

# 분리수거 로봇의 필요성

김영훈, 박선우, 안치형, 정예찬  
 송실대학교 전자정보공학부 전자공학과

kyh9561@naver.com, tjsdn7990@naver.com, anch728@gmail.com, yechani9703@naver.com

## Necessity of a Recycling Robot

Young-hoon Kim, Seon-woo Park, Chi-hyeong An, Ye-chan Jeong  
 Dept. of Electronic Engineering, Soong-Sil University

### 요 약

우리는 두 팔을 자유자재로 사용하여 수많은 일을 할 수 있다. 하지만 직접 하기보다 로봇을 이용하면 많은 시간을 단축할 수 있는 일들을 우리 주변에서 쉽게 찾아볼 수 있다. 대표적인 예로 세탁기, 전기밥솥, 청소기 등등 우리의 일을 대신해주는 로봇들이 많이 존재하지만, 아직 분리수거를 해주는 로봇은 실용화되어 있지 않다. 이에 우리는 분리수거를 대신해줄 수 있는 로봇을 구현해보고자 한다. 이는 작게는 편의를 위한 가정용 분리수거 로봇이 되기도 하지만, 크게는 분리수거가 잘 되어 있지 않은 분리수거장에서도 힘을 덜 들이고 쓰레기를 다시 분리할 수 있고 쓰레기양을 줄이는 데 이바지할 수 있을 것이다.

### 1. 배경

무분별한 쓰레기 사용을 통한 환경오염을 줄이기 위해 지구촌은 쓰레기 재활용을 하고 있다. 하지만, 상태에 따라 분리수거 기준이 전부 달라서 실제로 분리수거를 올바르게 하는 사람들은 많지 않다. 이러한 이유로 재활용품의 정확한 분리배출이 이루어지지 않아 세계적으로 재활용률이 낮은 것이 현실이다.

최근 전 세계적으로 소비량이 많아지는 추세를 보인 것에 더불어 코로나19 팬데믹 이후로 생활폐기물이 늘어나면서 전체적인 쓰레기양이 급증하였다. 결과적으로 쓰레기 처리에 대한 노동의 양 또한 늘어나게 되었다.

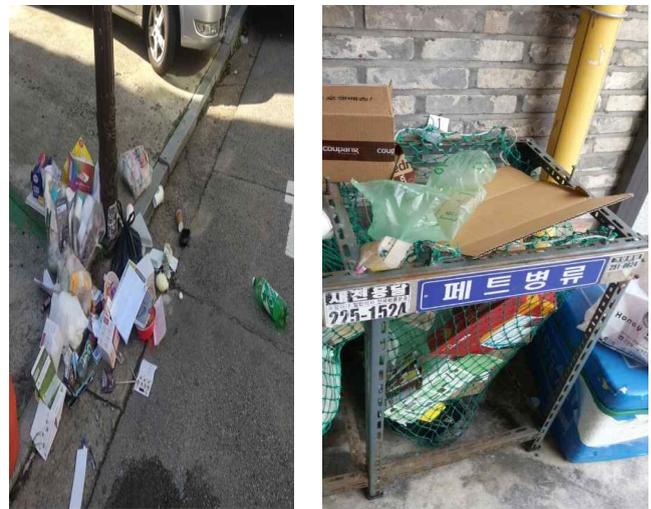
### 2. 로봇의 필요성

우리는 두 팔을 자유자재로 사용하여 수많은 일을 할 수 있지만, 로봇을 이용하면 많은 시간을 단축할 수 있는 일들을 우리 주변에서 쉽게 찾아볼 수 있다. 대표적으로 세탁기, 전기밥솥, 청소기 등등 우리의 일을 대신 해주는 로봇들이 많이 존재하지만, 분리수거를 해주는 로봇은 실용화되어 있지 않다.

실제로 지식의 부족 혹은 귀찮음으로 인해 분리수거가 잘 이루어지지 않고 있으며, 노약자나 장애인들은

일상생활을 하는데 불편한 점이 분명히 존재할 것이고 분리수거와 같이 번거로운 일들에 있어서 직접 하기에 일반인들보다 어려움이 있다.

이러한 로봇 기술들과 추가적인 S/W 기술을 병합하면 분리수거에 대한 지식이 많지 않아도 분리수거를 효율적으로 해낼 수 있을 뿐만 아니라 근래에 많이 부각이 되는 쓰레기 문제들과 더 나아가 환경문제 해결에도 도움이 될 것이다.



(그림 1) 무단투기와 분리수거의 현실

### 3. 적용 기술

#### 1) Arduino / Raspberry Pi

아두이노로 로봇팔(서보모터)과 레일(스텝모터) 제어 코드를 작성, 라즈베리파이에서 물체의 종류, 좌표등을 계산하여 시리얼 통신을 통해 아두이노로 전송한다.

#### 1) Labeling

Darknet으로 이미지를 학습시키기 전 학습 데이터가 되는 이미지의 전처리를 위한 소프트웨어이다. 각 이미지로부터 인식하려는 객체를 지정하고 Labeling 하여 학습에 필요한 텍스트 파일로 저장한다.

#### 2) YOLOv3

YOLO는 이미지를 한 번 보는 것으로 object의 종류와 위치를 추측하는 것으로 단일 네트워크를 통하여 여러 개의 테두리 상자에 대한 객체 확률을 계산하여 나타내는 기술이다.

YOLO를 이용하여 물체의 종류 판별에 가장 중요한 것은 판별의 기준이 되는 weights 파일이다. 이미 오픈소스로 주어진 weights 파일이 있지만, 우리 작품에 customizing된 파일이 필요했기 때문에 우리가 구별하고자 하는 캔, 플라스틱, 유리의 사진 여러 장을 가지고 머신러닝을 진행하여 weights 파일을 생성하였다.

#### 3) 머신러닝

기계학습의 한 종류인 머신러닝은 경험을 통해 자동으로 개선하는 컴퓨터 알고리즘이다. Labeling된 이미지 파일과 텍스트 파일을 이용해 사용자가 원하는 물체를 구분할 수 있도록 머신러닝을 계획하였고, 코드를 학습시키기 위한 도구로 Darknet을 사용했다. 다양한 방법을 통해 학습할 수 있지만, 여러 환경을 고려해 Colab에서 머신러닝을 진행했다. 러닝을 진행하면서 물체의 인식률이 가장 일반적이라고 판단되는 경우 학습은 종료되었다. 데이터 학습의 결과로 가중치 파일(weights 파일)을 얻을 수 있었다. 이 파일을 이용하여 라즈베리파이에 연결된 카메라 모듈을 통해 실시간 영상을 확인하고, 영상 속 물체를 인식하여 종류별로 구분할 수 있게 만들었다.

#### 4) Calibration

Calibration은 카메라를 통해서 물체를 볼 때 생기는

왜곡을 보정하는 과정으로 우리 작품에서는 Stereo Camera를 통해 물체의 이미지를 얻을 때 진행된다. 왜곡은 Depth를 포함한 3차원 공간을 계산할 때 실제와 다른 오차를 발생시키는 원인이 되기 때문에 Calibration이 요구된다.

이미지에 왜곡은 물체에서 반사된 빛이 카메라의 이미지 센서에 투영될 때, 렌즈의 형상, 수차, 주점 왜곡 등의 영향으로 상이 이미지 센서에 정확하게 맺히지 못하면서 발생한다. 왜곡 중 대표적으로 방사 왜곡(radial stortion)과 접선 왜곡(tangential distortion)이 존재한다.

방사 왜곡(radial distortion) : 반듯한 형상이 휘어지게 보이며, 이런 현상은 이미지의 중심으로부터 멀어질수록 심해진다.

접선 왜곡(tangential distortion): 이미지의 어떤 영역은 예상과 달리 물체가 더 가깝게 보이며, 원인은 카메라의 렌즈가 평면에 완전하게 수평이 아닌 이유로 발생한다.

왜곡 보정은 수식을 통하여 구현할 수 있다.

이론적으로 방사 왜곡(radial distortion)을 보정하는 수식은 다음과 같다.

$$\begin{aligned}x_{corrected} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\y_{corrected} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6)\end{aligned}$$

접선 왜곡(tangential distortion)을 보정하는 수식은 다음과 같다.

$$\begin{aligned}x_{corrected} &= x + [2p_1xy + p_2(r^2 + 2x^2)] \\y_{corrected} &= y + [p_1(r^2 + 2y^2) + 2p_2xy]\end{aligned}$$

이 두 값을 위해 사용되는 왜곡 보정 계수는 다음과 같다.

$$Distortion\ coefficients = (k_1, k_2, p_1, p_2, k_3)$$

우리가 사용한 Calibration은 OpenCV에 내장된 함수를 사용하여 코드 파일들을 생성하여 기능을 구현한다. Calibration\_image.py : 보정에 사용될 데이터를 위해 체스판과 같이 격자를 가진 물체의 이미지를 저장하는 파일

```
cv2.imwrite('images/stereoLeft/imageL' + str(num) + '.png', img_l)
cv2.imwrite('images/stereoright/imageR' + str(num) + '.png', img_r)
print("images saved!")
```

stereo\_calibration.py : Calibration\_image.py에서 얻은 이미지를 바탕으로 왜곡 보정 계수를 계산하여 이미지를

새롭게 mapping 하여 저장하는 기능을 하는 파일은 다음과 같다.

```
stereoLapl = cv.InitUndistortRectifyMap(newCameraMatrixL, distL, rectL, projMatrixL, grayL.shape[:1], cv.CV_16SC2)
stereoRapL = cv.InitUndistortRectifyMap(newCameraMatrixR, distR, rectR, projMatrixR, grayR.shape[:1], cv.CV_16SC2)

print("Saving parameters!")
cv_file = cv.FileStorage('stereoLapl.xml', cv.FILE_STORAGE_WRITE)

cv_file.write('stereoLapl_w', stereoLapl[0])
cv_file.write('stereoLapl_y', stereoLapl[1])
cv_file.write('stereoLapl_x', stereoLapl[0])
cv_file.write('stereoLapl_y', stereoLapl[1])
```

Calibration.py : stereo\_calibration.py에서 remapping한 파일을 불러들여 현재의 이미지에 적용하는 기능을 실행하기 위한 파일

```
def undistortRectify(frameR, frameL):
    # Undistort and rectify images
    undistortedL = cv2.remap(frameL, stereoMapL_x, stereoMapL_y, cv2.INTER_LANCZOS4, cv2.BORDER_CONSTANT, 0)
    undistortedR = cv2.remap(frameR, stereoMapR_x, stereoMapR_y, cv2.INTER_LANCZOS4, cv2.BORDER_CONSTANT, 0)

    return undistortedR, undistortedL
```

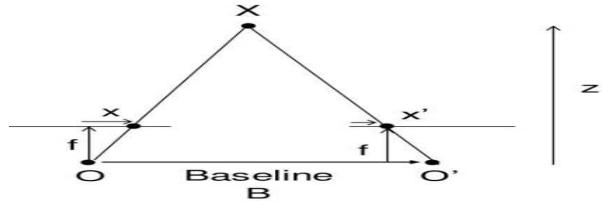
위의 파일을 사용하여 Calibration을 진행하여 이후 과정인 물체의 핵심 데이터로 거리를 측정하는 기능을 수행할 때, 계산한 거리와 실제 거리와의 오차를 줄여 Robot Arm이 물체에 정확하게 이동할 수 있게 하는 효과를 얻는다.

### 5) Triangulation(삼각측량)

우리는 로봇팔을 이용해 물체를 원하는 분리수거통으로 이동시킨다. 이때, 로봇팔은 자신이 집을 물체가 어디에 위치하였는지를 알아야 한다. 그래서 로봇팔과 물체 사이의 거리는 물체의 핵심 데이터로 간주하였다. 이때 현실 세계가 3D인 것과 달리, 카메라는 2D의 이미지를 보여주기 때문에 물체의 좌표는 간단히 구할 수 있으나, 좌표에 해당하는 Depth를 파악하지 못한다. 따라서 우리는 Depth를 측정할 수 있는 기능을 추가하기로 하였다.

Depth측정방법으로는 보편적인 방법인 Triangulation을 사용한다. Triangulation은 Epipolar Geometry에서 다루는 크게 다루는 개념 중 한 가지이다. Epipolar

Geometry은 한 사물 또는 장면에 대한 정보를 서로 다른 두 지점에서 획득하였을 때, 기하학적 관점으로 다루는 것을 의미한다.



Triangulation 방정식은 다음과 같이 얻어진다.

$$disparity = x - x' = \frac{Bf}{z}$$

x, x'은 서로 다른 지점에서 바라본 물체 X에 대한 각각의 x좌표이다. B는 이 Baseline으로 두 카메라 사이의 거리이다. f는 카메라의 초점거리이다. z는 카메라와의 Depth가 된다. 이처럼 Triangulation은 물체의 Depth를 구할 수 있기에 stereo vision에서 중요한 부분을 차지한다.

작품에서는 Triangulation을 구현하기 위해 함수로 구현하였다.

```
def find_depth(Lx, Ly, Rx, Ry, frame_right, frame_left, baseline, f, alpha):
    # CONVERT FOCAL LENGTH f FROM [mm] TO [pixel]:
    height_right, width_right, depth_right = frame_right.shape
    height_left, width_left, depth_left = frame_left.shape

    if width_right == width_left:
        f_pixel = (width_right * 0.5) / np.tan(alpha * 0.5 * np.pi/180)
    else:
        print('Left and right camera frames do not have the same pixel width')

    # CALCULATE THE DISPARITY:
    disparity = Lx - Rx #Displacement between left and right frames [pixels]

    # CALCULATE DEPTH x, y, z:
    X = round(float((baseline * (Lx+Rx)) / 2 * disparity), 2)
    Y = round(float((baseline * (Ly+Ry)) / 2 * disparity), 2)
    Z = round(float((baseline * f_pixel) / disparity), 2) #Depth in [cm]

    return X, Y, Z
```

물체의 서로 다른 관점을 수집하기 위해 Stereo Camera를 사용하였다.

Stereo Camera로 촬영된 좌우 화면 모두 YOLOv3를 통해서 물체의 x,y좌표를 추출하였다. 이를 바탕으로 물체의 3차원 좌표를 구하는데 이들의 실제 좌표를 더 정확하게 추출하기 위해 x ,y ,z를 모두 계산하였다.

계산을 위해 우리가 사용한 Stereo Camera의 스펙을 parameter로 받게 함수를 구성하였다.

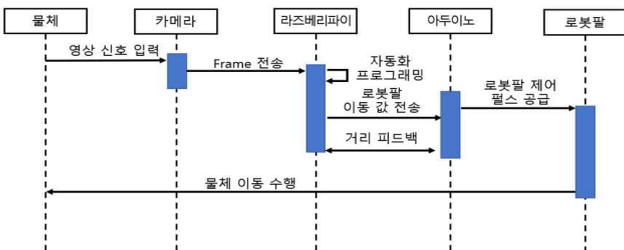
f\_pixel은 Triangulation 이론의 f를 구하는 방법으로 카메라로 들어온 것은 데이터의 해상도가 조절되어 들어온 것이기에 해상도 조절 전으로 돌리기 위해 계산을 해주는 과정이다. 좌우 화면에서 추출한 x좌표를 가지고 disparity를 도출하였다. 이를 바탕으로 분리수거 물체의 x, y, z 좌표를 소수점 첫 번째 자리까지만 계산하여 반환하는 함수로 구성하였다. 이렇게 구한 좌표를 가지고 피타고라스 정리를 이용하여 분리수거 할 물체와 Robot Arm 사이의 거리를 도출한다.

$$distance = \sqrt{(x_{Robot} - x_{Object})^2 + (y_{Robot} - y_{Object})^2 + (z_{Robot} - z_{Object})^2}$$

#### 4. 서비스 구성도 / 알고리즘



(그림 2) 서비스 구성도



(그림 3) 기능 흐름도



(그림 4) 알고리즘

- 알고리즘 -

- ① 사용자가 쓰레기통에 쓰레기통을 채운다.
- ② 카메라가 쓰레기통을 촬영하여 영상을 전송한다.
- ③ YOLOv3를 통해 분리수거가 가능할 물체를 뽑아낸다.
- ④ 인지된 물체의 좌표를 뽑아내어 거리를 계산하고, 물체에 대한 데이터를 저장한다.
- ⑤ 거리와 같은 정보를 기준으로 쓰레기통에서 꺼내는 순서를 계산하여 우선순위 변수에 저장한다.
- ⑥ 거리에 기반해서 레일이 이동하는 거리, 로봇팔이 움직이는 각도를 구한다.
- ⑦ 구한 행동데이터를 로봇팔을 움직이게 하는 아두이노에 전송한다.
- ⑧ 라즈베리파이에서 데이터가 왔는지 확인한다.
- ⑨ 라즈베리파이에서 온 데이터에 맞게 로봇팔이 행동한다.
- ⑩ 저장된 데이터와 실시간 영상을 가지고 분리수거를 더 수행할 수 있는지 판단한다.

#### 5. 기대 효과

- 로봇과 사람의 공존을 이루어낼 수 있다.
- 로봇의 스마트화에 도움이 되어 비용 절감 및 시스템 효율 향상을 기대할 수 있다.
- 영상인식, 원격제어 및 이동을 통해 다양한 활용이 가능하다.
- 유망시장 로봇 분야의 경쟁력을 확보할 수 있다.
- 국내 로봇 기술의 경쟁성 확보 및 로봇을 통한 혁신역량을 강화할 수 있다.
- 로봇팔을 활용하여 분리수거하는 제품을 제작함으로써 사람이 하던 일을 대신해서 할 수 있다.
- 사람이 일일이 조작할 필요 없이 자동화를 구축하여, 삶의 질을 운택하게 할 수 있을 것이다.

#### ○ 기존 서비스와의 차별성

-기존에 나온 자동 분리수거 기술들을 보면 쓰레기를 손으로 직접 넣고 그 쓰레기를 분리수거 한다. 우리가 만든 분리수거 로봇 팔은 쓰레기통에 쓰레기가 들어있으면 사람의 손길 필요 없이 자동으로 쓰레기통에 든 쓰레기들을 분리하게 된다.

-분리수거는 사람이 직접 하기에는 몸을 많이 움직이는 일이기 때문에 노약자나 장애인들에게 큰 도움을 줄 수 있다. 노약자나 장애인들이 아니더라도 요즘 시대에 분리수거에 신경을 쓰기 싫어하는 가구들이 많아, 상품성이 좋을 것이다.

-6DOF 로봇 팔을 사용함으로써 로봇 팔의 움직임이 제한되지 않아 자유롭고 넓은 동작을 할 수 있다.

#### ○ 자동화를 통한 생활 개선

현재 가정 내 또는 소규모 수준의 분리수거는 사람의 작업을 필요로 한다. 우리의 작품을 활용한다면 사람의 눈이 분류하던 작업을 영상처리를 통해서 대체 할 수 있고 로봇팔을 이용하여 사람이 분리수거를 알맞은 통에 넣는 과정을 대체하게 된다. 이를 통해 기존에 분리수거에 소비하던 시간을 자기 계발, 경제활동 등에 사용하여 삶의 질을 높이는 효과를 기대할 수 있다.