과제3 허프변환 Report

학번 : 2019202032

이름 : 이상현

1. 과제 개요

입력으로 주어진 영상에서 다수 개의 원을 검출한다. 이를 위해 캐니 에지 검출 알고리즘을 구현하여 에지 영상을 추출하고, 추출한 에지 영상에 대하여 허프 변환을 이용해 원을 검출한다. 원을 검출하는 과정에서는 두 가지의 방법을 사용한다. (a, b, r) 투표방법과 그래디언트 기반 방법을 사용하여 원을 검출하고, 검출한 원의 중심과 반지름을 구해 주어진 정답과 비교한다. 이후 원을 검출하는데 사용한 두 가지 방법에 대해 연산량, 메모리 사용량, 수행 시간 등의 관점에서 비교한다.

2. 과제 수행 방법

[캐니 에지 검출 알고리즘]

- 가우시안 필터

: 5x5 크기의 가우시안 커널을 만들고, 이를 원본 픽셀과 연산하여 노이즈를 제거한다.

- Sobel 필터

: Sobel 필터를 사용하여 영상의 가로축과 세로축 미분 결과를 구하고, 이를 통해 기울기의 크기와 방향을 계산한다.

- normalize angle

: atan 함수를 사용하여 gradient_x와 gradient_y 사이의 각도를 구하고, 해당 각도를 0, 45, 135, 90의 값으로 normalize한다.

- Non-maximum suppression

: 위에서 nomalize한 angle값이 0이라면 해당 픽셀의 값을 좌우 픽셀의 gradient 크기 값과 비교한다. 만약 해당 픽셀의 값이 가장 크다면 그대로 두고, 그렇지 않다면 해당 픽셀의 gradient 크기 값을 0으로 바꾼다.

위와 같은 논리로 만약 angle이 90이라면 위아래 픽셀과 비교하고, 45라면 135도 각도에 있는 픽셀들과 비교한다.

- Two-level- threshold

: 임계값을 사용하여 약한 경계선과 강한 경계선으로 구분한다. 과제에서는 low_threshold의 값을 4500, high_threshold의 값을 5000으로 설정하였다.

- Edge tracing

:위에서 약한 경계선과 강한 경계선으로 나눈 것을 토대로 edge tracing을 진행하였다. 과제에서는 bfs 알고리즘을 통해 강한 경계선으로 구분된 픽셀과 연결된 약한 경계선 픽셀들을 방문처리하였다. 이후 방문처리가 되지 않은 픽셀들, 즉 강한 경계선과 연결되지 못한 약한 경계선들의 값은 0으로 down 시켰다.

[허프 변환]

- gradient 기반 원 검출

먼저 캐니 에지 검출 알고리즘을 적용한 영상에서 gradient 크기가 0이 아닌 픽셀이 있다면 해당 픽셀의 좌표를 시작점으로 설정한다. 이후 해당 픽셀에 gradient_x, gradient_t 값을 더해서 나온 좌표를 끝점으로 설정한다. 이후 시작점과 끝점을 지나는 직선의 방정식을 세운다. for문을 통해 직선의 방정식의 x 값에 0부터 399까지 대입하며 y 좌표를 구해낸다. 만약 y 좌표가 0보다 작거나 396 이상의 값이라면 window size를 over하므로 해당 값은 버린다. 이렇게 구해낸 직선상의 모든 x, y 좌표에 해당하는 int 형 2차원 배열의 위치에 +1 값을 해준다. 이러한 방식을 통해 캐니 에지 검출 알고리즘을 적용한 영상 속 원 위의 한점에서 원의 중심을 지나는 모든 직선을 얻어낼 수 있고, 해당 직선들이 많이 지나는 픽셀, 즉 int형 2차원 배열의 각각의 임계구역에서 크기가 가장 큰 좌표 4개를 검출한다. 이후 해당 점을 중심으로 가능한 모든 반지름을 가지는 원을 그리고, 그려진 원들 중 캐니 에지 검출 알고리즘을 적용한 영상속의 원과 가장 많이 겹치는 원을 정답으로 추출한다.

- (a, b, r) 투표 방법

3. 결과 분석

[캐니 에지 검출 알고리즘]

- 원본 코인 이미지

- 가우시안 필터를 적용한 이미지

- sobel 필터를 적용한 gradient_x와 gradient_y


gradient_x

- gradient 절대값을 구한 영상

- non maximum suppression을 적용한 영상

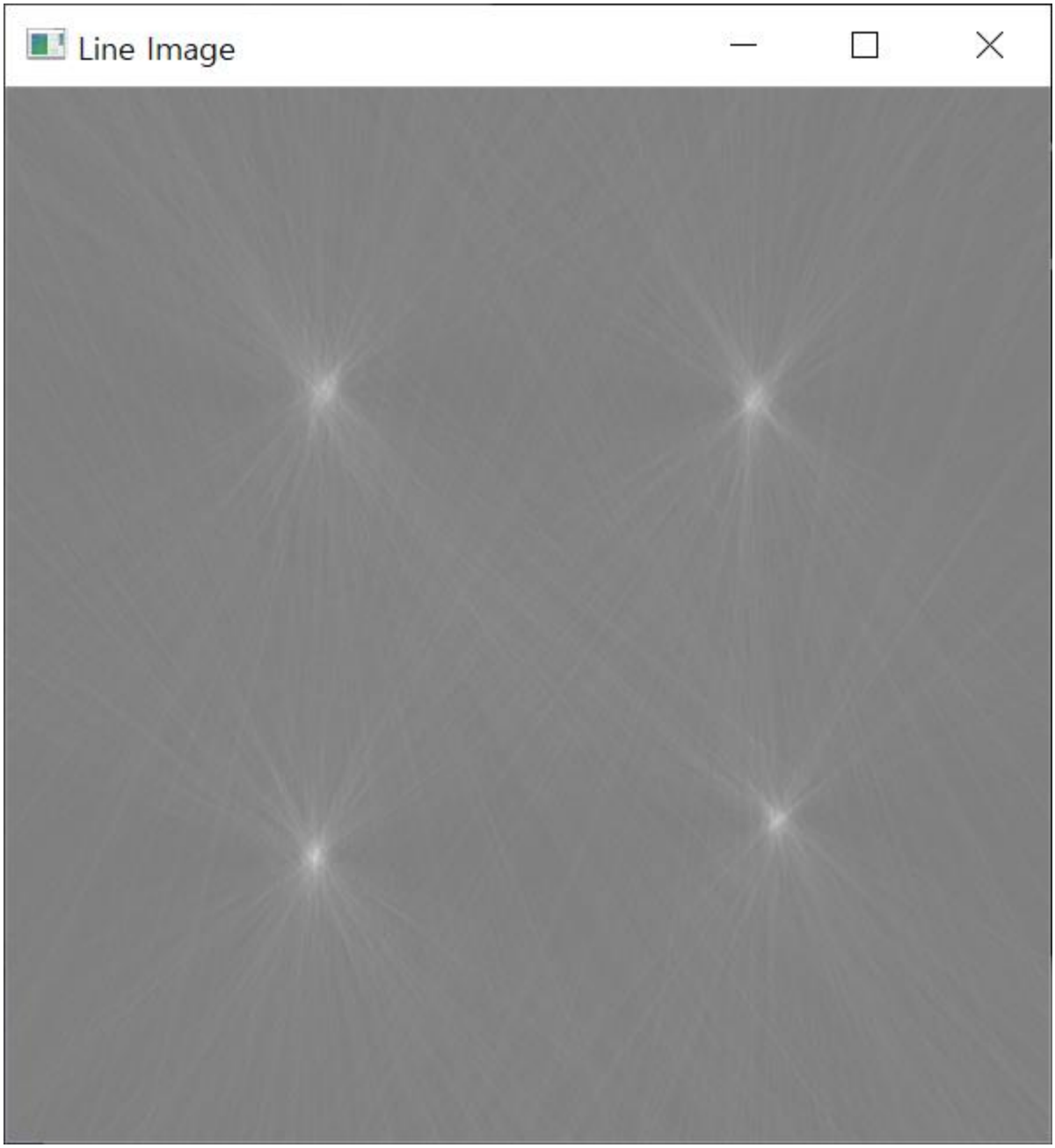
위의 영상과 비교하여 원의 테두리를 제외한 부분은 육안으로 봤을 때 흐릿해진 것을 확인할 수 있다.

- edge tracing을 적용한 영상

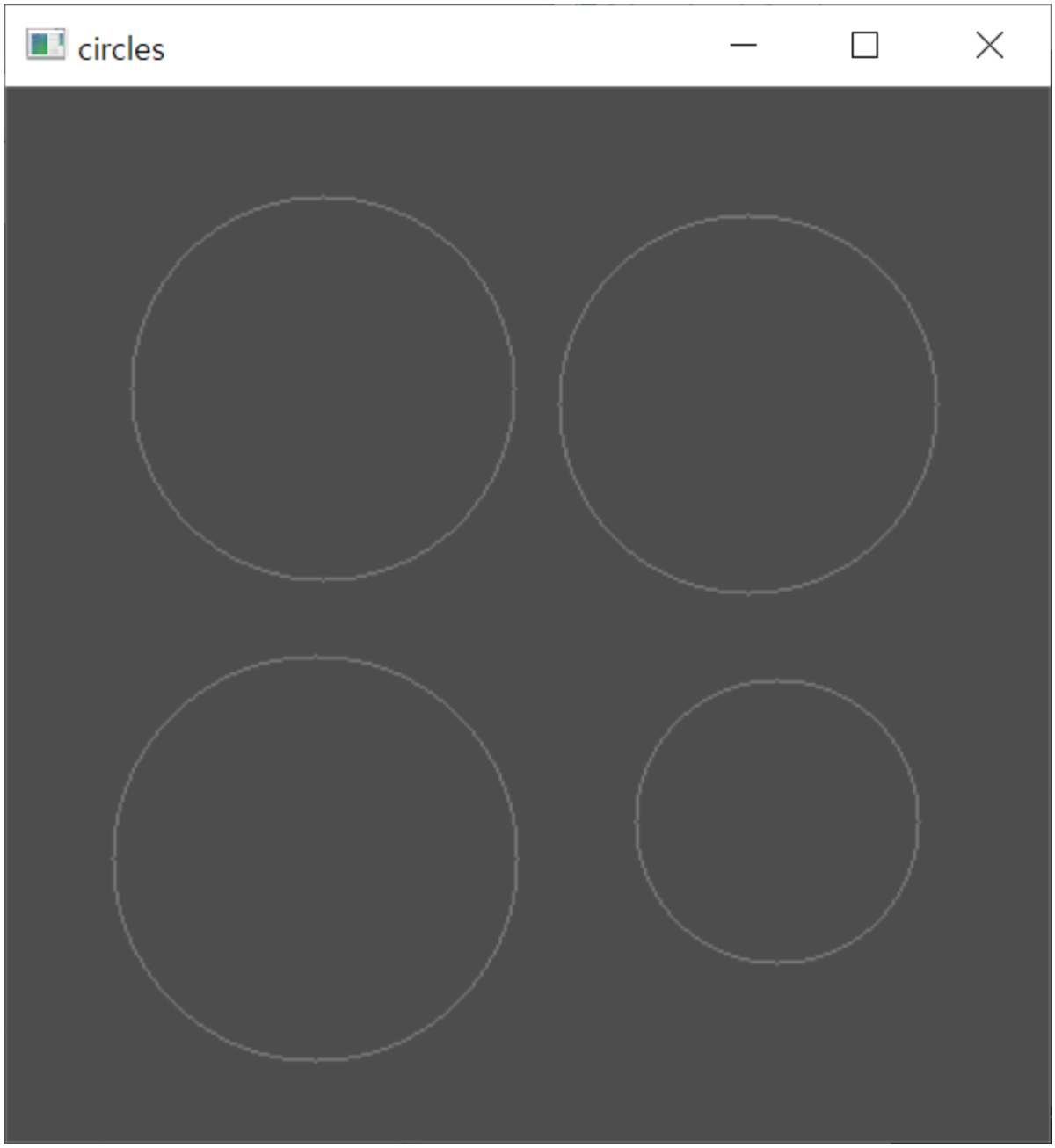
다음과 같이 부분적으로 제거되지 않은 noise 들이 있지만 해당 noise는 gradient의 값 자체가 커서 제거되지 않은 것을 생각되었다.

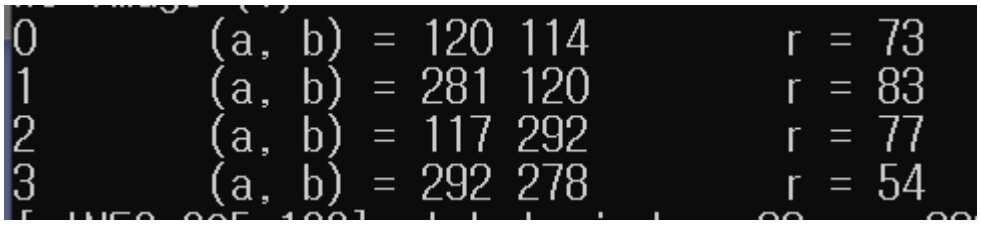
[허프변환]

- gradient 기반의 원 검출



다음과 같이 gradient를 기반으로 원의 중심을 지나는 직선을 그렸고, 직선이 가장 많이 지나가는 점이 가장 밝은 값을 가지는 것을 확인할 수 있다.



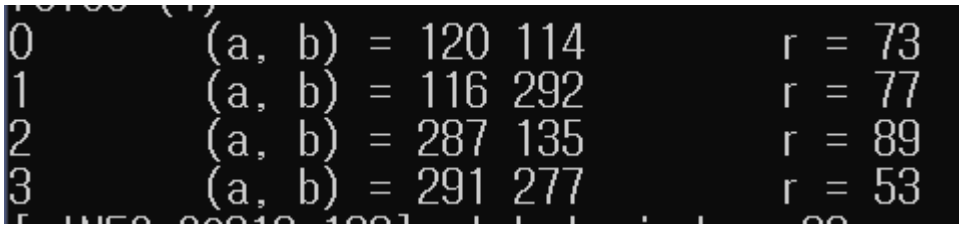


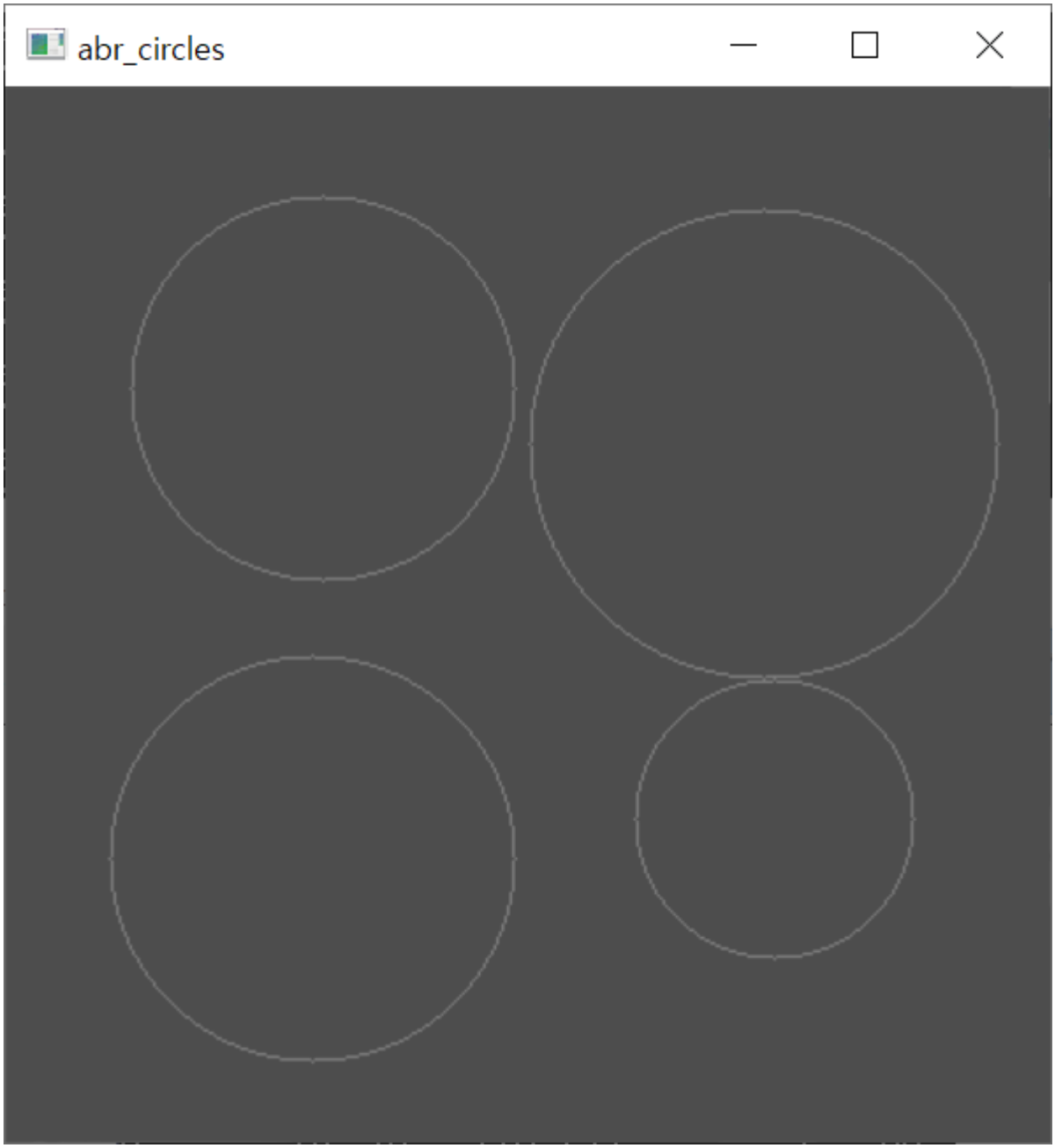
다음과 같이 원을 검출하였다.

Ground truth와 비교하였을 때, 원의 중심점 좌표가 같으면 원의 반지름의 기울기도 정답과 유사하게 나타나는 것을 확인할 수 있다.

		1번 원	2번 원	3번 원	4번 원
Ground truth	원의 중심	(116, 114)	(284, 116)	(118, 290)	(292, 278)
	반지름	70	76	75	56
Hough transform results	원의 중심	(120, 114)	(281, 120)	(117, 292)	(292, 278)
	반지름	73	83	77	54

- (a, b, r) 투표 방식





		1번 원	2번 원	3번 원	4번 원
Ground truth	원의 중심	(116, 114)	(284, 116)	(118, 290)	(292, 278)
	반지름	70	76	75	56
Hough transform results	원의 중심	(120, 114)	(287, 135)	(116, 292)	(212, 277)
	반지름	73	83	77	54

전체적으로 잘 추출이 되지만 두번째 원의 결과가 잘 나오지 않는 것을 확인할 수 있었다. 여러 가지를 실험하며 이유에 대해 알아보려 하였지만 이유를 알지 못했다.

4. 고찰

[issue1]

해당 과제를 진행하는 과정에서 어려웠던 점은 허프변환의 마지막 단계인 정답에 가까운 원을 찾아가는 과정이었다. 먼저 앞서 구한 중심점 후보를 중심으로 모든 반지름에 해당하는 원을 그리고, 원의 테두리의 좌표롸 앞서 구한 magnitude 영상에서 값이 0이 아닌 좌표의 값을 비교해, 좌표가 일치하는 수가 많은 것을 찾을 수 있도록 설계하였다. 그러나 이 과정에서 문제가 발생하였는데, 아무래도 앞서 구한 원의 중심점이 정답과 완벽하게 일치하지 않기 때문에 원이 magnitude 영상 속 원의 모양과 정확하게 들어맞지 않는 것이었고, 때문에 정답으로 주어진 원보다 더 큰 원이 그려질 경우 다른 원의 좌표와 일치하면서 해당 원이 정답에 가까운 원으로 채택된다는 것이었다. 이를 해결하기 위해서 임의의 임계영역을 설정하고, 해당 임계영역 안에서만 원이 그려질 수 있도록 설계하였고, 이를 통해 해당 문제를 해결할 수 있었다.

[issue2]

Gradient에 기반하여 원을 검출하는 과정에서 직선을 긋기 위해서 시작점과 끝점을 구하고, 두 점을 지나는 직선의 방정식을 설계하여 직선을 구했다. 그런데 이 과정에서 얻어낸 직선이 기울기가 역수에 해당하는 직선으로 만들어지는 문제가 발생하였다. 이 문제에 대해서 배열과 2차원 좌표계가 1대1로 대응하지 않아서 발생하는 문제라고 생각하였고, 이러한 생각을 바탕으로 문제를 해결할 수 있었다.

[(a, b, r) 투표 방식 vs 그래디언트 기반 방법]

(a, b, r) 투표 방식의 경우 모든 케이스에 대해 조사하는 완전 탐색 알고리즘이므로 그래디언트 기반 방법에 비해 연산량이 많고 수행시간이 길다. 이번 과제에서처럼 400x396 크기의 이미지에 대해 두 알고리즘을 실행한다면, (a, b, r)투표 방법의 경우 400x396(전체순회) x 반지름 x 400x396(원 확인)의 연산량을 가지는 반면, 그래디언트 기반의 알고리즘의 경우 반지름x 400x396(원 확인)의 연산량만을 필요로 하므로 (a, b, r)에 비해 훨씬 적은 연산량을 가질 수 있다. 그러나 정확도의 경우 그래디언트 기반의 방법은 원의 중심점이 잘못 구해질 경우 오차가 발생할 수 있기 때문에 정확도가 떨어지는 경우가 발생할 수 있다.