

```
In [1]: import sys
[sys.executable] -m pip install rawpy

Requirement already satisfied: rawpy in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (0.18.0)
Requirement already satisfied: numpy in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from rawpy) (1.24.3)

[notice] A new release of pip is available: 23.0 -> 23.1.2
[notice] To update, run: pip install --upgrade pip

In [2]: import sys
[sys.executable] -m pip install matplotlib opencv-python-headless

Requirement already satisfied: matplotlib in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (3.7.1)
Requirement already satisfied: opencv-python-headless in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (4.7.8.72)
Requirement already satisfied: pyrsimg>=2.3.1 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (1.0.7)
Requirement already satisfied: packaging>=20.0 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (23.1)
Requirement already satisfied: numpy>=1.20 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: fonttools>=4.22.0 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (4.39.3)
Requirement already satisfied: cycler>=0.10 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: kiwisolver>=3.0.1 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from matplotlib) (9.5.0)
Requirement already satisfied: six>=1.5 in /mnt/c/Users/harri/tasks/jupyter/.jup/lib/python3.10/site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

[notice] A new release of pip is available: 23.0 -> 23.1.2
[notice] To update, run: pip install --upgrade pip

In [3]: import os
import rawpy

directory = "Pictures"

images = []
for filename in os.listdir(directory):
    if filename.lower().endswith(".cr2"):
        with rawpy.imread(os.path.join(directory, filename)) as raw:
            images.append(raw.postprocess())
```

```
In [4]: import numpy as np

def mean_color(coords):
    mean_colors = []
    for img in images:
        img_shape = img.shape[:2]
        x_coords, y_coords = zip(*coords)
        x_min, x_max = min(x_coords), max(x_coords)
        y_min, y_max = min(y_coords), max(y_coords)
        img_slice = (slice(y_min, y_max), slice(x_min, x_max), slice(None))
        color_values = img[img_slice].reshape(-1,3)
        mean_colors.append(np.mean(color_values, axis=0))
    return mean_colors
```

```
In [8]: import cv2
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle

coords_exp = ((700, 1880), (2000, 2600))
coords_control = ((3450, 4750), (2000, 2600))

xy1 = (coords_exp[0][0], coords_exp[1][0])
length1 = coords_exp[0][1] - coords_exp[0][0]
height1 = coords_exp[1][1] - coords_exp[1][0]

xy2 = (coords_control[0][0], coords_control[1][0])
length2 = coords_control[0][1] - coords_control[0][0]
height2 = coords_control[1][1] - coords_control[1][0]

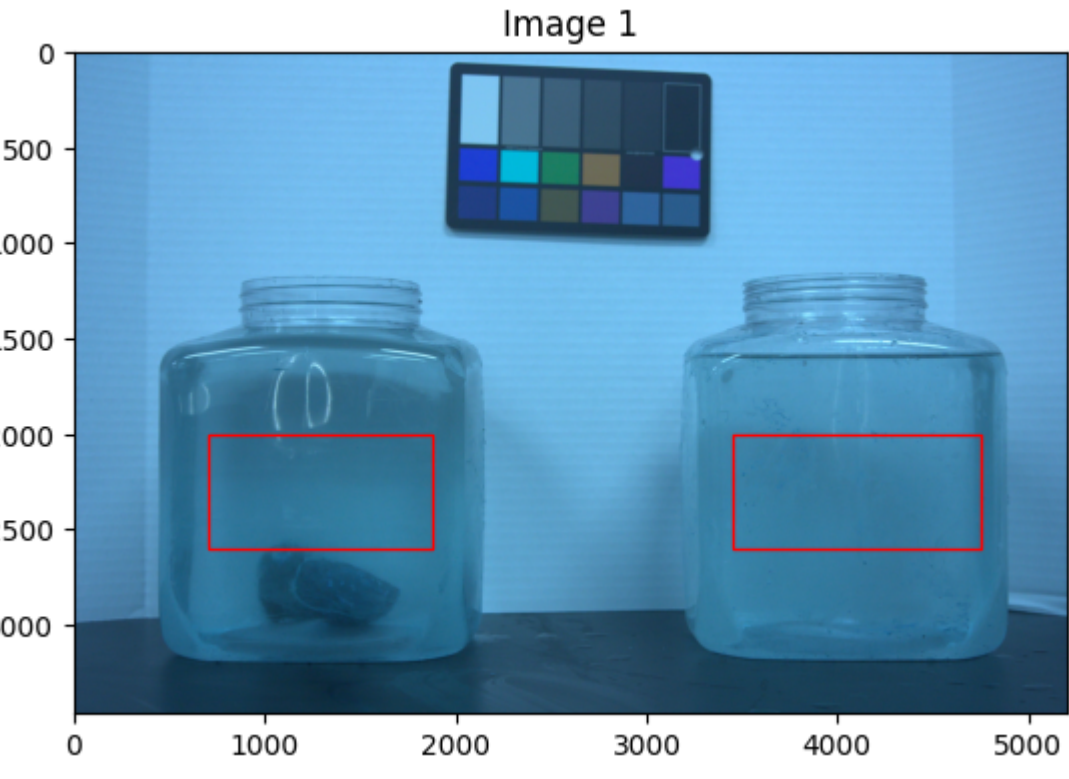
for i, img in enumerate(images):

    img_8bit = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8U)
    fig, ax = plt.subplots()
    ax.imshow(cv2.cvtColor(img_8bit, cv2.COLOR_BGR2RGB))

    rect1 = Rectangle(xy1, length1, height1, linewidth=1, edgecolor='r', facecolor='none')
    rect2 = Rectangle(xy2, length2, height2, linewidth=1, edgecolor='r', facecolor='none')
    ax.add_patch(rect1)
    ax.add_patch(rect2)

    plt.title(f"Image {i + 1}")

    #just visualize one
    break
plt.show()
```



```
In [6]: import matplotlib.pyplot as plt
import numpy as np

import matplotlib.pyplot as plt
import numpy as np

coords_exp = ((700, 1880), (2000, 2600))
coords_control = ((3450, 4750), (2000, 2600))

# Your RGB data for the two datasets
rgb_data1 = mean_color(coords_exp)
rgb_data2 = mean_color(coords_control)

# Separate the data into R, G, and B channels for each dataset
r_data1 = [point[0] for point in rgb_data1]
g_data1 = [point[1] for point in rgb_data1]
b_data1 = [point[2] for point in rgb_data1]

r_data2 = [point[0] for point in rgb_data2]
g_data2 = [point[1] for point in rgb_data2]
b_data2 = [point[2] for point in rgb_data2]

# Set the x-axis range (num1, num2) for the data you want to display
num1 = 18
num2 = 40

# Slice the data arrays
r_data1 = r_data1[num1:num2]
g_data1 = g_data1[num1:num2]
b_data1 = b_data1[num1:num2]

r_data2 = r_data2[num1:num2]
g_data2 = g_data2[num1:num2]
b_data2 = b_data2[num1:num2]

# Set the y-axis range for each plot
y_range_r_1 = (152, 155)
y_range_r_2 = (132, 135)
y_range_g_1 = (118, 122)
y_range_g_2 = (101, 105)
y_range_b_1 = (52, 55)
y_range_b_2 = (52, 55)

# Create the subplots in a 3x2 grid
fig, ((ax1, ax4), (ax2, ax5), (ax3, ax6)) = plt.subplots(3, 2, figsize=(15, 15), sharey='none')

# Plot the R channel data for dataset 1
ax1.plot(r_data1, color='red')
ax1.set_ylim('y_range_r_1')
ax1.set_title('Red Channel - Dataset 1')
ax1.set_xlabel('Index')
ax1.set_ylabel('Value')

# Plot the G channel data for dataset 1
ax2.plot(g_data1, color='green')
ax2.set_ylim('y_range_g_1')
ax2.set_title('Green Channel - Dataset 1')
ax2.set_xlabel('Index')
ax2.set_ylabel('Value')

# Plot the B channel data for dataset 1
ax3.plot(b_data1, color='blue')
ax3.set_ylim('y_range_b_1')
ax3.set_title('Blue Channel - Dataset 1')
ax3.set_xlabel('Index')
ax3.set_ylabel('Value')

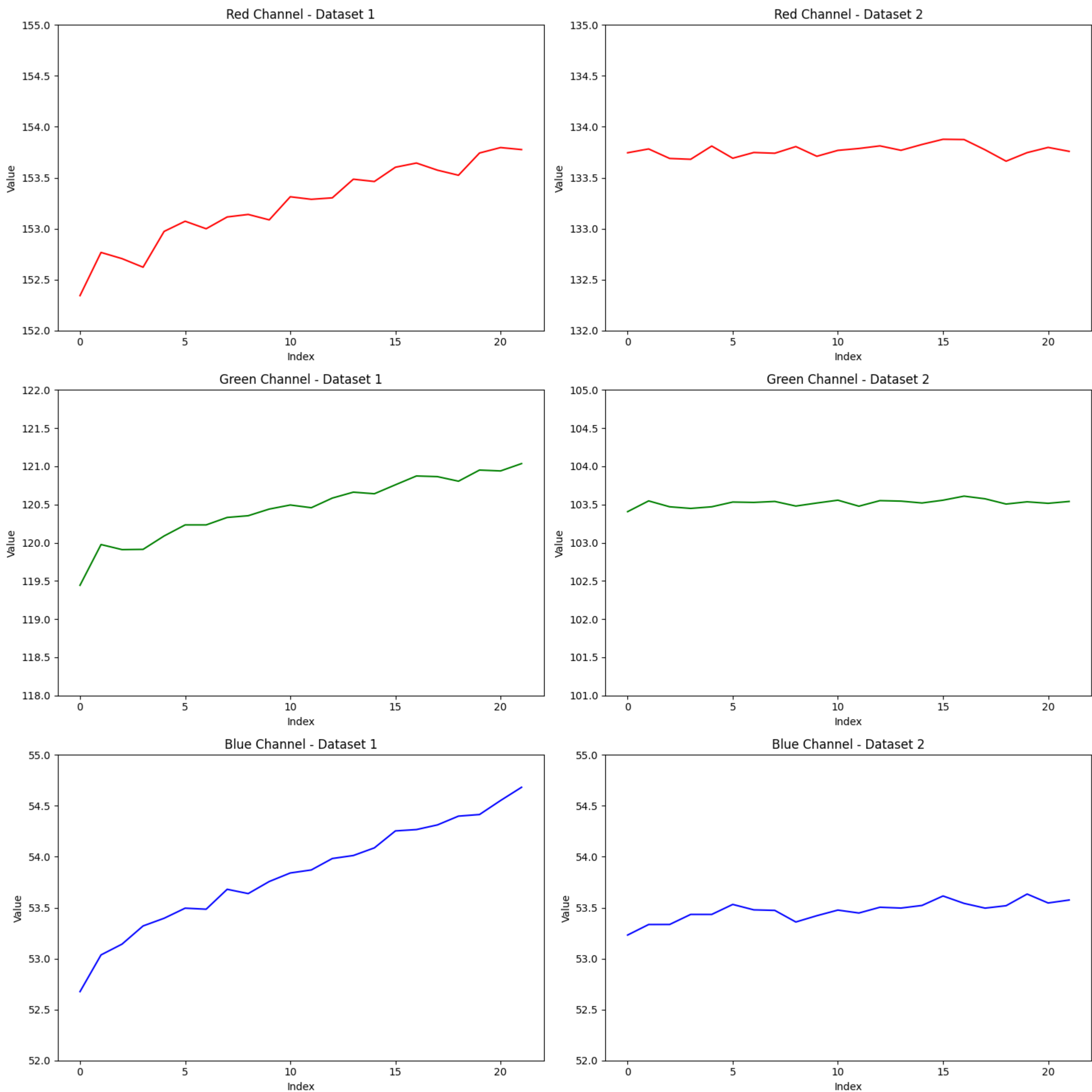
# Plot the R channel data for dataset 2
ax4.plot(r_data2, color='red')
ax4.set_ylim('y_range_r_2')
ax4.set_title('Red Channel - Dataset 2')
ax4.set_xlabel('Index')
ax4.set_ylabel('Value')

# Plot the G channel data for dataset 2
ax5.plot(g_data2, color='green')
ax5.set_ylim('y_range_g_2')
ax5.set_title('Green Channel - Dataset 2')
ax5.set_xlabel('Index')
ax5.set_ylabel('Value')

# Plot the B channel data for dataset 2
ax6.plot(b_data2, color='blue')
ax6.set_ylim('y_range_b_2')
ax6.set_title('Blue Channel - Dataset 2')
ax6.set_xlabel('Index')
ax6.set_ylabel('Value')

# Adjust the layout
plt.tight_layout()

# Display the graphs
plt.show()
```



```
In [7]: """
The comparison of the 3 pairs of graphs, which represent the average RGB values of the experimental (left)
and control (right)tubs, demonstrates a noticeable impact of the oysters' filtration on the water quality
within an hour. In the provided code, images are first loaded and processed from a specified directory.
Then, the average color values of designated regions within the experimental and control groups are
calculated. These regions correspond to the areas where the oysters are located.
"""
```

```
Out[7]: "\n\nThe comparison of the 3 pairs of graphs, which represent the average Rgb values of the experimental (left)\n\nand control (right)\n\nsubs, demonstrates a noticeable impact of the oysters' filtration on the water quality\n\nwithin an hour. In the provided code, images are first loaded and processed from a specified directory.\n\nThen, the average color values of designated regions within the experimental and control groups are\n\n\nrecalculated. These regions correspond to the areas where the oysters are located.\n\n\n"
```