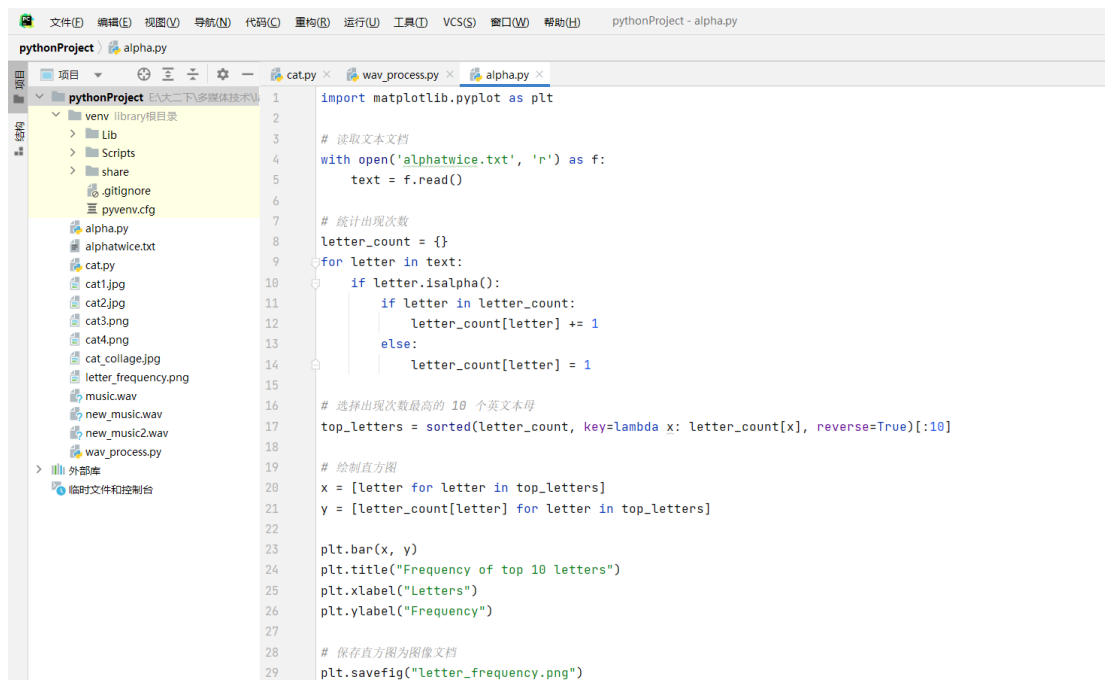
2) 问题 2

在这段代码中，我首先使用 `AudioSegment.from_file()` 方法读取音频文档，并使用 `len()` 方法获取音频的时长，单位为毫秒。然后，我们使用 `AudioSegment.empty()` 方法定义一个新的空音频对象，用于存储去掉偶数秒的音频。

接下来，我使用 `range()` 函数循环遍历每一秒的音频。在循环中，我使用模运算判断当前

秒数是否为偶数秒。如果是奇数秒，则获取该秒的音频并添加到新的音频对象中。最后，我使用 `export()` 方法将新的音频保存为 WAV 格式的文档。

3) 问题 3



我的这段代码主要使用了 `matplotlib` 库来绘制直方图和保存图像文档，流程如下：

- 通过 `open` 函数读取 "alphanat.txt" 文本文档，
- 使用一个字典 `letter_count` 来统计每个字母在文本中出现的次数。使用 `isalpha` 方法来过滤掉非字母字符。
- 使用 `sorted` 函数对 `letter_count` 字典中的键值进行排序，根据字母出现次数的大小进行排序，从而得到出现次数最高的前 10 个字母。
- 将得到的前 10 个字母以及它们的出现次数分别存储在两个列表 `x` 和 `y` 中，用于绘制直方图。
- 使用 `plt.bar` 函数绘制直方图，并使用 `plt.title`、`plt.xlabel` 和 `plt.ylabel` 函数设置图像的标题、X 轴和 Y 轴标签。
- 使用 `plt.savefig` 函数将绘制的直方图保存为一个图像文档 "letter_frequency.png"。

3.其他

本次实验涉及到了三个不同的任务，分别是图像处理、音频处理和文本处理，它们分别使用了不同的库和方法，但它们都有一个共同点，即利用计算机处理不同类型的数据。这些任务的完成不仅需要良好的编程能力，还需要对不同类型数据的处理方法有深刻的理解。通过本次实验，我们可以更深入地理解计算机对不同类型数据的处理方式，以及不同库和方法的具体指示，有很大的帮助。