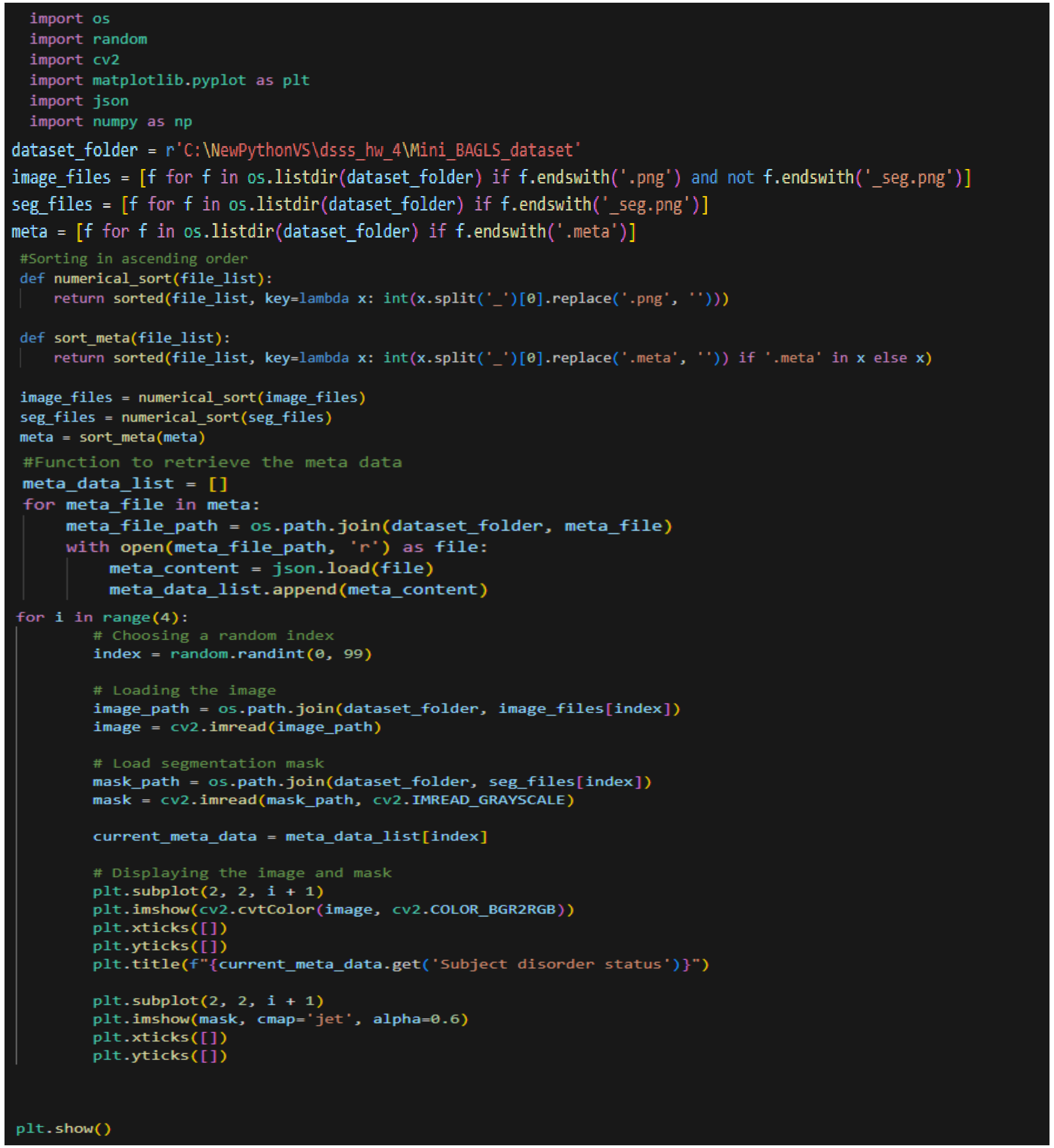
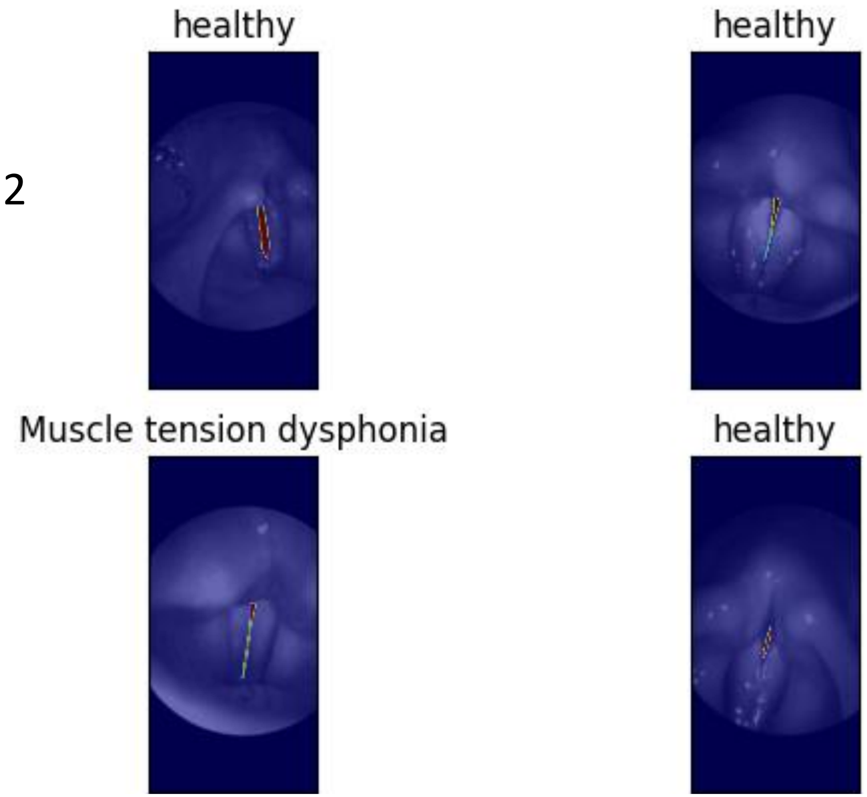
Task 1



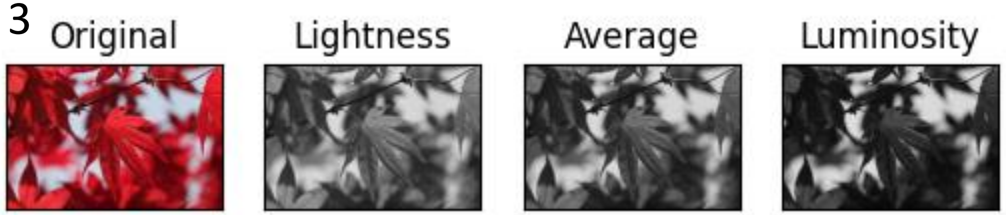
```
 import os
 import random
 import cv2
 import matplotlib.pyplot as plt
 import json
 import numpy as np
dataset_folder = r'C:\NewPythonVS\dsss_hw_4\Mini_BAGLS_dataset'
image_files = [f for f in os.listdir(dataset_folder) if f.endswith('.png') and not f.endswith('_seg.png')]
seg_files = [f for f in os.listdir(dataset_folder) if f.endswith('_seg.png')]
meta = [f for f in os.listdir(dataset_folder) if f.endswith('.meta')]
#Sorting in ascending order
def numerical_sort(file_list):
    return sorted(file_list, key=lambda x: int(x.split('_')[0].replace('.png', '')))

def sort_meta(file_list):
    return sorted(file_list, key=lambda x: int(x.split('_')[0].replace('.meta', '')) if '.meta' in x else x)

image_files = numerical_sort(image_files)
seg_files = numerical_sort(seg_files)
meta = sort_meta(meta)
#Function to retrieve the meta data
meta_data_list = []
for meta_file in meta:
    meta_file_path = os.path.join(dataset_folder, meta_file)
    with open(meta_file_path, 'r') as file:
        meta_content = json.load(file)
        meta_data_list.append(meta_content)
for i in range(4):
        # Choosing a random index
        index = random.randint(0, 99)

        # Loading the image
        image_path = os.path.join(dataset_folder, image_files[index])
        image = cv2.imread(image_path)

        # Load segmentation mask
        mask_path = os.path.join(dataset_folder, seg_files[index])
        mask = cv2.imread(mask_path, cv2.IMREAD_GRAYSCALE)

        current_meta_data = meta_data_list[index]

        # Displaying the image and mask
        plt.subplot(2, 2, i + 1)
        plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
        plt.xticks([])
        plt.yticks([])
        plt.title(f"{current_meta_data.get('Subject disorder status')}")

        plt.subplot(2, 2, i + 1)
        plt.imshow(mask, cmap='jet', alpha=0.6)
        plt.xticks([])
        plt.yticks([])


plt.show()
```

Alap Mundayoor, M.N. 23351429, IdM: ow07ysym

Task 2



Task 3



Task 4

From a computing point of view, I think the lightness method is preferred owing to the lesser number of floating point arithmetic operations being used when compared to the other two methods. This would hence, ultimately lead to lesser round off errors. The other two methods depend on the resolution of an image (since the weighted average for luminosity and average method take into account number of pixels) and more round off errors could arise.