컴퓨터비전 프로젝트

최종 발표

컴퓨터 비전을 이용한 3쿠션 가이드

2022. 6. 08

컴퓨터공학부 20172864 서정현
컴퓨터공학부 20176342 송민준

1. 문제 제기 및 필요성

당구 3쿠션은 어렵다?

[출처 - [당구 3쿠션] 초구 잘치는 방법] https://www.youtube.com/watch?v=xTffcw4VIBo&ab_channel=%EB%8B%B9%EA%B5%AC%EA%B0%95%EC%A2%8C4%EA%B5%AC%263%EC%BF%A0%EC%85%98

1. 문제 제기 및 필요성

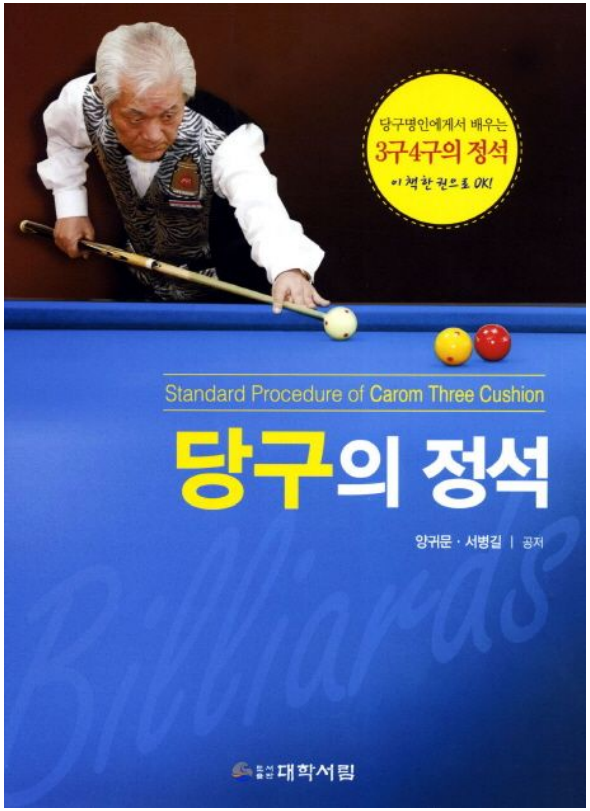
당구에서 3쿠션으로 성공적으로 득점할 수 있는 경로를 아는 방법에는 당구 프로의 경기를 시청하여 비슷한 공의 배치에서 득점하는 방법을 따라하는 것이 있습니다.

[출처 : 3쿠션 4대천왕]프레드릭 쿠드롱

1. 문제 제기 및 필요성

혹은 당구 서적을 통해서 여러 가지 득점할 수 있는 경로에 대해

학습을 할 수 있습니다.



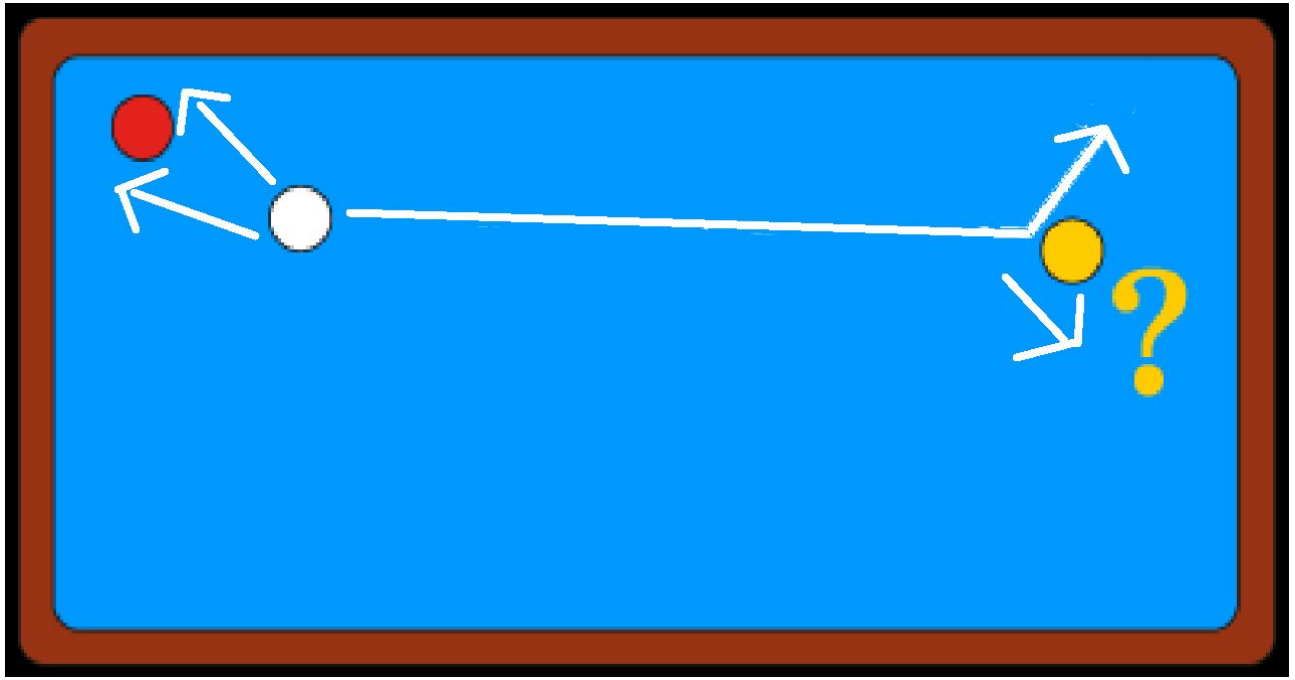
[출처 : 당구의 정석양귀문 | 대학서림 - 교보문고]

1. 문제 제기 및 필요성

근본적인 문제

눈 앞의 공 배치를 경기 영상이나, 당구 서적에서 보지 못했다면,

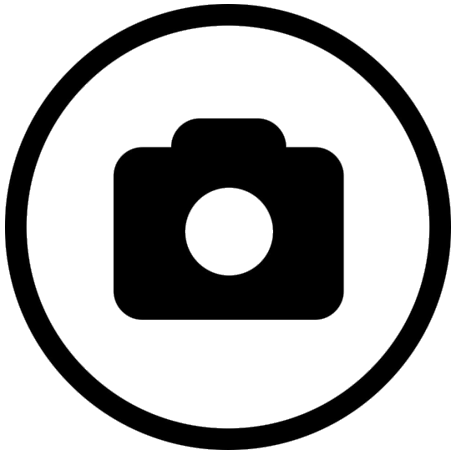
3쿠션 입문자의 입장에서는 득점 경로를 알기 어렵습니다.



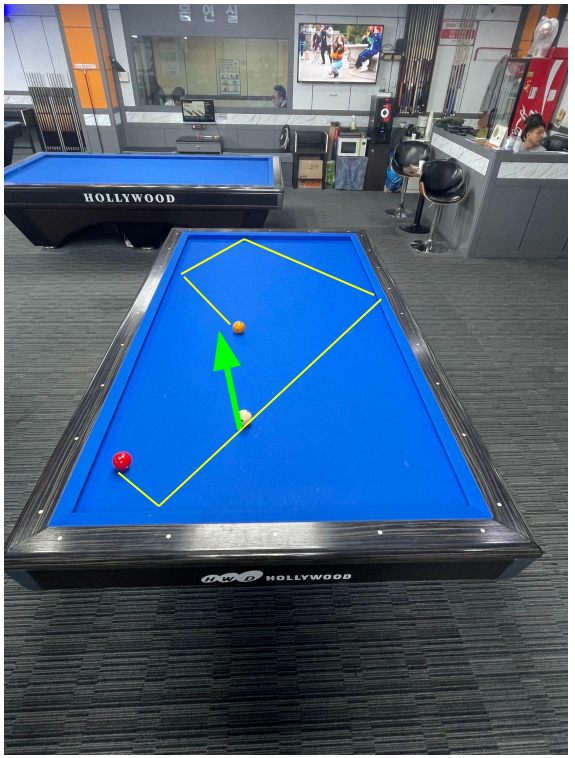
1. 문제 제기 및 필요성

따라서 입문자를 위해, 당구장에서 사진을 찍어서

즉석에서 경로를 알려주는 프로그램이 있다면 도움이 될 것 입니다.



당구대 사진 촬영



최적의 득점 경로 표시

2 - 1. 사전 연구 조사

https://github.com/choonguri/dl-3cushion-hint

기존에 당구대와 공을 input으로 입력하면

3쿠션 배치를 치는 방법(옆돌리기 ,제각돌리기 등..)

과 확률을 결과로 보여주는 program이 있다.

2 - 2. 기존 방법의 문제

1. 수구와 적구의 위치를 사용자가 직접 설정해야 하는데, 이는 불편할 뿐 아니라 정밀한 설정에 어려움이 있다.

2. 기존 프로그램은 공 3개의 배치를 설정하면 단순히 득점 할 수 있는 경로의 이름(앞돌리기, 제각돌리기 등..)만 알려주어 해당 용어를 모르는 사람은 어떻게 쳐야 될 지 모를 수 있다.

2 - 3. 해결 방법 제안

1. 수구와 적구의 위치를 사용자가 직접 설정해야 하는데, 이는 불편할 뿐 아니라 정밀한 설정에 어려움이 있다.

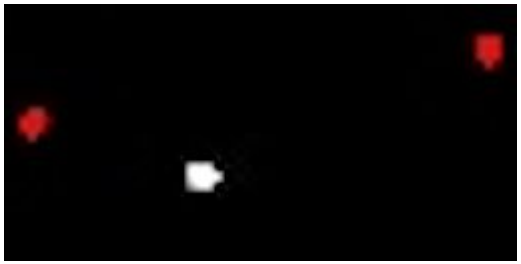
=> 컨투어 분석 또는 HSV 색 공간 추출을 통해 당구대의 영역을 구한뒤 Perspective Transformation을 통해 당구공들의 위치를 계산한다.

2. 기존 프로그램은 공 3개의 배치를 설정하면 단순히 득점 할 수 있는 경로의 이름(앞돌리기, 제각돌리기 등..)만 알려주어 해당 용어를 모르는 사람은 어떻게 쳐야 될 지 모를 수 있다.

=> 수구가 진행될 경로를 같이 학습시켜 시각화해서 보여준다.

2 - 4. 데이터셋

Tensorflow, keras 사용하기 위해 데이터셋 제작

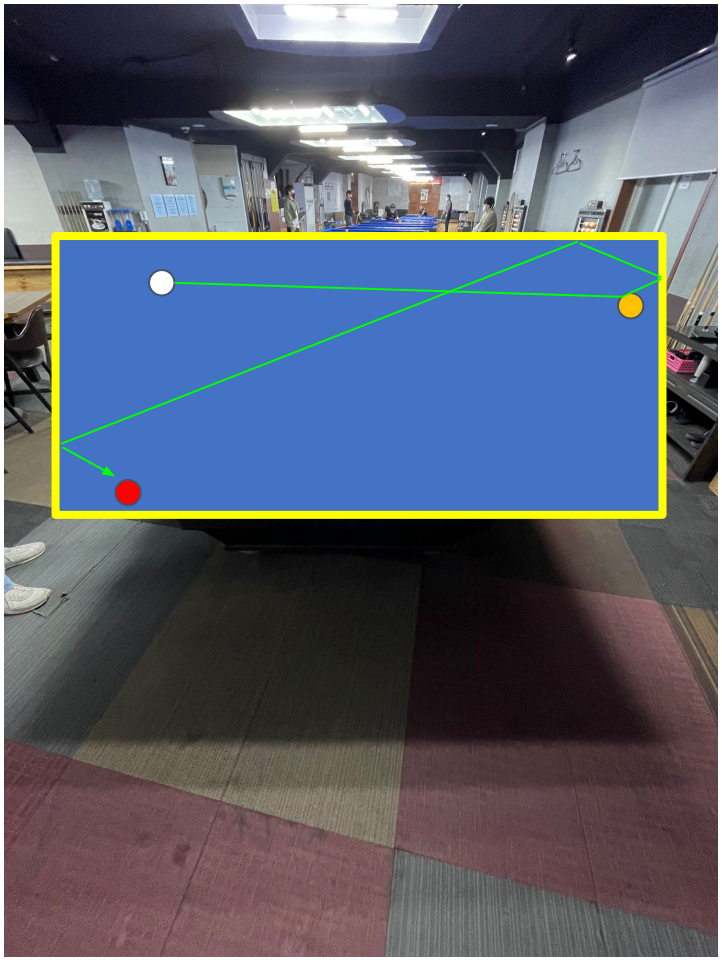


공 3개를 무작위로 찍어주는 프로그램 제작 후 학습 데이터 생성 자동화

↓

각 학습 데이터별로 레이블링 진행(진행 방향, 먼저 쳐야되는 공 등..)

2 - 5. 플로우 차트



당구대와 당구공이 포함된 사진 업로드

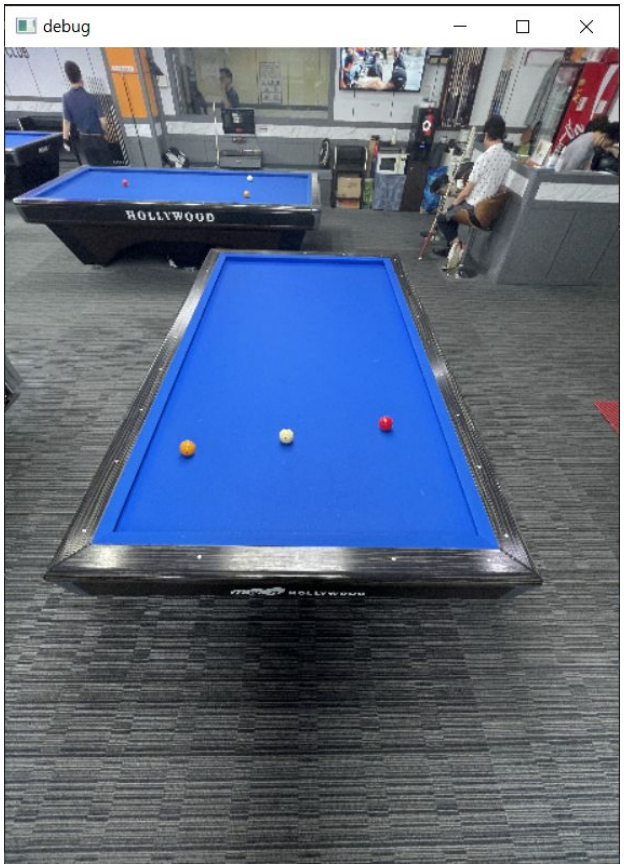
당구대의 영역을 2d 직사각형으로 변환

각 당구공을 찾아서 변환시킨 2d 직사각형에 표시

당구공 3개의 위치를 이용해서 경로를 계산하고 확률 표시, 공의 경로 시각화해서 표시

3. 구현 내용 및 결과 분석

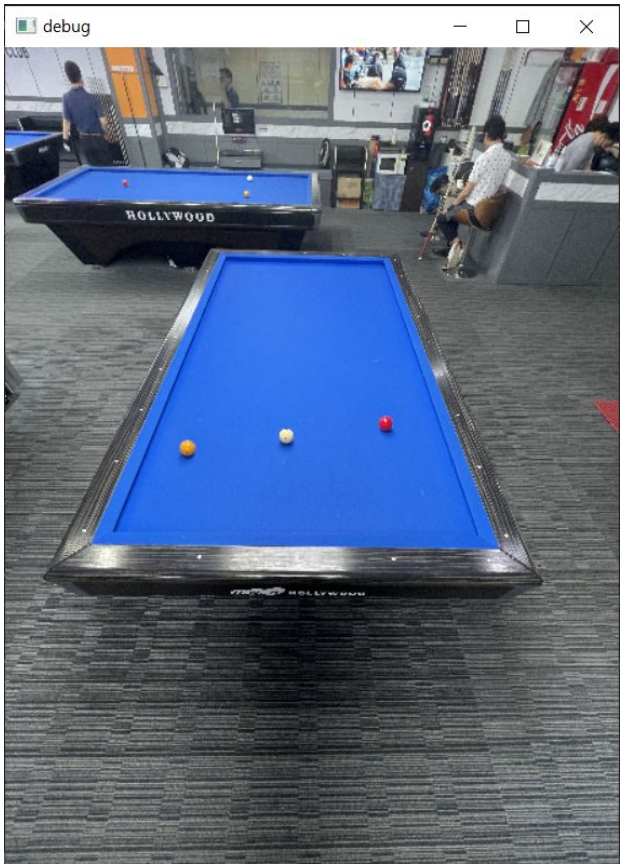
1. 당구대 평면 사진 변환



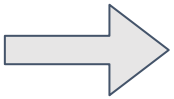
Source Image

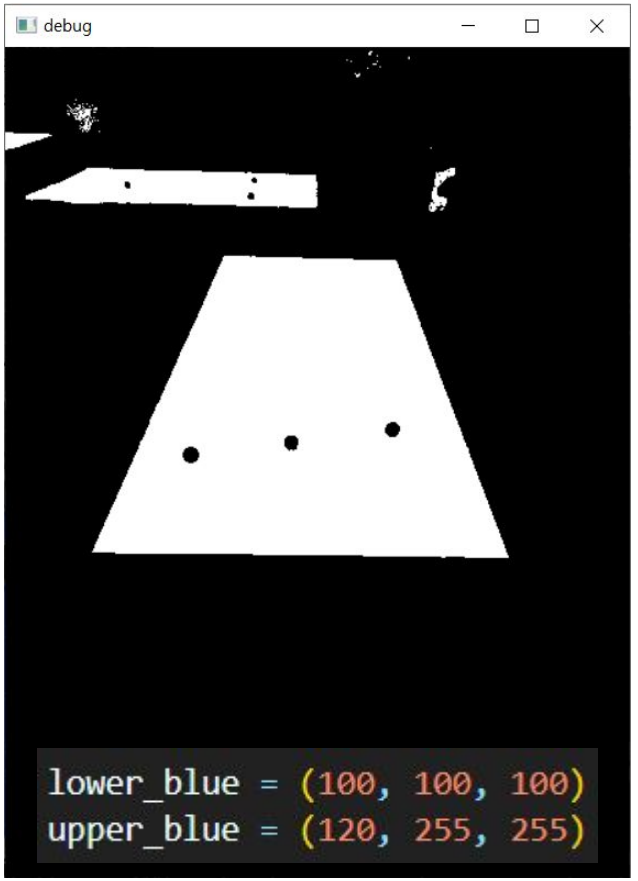
3. 구현 내용 및 결과 분석

1. 당구대 평면 사진 변환 - HSV 색 공간 변환 후 파란색 영역 추출



Source Image

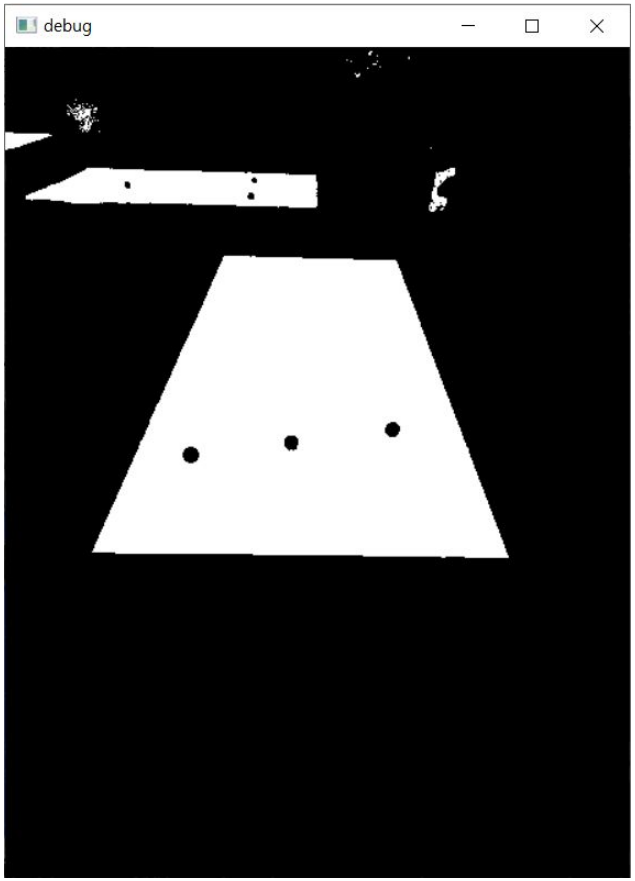




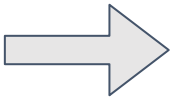
Blue area binary image

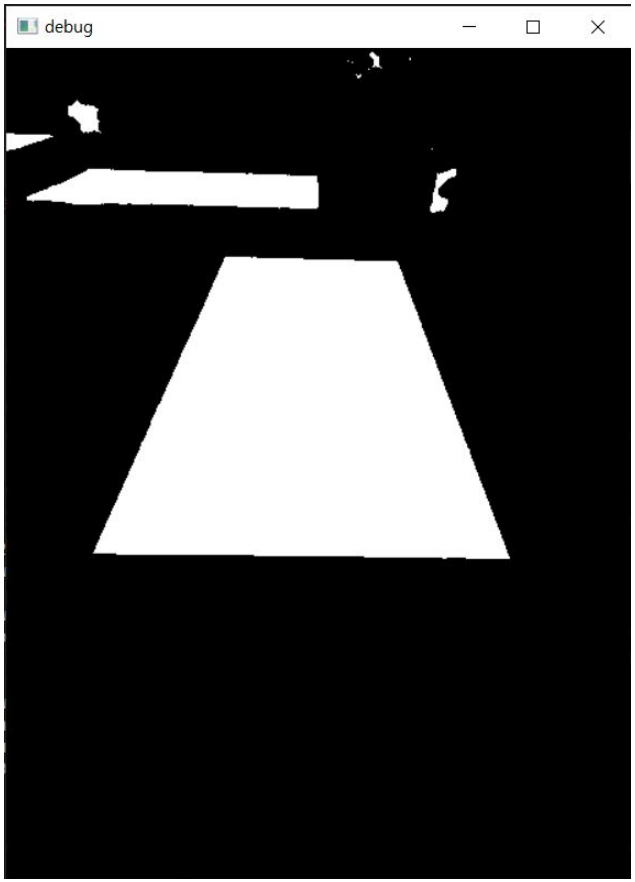
3. 구현 내용 및 결과 분석

1. 당구대 평면 사진 변환 - 모폴로지 닫힘 연산



Blue area binary image

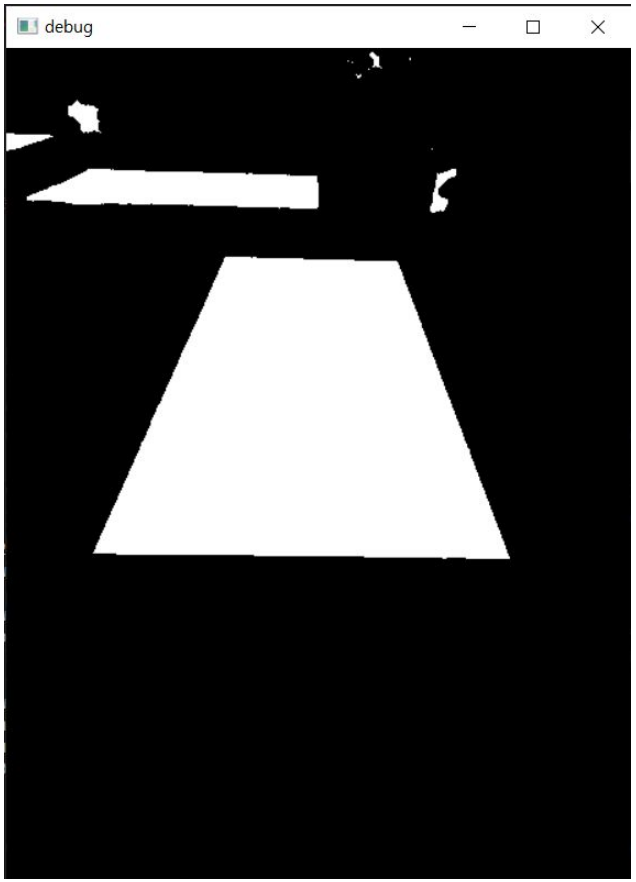




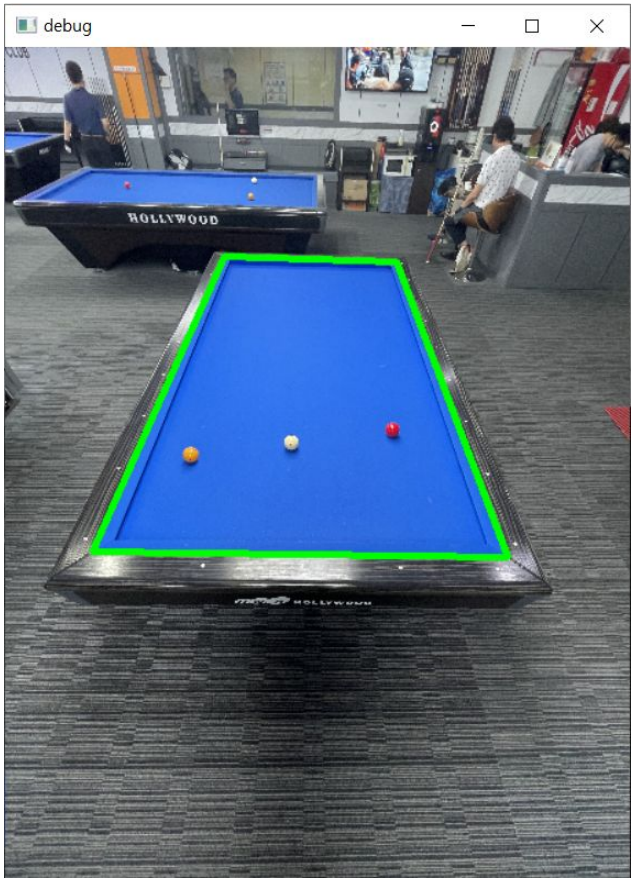
Blue area (morphology - close)

3. 구현 내용 및 결과 분석

1. 당구대 평면 사진 변환 - 컨투어 추출



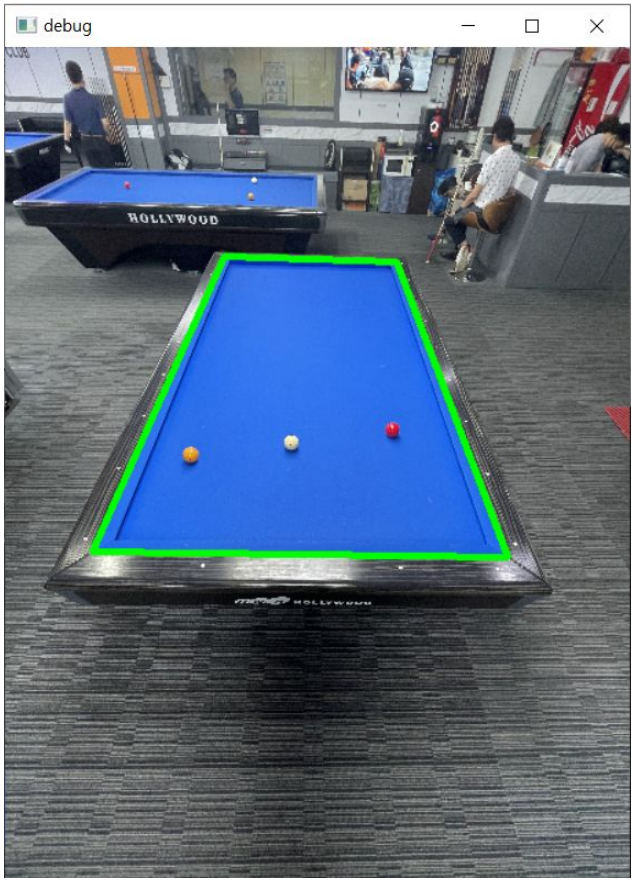
Blue area (morphology - close)



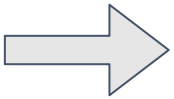
Blue rectangle area contour

3. 구현 내용 및 결과 분석

1. 당구대 평면 사진 변환 - 원근 변환(Perspective Transform)



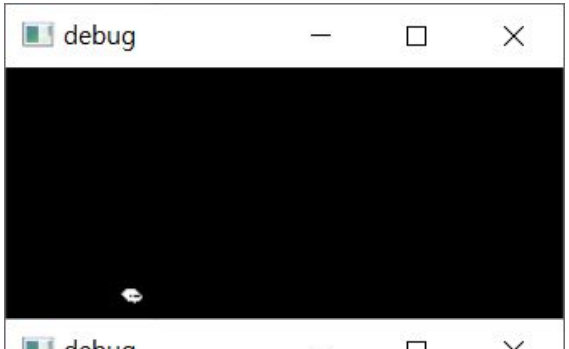
Blue rectangle area contour

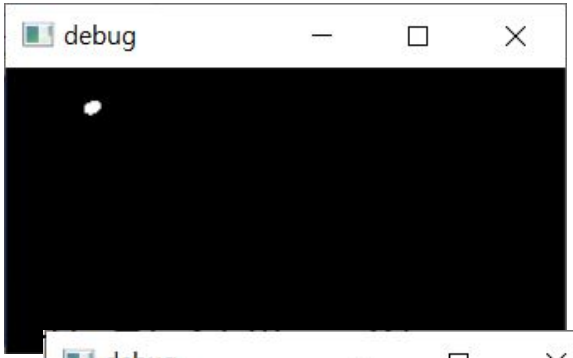


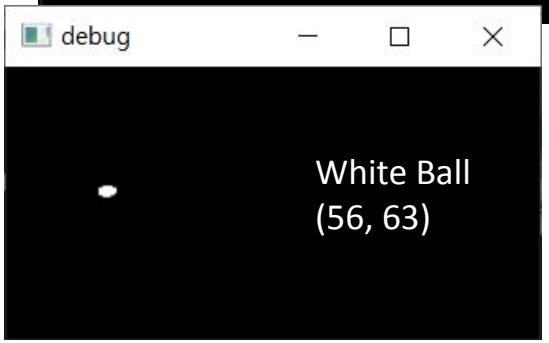
Warped table

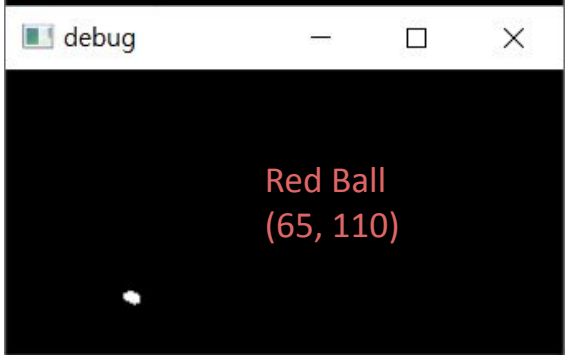
3. 구현 내용 및 결과 분석

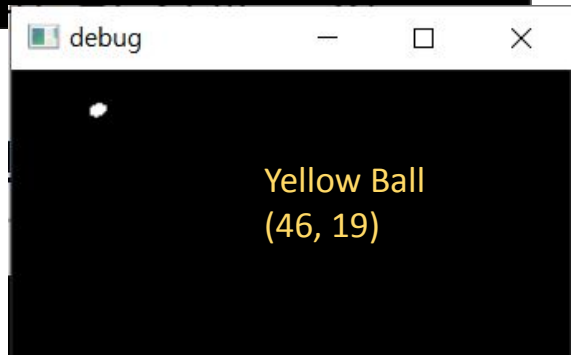
1. 당구대 평면 사진 변환 - 공 위치 추출 (HSV, Morphology, Contour)





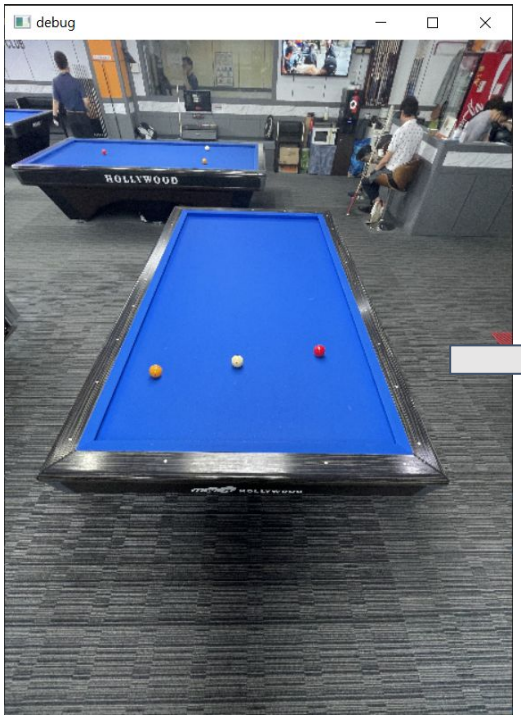






3. 구현 내용 및 결과 분석

1. 당구대 평면 사진 변환 - 결과 테이블 그려보면...



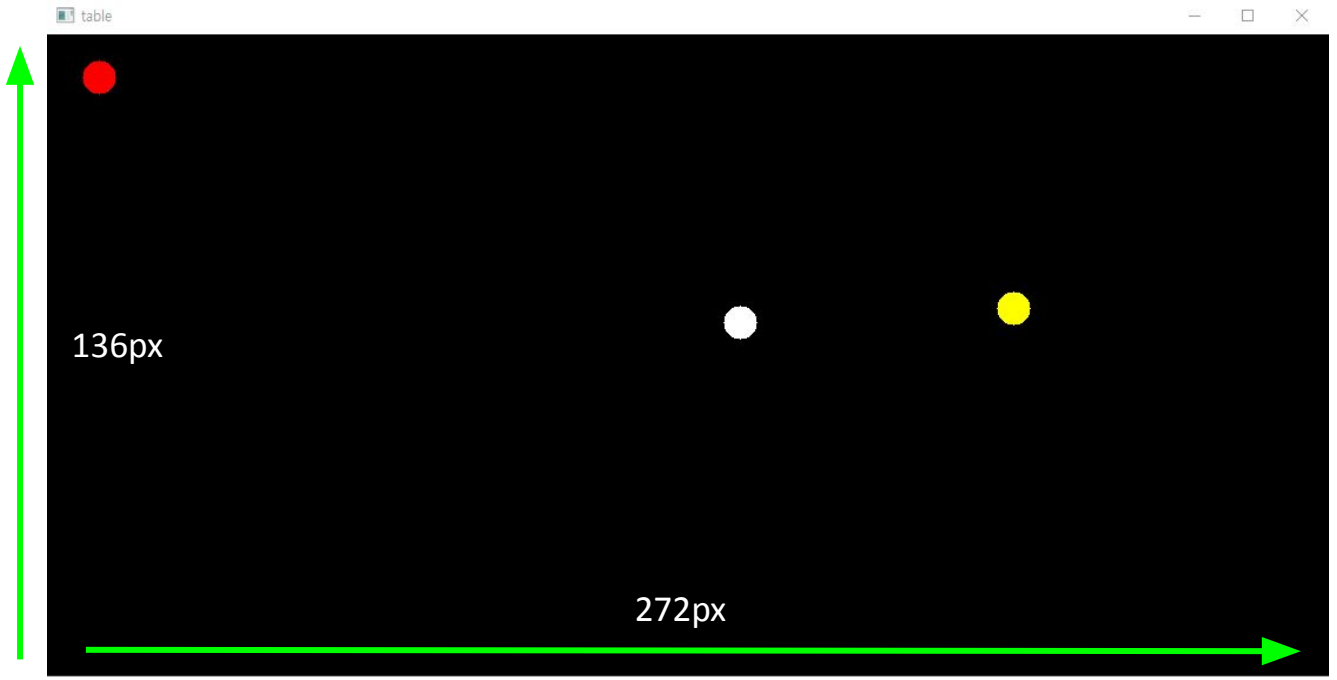
White Ball
(56, 63)

Red Ball
(65, 110)

Yellow Ball
(46, 19)

3. 구현 내용 및 결과 분석

2. 데이터셋 학습



공 3개를 자유롭게 혹은 무작위로 배치하고,
해당 공 3개의 좌표와 수동으로 라벨링(직접 라벨 값 입력)해서 csv에 추가하는 프로그램 제작

3. 구현 내용 및 결과 분석

2. 데이터셋 학습

내 공은 무조건 흰공 기준

라벨링

빨간공 왼쪽 => 0 노란공 왼쪽 => 2
빨간공 오른쪽 => 1 노란공 오른쪽 => 3

빈쿠션 => 4

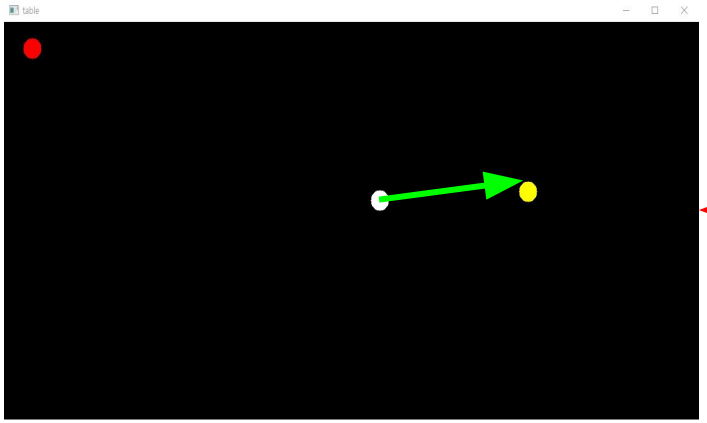
	A	B	C	D	E	F	G
1	x0	y0	x1	y1	x2	y2	label
2	179	131	264	33	91	76	3
3	159	36	120	73	143	115	2
4	104	87	18	95	200	123	1
5	74	29	180	57	223	62	0
6	121	19	158	111	97	100	0
7	142	60	178	4	206	125	1
8	176	109	105	53	188	119	0
9	175	34	102	26	31	115	0
10	164	98	161	117	23	28	1
11	57	27	207	23	187	39	4

3. 구현 내용 및 결과 분석

2. 데이터셋 학습

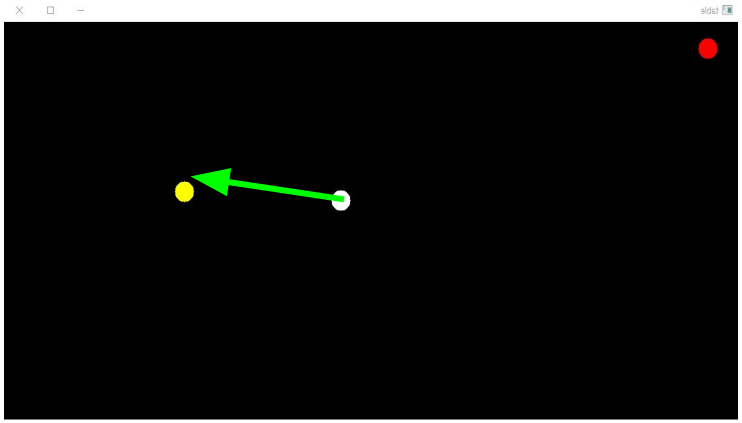
라벨링

약 450개의 무작위 데이터 학습
=> 당구대와 당구공은 상하, 좌우, 상하좌우로 뒤집어서 데이터를 총 4배로 만들 수 있습니다.
=> 약 1800개 학습



노란공 좌측 겨냥

좌우반전

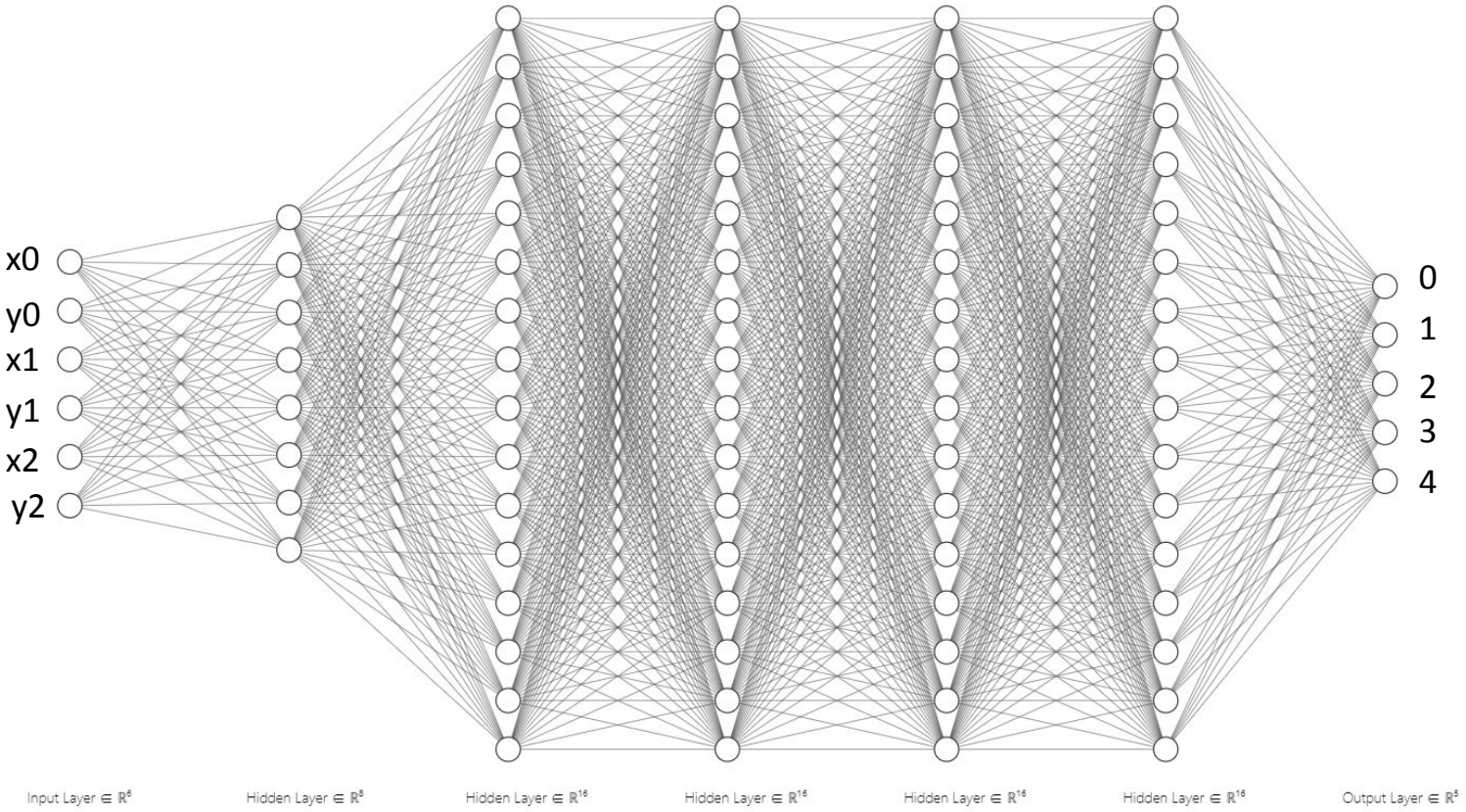


노란공 우측 겨냥

3. 구현 내용 및 결과 분석

2. 데이터셋 학습

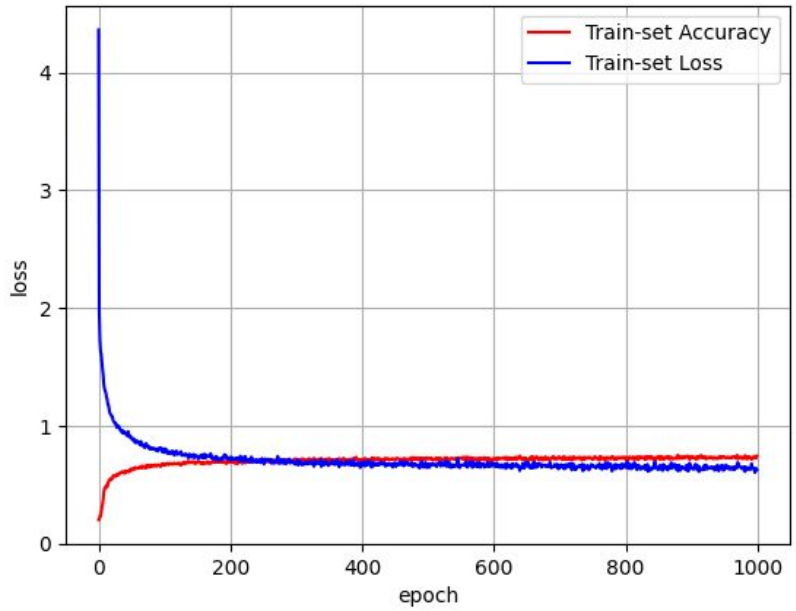
뉴럴 네트워크 구조



3. 구현 내용 및 결과 분석

2. 데이터셋 학습

1800개 데이터 학습, epoch = 1000, batch size = 16
train set 정확도 = 0.74



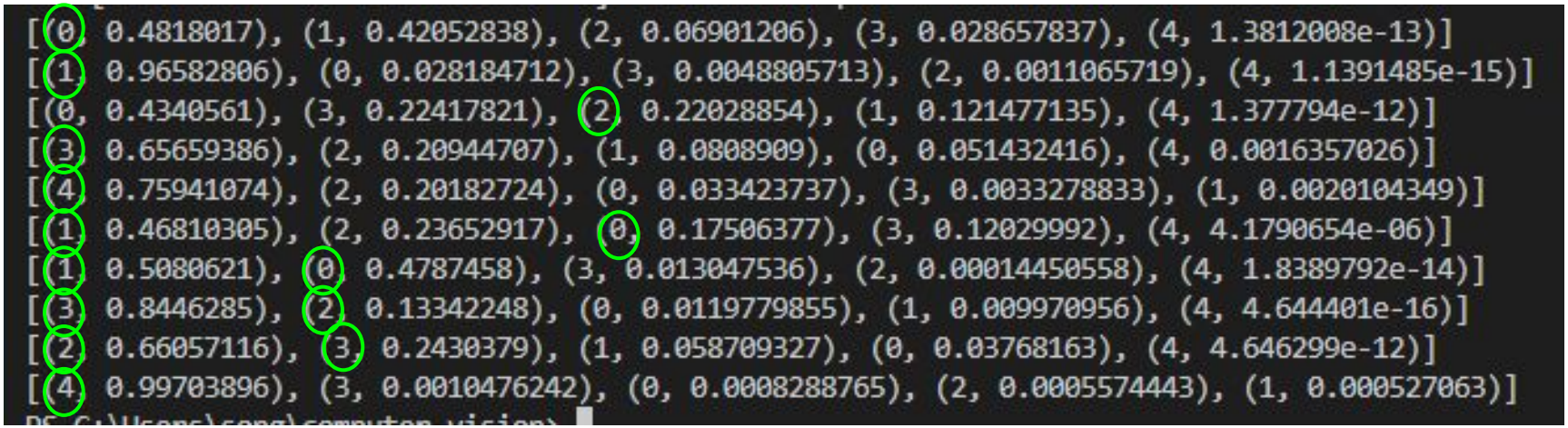

```
116/116 [==============================] - 0s 540us/step - loss: 0.6804 - accuracy: 0.7143
Epoch 993/1000
116/116 [==============================] - 0s 548us/step - loss: 0.6421 - accuracy: 0.7294
Epoch 994/1000
116/116 [==============================] - 0s 540us/step - loss: 0.6414 - accuracy: 0.7332
Epoch 995/1000
116/116 [==============================] - 0s 557us/step - loss: 0.6431 - accuracy: 0.7392
Epoch 996/1000
116/116 [==============================] - 0s 540us/step - loss: 0.6228 - accuracy: 0.7359
Epoch 997/1000
116/116 [==============================] - 0s 540us/step - loss: 0.6040 - accuracy: 0.7419
Epoch 998/1000
116/116 [==============================] - 0s 540us/step - loss: 0.6302 - accuracy: 0.7289
Epoch 999/1000
116/116 [==============================] - 0s 540us/step - loss: 0.6410 - accuracy: 0.7284
Epoch 1000/1000
116/116 [==============================] - 0s 531us/step - loss: 0.6280 - accuracy: 0.7462
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 8)	56
dense_1 (Dense)	(None, 16)	144
dropout (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 16)	272
dropout_1 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 16)	272
dense_4 (Dense)	(None, 16)	272
dense_5 (Dense)	(None, 5)	85

Total params: 1,101
Trainable params: 1,101
Non-trainable params: 0

3. 구현 내용 및 결과 분석

3. 학습 결과



test set 10개로 예측한 결과

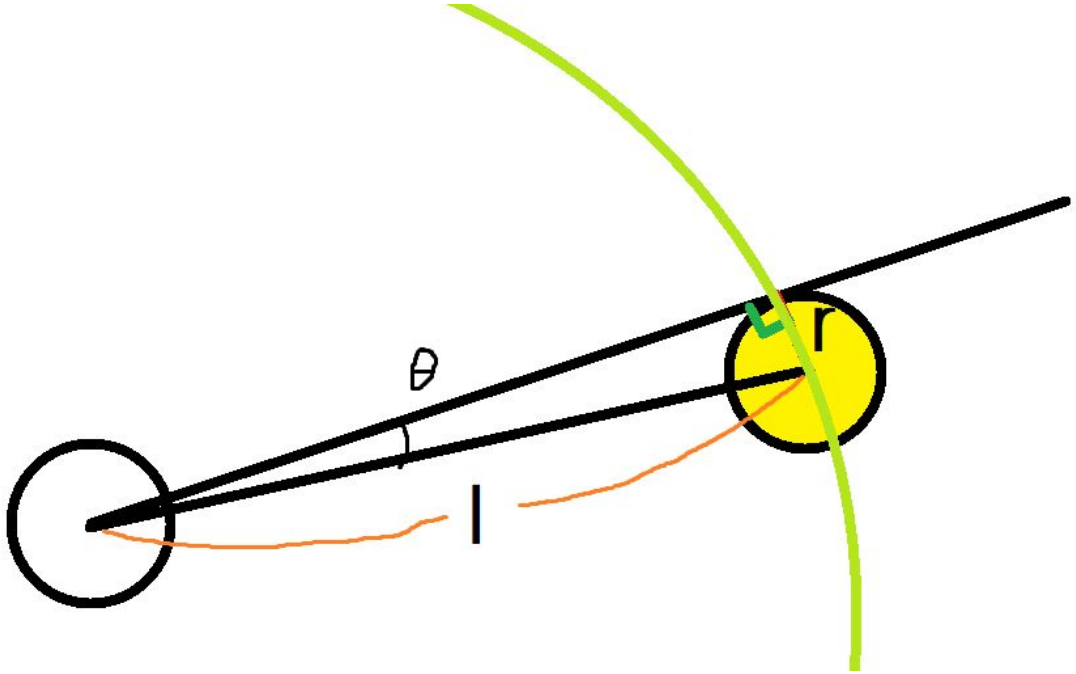
3. 구현 내용 및 결과 분석

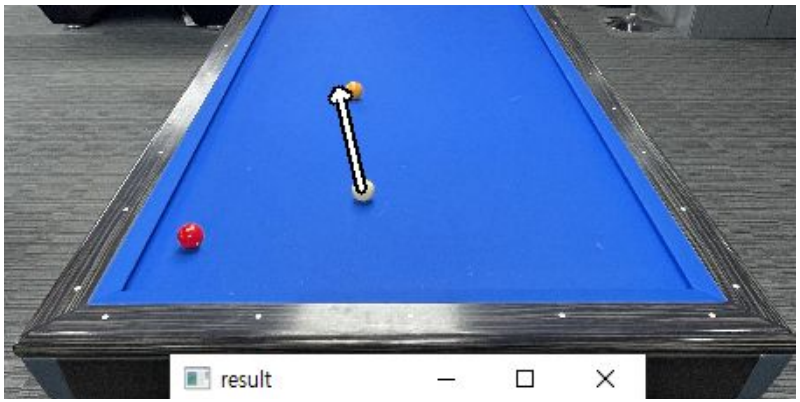
4. 결과 표시

예측한 결과를 토대로

흰 공=>빨간 공 (왼쪽 or 오른쪽)
흰 공=>노란 공 (왼쪽 or 오른쪽)
빈 쿠션

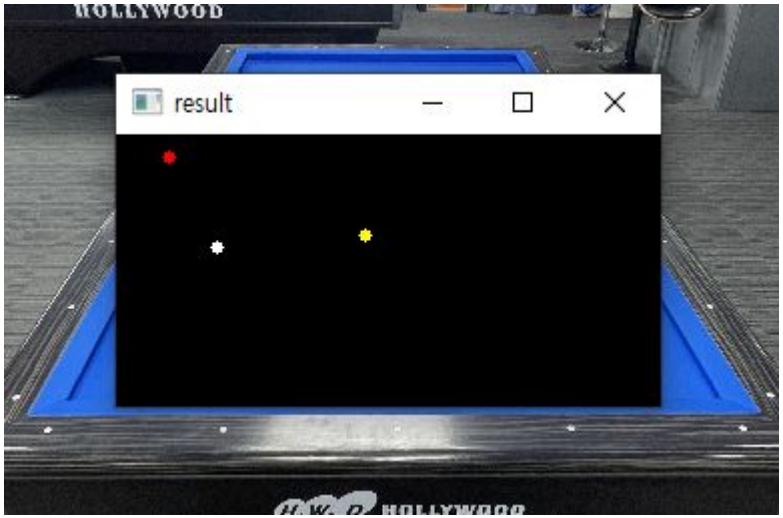
sin θ = r / l
radian = asin(sin θ)

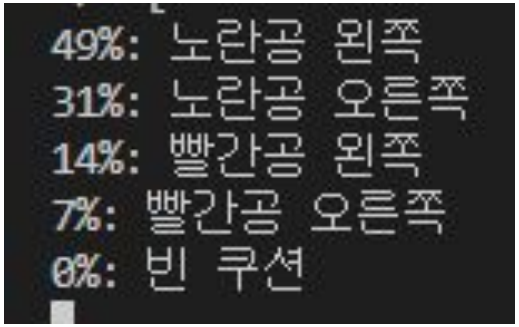




3. 구현 내용 및 결과 분석

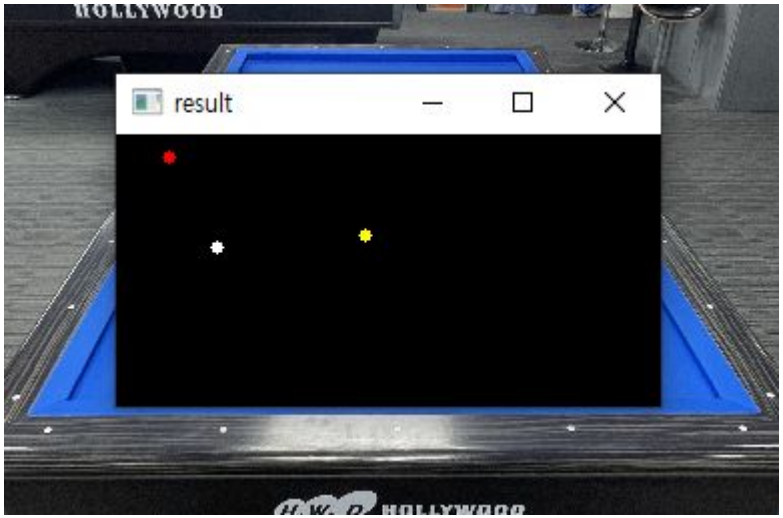
4. 실제 사용 결과 (1) 뒤돌리기

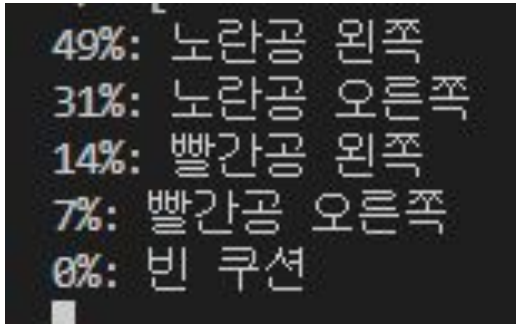


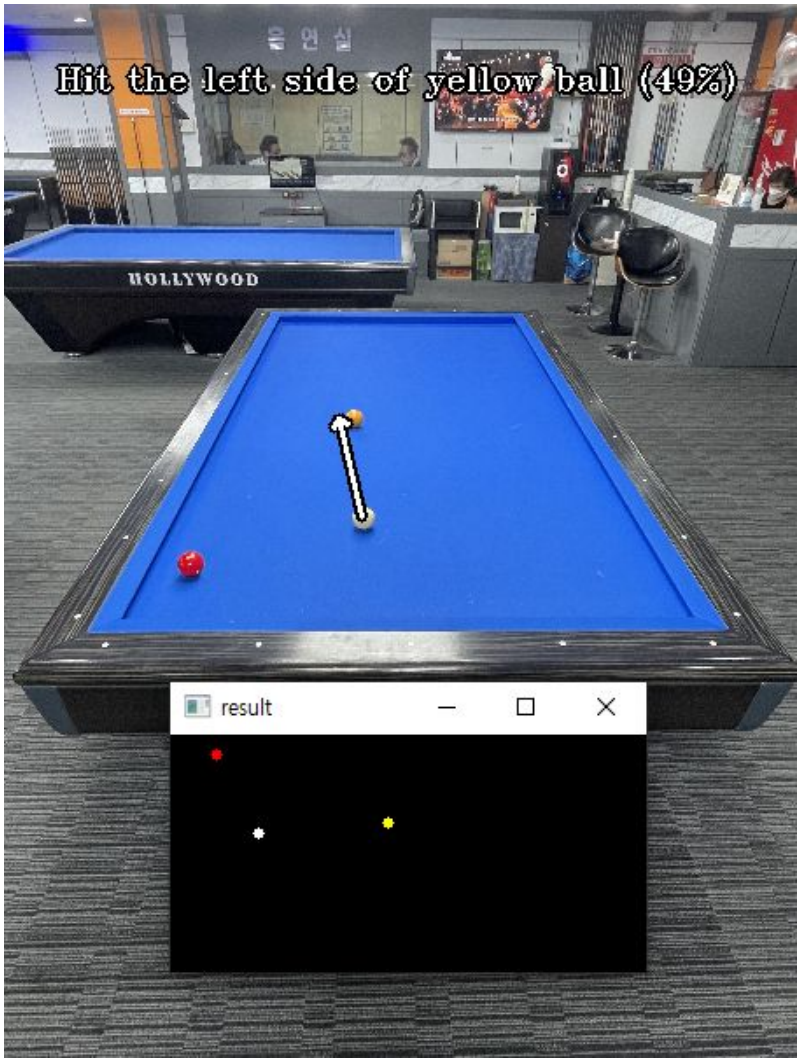


3. 구현 내용 및 결과 분석

4. 실제 사용 결과 (1) 뒤돌리기

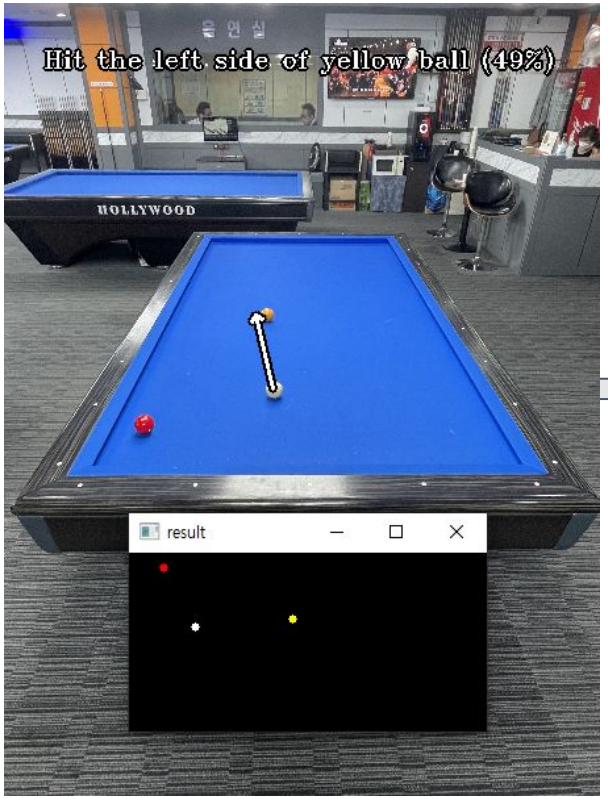






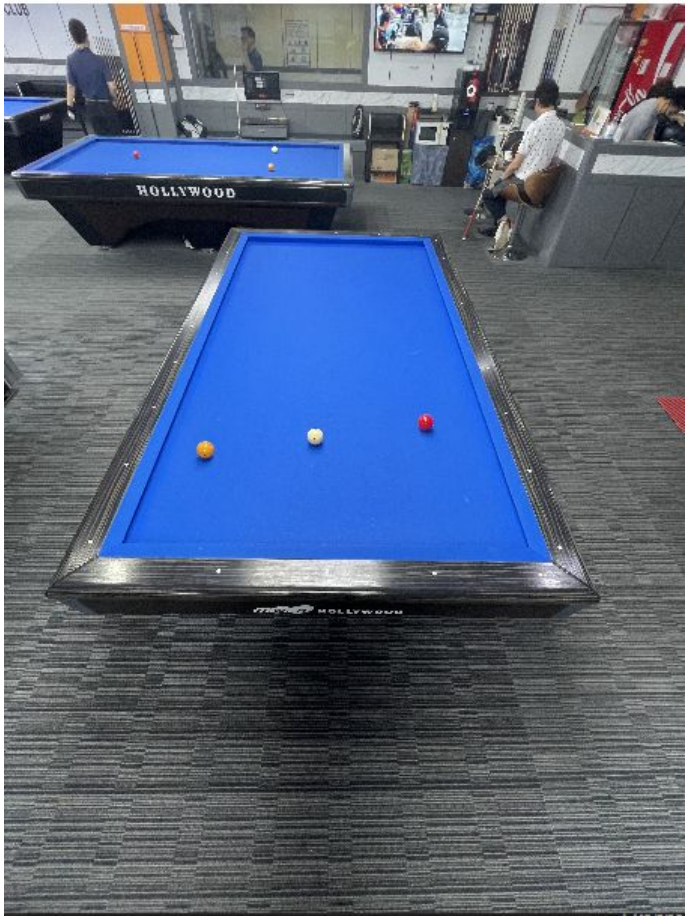
3. 구현 내용 및 결과 분석

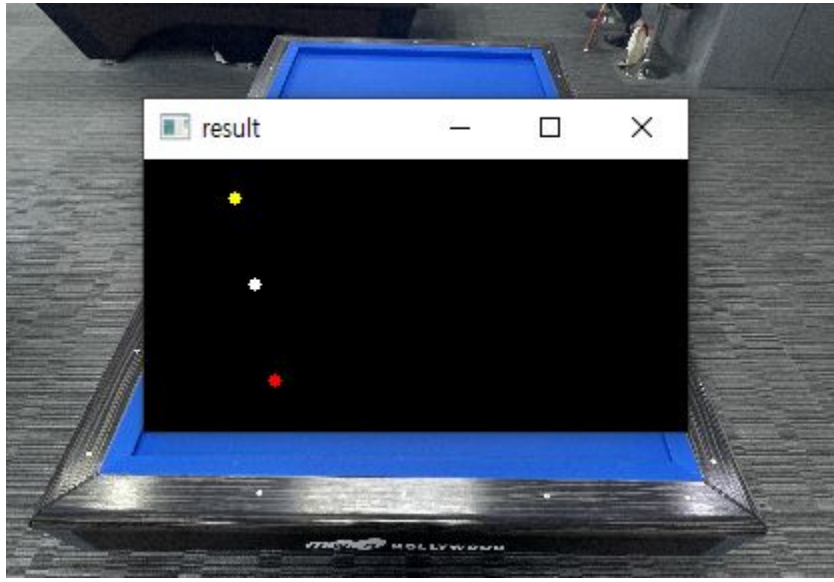
4. 실제 사용 결과 (1) 뒤돌리기



3. 구현 내용 및 결과 분석

4. 실제 사용 결과 (2) 안돌리기

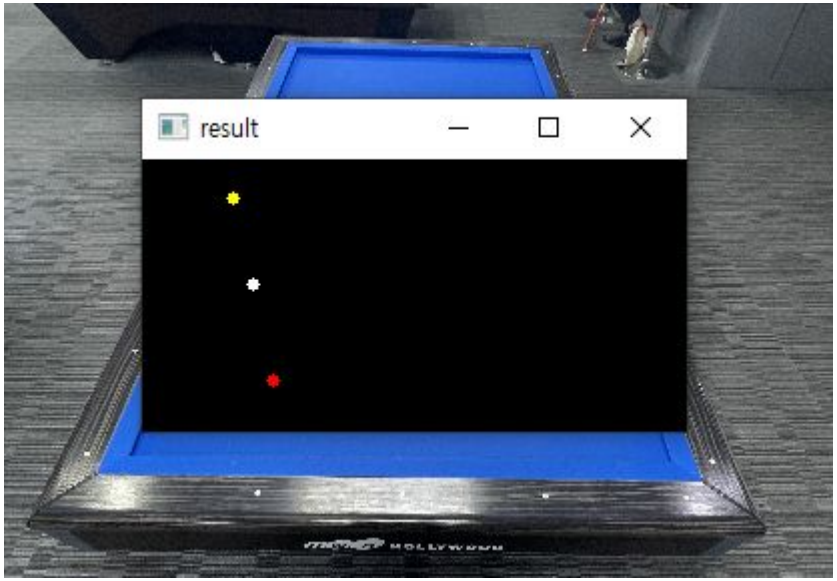


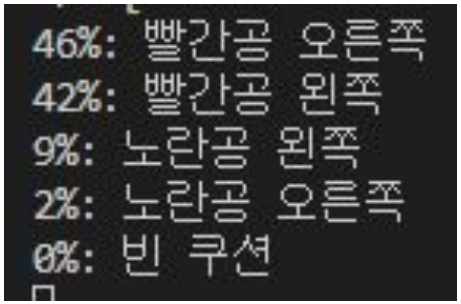


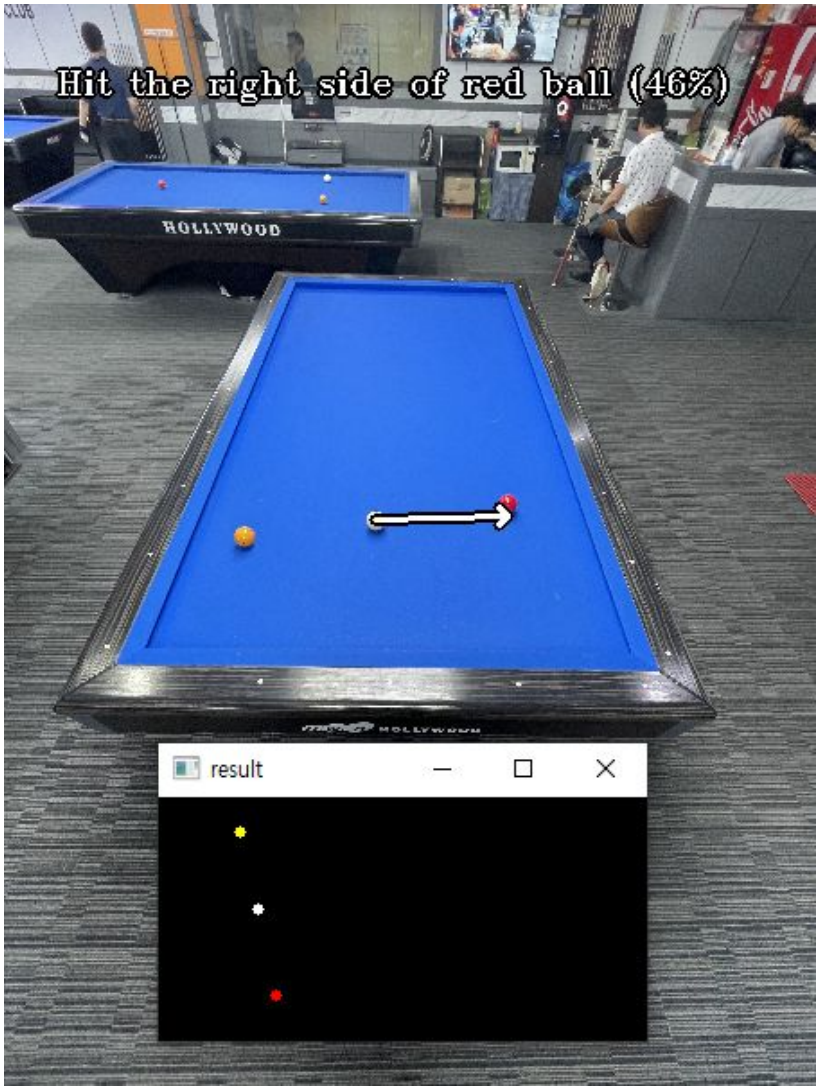
46%: 빨간공 오른쪽
42%: 빨간공 왼쪽
9%: 노란공 왼쪽
2%: 노란공 오른쪽
0%: 빈 쿠션

3. 구현 내용 및 결과 분석

4. 실제 사용 결과 (2) 안돌리기

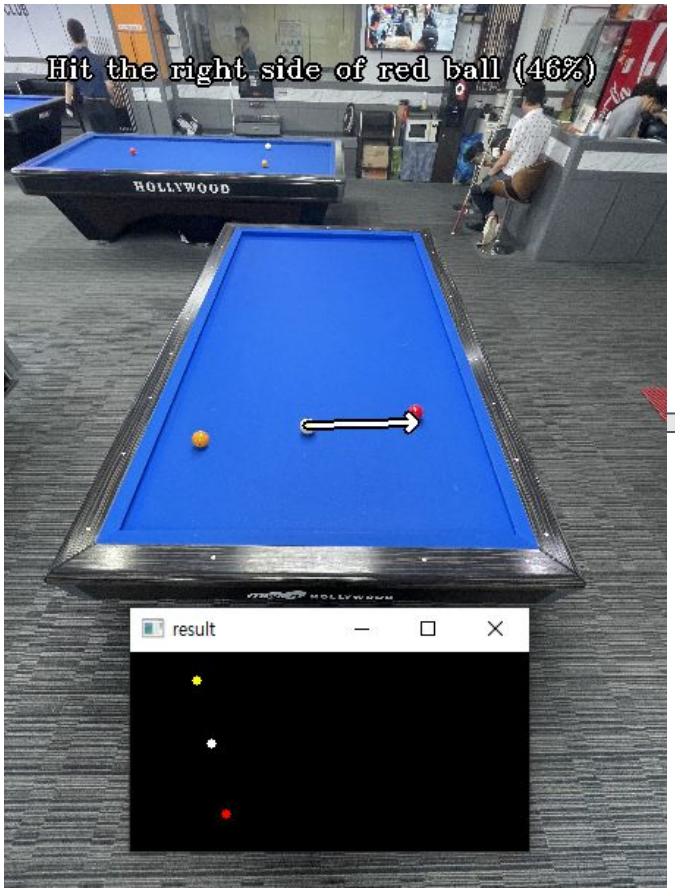






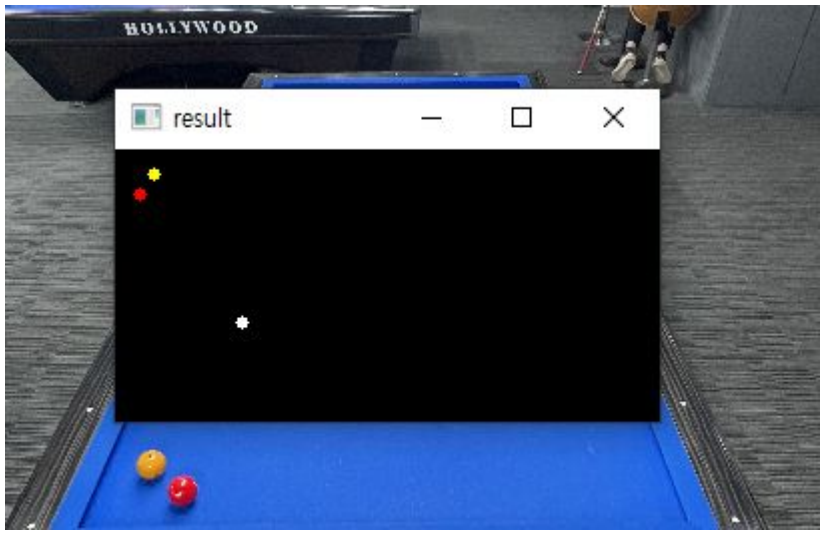
3. 구현 내용 및 결과 분석

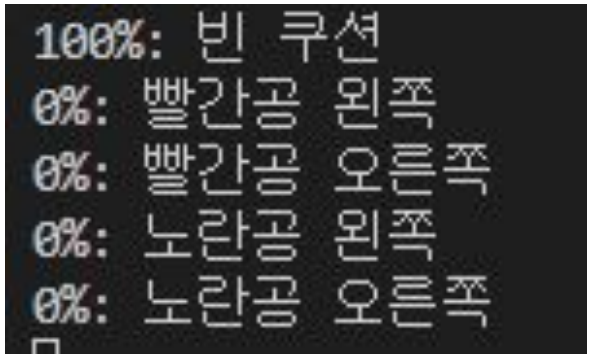
4. 실제 사용 결과 (2) 안돌리기



3. 구현 내용 및 결과 분석

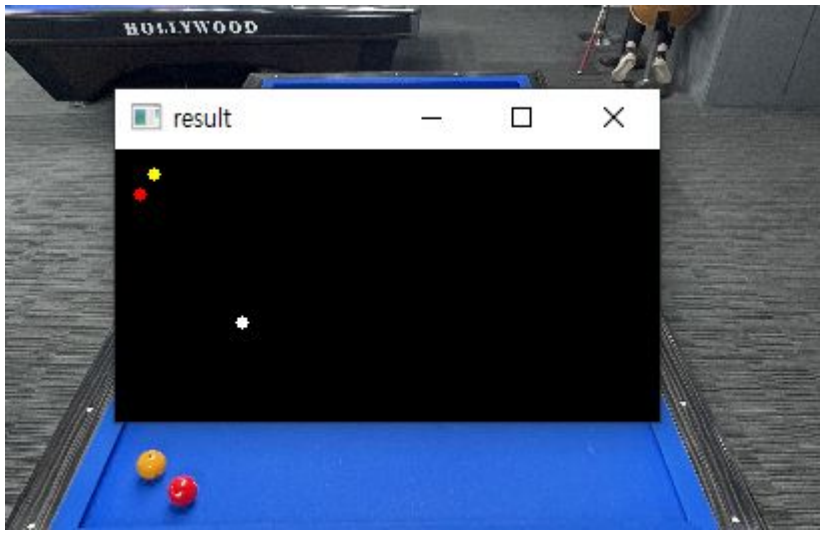
4. 실제 사용 결과 (3) 빈 쿠션

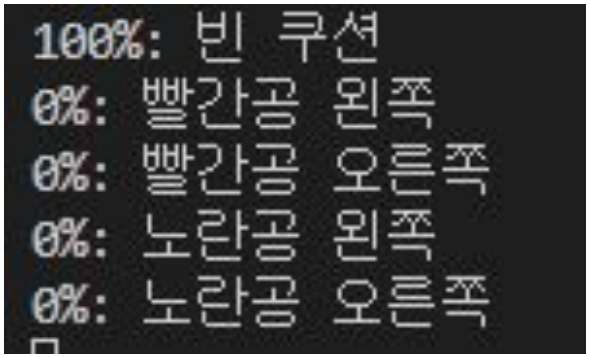


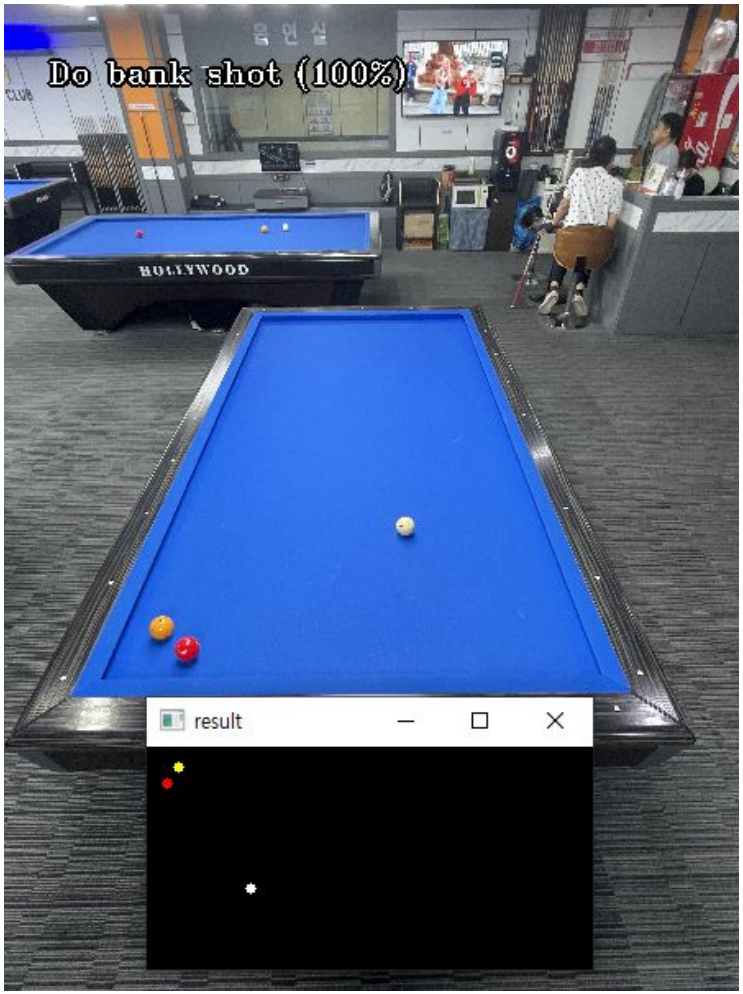


3. 구현 내용 및 결과 분석

4. 실제 사용 결과 (3) 빈 쿠션

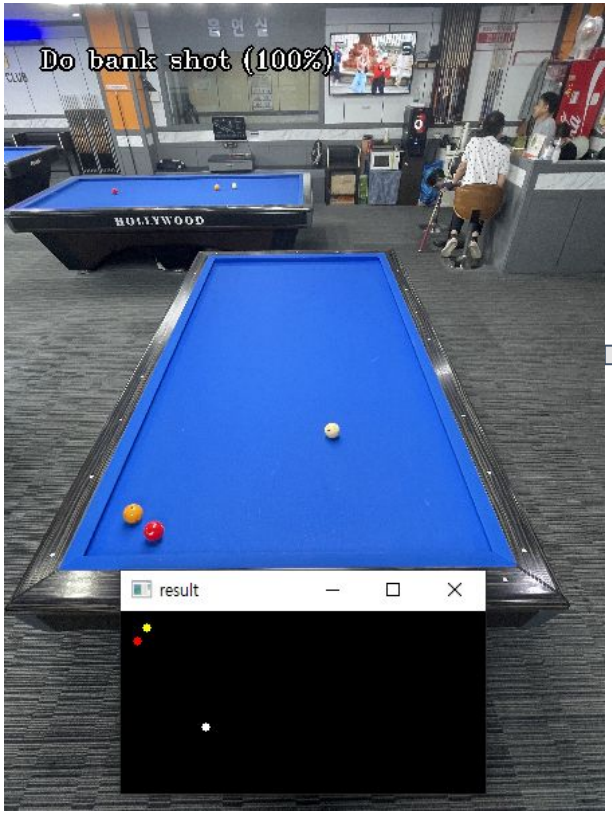






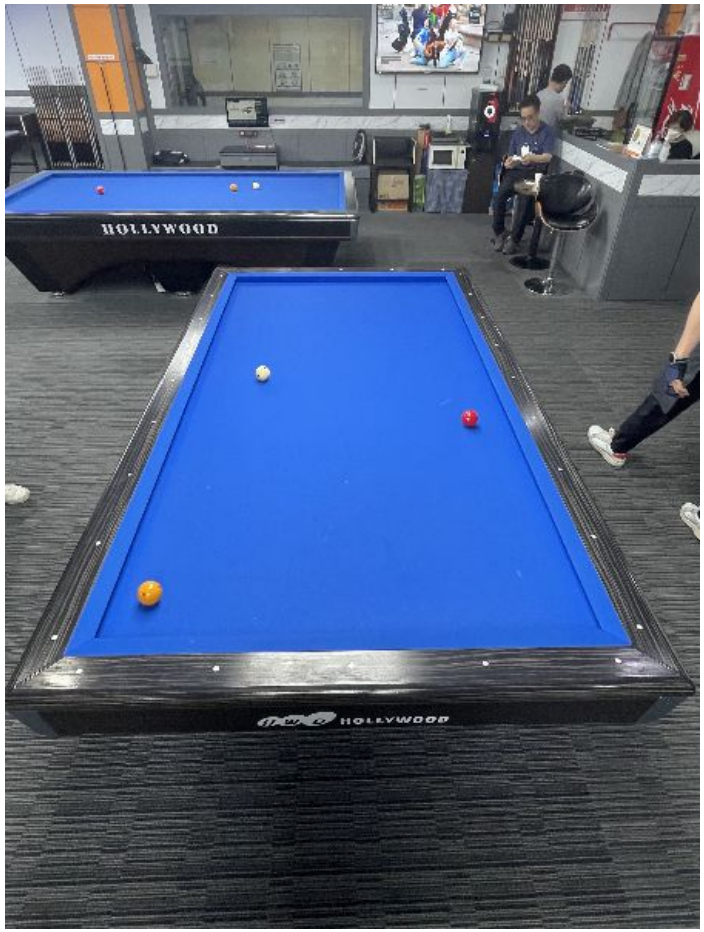
3. 구현 내용 및 결과 분석

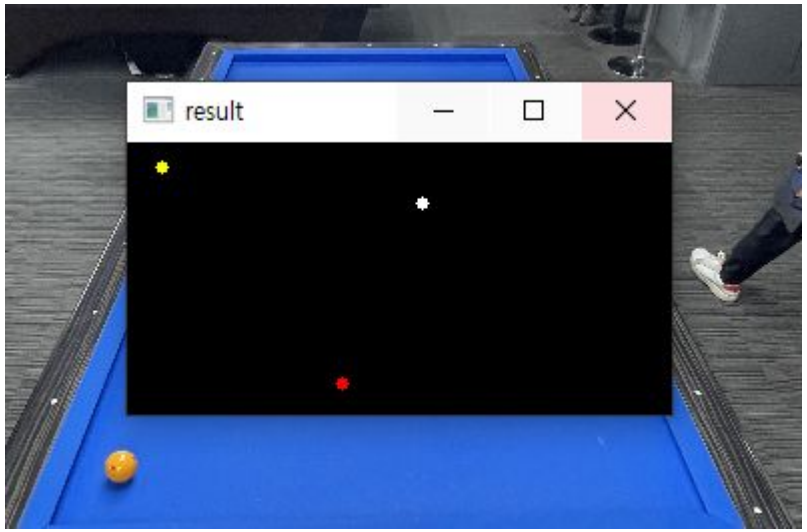
4. 실제 사용 결과 (3) 빈 쿠션



3. 구현 내용 및 결과 분석

4. 실제 사용 결과 (4) 대회전

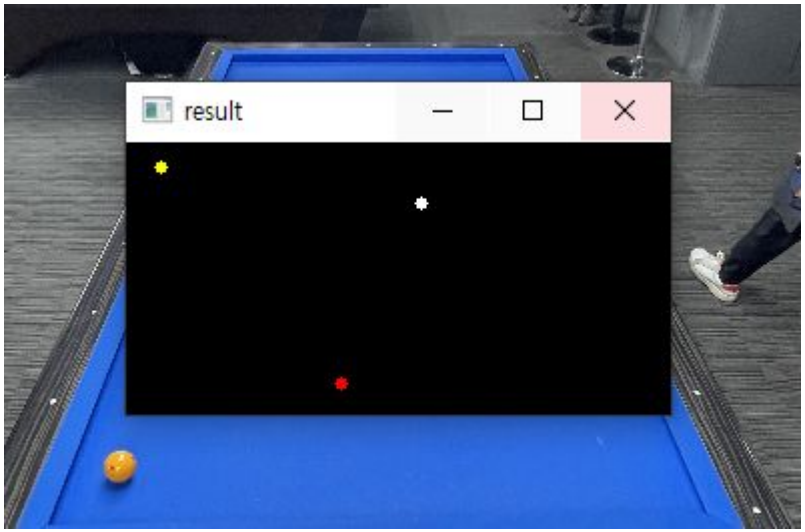


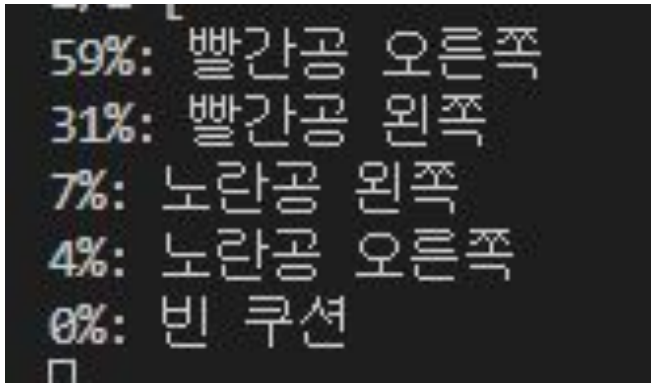


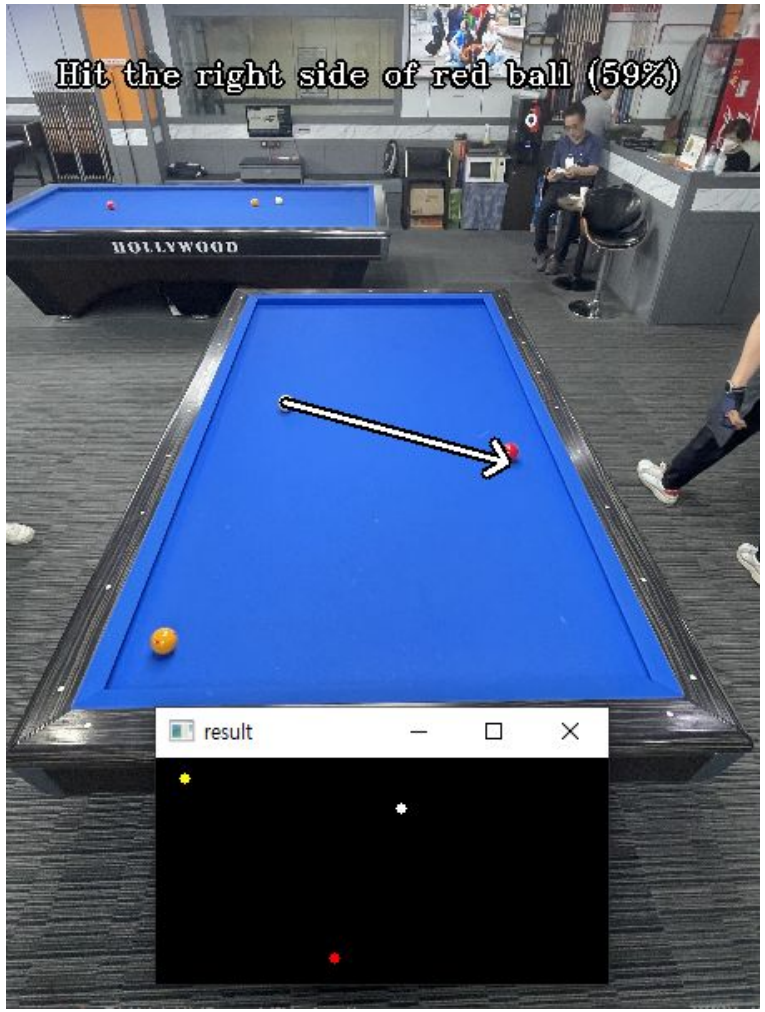
59%: 빨간공 오른쪽
31%: 빨간공 왼쪽
7%: 노란공 왼쪽
4%: 노란공 오른쪽
0%: 빈 쿠션

3. 구현 내용 및 결과 분석

4. 실제 사용 결과 (4) 대회전

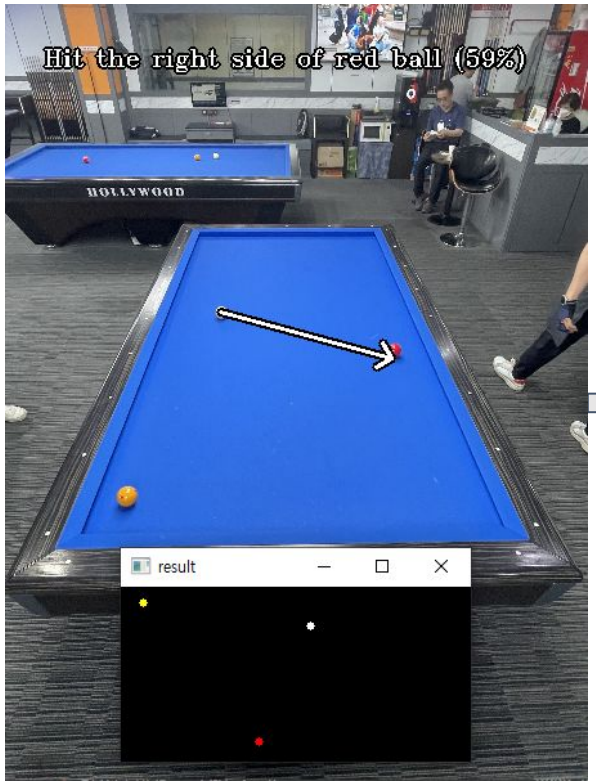




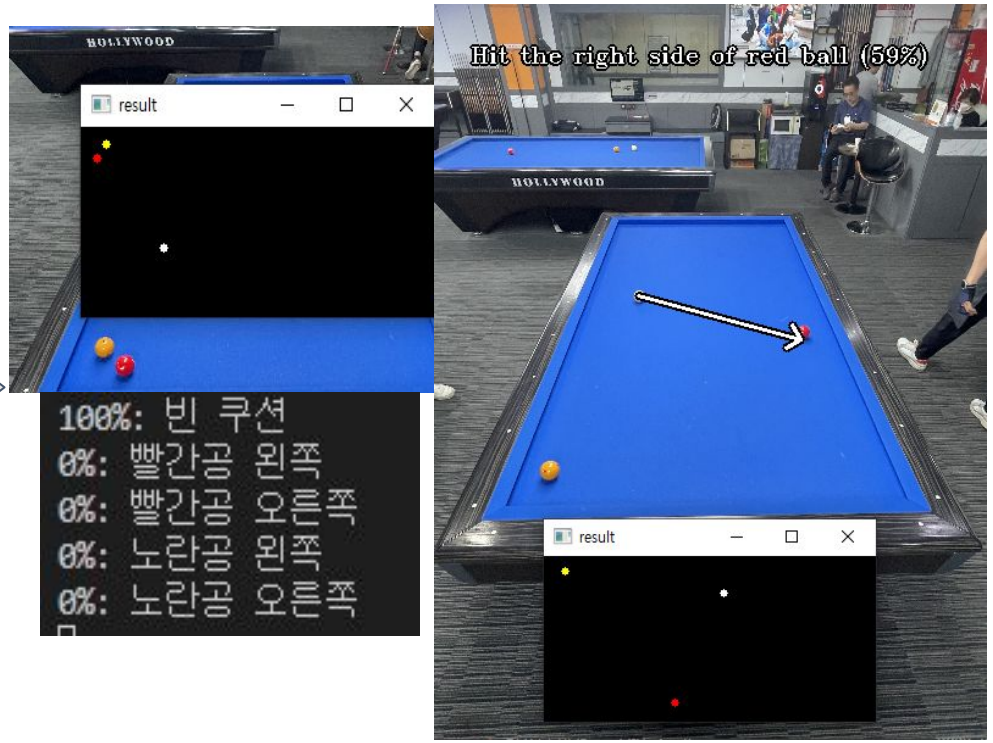


3. 구현 내용 및 결과 분석

4. 실제 사용 결과 (4) 대회전



4. 개선 내용



기존

개선 결과

5. 결론

- 컨투어 분석과 HSV 색 공간 추출, Perspective Transformation 등에 대해 탐구하고 허프 변환을 활용해서 변환된 이미지에서 당구공을 찾을 수 있다.
- 딥러닝을 활용해 당구대와 당구공 좌표, 수구의 경로를 학습시켜 적합한 득점 방법과 경로를 획득할 수 있다.
- 신형 당구대에 설치된 디지털 점수판(버드 아이 뷰 카메라가 연동되어 있음)에 이 기술을 접목하면 보다 효용성이 높아질 수 있을 것이다.

6. 역할 분담

- 컴퓨터 비전

  컨투어 분석, HSV 색 공간 추출, Perspective Transformation

  20172864 서정현

- 머신 러닝

  진행 방향, 경로 예측 및 가이드

  20176342 송민준

7. 참고문헌

https://github.com/choonguri/dl-3cushion-hint

(dl-3cushion-hint 딥러닝을 이용한 3쿠션 힌트 안드로이드앱)