HW#2　　學號：110810006　　姓名：林君曆

Component Labeling

1. Convert the color image to a binary image

```python
image_paths = [f'./images/img{i}.png' for i in range(1, 5)]
# print(image_paths)
binary_images = []
thresholds = [120, 190, 210, 210]

i = 1
for path in image_paths:
    img = cv2.imread(path)
    grey_img = ImageConverter.convert_color_to_greyscale(img)
    binary_img = ImageConverter.convert_greyscale_to_binary(grey_img, threshold=thresholds[i-1])
    # cv2.imwrite(f'./binary_img/binary_image_{i}.jpg', binary_img)
    binary_images.append(binary_img)
    i += 1
```

Reuse the defined static function of image converter done by HW#1.
After doing experiments, found out the best threshold, stored in the threshold array, to turn the grey to binary. In the for loop, I process the images to binary images and store them.

2. Labeling components using 4-connected and 8-connected
   - Function structure
     def process_img
     | def generate_random_color
       : As the name of function, it generate the unique RGB value for a label
     | def get_4neighbors_marks
       : Because the neighbors are symmetric, only top and left marks of the specific pixel are needed. But, we need to deal with the several edge cases to avoid “index out of range” error.
     | def get_8neightbors_marks
       : We only need the four marks top, top_left, top, top_right because of the symmetry. But, we need to deal with the several edge cases to avoid “index out of range” error.
   - Main part of the algorithm in function process_img
     In general, there are two cases when the pixel we are look at is a foreground pixel.
     1) The neightbors are all unmarked. Then we increase the label count and assign the new label number to this pixel.
     2) There are some marked neighbor. Then these marks are all in the same group(object). In this case, we have to record the group, and mark a representative mark of that group to the current pixel. To record this kind of relationship, I apply a union_find data structure here with better time complexity.

After marking all the pixel, we iterate throught all the pixel again and assign a unique RGB color to each group. I declare a color_map to record the existing color and the RGB corresponding to the certain representative mark of groups.

3. Output color image
   After all I done above, I process all the image and save them. Because I run the 4-connected and 8-connected separated, the two version of same image will have different color mapping.

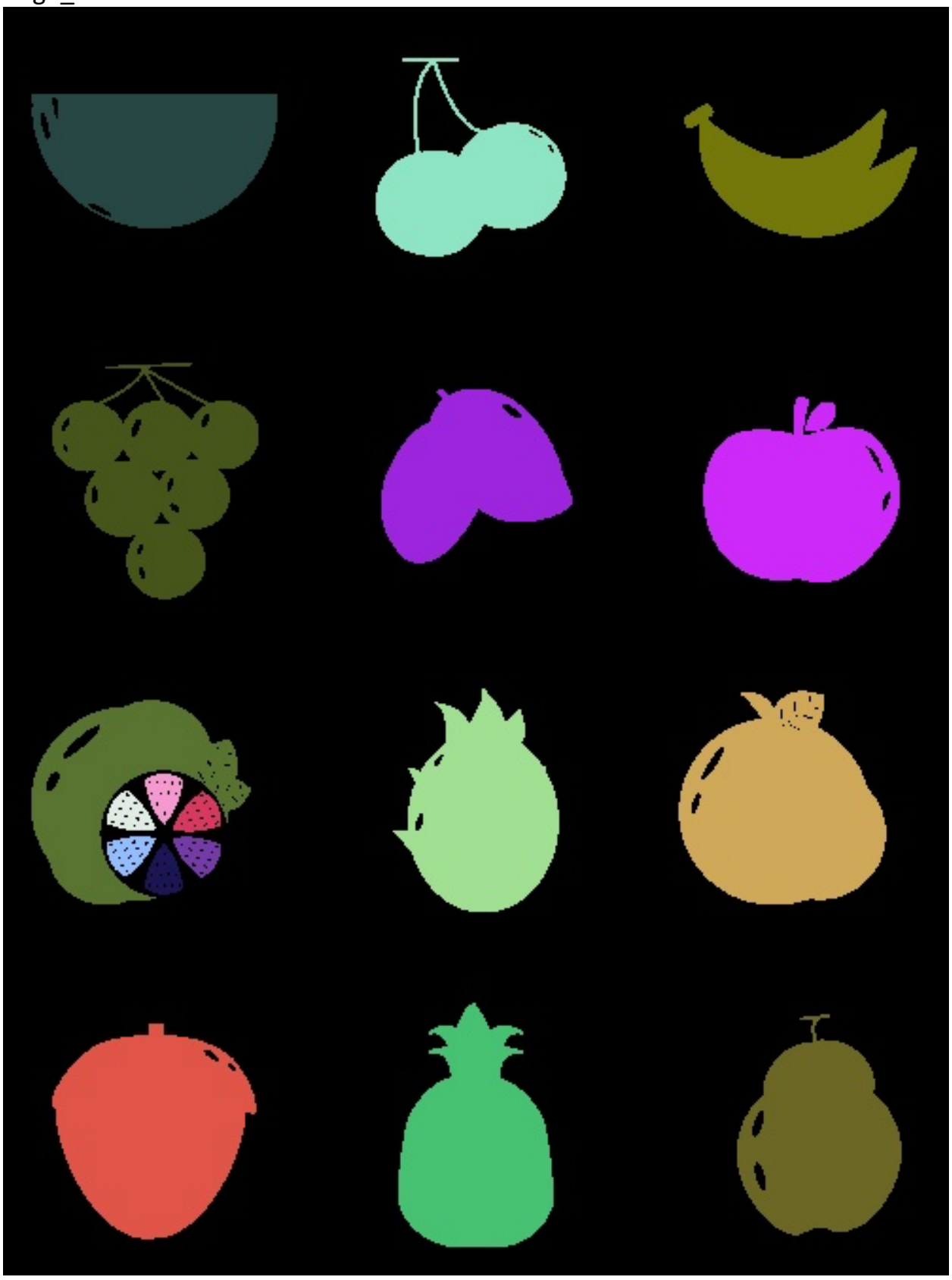
   Img1_4:

Img1_8

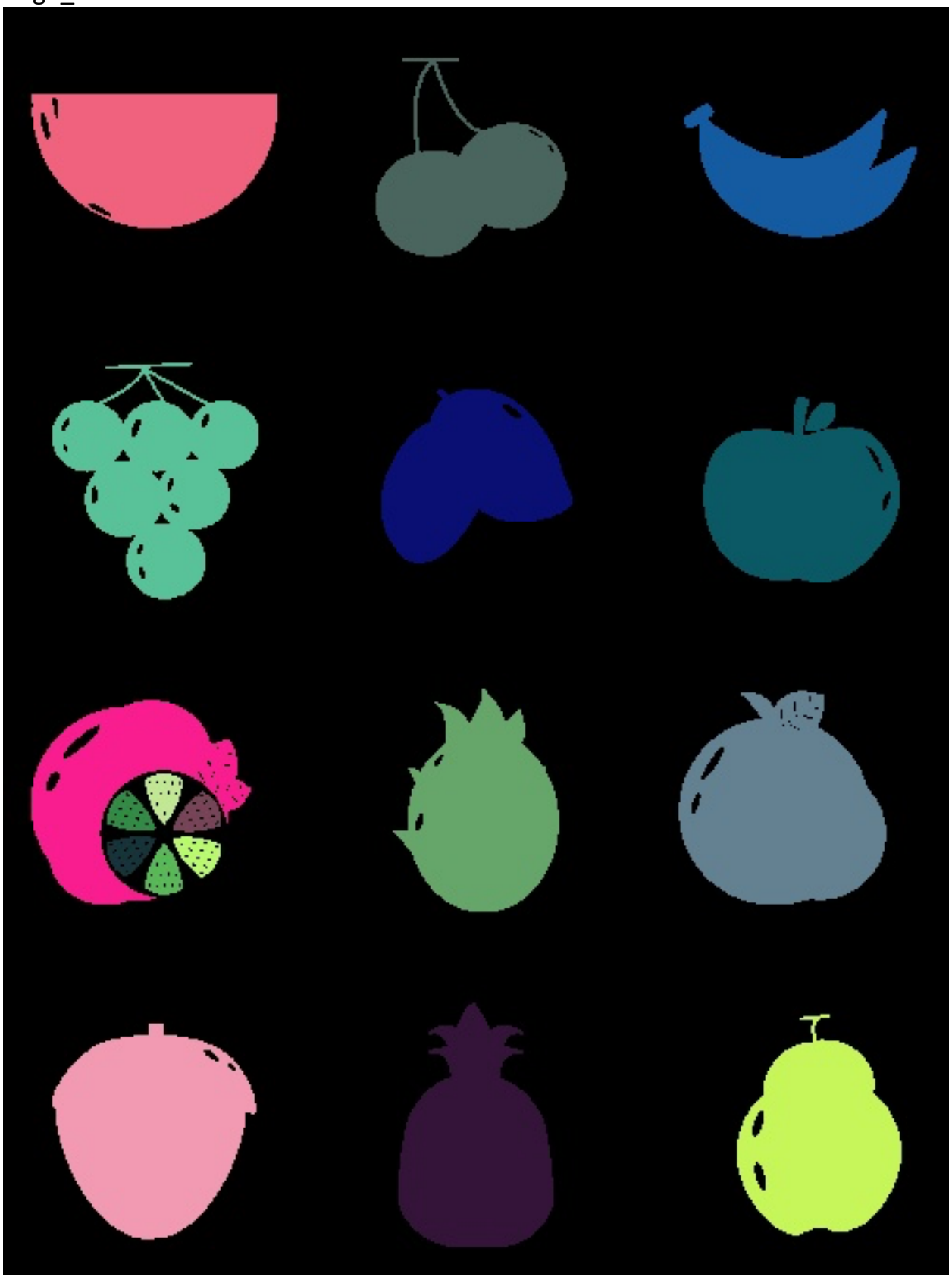
Img2_4

Img2_8

Img3_4



Img3_8



Img4_4

Img4_8