Smart Gripper

캡스톤 디자인 (1)

김진용 임동훈

A Table of Contents.

Smart Factory

- 공장 내 설비와 기계에 센서(IoT)가 설치되어 데이터가 실시간으로 수집, 분석
- 공장 내 모든 상황들이 일목요연하게 보여지고, 이를 분석해 목적된 바에 따라 스스로 제어되는 공장

Smart Factory

현재 사용되는 대다수의 그리퍼는 정해진 작업의 반복

Purpose

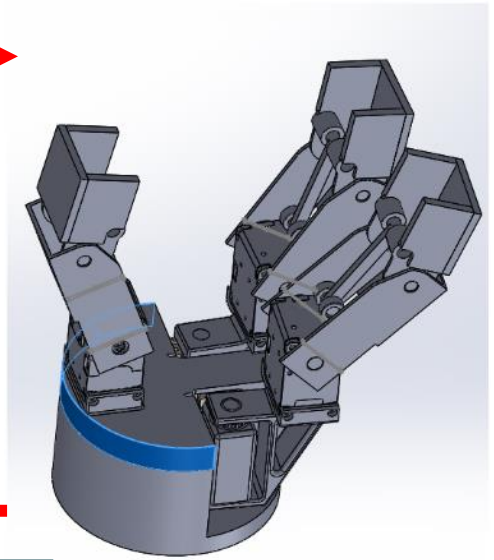
- 4차 산업 혁명 시대에 따른 스마트 팩토리가 발전
- 여러 환경 요인에 맞춰 다양하게 사용할 수 있는 자동화 그리퍼 필요
- 첫 번째 단계로 물체의 형상에 따라 물체를 잡는 모션을 변화시키는 그리퍼 개발 목표

Method

Frame 전달

오브젝트
분류 결과 값 전달

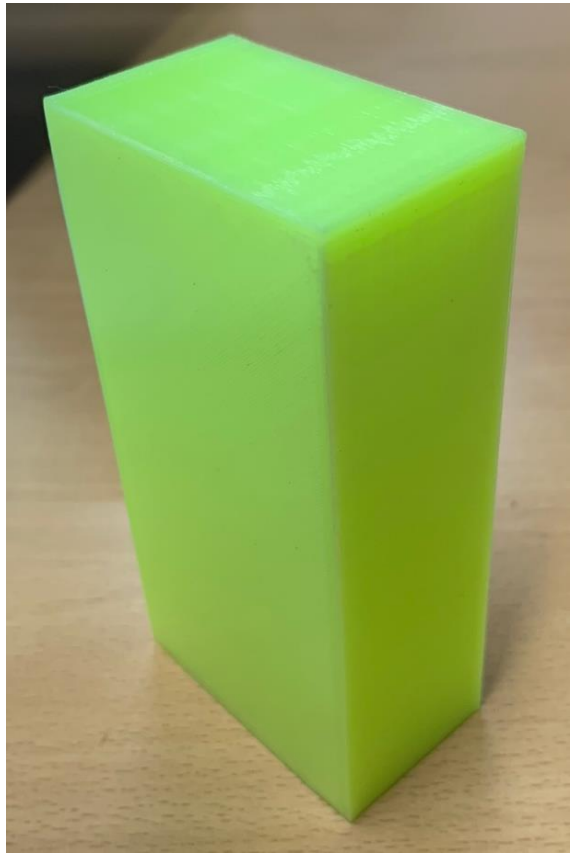
모터 제어



피드백

Part 2, Method

Object

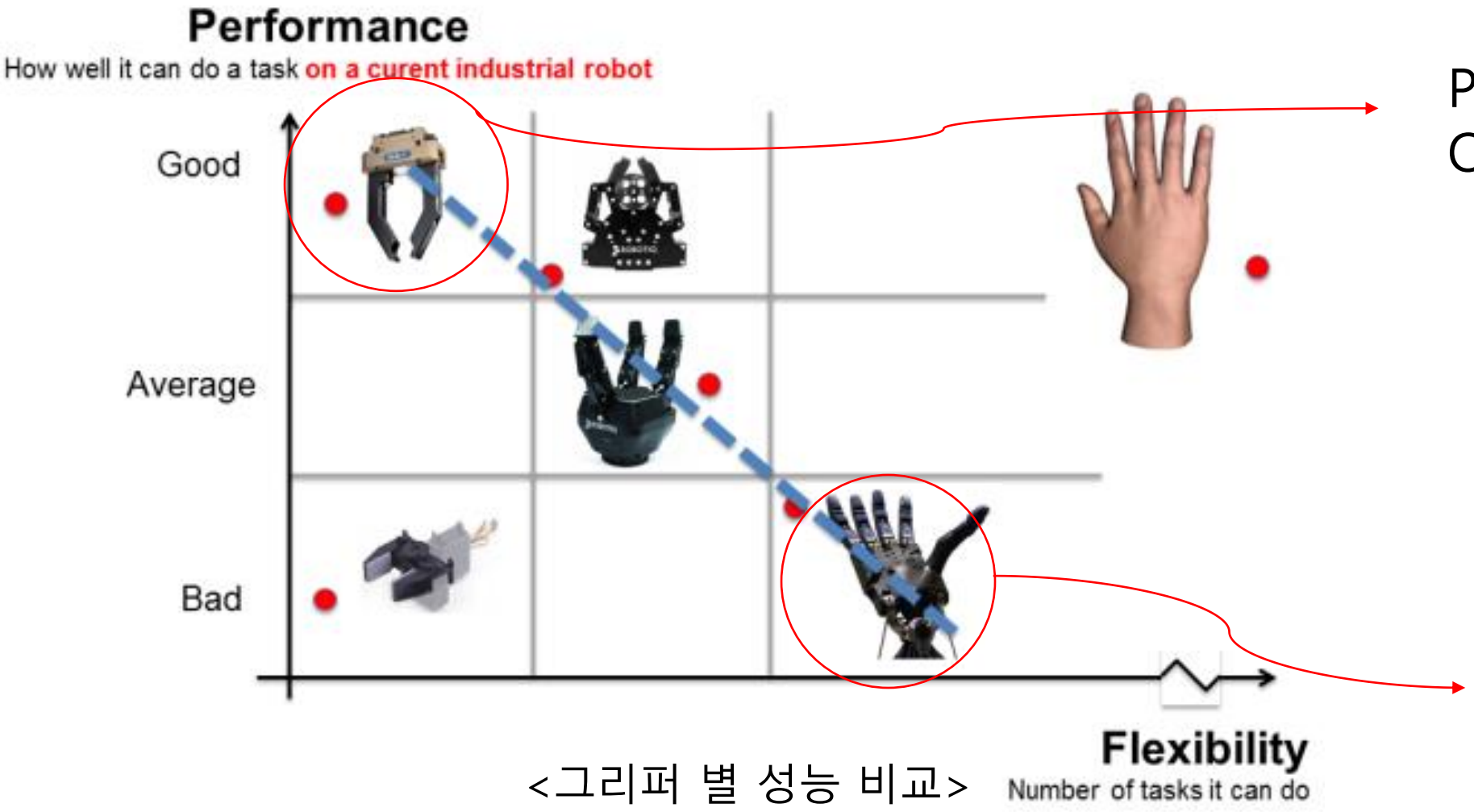


<직육면체>

<구체>

<원기둥>

설계



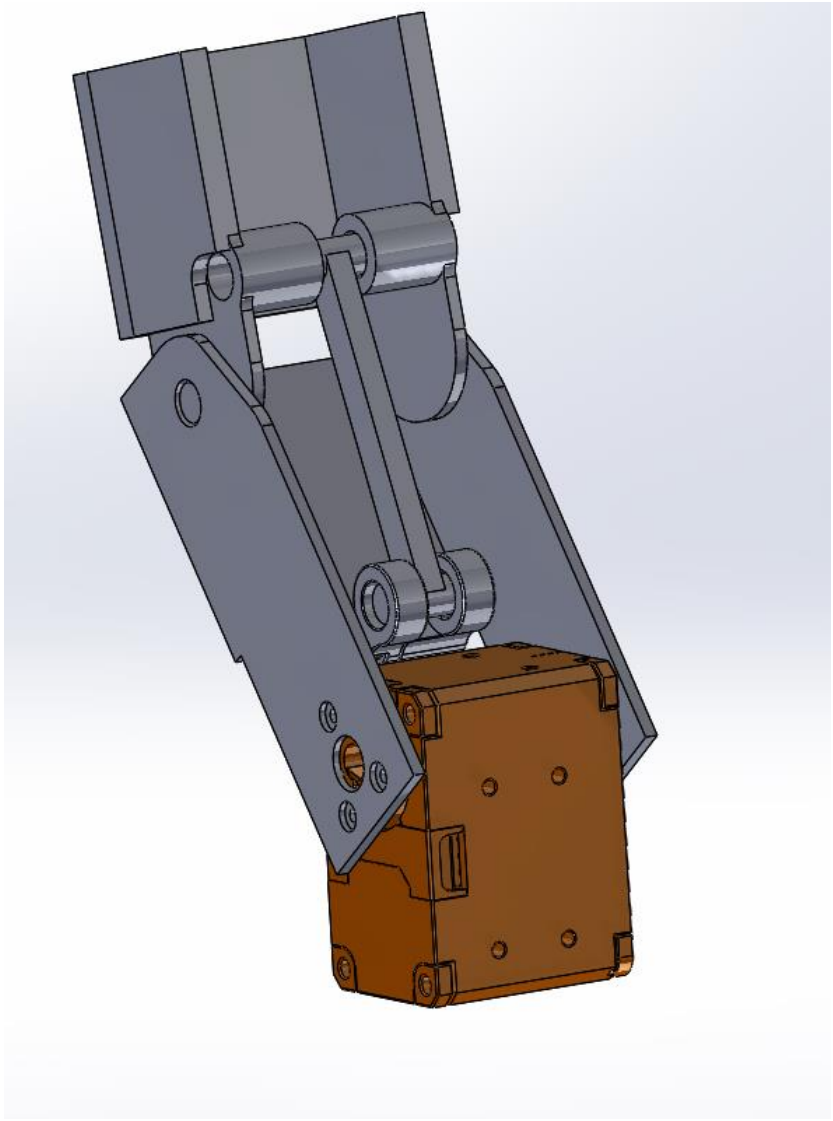
<그리퍼 별 성능 비교>

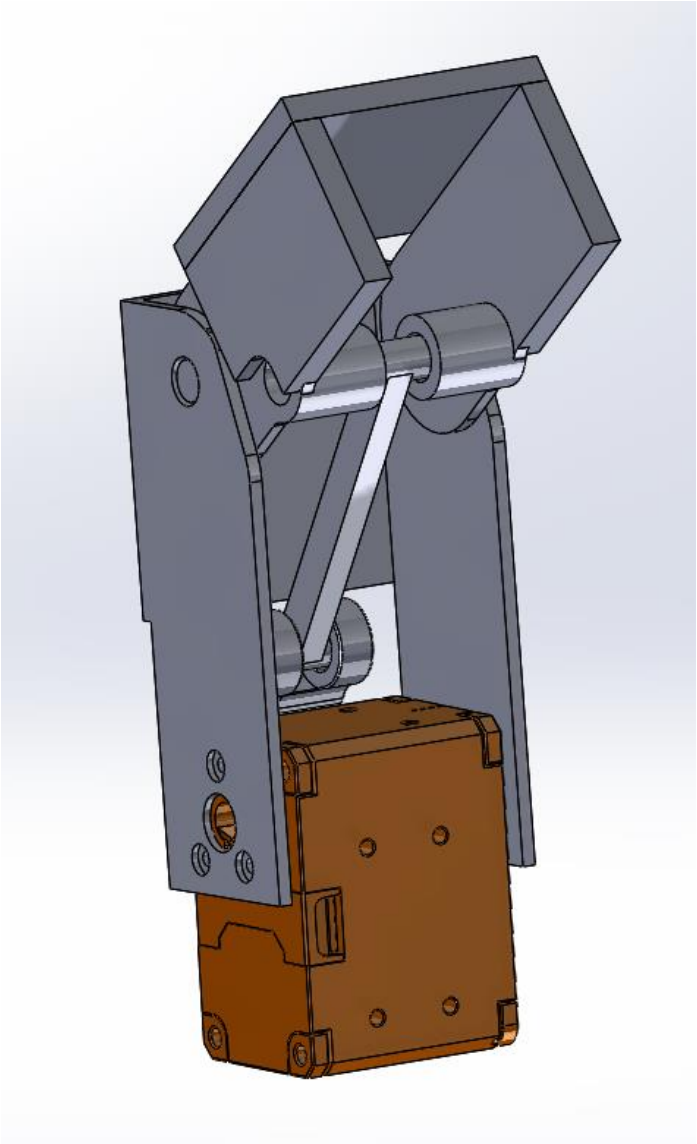
Parallel Jaw Gripper
Only for simple task

"Three Finger Gripper"

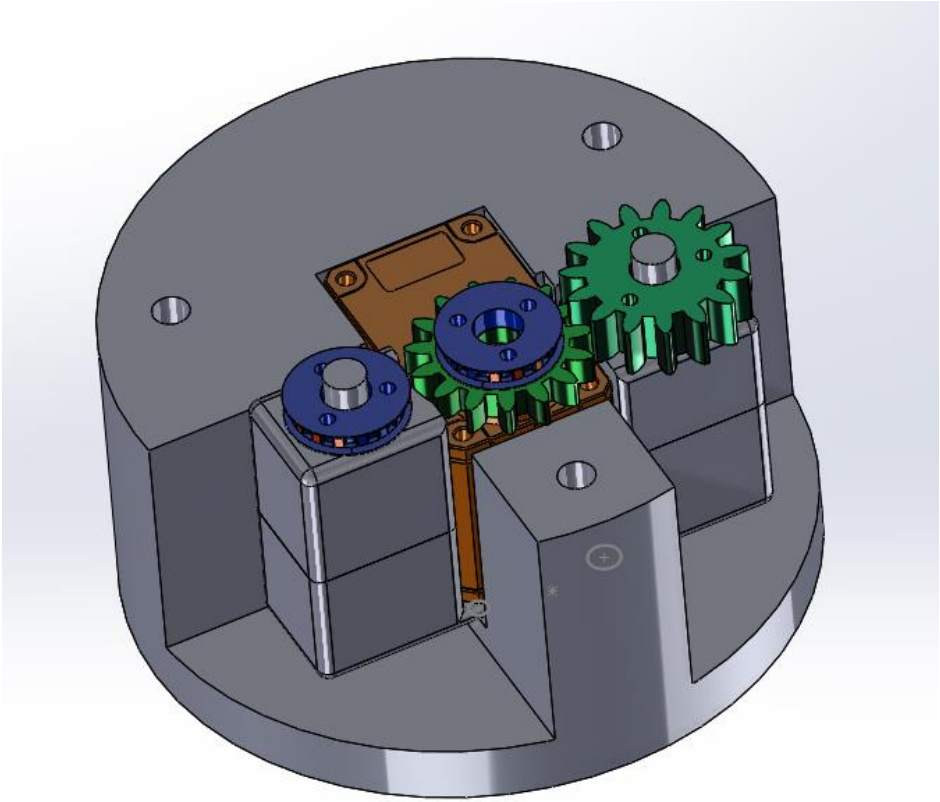
Hard to control and heavy

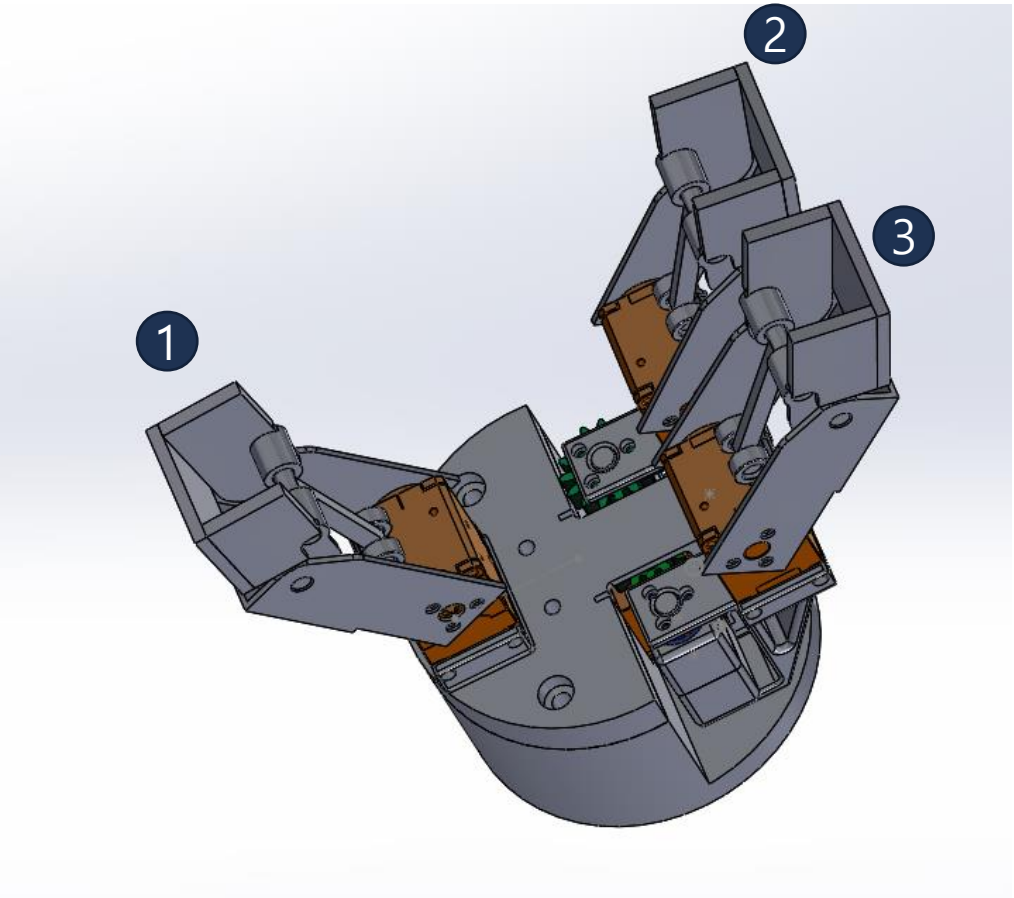
설계





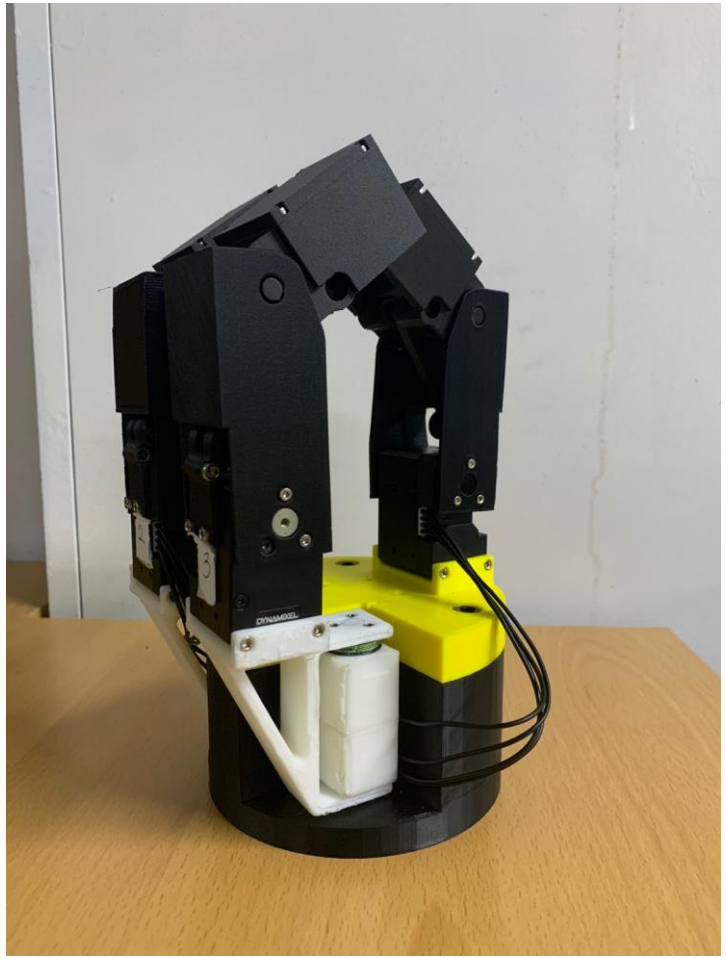
설계



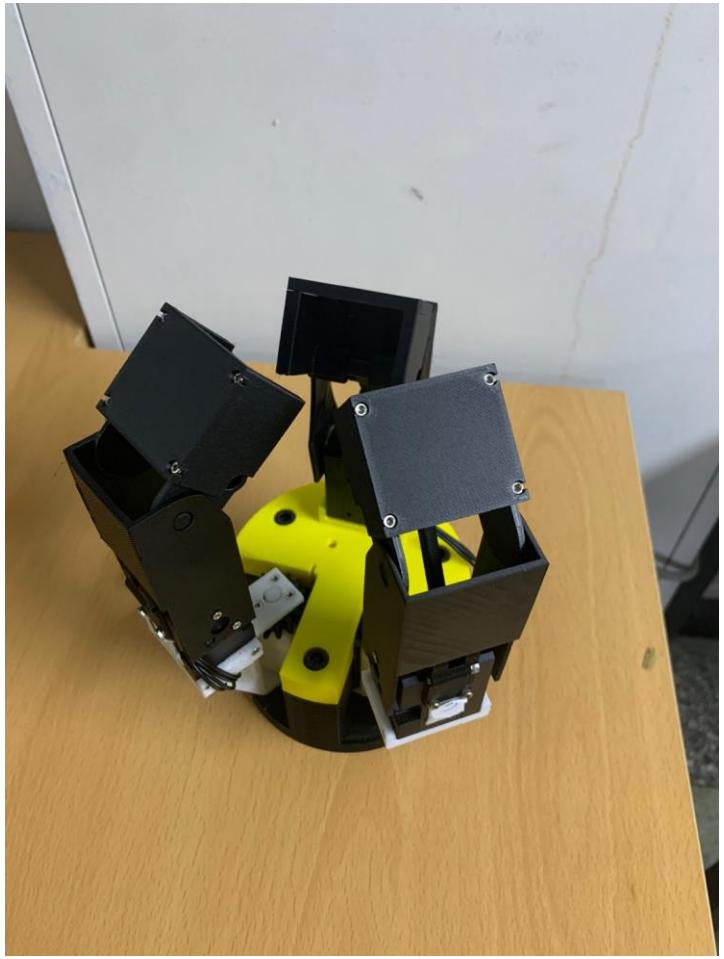


Part2, Method

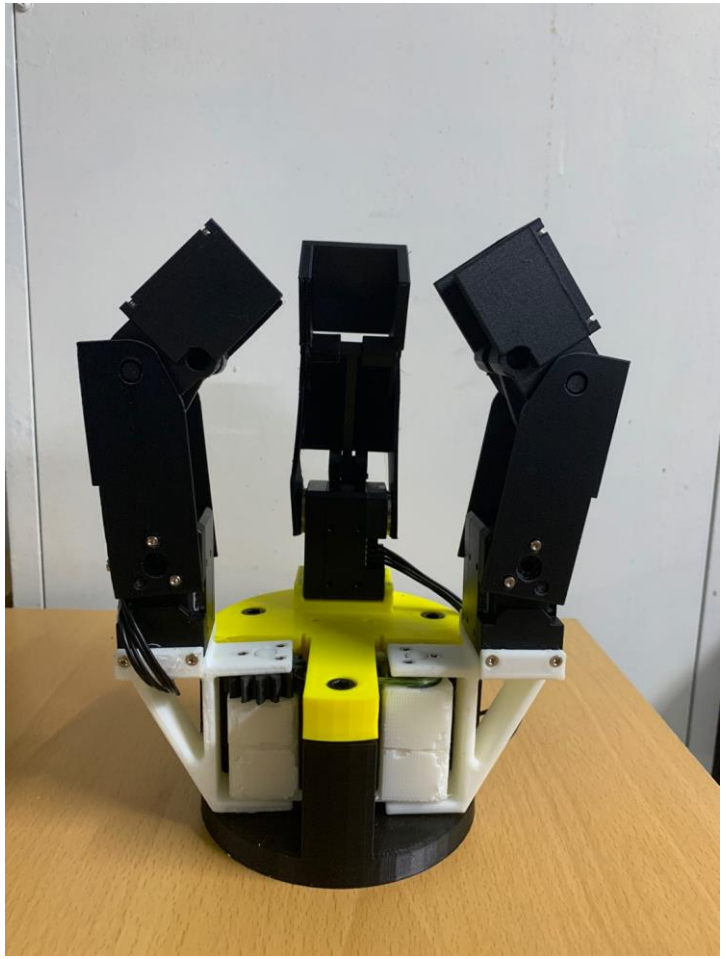
제작



<직육면체>



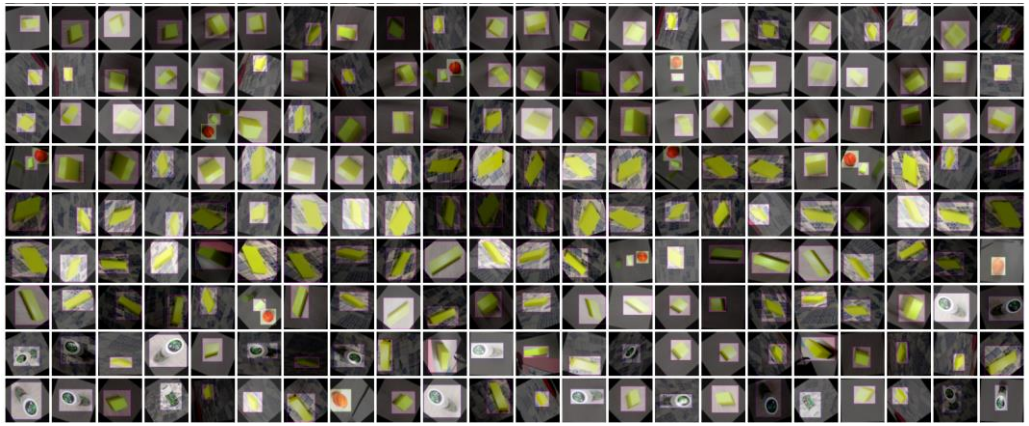
<구체>



<원기둥>

비전 인식

1) Data Augmentation



원본 데이터 약 600장 -> 약 1550장

Augmentations

90° Rotate: Clockwise, Counter-Clockwise, Upside Down

Crop: 0% Minimum Zoom, 17% Maximum Zoom

Rotation: Between -45° and +45°

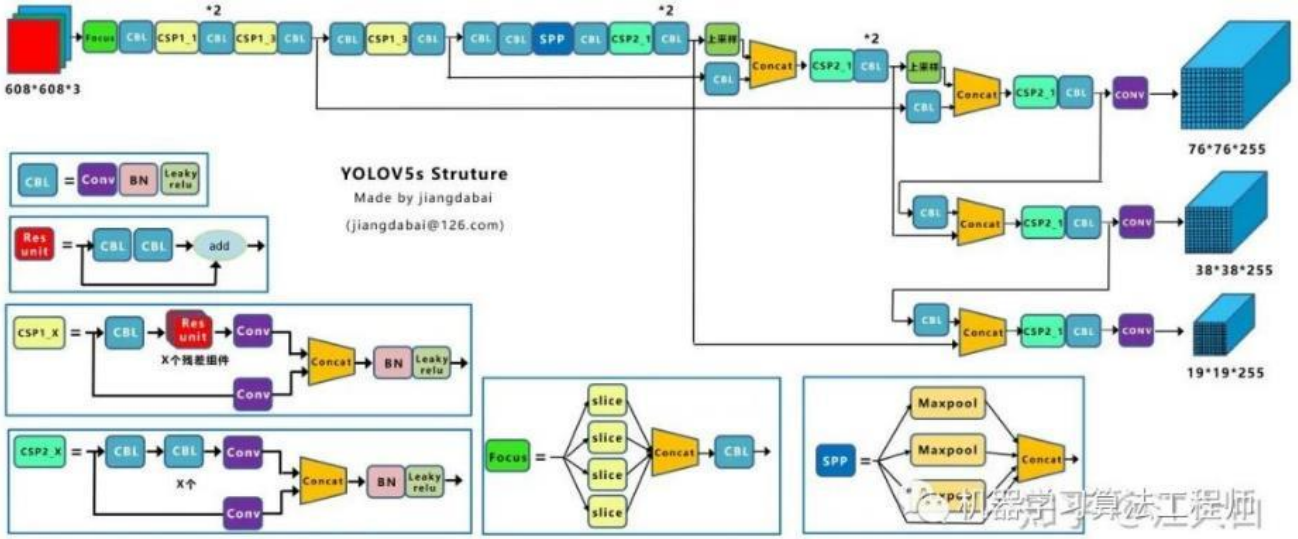
Brightness: Between -25% and +25%

Exposure: Between -25% and +25%

비전 인식

2) Yolo V5 custom training

Yolo v5는 실시간 객체 탐지에 매우 유리한 모델로, 가장 가벼운 v5s 모델을 전이학습 하였다. 실시간으로 영상을 받아와 프레임 안에 객체가 있다면 시리얼 통신을 사용하여 openCR 제어기에 분류한 객체 값을 전달한다.

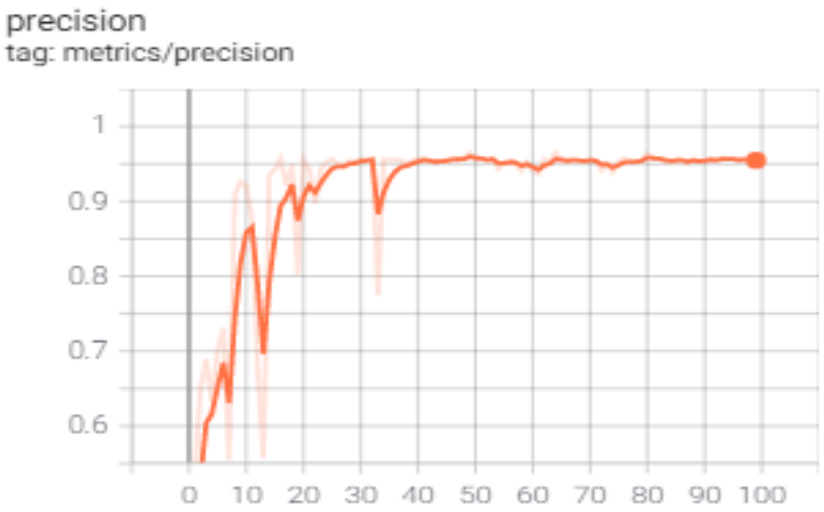


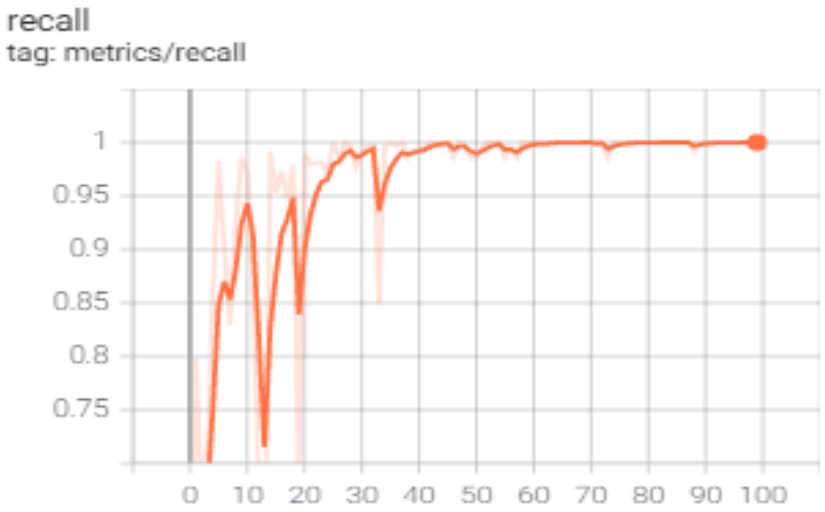
비전 인식

2) Yolo V5 custom training

정확도 / 재현율

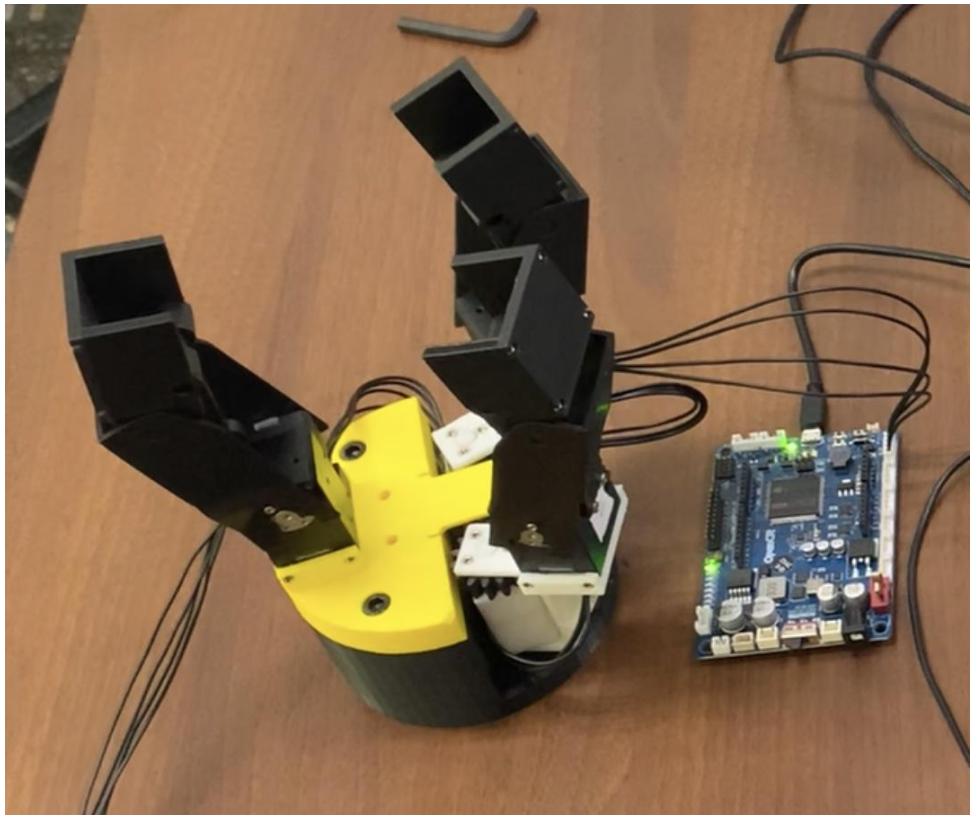
결과





Part2, Method

제어



Smart Factory

YOLO V5

detect.py

```
        # Process detections
        for i, det in enumerate(pred):  # detections per image
            if webcam:  # batch_size >= 1
                p, s, im0, frame = path[i], '%g: ' % i, im0s[i].copy(), dataset.count
            else:
                p, s, im0, frame = path, '', im0s, getattr(dataset, 'frame', 0)

            p = Path(p)  # to Path
            save_path = str(save_dir / p.name)  # img.jpg
            txt_path = str(save_dir / 'labels' / p.stem) + ('' if dataset.mode == 'image' els
            s += '%gx%g ' % img.shape[2:]  # print string
            gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]  # normalization gain whwh
            if len(det):
                # Rescale boxes from img_size to im0 size
                det[:, :4] = scale_coords(img.shape[2:], det[:, :4], im0.shape).round()

                # Print results
                for c in det[:, -1].unique():
                    n = (det[:, -1] == c).sum()  # detections per class
                    s += f"{n} {names[int(c)]}{'s' * (n > 1)}, "  # add to string
                    a=f"{n}"
                    print('a',a,names[int(c)],n,n.type())

                    if names[int(c)] =='cylinder':
                        # while 1:
                            # time.sleep(1)
                            c=''
                            c='A'
                            if c=='q':
                                break
                            else:
                                c=c.encode('utf-8')
                                ard.write(c)
                    elif names[int(c)] =='cube':
                        # while 1:
                            # time.sleep(1)
                            c=''
                            c='B'
                            if c=='q':
```

OPENCR(ARDUINO)

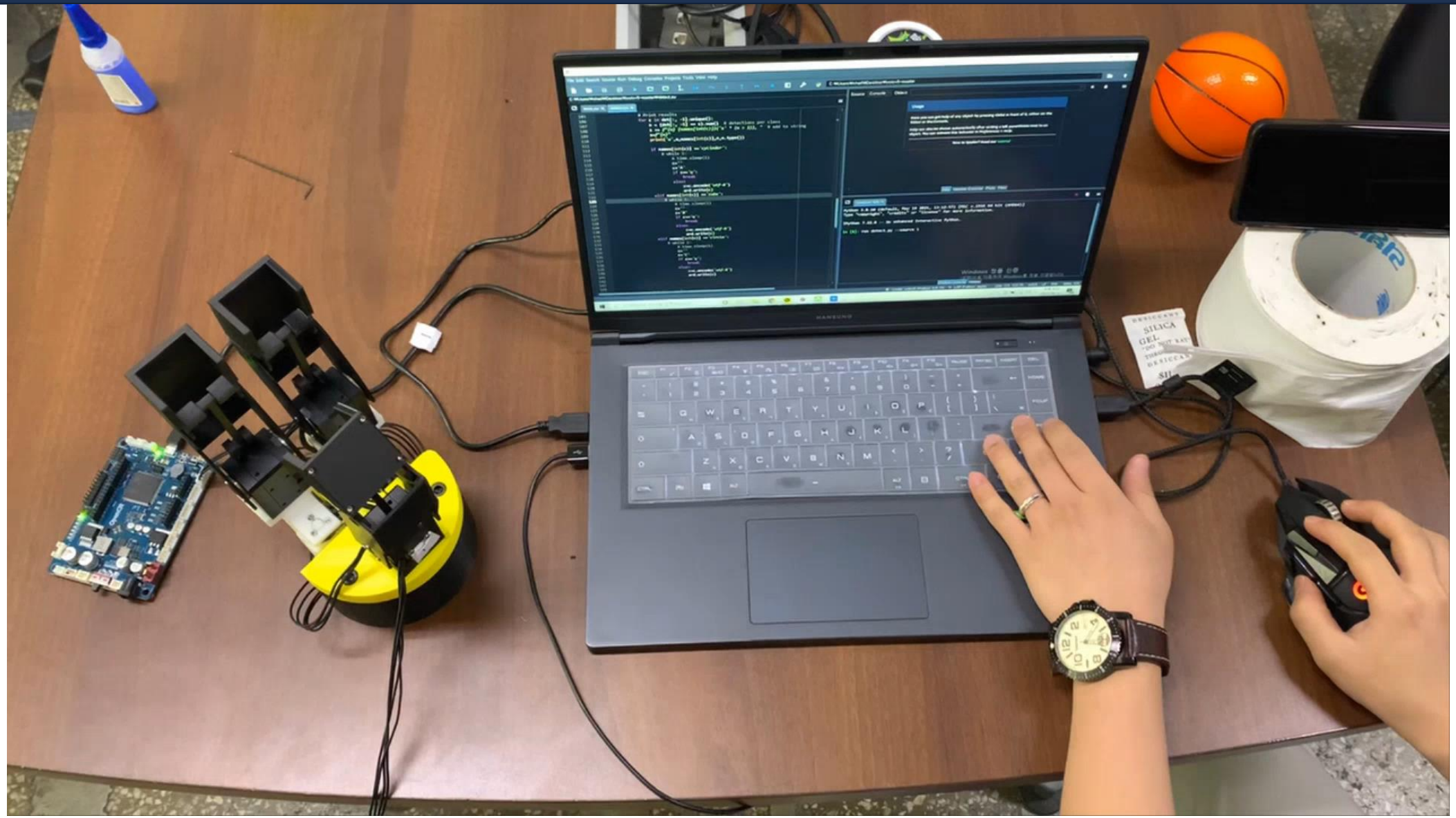
DH | 아두이노 1.8.15 (Windows Store 1.8.49.0)

파일 편집 스케치 툴 도움말

DH §

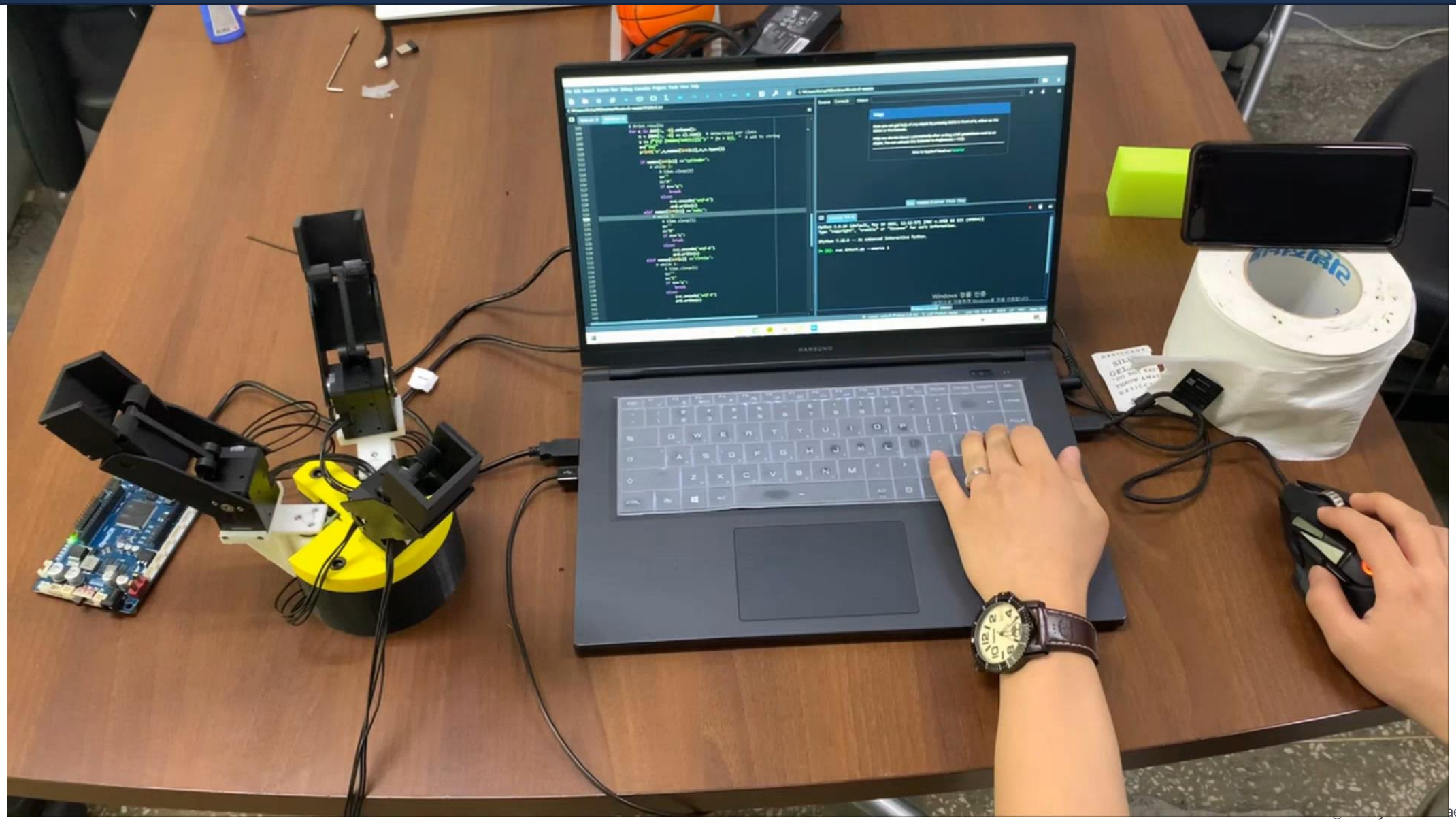
```
  if (dxl_addparam_result != true)
  {
    Serial.print("[ID:"); Serial.print(DXL4_ID); Serial.println("groupSyncRead getdata failed");
    return;
  }

  // Get Dynamixel#1 present position value
  dxl1_present_position = groupSyncRead.getData(DXL1_ID, ADDR_PRO_PRESENT_POSITION, LEN_PRO_PRESENT_P

  // Get Dynamixel#2 present position value
  dxl2_present_position = groupSyncRead.getData(DXL2_ID, ADDR_PRO_PRESENT_POSITION, LEN_PRO_PRESENT_P

 // Get Dynamixel#3 present position value
  dxl3_present_position = groupSyncRead.getData(DXL3_ID, ADDR_PRO_PRESENT_POSITION, LEN_PRO_PRESENT_P

 // Get Dynamixel#4 present position value
  dxl4_present_position = groupSyncRead.getData(DXL4_ID, ADDR_PRO_PRESENT_POSITION, LEN_PRO_PRESENT_P

  Serial.print("[ID:"); Serial.print(DXL1_ID);
  Serial.print("] GoalPos:"); Serial.print(dxl_goal_position);
  Serial.print("  PresPos:"); Serial.print(dxl1_present_position);
  Serial.print(" [ID:"); Serial.print(DXL2_ID);
  Serial.print("] GoalPos:"); Serial.print(dxl_goal_position2);
  Serial.print("  PresPos:"); Serial.print(dxl2_present_position);
  Serial.println(" ");
  Serial.print("[ID:"); Serial.print(DXL3_ID);
  Serial.print("] GoalPos:"); Serial.print(dxl_goal_position3);
  Serial.print("  PresPos:"); Serial.print(dxl3_present_position);
  Serial.print(" [ID:"); Serial.print(DXL4_ID);
  Serial.print("] GoalPos:"); Serial.print(dxl_goal_position4);
  Serial.print("  PresPos:"); Serial.print(dxl4_present_position);
  Serial.println(" ");

}while((abs(dxl_goal_position - dxl1_present_position) > DXL_MOVING_STATUS_THRESHOLD) || (abs(dxl_goa
```

Balls

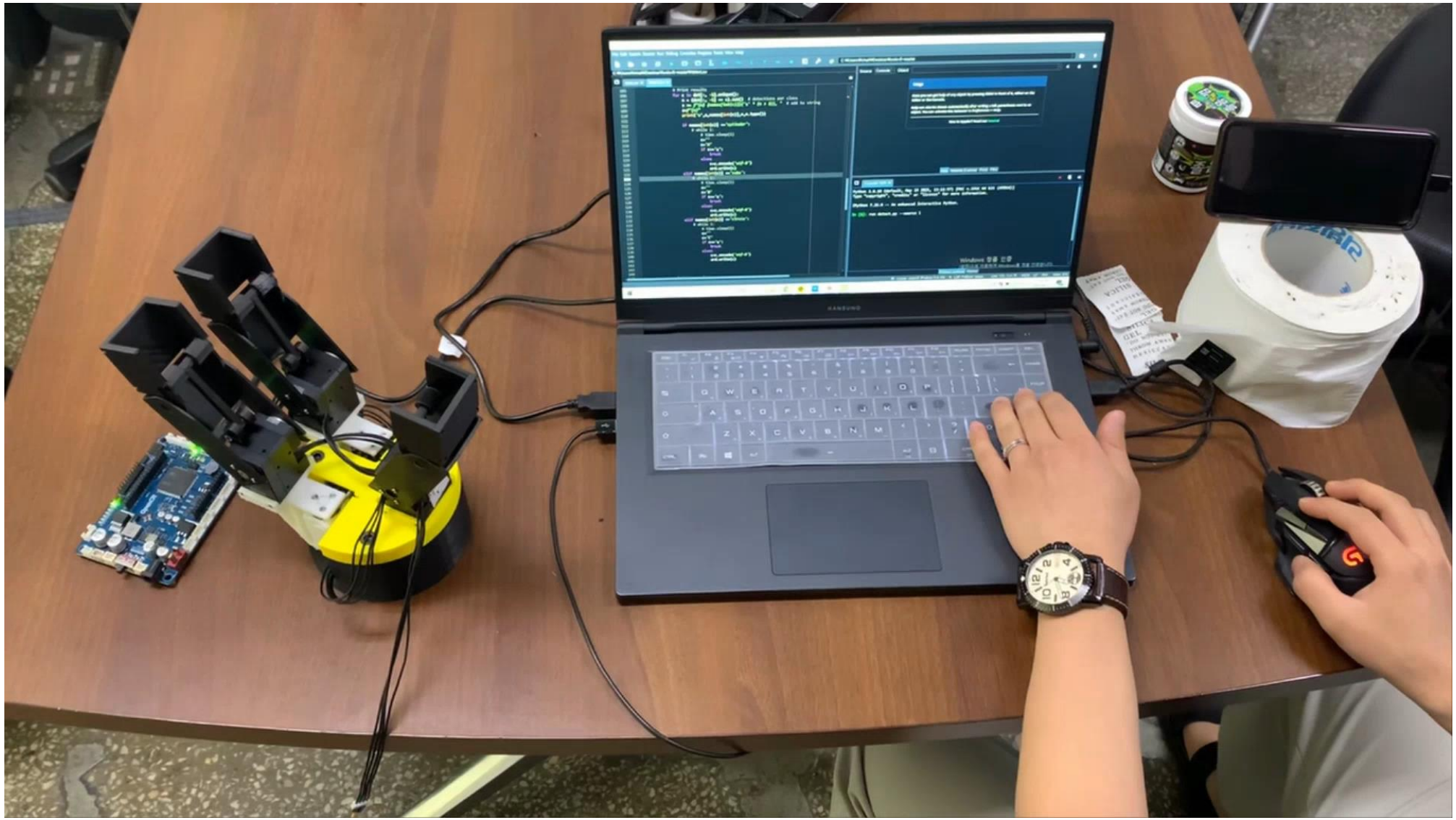


Part 2, Method

Cube



Cilynder



Part 3,

Conclusion &Future Work

- 4차 산업 혁명시대에 발맞추어 스마트 그리퍼가 필요
- 영상인식을 활용한 3지 그리퍼 제작
- 설계 – 핑거팁 추가
- PID제어를 이용한 토크 컨트롤을 활용하여 실제 object 잡기 수행
- 물체의 크기에 따른 그립 모션 차별화

감사합니다