CP467: Assignment 3 Report

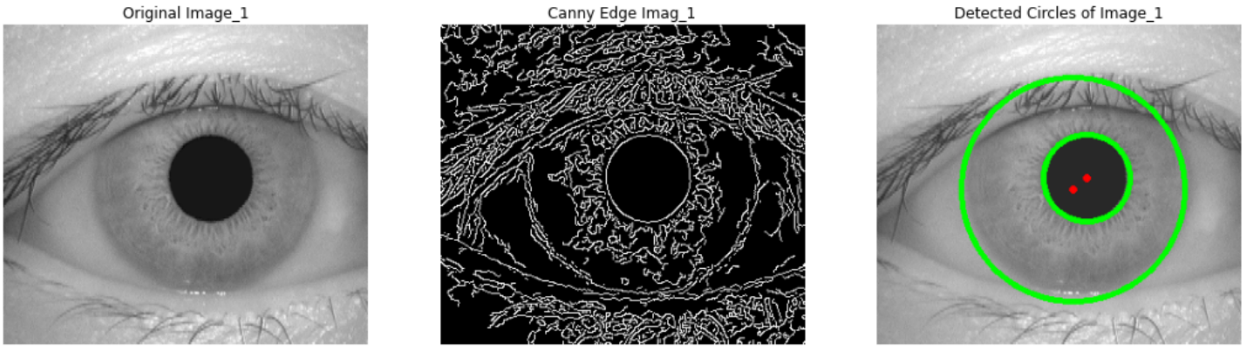
Duc Minh Nguyen - 203009140

I. Approach and algorithms uses

- The process of identifying the circular boundary between the pupil and the iris, as well as between the iris and the sclera, involves the following steps:
    - Filtering to reduce image noise.
    - Conducting edge detection and applying various threshold values to determine which yields the most accurate edge results.
    - Utilizing the Hough Circle detection algorithm on the processed images.
- The rationale behind this procedure is that by first minimizing noise in the images and then optimizing edge detection through thresholding, the Hough Circle algorithm's performance is enhanced.
- Helper function: I also created a helper function to help me tune the parameters by binding each parameter to the key, it helps me reduce time. The code is attached along with the assignment

II. Result Images

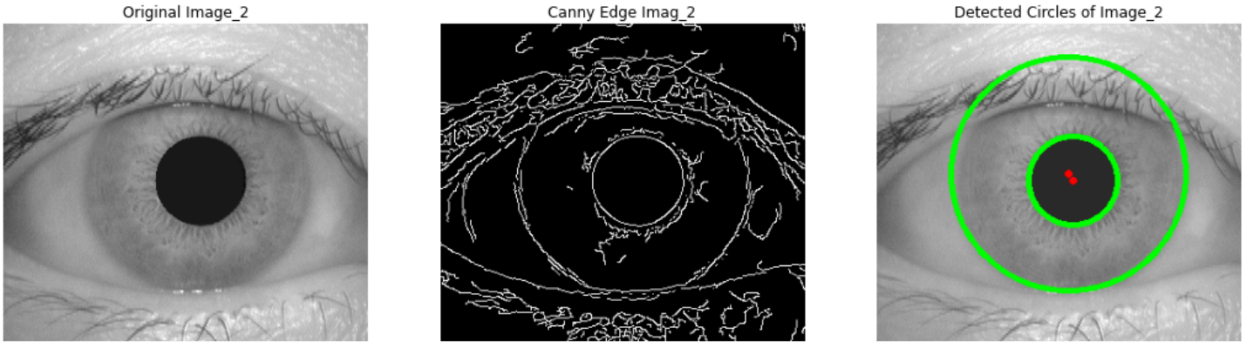
A. Image 1



- **Source image**: this grayscale of an eye displays typically high resolution eye image, it has clear contrast between the pupil, iris and sclera. However, the texture of the eyelids and eyelashes might introduce “noise”
- **Canny edge image**: Before edge detection, I applied medianBlur with value of 5 to help smoothing out noise and then I applied the edge detection with threshold value of (10,30), now we can see the patterns around the iris. Strong contrast between pupil, iris and sclera make edge detection quite effective despite the noise

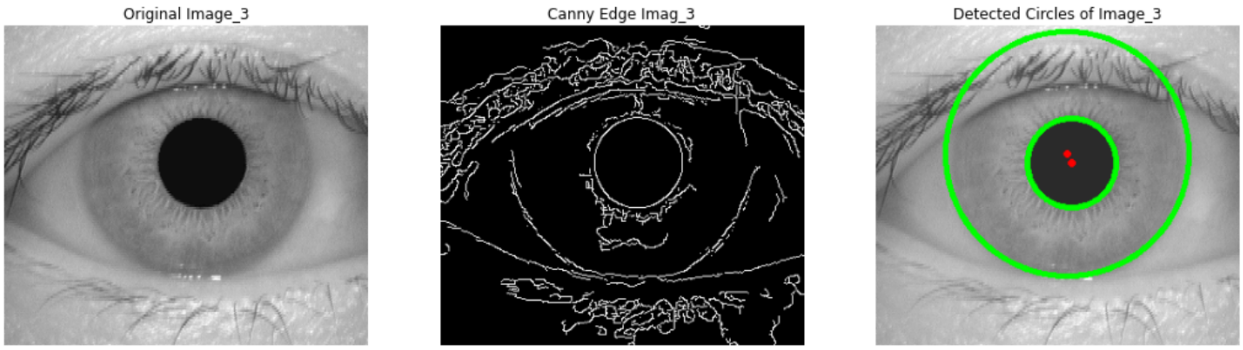
- **Challenges face**: Not much
- **Detected circle of image**: Using Hough circle transform, the system has identified and highlighted the boundaries of the pupils and iris with green circles. Additionally, the red dots denote the centers of detected circles, I apply the value as follow (**dist= 14, param2 = 71** and pram1, minRadius, maxRadius don't affect much)

B. Image 2



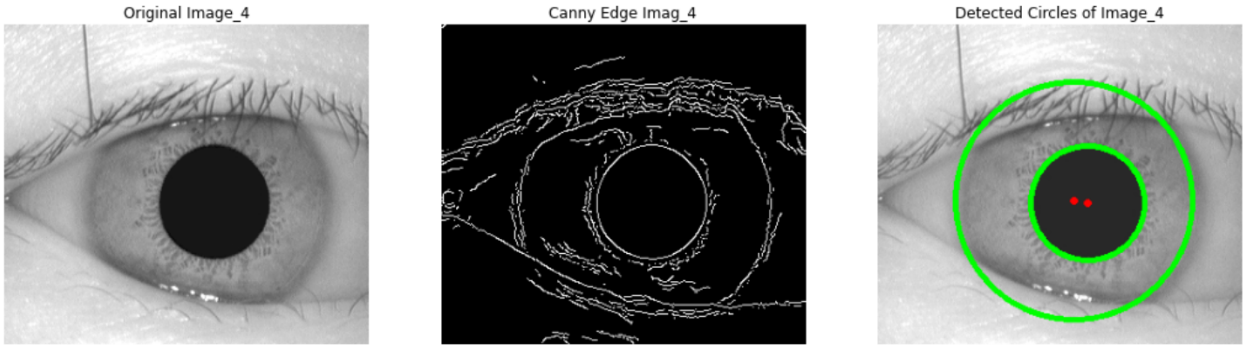
- **Source image:** this grayscale of an eye displays typically high resolution eye image, it has clear contrast between the pupil, iris and sclera, it's almost the same as image 1. However, the texture of the eyelids and eyelashes might introduce "noise"
- **Canny edge image**: Before edge detection, I applied medianBlur with value of 9 to compare between image 1 and 2, which now significantly reduce in noise to help smoothing out noise and then I applied the edge detection with threshold value of (10,30), now we can see the patterns around the iris. Strong contrast between pupil, iris and sclera make edge detection quite effective with less noise
- **Challenges face**: images has more noise, therefore I smooth image more
- **Detected circle of image**: Using Hough circle transform, the system has identified and highlighted the boundaries of the pupils and iris with green circles. Additionally, the red dots denote the centers of detected circles, I apply the value as follow (**dist= 8, param2 = 85** and pram1, minRadius, maxRadius don't affect much)

C. Image 3



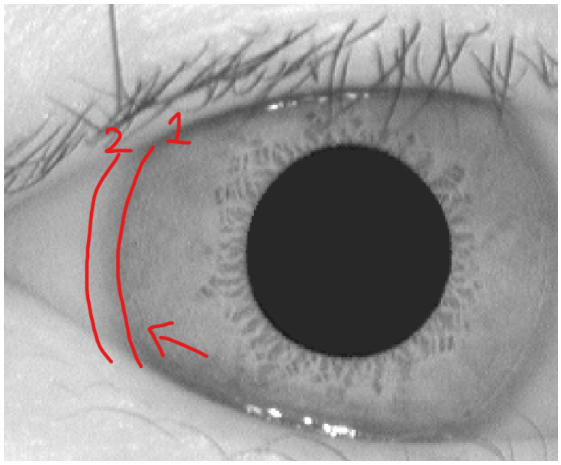
- **Source image**: this grayscale image shows that it might have lower resolution compared to image 1 and 2
- **Challenges face**: edge map don't show clear circle, its not connected continuously
- **Canny edge image**: Before edge detection, I applied medianBlur with value of 9 to help smoothing out noise and then I applied the edge detection with threshold value of (10,35), now we can see the patterns around the iris. However, the circles in the eyes is now not too obvious to figure out, we can see that iris edge is not connected like images 1 and 2
- **Detected circle of image**: Using Hough circle transform, the system has identified and highlighted the boundaries of the pupils and iris with green circles. Additionally, the red dots denote the centers of detected circles, I apply the value as follow (**dist= 8, param2 = 75** and pram1, minRadius, maxRadius don't affect much)

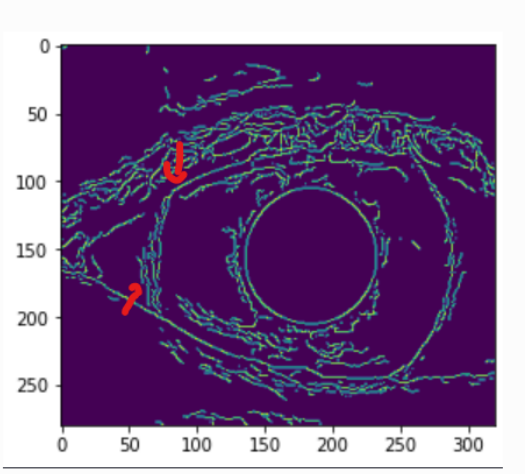
D. Image 4



- **Source image**: this grayscale image shows the iris is missing the top and bottom part, it would result in harder detection, moreover the sclera and iris doesn't stand out from each other

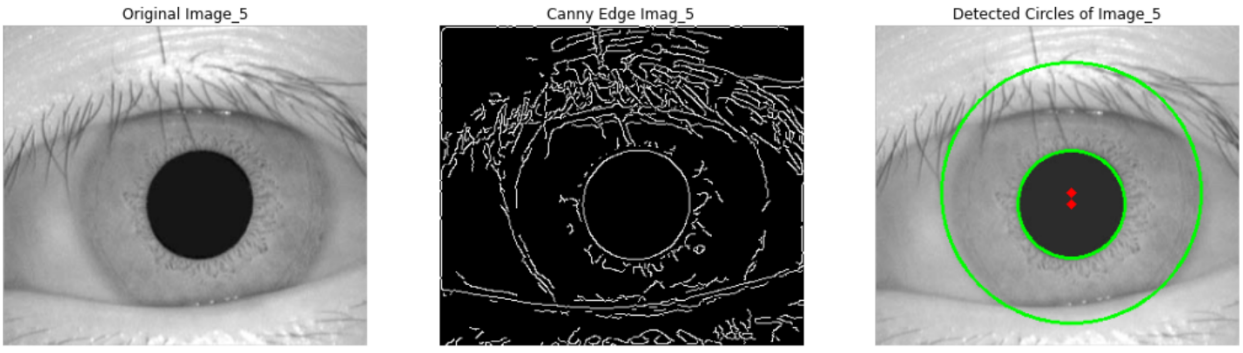
- **Canny edge image**: Before edge detection, I applied medianBlur with value of 19 to help smoothing out noise and then I applied the edge detection with threshold value of (15,20), now we can see the patterns around the iris.
- **Challenges face**: while apply canny on this image, I keep getting the false circles, as the image bellowed shown that canny edge with low blur window will detect false circle therefore, I preprocess the image so it blur out the outer part of the iris to get the true circle





- Detected circle of image: Using Hough circle transform, the system has identified and highlighted the boundaries of the pupils and iris with green circles. Additionally, the red dots denote the centers of detected circles, I apply the value as follow (**dist= 8, param2 = 85** and pram1, minRadius, maxRadius don't affect much). The output now can detect the pupils and iris perfectly

E. Image 5



- **Source image**: this grayscale image shows that it might have lower resolution, the iris does not stand out obviously to the scalar, therefore if we apply too much smoothing it might not be able to show the iris

- **Canny edge image**: Before edge detection, I applied medianBlur with value of 7, less than the other images to not smoothing out too many details and then I applied the edge detection with threshold value of (20,37), now we can see the patterns around the iris.
- **Challenges face**: at first, I apply too much blur to the image, therefore edge detection could not plot out the edge between pupils and iris, however after modify it, it works well
- **Detected circle of image**: Using Hough circle transform, the system has identified and highlighted the boundaries of the pupils and iris with green circles. Additionally, the red dots denote the centers of detected circles, I apply the value as follow (**dist= 9, param2 = 92** and pram1, minRadius, maxRadius don't affect much)

III. Potential improvements

- For this assignment, I create tools to tune each parameter to find out which would fit best to generate the best outcome, however, this is time consuming and not effective in industrial fields.
- Due to the variability in results when using the Hough Circle algorithm on different images, it's essential to optimize its parameters. Integrating machine learning and deep learning in the future. Especially using diverse eye images seems to be a promising improvement. This approach could enhance accuracy and efficiency, allowing the algorithm to deliver superior outcomes in less time

IV. References

- *OpenCV: Hough Circle Transform*. (2023). Opencv.org.

  https://docs.opencv.org/4.x/da/d53/tutorial_py_houghcircles.html