1. Create a new conda environment

conda create -n imagebc

2. Activate the created environment

conda activate imagebc

3. Install all packages from requirements.txt

- We need to install 'pip' first, which helps us install other packages.
  - Code: *conda install pip*
- Install all required packages: *pip install -r requirements.txt*

4. Change the path to all input pictures

- Go to "*config.yaml*" and assign the path to the correct folder
  - input_images: data/input_images/**sample**
  - calibration_images: data/input_images/**sample**

5. Navigate to the "app" folder

- cd path/to/image_bc /filter_extraction/app
- Code: *python main.py extract*

Changing radius for offline python analysis

"utils.py"- line 387 => default radius = 24

"filter_extraction.py" – line 77 => insert (color_corrected_image, radius = 24, show_circle = show_extracted_circles)

Visualizing with "Streamlit" function

- To install streamlit in the right environment
  - Go to the *imagebc* environment: *conda activate imagebc*
  - *pip install -r requirements.txt* (only when running the package for the first time in an env)
- The apps are in folder – path/to/streamlit
  - Enter into the folder: **cd /d D:\streamlit**
- Run code: **streamlit run app.py**
- **Note:** If you want to change the radius of the extraction circle,
  - Go to *app.py* in Spyder
  - Line 121 – modify *radius*

How to change the change RGB outputs

After Eniola changed the Lowe's ratio in the code, I observed a change in the RGB outputs when the change was made. I see LOWE_RATIO in the following places:

- ***utils.py*** (line **101**, 124, 137)

```
def run_sift(img1, img2, SHOW_PLOT=False, LOWE_RATIO=0.9):
    sift = cv.SIFT_create()
    kp1, des1 = sift.detectAndCompute(img1, None)
    res1 = cv.drawKeypoints(
        img1, kp1, img1.copy(), flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

    sift = cv.SIFT_create()
    kp2, des2 = sift.detectAndCompute(img2, None)
    res2 = cv.drawKeypoints(
        img2, kp2, img2.copy(), flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```

- ***filter_extraction.py*** (line **46**, 48, 51, 67)

```
46  def extract_artifacts(original_input_image, LOWE_RATIO = 0.85):
47
48      if(LOWE_RATIO > 1):
49          raise Exception("Failed to extract artfacts")
50
51      input_image = utils.run_sift(reference_image, original_input_image, SHOW_PLOT=show_sift_plot, LOWE_RATIO=LOWE_RATIO
52      input_image_grayscale = utils.convert_to_grayscale(input_image)
53
54      # if use_ui for calibration is false: low = 127, high = 255
55      input_low, input_high = utils.calibrate_threshold(input_image_grayscale, use_ui=use_ui_for_calibration)
56
57      # Convert grayscale image to B/W
58      (input_thresh, input_threshold) = cv2.threshold(input_image_grayscale, input_low, input_high, cv2.THRESH_BINARY_INV
59
60      # Load up image, extract the positions and colors of all 30 boxes
61      (marked_image, target_contours, target_colors) = utils.extract_all_points(input_image, input_threshold)
62
63      if((len(target_contours) == 30) and (len(target_colors) == 30)):
64          return (input_image, marked_image, target_contours, target_colors)
65
66      print("Increasing lowe ratio by 0.05")
67      return extract_artifacts(original_input_image, LOWE_RATIO=LOWE_RATIO+0.05)
```