

## 캡스톤디자인(종합설계) 결과보고서

소속학부(과)	디지털콘텐츠공학과	팀명	지구지켜조		
개설 연도 및 학기	2022학년도 □1학기 □2학기	교과목명	캡스톤디자인2		
과제명	자동분리수거시스템				
과제유형	■기업연계형 캡스톤디자인	□기술이전형 캡스톤디자인			
희망금액	(기술이전금액)천원				
참여기업현황	기업	기업명	유)이티컴퍼니	소재지	익산
		사업자번호	262-88-00928	주요생산품목	소프트웨어 개발 및 공급
	담당자	성명	정영호	소속부서	대표
		H.P	010-2437-0335	E-mail	project307@naver.com

### 참여 학생 현황

구분	이름	학부(과)	학년	학번	H.P	E-mail
팀장	임소정	디지털콘텐츠공학과	4	20192875	010-7206-9901	aaas03012@naver.com
팀원1	박준하	디지털콘텐츠공학과	4	20192857	010-8728-0873	wmsgk0873@naver.com
팀원2	이환규	디지털콘텐츠공학과	4	20162925	010-6688-6207	free44556@naver.com
팀원3	임은서	디지털콘텐츠공학과	4	20183139	010-8597-2379	yeon1866@naver.com
팀원4						
팀원5						
팀원6						
팀원7						

집행경비내역	비목	집행내역	금액
	재료비	없음	0천원
	인쇄비	없음	0천원
	학생여비	자세히 작성	
	학생회의비	( )천원 × ( )인 × ( )회	0천원
			0천원
			0천원
	<b>총액</b>		

위와 같이 캡스톤디자인(종합설계) 결과보고서를 제출합니다.

첨부 : 캡스톤디자인(종합설계) 과제 상세 결과보고서[별첨 1호]

2022년 06월 10일

지원학생(팀장) 임소정

사업책임자(지도교수) 이현창

참여기업 담당자 정영호



원광대학교 LINC 3.0 사업단장 귀하

## 캡스톤디자인(종합설계) 상세 결과보고서

### 1-1 과제설계의 필요성

#### 1. AI(인공지능)이란?

- 가. AI는 'Artificial Intelligence'의 약자로 인공지능이라고도 함.
- 나. AI는 인간 지능을 모방하여 작업을 수행하고 수집한 정보를 기반으로 반복적으로 개선할 수 있는 시스템 또는 기계를 의미함.
- 다. 인공지능은 4차산업혁명인 시대에 '더 나은 삶'을 위한 필수조건으로 우리 삶에 깊숙이 자리를 잡고 있으며 수많은 발전단계를 거쳐왔음.
- 라. 현대 사회에서 AI(인공지능)는 중요한 분야 중이 하나임.
- 마. AI(인공지능)에 대해 긍정적인 견해와 부정적인 견해가 있음.



#### 1. AI(인공지능) 머신러닝(machine learning)

- 가. 4차 산업 혁명의 주역으로 떠오른 인공지능(artificial intelligence, 이하 'AI') 중 구조화 학습(deep structured learning) 기법을 사용함.
- 나. 머신러닝(machine learning) AI는 빅 데이터(Big Data)로 일컬어지는 통상적인 데이터베이스 소프트웨어가 보관하고 분석할 수 있는 한계를 넘는 데이터를 기반으로 하여 추상화 정도에 따라 하위 레벨부터 상위 레벨까지의 수직구조로 구성된 다수의 계층(layer)으로 구성된 소프트웨어임.
- 다. 각 하위 레벨의 데이터를 상위 레벨의 입력값으로 사용하며 각 하부를 구성하는 응용프로그램은 패턴분석과 분류 기능을 갖추고 최종적인 결과를 내는데 각 구성부분이 마치 인간 신경망(neural network)처럼 연결되어 최종 결과를 내기 전까지 반복적인 출력 데이터의 입력 데이터 전환 과정을 거치며 부분적 정보나 추상적 정보, 매개변수가 다양한 정보도 학습을 통해 정확한 추측 값을 내놓을 수 있음.
- 라. 여기에 선택의 범위를 제한하는 가치판단 선택 프로그램(tree program)을 더하면 인간의 전문적인 결정과 비슷한 결정을 빠른 시간 내에 수행하는 모습을 보여줌.
- 마. 머신러닝 AI 모델 중 알 수 없는 환경과 반응하며 그 단기적 성과를 장기적인 성과와 대비시켜

적응해가는 강화 학습모델(reinforcement learning)을 채용한 AI는 2016년과 2017년 알파고가 한국과 중국에서 벌인 바둑 관련 이벤트에서 보여주듯이 게임과 같이 반복하여 스스로 데이터를 쌓을 수 있는 환경에서는 인간이 생각해 내지 못한 방식으로 움직임으로써 인간의 지능에 의한 성과를 뛰어넘는 모습을 보여주었음.

### Traditional Programming



### Machine Learning



## 2. AI(인공지능) 딥러닝(deep learning)

가. 딥러닝은 머신러닝의 한 방법으로, 학습 과정 동안 인공 신경망으로서 예시 데이터에서 얻은 일반적인 규칙을 독립적으로 구축(훈련)함. 특히 머신 비전 분야에서 신경망은 일반적으로 데이터와 예제 데이터에 대한 사전 정의된 결과와 같은 지도 학습을 통해 학습함.

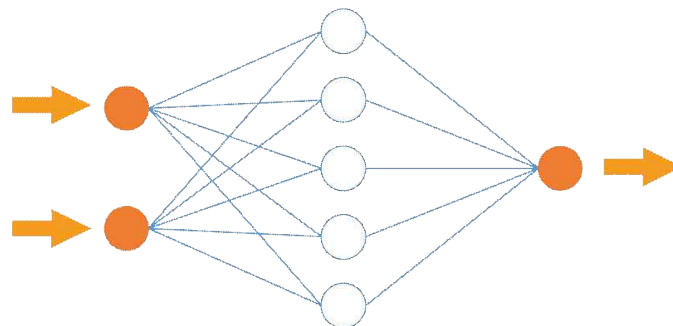
나. 딥러닝의 작동원리

### 1) 인공 신경망

가) 딥러닝은 특정 형식의 인공 신경망(ANN)을 사용하며, 우선 샘플 데이터를 통한 훈련 작업이 필요함. 그 이후에는 훈련된 인공 신경망(ANN)을 해당 작업에 사용할 수 있음.

나) 훈련된 인공 신경망(ANN)을 사용하는 것을 '추론'이라고 함.

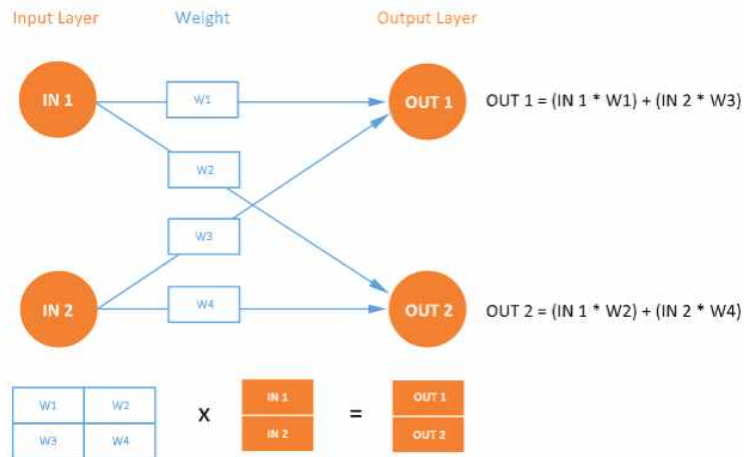
다) 추론이 진행되는 동안 인공 신경망(ANN)은 학습된 규칙에 따라 제공된 데이터에 대한 평가 결과를 다시 보고함. 이러한 평가 결과는 입력 이미지에 결함이 있는지 또는 오류가 없는 객체를 나타내는지에 대한 추정 등이 될 수 있음.



### 2) 뉴런, 계층 및 연결

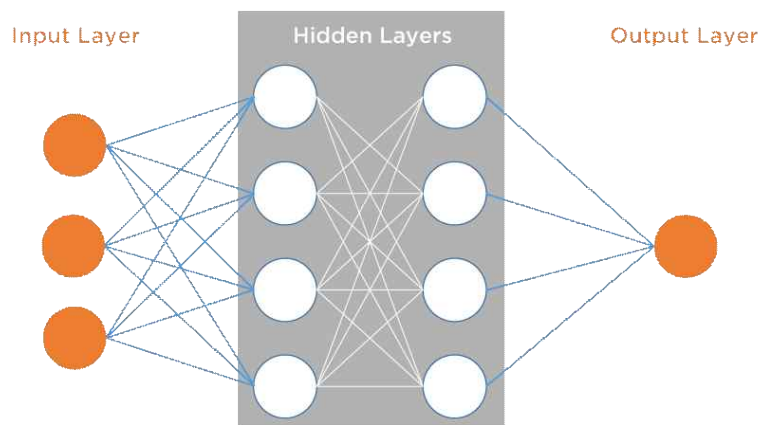
가) 인공 신경망(ANN)은 서로 연결된 '뉴런' 계층으로 구성됨

- 나) 가장 간단한 경우, 이 계층은 입력 계층과 출력 계층으로 구성됨
- 다) 뉴런과 연결(link)은 매트릭스에 비유할 수 있음
- 라) 링크 매트릭스는 입력 매트릭스 개별 값과 결과 매트릭스의 값 사이의 연결을 포함함.
- 마) 연결 매트릭스의 값에는 각 연결의 가중치가 포함됨.
- 바) 입력값과 논리 매트릭스의 값에 가중치를 반영하면 결과 매트릭스의 개별 값이 생성됨.



### 3) 심층 인공 신경망

- 가) 딥러닝이라는 용어는 소위 '심층' 인공 신경망(ANN)에 대한 훈련을 의미함. 이 인공 신경망(ANN)은 입력 및 출력 계층뿐만 아니라 입력 및 출력을 위한 가시적인 계층 사이에 존재하는 수백 개의 추가적인 '숨겨진' 계층으로 구성됨.
- 나) 숨겨진 계층의 결과 매트릭스는 다음 계층의 입력 매트릭스로 사용됨. 이 경우에는 마지막 계층의 출력 매트릭스에만 결과가 포함됨.

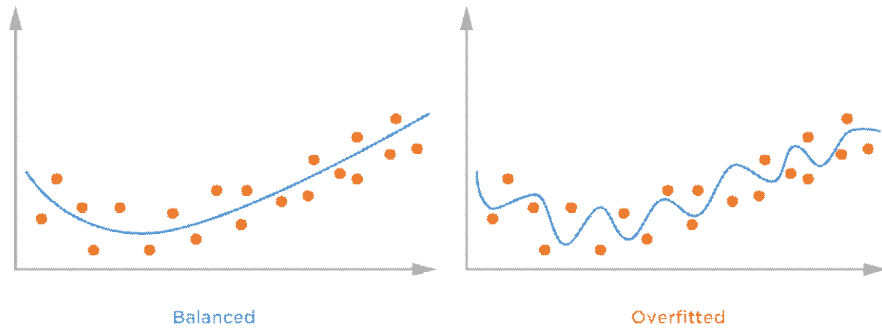


#### 1) 훈련

- 가) 인공 신경망(ANN)을 훈련할 때 초기 초점은 무작위로 설정됨. 후에 샘플 데이터가 서서히 추가됨
- 나) 학습 규칙은 입력 데이터 및 예상 결과에 따라 관계의 가중치를 조정하는데 사용됨.
- 다) 결과에 대한 평가의 정확성을 의미하는 인공 신경망(ANN)의 궁극적인 효과는 훈련에서 사용되는 예시 데이터에 큰 영향을 받음.

라) 일반적으로 훈련 내용에 변동성이 높은 예시 데이터가 많이 포함될수록 추론에서 더 정확한 결과를 얻을 수 있음.

마) 매우 유사하거나 반복적인 데이터를 사용하여 훈련을 수행하는 경우, 인공 신경망(ANN)은 예시 데이터와 다른 분야의 데이터를 추정할 수 없게 됨. 이 경우 인공 신경망(ANN)이 '과적합(overfit)'하다고 표현함.



#### 나. 딥러닝의 활용 용도

- 1) 딥러닝의 적용 영역은 매우 다양함.
- 2) 특히 머신 비전 분야에서 딥러닝은 매우 다양한 작업에 널리 사용되는 방법임.
- 3) 시각 분야에서 딥러닝에 맞는 가장 일반적인 작업은 이미지 데이터의 분류와 세분화를 위한 이미지 분석 작업임.

#### 4) 이미지 분류

- (1) 이미지 분류 작업에서는 이미지를 결함의 있는 구성 요소와 정상적인 구성 요소로 분리한 후 결함의 종류에 따라 정렬하거나 빈 이미지를 다른 카테고리에 할당하는 등 이미지들이 서로 다른 클래스에 할당됨.
- (2) 쿠키 제조 과정에서 쿠키에 아무런 문제가 발생하지 않았는지 또는 쿠키의 일부가 부서졌는지를 확인하는 작업이 이뤄지는 것과 유사함.

#### 5) 이미지 세분화 및 오브젝트 인식

- (1) 이미지 세분화 작업은 이미지의 각 픽셀을 하나의 클래스로 할당함. 이를 통해 이미지에서 여러 가지 오브젝트를 식별할 수 있음.
- (3) 예를 들어 쇼핑 바구니에서 서로 다른 과일 조각을 식별하거나 교통 표지판, 도로 및 사람을 구별할 수 있음.
- (4) 이미지 세분화를 통해 어셈블리 라인에서 원형, 정사각형, 육각형 등 어떤 모양의 쿠키가 운반되는지 등을 확인할 수 있음.

### 3. AI(인공지능)의 응용 분야

가. AI(인공지능)의 주요 응용 분야는 크게 6가지로 구분할 수 있음.

나. 전문가 시스템, 자연어처리, 데이터 마이닝, 음성인식, 컴퓨터 비전, 지능로봇 등의 영역에서 인

공지능이 활발하게 응용됨.

1) 전문가 시스템

가) 특정 분야에 대한 전문지식을 체계화해서 시스템화하고, 어떤 문제가 발생해서 해결책이 필요하면 인공지능이 인간을 대신해서 해법을 제공하는 시스템을 말함.

2) 자연어처리

가) 사람이 사용하는 일반언어로 작성된 문서를 처리하고 이해하는 분야임.

3) 데이터 마이닝

가) 방대한 데이터에서 유용한 정보를 캐내는걸 말함.

나) 대표적인 예로는 암호화폐가 있음.

4) 음성인식

가) 사람의 음성언어를 컴퓨터가 해석해서 그 내용을 문자데이터로 보여주는 것을 말함.

5) 컴퓨터 비전

가) 기계의 시각에 해당하는 부분을 연구하는 기술을 말하는데 디지털 영상을 처리하거나 움직이는 물체를 인식해서 컴퓨터가 그 결과값을 도출해내는 것을 의미함.

6) 지능로봇

가) 로봇에 관련된 기술 분야를 뜻하는데, 인공지능 기술을 활용해서 인간이 해야 할 영역을 로봇이 대신하는 것을 의미함.

1. 전 세계 AI(인공지능) 시장 동향

가. 전 세계에서 인공지능 시장이 빠르게 성장하고 있음.

나. 한국IDC<sup>1)</sup>의 보고서에 따르면 2022년 전 세계의 인공지능 매출이 전년 대비 18.8% 증가할 것으로 전망하였고, 2024년에는 5000억달러(583조1000억원)을 돌파할 것으로 전망하였음.

다. IDC는 AI 시장의 성장세가 모든 IT 관련 산업 분야에 걸쳐 계속될 것으로 예측함.

라. IDC는 AI 시장을 소프트웨어, 하드웨어 및 서비스 시장으로 구분했음.

마. AI 소프트웨어 시장이 전체 AI 시장의 88%에 달하지만, 성장률 측면에서는 AI 하드웨어 시장이 향후 몇 년간 가장 빠른 성장을 보일 것으로 예상됨.

바. 2023년부터는 AI 서비스 시장이 가장 빠르게 성장할 전망이다.

사. 세부 시장별로 살펴보면 AI 소프트웨어 시장 내에서 AI 애플리케이션이 매출의 50%에 가까운 점유율을 차지함.

아. 5년간 연평균 성장률(CAGR) 측면에서는 AI 플랫폼이 33.2%로 가장 높은 성장률을 기록할 전망이다.

자. AI 시스템 인프라스트럭처 소프트웨어가 14.4%로 가장 느린 성장률을 보일 것으로 예상되지만, 전체 AI 소프트웨어 매출의 약 35%를 차지할 것으로 분석됐음.

차. AI 애플리케이션 시장 내에서는 AI ERM이 AI CRM보다 향후 5년간 소폭 성장할 예정임.

카. AI 라이프사이클 소프트웨어는 AI 플랫폼 시장 내에서 가장 빠르게 성장할 것으로 보임.

타. AI 서비스 시장은 2020년 194억달러(22조6000억원)를 기록하며 하드웨어와 소프트웨어를 제치고 가장 가파른 성장을 보였음.

1) 2021년에는 19.3% 성장이 예상되며 향후 5년간 연평균 21%의 성장률을 유지하며 계속해서 성장할 전망이다.

2) AI 서비스 시장은 IT 서비스와 비즈니스 서비스 두 부문으로 나뉘는데, IT서비스는 전체 AI 서비스 매출의 80%쯤을 차지할 정도로 규모가 큼. 두 시장 모두 5년간 21% 성장할 것으로 보임.

3) IDC는 AI 서비스 시장이 2025년 500억달러(58조3000억원) 규모의 시장이 될 것으로 내다봄.

가. AI 하드웨어 시장은 전체 AI 시장의 5% 정도로 가장 작은 시장이지만 2021년에는 전년 대비 29.6% 성장을 기록하며 가장 빠르게 성장할 시장으로 주목받았음.

1) 해당 시장은 2022년에도 성장률이 가장 높을 것으로 예상되며, 향후 5년간의 연평균 성장률 19.4%에 이를 것으로 추정된다.

나. 하드웨어 시장은 서버와 스토리지 두 개로 나뉨.

1) IDC는 서버 시장이 약 82%로 더 높은 점유율을 보이고 있지만, 스토리지 시장이 5년간 연평균 22.1%의 성장률을 유지하면서 더 높은 성장세를 보일 것으로 전망함.

파. 리투 조티 IDC의 AI 및 오토메이션 리서치 부서의 부사장은 "2020년은 디지털 트랜스포메이션이 한 층 가속화되고 기업 AI의 가치가 강화한 해였으며, 급격한 변화는 불안하지만, 혁신과 발전의 촉매 역할을 하기도 한다"며 "빠르게 상황을 감지·예측·대응 및 적응할 수 있는 AI 솔루션의 개발 및 활용은 이제 비즈니스의 필수 과제가 됐다"고 말했음.

하. 김경민 한국 IDC 수석연구원은 "국내 AI시장도 글로벌과 마찬가지로 빠르게 성장하고 있다"며 "기업의 모든 비즈니스 의사결정 과정에 AI가 활용되면서 내부 직원, 외부 파트너, 그리고 최종 소비자단의 경험 향상에 적용되는 과정에서 전반적인 비즈니스 생태계 혁신을 이끄는 필수 기술이 되고 있다"고 말했음.

## 2. 국내 AI 시장(인공지능) 동향

가. 한국IDC는 최근 발간한 '국내 인공지능 2019-2023 시장 전망' 연구 보고서에서 국내 인공지능 시장이 향후 5년간 연평균 17.8% 성장해 2023년에 6400억원 이상의 규모를 형성할 것이라고 밝혔음.

나. IDC 해당 보고서를 통해 인공지능 전체시장에서 가장 큰 매출을 차지하는 하드웨어 시장이 2022년 이후부터는 인공지능 구축을 위한 컨설팅 및 개발 관련 서비스 시장보다 작은 비중을 차지할 것으로 예상함.

다. 반면, 업무 프로세스 효율화 및 비즈니스 자동화를 위한 AI 애플리케이션 및 플랫폼 구현 사업이 크게 증가하며 서비스와 소프트웨어 시장 모두 향후 5년간 연평균 30% 이상의 성장률을 보일 것으로 전망함.

라. AI(인공지능) 기술의 도입은 단순 업무 효율화뿐만 아니라 사람이 해결하기 힘든 복잡한 현상의 데이터 인사이트를 발굴하고 즉각적인 비즈니스 의사결정과 미래 예측 및 추천 등에 활용되어 막강한 효과를 창출할 수 있다고 강조함.

마. 국내 AI(인공지능) 분야는 기술 및 인력 모든 측면에서 글로벌 시장에서 후발주자이지만, 국내의 높은 ICT 인프라 및 교육 수준과 신기술에 대한 빠른 수용성을 고려해 보았을 때 공공 및 기업의 과감한 투자와 도전을 통해 빠르게 성장할 수 있다고 전망하였음.

## 3. 분리수거의 필요성

가. 분리수거의 필요성 개요

- 1) 폐기물은 우리의 생활 또는 산업 활동에 의해 발생된 필요 없게 된 물질로써 고철, 폐유, 폐합성수지 등 고체상 또는 액상으로 발생하게 됨.
- 2) 환경부 조사에 따르면 대한민국 국민은 일생동안 평균 55톤의 쓰레기를 배출한다고 함.
- 3) 특히 현대의 경제 규모의 확대와 산업구조의 고도화 및 소비자들의 다양한 구매욕들로 폐기물량이 급격히 증가되었음.
- 4) 편리함을 얻기 위해 사용되는 플라스틱 등의 제품 등이 난분해성이라 폐기물 처리에 곤란함을 겪고 있음.
- 5) 대량생산과 대량 공급체제에 따라 이러한 난분해성 폐기물들의 양이 기하급수적으로 늘어나고 있음.
- 6) 폐기물도 귀중한 자원이 될 수 있음.
  - 가) 유리병의 재사용(재활용), 알루미늄캔과 페트병의 재활용 등은 폐기물을 줄임과 동시에 재사용을 할 수 있어 매우 좋은 예시임.
  - 나) 유리병과 알루미늄 캔의 재사용은 경제적 가치가 가장 높음.
  - 다) 폐기물 오염을 방지하는 가장 첫 번째 길은 절약으로 발생 자체를 줄이는 것과 철저한 분리수거로 재사용, 재활용 자원을 늘려가는 것임.

4. 국내 플라스틱 분리수거 실태

가. 대한민국 가정과 사무실 등에서 배출하는 쓰레기는 연간 323만 톤임.

- 1) 종량제 봉투에 담겨버려지는 178만 톤(55.1%)은 소각 또는 매립됨.
- 2) 나머지 145(44.9%)만이 재활용품으로 분리수거 됨.

나. 재활용품의 재활용 실태를 파악해 보면 현재 가장 문제로 지적되는 플라스틱을 예로 들었을 때 환경부의 조사에 따르면 선별장을 거쳐 재활용되는 플라스틱은 분리수거 된 물량의 53.6%(연 77만 6000톤)에 그쳤다고 함.

- 1) 이마저도 고형연료제품(SRF)을 제외하면 실제로 재활용되는 양은 총 44만 6000톤임.

다. 잘 분리배출 된 플라스틱 중 30.7%만이 다시 플라스틱으로 재사용됨.



가. 가정에서 잘 분리수거된 플라스틱들은 운송과정에서 운송방법 등의 문제로 분리된 쓰레기들이 다시 뒤섞이고 분리수거물량의 30%는 다시 소각장으로 가게 됨.

- 1) 그 외 선별장에 도착한 플라스틱들은 사람의 손으로 일정량 채 분리를 하지만 한계가 있기 때문에 완전하게 분리가 되지는 않는다고 함.
- 2) 특히 일회용 플라스틱 같은 경우 플라스틱의 종류(플라스틱의 세부 종류)가 다양하고 그것을 사람의 눈으로 구별하는 것이 어렵기 때문에 재활용이 불가능하다고 함.
- 3) 이러한 이유로 선별장에 온 물량 중 대략 30% 이상은 재활용을 할 수 없게 됨.
- 4) 환경부 조사에 따르면 재활용품으로 잘 분리수거 된 페플리스 틱크, 페비닐의 46.4%가 선별장 또는 재활용업체에서 다시 쓰레기로 버려진다고 함.

## 5. 초음파 센서

가. 초음파라는 용어는 인간의 청력 범위(20kHz)를 벗어난 오디오 주파수를 나타냄.

나. 가청영역 이상의 모든 소리는 그 주파수 대역이 매우 넓으므로 초음파는 중심 주파수의 크기에 따라 각각 다른 용도로 응용될 수 있음.

다. 초음파의 예로는 박쥐나 돌고래 등이 통신이나 거리 감지를 위해 발생시키는 것과 산부인과에서 태아의 건강 상태를 검사하기 위해 사용되는 의료 진단기 등에서 찾아볼 수 있음.

라. 초음파는 알게 모르게 우리 주위에 널리 사용이 되고 있는데, 최근의 급격한 기계와 전자공학의 발달로 인해 그 응용범위 또한 날로 넓어지고 있음.

마. 산업이 발달함에 따라 각종 산업에서, 주어진 일 및 서비스를 하기 위해 지정된 일을 하는 로봇이 점점 증가하는 추세임.

- 1) 그중 물건을 분리하는 로봇은 물건을 정확히 분리해야 함.
- 2) 물건을 감지하기 위한 센서로 가격이 저렴하고 성능이 우수한 초음파 센서를 많이 사용함.

바. 초음파의 발생 기구는 크게 트랜스듀서(electromrchanical)와 고주파 전원(high frequency)으로 구성이 됨.

사. 현재 사용되고 있는 트랜스듀서들의 종류들로는 압전 센서, 전자기 센서, 그리고 Lase Device 등이 있음.

- 1) 특수한 목적일 경우를 제외하고는 우수한 초음파 발진, 수신 효율과 구조의 간단함, 그리고 작은 size를 이룰 수 있는 압전소자를 이용한 압전 센서가 사용되고 있음.
- 2) 이들 재료에는 공급되는 전자기장의 크기에 의해 발생하는 초음파의 형태 및 크기를 쉽게 조절할 수 있다는 장점이 있음.

## 가. 초음파 센서 원리

- 1) 초음파의 기본적인 원리는 비접촉식 방식으로 물체를 감지하거나 초음파 센서로부터의 거리를 측정함.
- 2) 초음파 센서의 송신부(Trig)에서 일정한 시간의 간격을 둔 짧은, 초음파 펄스를 방사하고, 대상물에 부딪혀 돌아온 에코 신호를 수신부(Echo)에서 받아, 이에 대한 시간차를 기반으로 거리를 산출함.

3) 1cm를 이동하는데 걸리는 시간은 다음과 같이 구할 수 있음.

$$t = \frac{2 \times L(\text{물체와의 거리 } m)}{Vs(\text{음속 } m/s)}$$

$t$  : 신호가 되돌아올 때 까지 걸리는 시간(s)

4) 초음파 센서는 모양이나 색상에 상관없이 나무, 메탈 및 플라스틱으로 제작된 물체를 감지할 수 있다고 함.

가) 이러한 물체는 고체, 액체 또는 분말 상태로 될 수 있으나 반드시 음파를 반사할 수 있어야 함.

나) 감지 거리를 감소할 수 있는 물체도 있는데, 대형의 매끈하거나 기울어진 표면을 갖추거나 다공질 물체, 예를 들면 펠트, 모직품 또는 스펀지 고무 등이 포함됨.

5) 초음파 센서의 측정 거리는 물체의 표면 속성 및 각도에 의하여 결정됨.

6) 플랫폼 표면(표준 반사판)을 갖추며 정확히 센서 축에 수직으로 위치한 물체를 감지할 때, 감지 거리가 가장 큼.

7) 아주 작은 물체 또는 부분적으로 음파 방향을 돌릴 수 있는 물체는 감지 거리를 줄임.

8) 매끈한 표면을 가진 물체는 가능한 센서와 90° 위치로 설치되어야 함.

9) 굴곡 표면의 물체는 설치 각도 편차를 크게 할 수 있음.

10) 초음파 센서를 사용하는 주된 이유는 다양한 재질 및 표면을 인식할 수 있는 이들의 특성임. 색상에 상관없이 고체, 액체 및 분말 등 재질의 물체를 감지함.

11) 물체 표면 특성 또한 감지 안정성에 영향이 없음.

12) Pepperl+Fuchs의 고품질 초음파 센서는 거친/매끄러운/고광택/투명한/어지러운/습한 또는 건조한 표면 전부를 안정적으로 감지한다고 함.

나. 초음파 센서 응용 사례

가) 드론

(1) 송신부와 수신부, 두 부분을 갖춘 센서를 통해 송신부에서 초음파를 쏘면 수신부에서 수신 후 거리를 측정하여 고도를 유지하도록 돕는 원리임. 이 과정에서 초음파는 GPS와 협력해 안정성을 극대화 함.

나) 로봇 청소기

(1) 로봇 청소기에 부착된 다수의 초음파 센서를 일정 시간 동안 두 개씩 병용하여 운용함으로써, 목표물까지의 거리를 정확하게 측정할 수 있도록 함.

다) 초음파 주차 유도 시스템

(1) 주차면에 설치된 초음파 센서가 각각의 설정값을 가지고 순서를 정하여 초음파를 발산하고 수신을 받는 방식으로 초음파 신호를 발산할 때만 반사되는 신호를 받아 동작함.

## 1-2 선행연구 및 제품 관련 자료조사

### 가. 네프론 쓰레기통

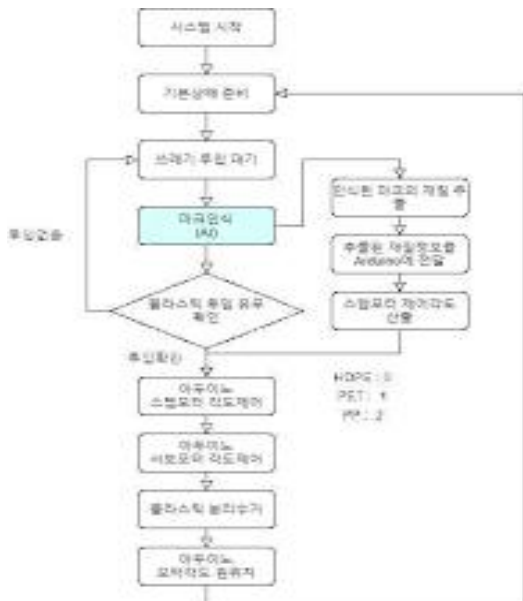
- 1) '뉴로지니'라는 인공지능(AI) 기술, 뉴로지니는 비전(Vision) AI 기술임.
- 2) 자원이 새롭게 투입될 때마다 딥러닝으로 끊임없이 자원이 무엇인지 학습함. 이렇게 모은 자원의 이미지는 빅데이터로 구축되어 있어서 외형이나 바코드가 훼손되어도 종류를 구분할 수 있음.

### 나. AI 자원순환선별로봇 '에이트론'

- 1) 자율주행 자동차의 객체 인식에서 영감을 받아 재활용 폐기물의 종류와 특징을 인식하여 자동으로 분류하는 인공지능 로봇임.

다. 영상 처리 기술을 이용한 재활용 분리기

- 1) 노트북과 PLC를 연결하여 전체 시스템을 제어하면서 근접센서와 포토 센서를 이용해서 병의 높이를 측정해 크기별로 분류함.
- 2) 물체 인식기는 물건을 영상 인식하는 라즈베리 파이와 카메라 모듈, 물건의 무게를 재는 무게 센서, 물체 인식기에서 물체 분류기로 물체를 이동시켜주는 컨베이어벨트와 서보모터로 구성되어 있음.
- 3) 해당 검출 데이터는 라즈베리파이(Raspberry Pi)를 통해 아두이노(Arduino)를 거쳐 데이터베이스에 저장됨.
- 4) 물체 분류기는 물체가 들어오면 서보모터를 통해 물체에 해당하는 보관함으로 자동으로 분류해 주고, 초음파 센서를 통해 보관함이 꽉 찼는지 등의 여부를 알려 줌.



라. AI와 로봇을 물자 재활용 컨베이어 라인에 접목한 AMP 로봇틱스

- 1) AMP 코텍스(AMP Cortex)시스템은 인공지능(AI) 학습을 바탕으로 혼합된 자원 재활용 공장의 컨베이어 라인에서 뒤 섞여 있는 재활용품 식별과 분류 작업을 자동화함.
- 2) AMP 뉴런 AI(AMP Neuron AI) 플랫폼이 다양한 색상, 질감, 모양, 크기, 패턴, 심지어는 브랜드 라벨까지 인식해 소재와 재활용성을 파악해 가면서 지속적으로 자체 훈련함.
- 3) 뉴런은 재활용될 물질을 선별해 옮겨서 재배치하도록 안내함.

마. 닥터 비(B) 인공지능 로봇

- 1) 딥러닝을 통한 다층 신경망과 AI 비전 시스템을 활용하여 사람의 판단 방식과 유사하게 객체를 식별하고 컨베이어 벨트 위의 거의 모든 재료를 실시간으로 인식함.
- 2) 종이, 비닐, 알루미늄 등 58가지 항목을 구분할 수 있고 폐기물의 모양과 색깔을 감지하여 숨어

냄.

3) 딥러닝 기술을 장착해서 폐기물을 접할수록 선별 능력도 올라감.

### 1-3 과제설계의 목표

- 가. AI(인공지능)의 이미지 인식 기술을 통해 쓰레기를 분리수거 할 수 있도록 머신러닝과 딥러닝을 함.
- 가. 레고마인드스톰(EV3)을 이용하여 자동분리수거를 할 수 있는 장치를 만들.
- 가. 아두이노의 여러 센서를 활용하여 완성도를 높임.

### 1-4 현실적 제한 요건

- 새로운 지식을 습득하고 완벽하게 구현하기에는 짧은 시간이었음.
- 라즈베리파이와 레고마인드스톰과의 연결에서 자료가 부족했기 때문에 구현하지 못함.
- 학과에서 제공하는 레고마인드스톰 재료로는 완벽하게 구현하기 어려웠음.

### 1-5 작품의 특징 및 기대효과

- 가. 모터를 활용하여 컨베이어 벨트를 구현.
- 가. 카메라나 센서를 통해 쓰레기를 인식하여 종류별로 분리수거 할 수 있게 함.
- 가. 인식한 쓰레기를 컨베이어 벨트가 움직이는 속도나 시간을 고려하여 알맞은 쓰레기통에 들어갈 수 있게 함.
- 가. 쓰레기는 쓰레기통이 있는 앞으로 밀어서 들어갈 수 있게 함.
- 가. 쓰레기통에 쓰레기가 어느 정도 차면 센서를 이용해 경고음이나 불빛이 깜빡거리게 만들어 쓰레기통 교체할 수 있게 함.

### 기대효과

- AI를 활용했기 때문에 정확하게 쓰레기를 분리수거 할 수 있음
- 올바른 분리수거로 재활용을 하여 에너지 자원을 절약할 수 있음
- 올바르게 분리된 쓰레기는 소각이나 매립 시에 비용을 줄일 수 있어 경제적이고 매립지의 수명도 연장됨
- 올바른 분리수거는 대기 오염 배출물질을 많이 줄일 수 있어 환경 오염도 줄일 수 있음
- 사람들이 헛갈려 하는 부분도 자동 분리수거 시스템을 통해 좀 더 분리수거에 대한 지식을 알릴

수 있음

## 2-1 문제 정의 및 아이디어 스케치

요즘 환경에 대한 관심이 높아짐에 따라 '분리수거'의 중요성이 높아지고 있다. 하지만 사람들이 분리수거를 잘하지 못하는 경우가 발생한다. 그래서 분리수거를 정확하게 하기 위해 인공지능을 활용하여 자동으로 분리수거를 할 수 있는 시스템을 개발하고자 한다.

## 2-2 개념설계

- 가. 아두이노 센서를 이용해 분리수거 된 쓰레기의 쌓임 정도를 확인함. 만약 쓰레기가 가득 찼다면 경고음을 내거나 불빛을 깜빡이게 함.
- 가. 레고마인드스톰을 활용하여 컨베이어 벨트를 구현함. 모터를 이용해 컨베이어 벨트가 작동하게 할 예정임.
- 가. AI(인공지능)의 이미지 인식 기술을 활용하여 쓰레기의 종류를 파악함. 이때 카메라나 센서를 통해 인식하도록 함.

## 2-3 설계 제작 과정

### • 카메라

- 어떤 카메라여도 상관x → USB 꽂으면 인식
- 연동 문제 → 라즈베리파이와 카메라 서로 연동(휴대폰 카메라 가능(?))
- ↳ 관련 서적은 많이 없지만, 인터넷에 많이 나옴
- 라즈베리파이 구매사이트 → 디바이스마트(<https://www.devicemart.co.kr/main/index>)

\* 라즈베리파이에 카메라 연결하는 방법 :

<https://blog.naver.com/naininfo/221563847847>

<http://www.3demp.com/community/boardDetails.php?cbID=233>

<https://rudalskim.tistory.com/275>

<https://neosarchizo.gitbooks.io/raspberrypiforsejonguniv/content/chapter2.html>

### • 프로그램

- 주피터 노트북(Jupyter Notebook) or 스파이더(spyder) ⇒ 파이썬(Python) 기반 프로그램
- ↳ 어떤 코드를(쉬운 코드) 쓸지에 따라 프로그램 결정

- 깊게 공부하지 말고 대략적으로 특징만 분석

YOLO(You only Look Once)	텐서플로우(Tensorflow)
<p>&lt;특징&gt;</p> <ul style="list-style-type: none"> <li>· 이미지를 여러장 분할하지 않고 한번만 봄.</li> <li>· 실시간으로 객체를 탐지할 수 있음.</li> </ul> <p>&lt;장점&gt;</p> <ul style="list-style-type: none"> <li>· 간단한 처리과정으로 속도가 빠름.</li> <li>· 모든 학습 과정이 이미지 전체를 통해 일어나기 때문에 단일 대상의 특징뿐 아니라 이미지 전체의 맥락을 학습.</li> </ul> <p>&lt;단점&gt;</p> <ul style="list-style-type: none"> <li>· 상대적으로 작은 개체들이 한곳에 모여있는 경우, 정확도가 낮음.</li> </ul>	<p>&lt;특징&gt;</p> <ul style="list-style-type: none"> <li>· 현재 가장 범용적으로 사용되는 딥러닝 프레임 워크임.</li> <li>· 파이썬 언어를 활용함.</li> <li>· 구글에서 공개한 머신러닝 라이브러리임 (전문성).</li> </ul> <p>&lt;장점&gt;</p> <ul style="list-style-type: none"> <li>· 알고리즘 구현의 세부적인 면은 텐서플로우가 알아서 처리함(추상화).</li> <li>· 파이썬과 C++를 지원함.</li> <li>· 구글에서 공식 배포하였기 때문에 레퍼런스 및 전문성이 보장됨.</li> </ul> <p>&lt;단점&gt;</p> <ul style="list-style-type: none"> <li>· 메모리를 효율적으로 사용하지 못함.</li> </ul>

\* 라즈베리파이를 이용하여 얼굴 or 사물 인식하기 :

★① Open CV + Dlib 툴킷 사용(인물 인식)

<https://blog.naver.com/PostView.nhn?isHttpsRedirect=true&blogId=naininfo&logNo=221563862456>

② 라즈베리파이 기반 딥러닝 객체 인식 개발 방법(Open CV + YOLO 알고리즘)

<http://daddynkidsmakers.blogspot.com/2019/01/blog-post.html>

★③ TensorFlow Lite + Open CV(사물 인식)

<https://seo-dh-elec.tistory.com/32>

<https://blog.naver.com/PostView.nhn?blogId=cosmosjs&logNo=222039217564&parentCategoryNo=&categoryNo=56&viewDate=&isShowPopularPosts=false&from=postView>



★④ 라즈베리파이 기반 YOLO 사물 인식

[https://github.com/leehaesung/YOLO-Powered\\_Robot\\_Vision/blob/master/README\\_Korean.md](https://github.com/leehaesung/YOLO-Powered_Robot_Vision/blob/master/README_Korean.md)

(라즈베리파이3에 YOLO 설치하는 법)

<https://j-remind.tistory.com/53>

<https://jvvp.tistory.com/1180>

• 레고스툼 - 컨베이어벨트

- 나중에 필요하면 말씀드리기 → 크게 만들 필요x, 작은(가벼운) 것부터 테스트

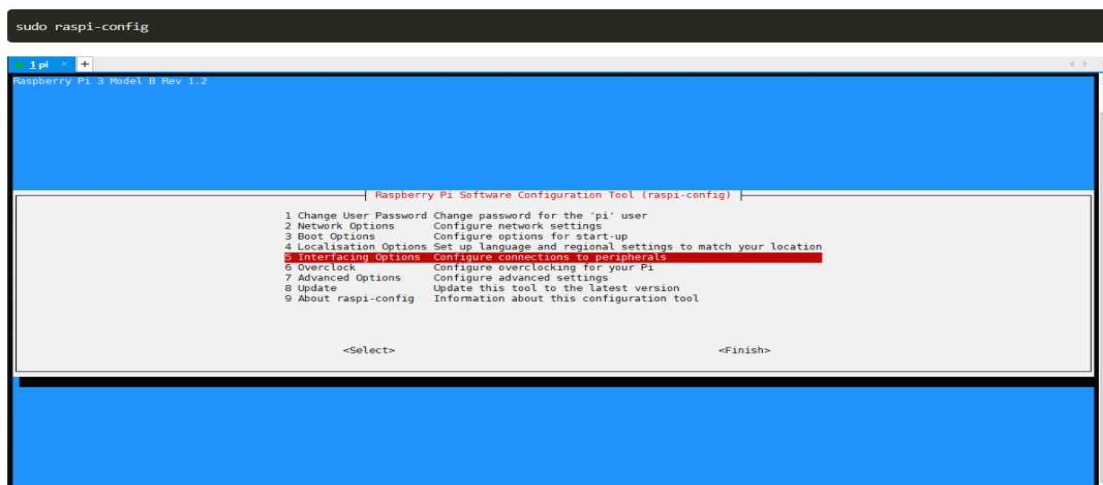
라즈베리파이 환경

-프로그램을 설치하기 위해 카메라 모듈과 OpenCV 설치 필요

라즈베리파이 카메라 모듈 사용하기

-카메라 모듈 설정

다음 명령어로 설정환경 진입



## 모듈 사용하기

-사진 찍는 명령어

```
raspistill -o picture.jpg
```

-동영상 찍는 명령어

h264에서 mp4로 변환하기

-필요한 패키지 설치

```
pi@raspberrypi:~/pi_video $ sudo apt-get install -y gpac
```

다음 명령어를 통해 변환

## 라즈베리파이 OpenCV 설치

OpenCV란?

-Intel에서 제공하는 다양한 영상 처리에 사용할 수 있는  
오픈소스 이미지 처리 라이브러리이다.

-python을 공식 지원하고 있다.

## 설치하기전

라즈베리파이에 카메라 연결 > 라즈베리파이에서 카메라를 사용할 수 있도록 설정  
[기본설정(preferences)]-[Raspberry Pi Configuration]으로 진입

[Interfaces]-Camera 항목을 Enable 바꿔주기

[ok]버튼을 누르면 재부팅되며 카메라 설정이 적용된다.

## OpenCV 설치하기

1) 라즈베리파이의 패키지 목록 업데이트 하기

```
sudo apt-get update  
sudo apt-get upgrade
```

2) openCV 패키지 설치해주기

3)openCV 소스코드 다운받기

-임시 폴더 생성

```
mkdir opencv
```

생성한 폴더로 이동

```
cd opencv
```

openCV 소스코드 다운 후 압축 풀기

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.1.2.zip  
unzip opencv.zip
```

openCV\_contrib 소스코드 다운 후 압축 풀기

```
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.1.2.zip  
unzip opencv_contrib.zip
```

4)컴파일 준비

아래 폴더로 이동

```
cd opencv-4.1.2
```

컴파일을 위한 폴더 생성

```
mkdir build
```

생성한 폴더로 이동

```
cd build
```

패키지에서 설치한 cmake를 통해 openCV에 대한 컴파일 설정

정상설치 완료

컴파일 시 메모리 부족 현상을 방지하기 위해 주저장장치를 메모리로 사용할 수 있도록 설정  
아래 명령어를 통해 [dphys-swapfile] 설정파일에 접속

```
sudo nano /etc/dphys-swapfile
```

CONF\_SWAPSIZE = 100을 #으로 주석처리 후 CONF\_SWAPSIZE = 2048 입력

```
# where we want the swapfile to be, this is the default
#CONF_SWAPFILE=/var/swap

# set size to absolute value, leaving empty (default) then uses computed value
# you most likely don't want this, unless you have a special disk situation
#CONF_SWAPSIZE=100
CONF_SWAPSIZE=2048

# set size to computed value, this times RAM size, dynamically adapts,

^G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos
^X Exit        ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell   ^_ Go To Line
```

저장하고 빠져나오기

설정파일을 재시작해 변경된 값 적용

```
sudo /etc/init.d/dphys-swapfile restart
```

```
pi@raspberrypi:~/opencv/opencv-4.1.2/build $ sudo /etc/init.d/dphys-swapfile restart
[ ok ] Restarting dphys-swapfile (via systemctl): dphys-swapfile.service.
pi@raspberrypi:~/opencv/opencv-4.1.2/build $
```

### 5)컴파일 시작

아래 명령어를 통해 컴파일 진행

```
make -j4
```

컴파일이 완료되면 아래와 같은 화면

### 6)완료된 컴파일 설치

```
sudo make install
```

openCV 라이브러리를 찾을 수 있도록 아래 명령어 실행

[dphys-swapfile] 설정파일 원래대로 수정

CONF\_SWAPSIZE = 100으로 수정

설정파일 재시작

```
sudo /etc/init.d/dphys-swapfile restart
```

카메라 테스트

1) 카메라 설정하기

파일 탐색기를 열고 상단에 아래와 같이 입력



[현재 폴더를 터미널에서 열기] 클릭

아래 명령어를 통해 예제파일 열기

```
sudo nano video.py
```

루트 디렉토리로 예제파일 열기

다음 코드에서 cap=cv.VidioCature(sourece)를 cap = cv.VidioCapture(-1)로 수정

카메라 실행

아래 명령어를 입력해 카메라 실행

```
python video.py
```

```
pi@raspberrypi:~/usr/local/share/opencv4/samples/python $ sudo nano video.py
```

```
pi@raspberrypi:~/usr/local/share/opencv4/samples/python $ python video.py
```

루트 디렉토리 명령어로 카메라 실행

```
python /usr/local/share/opencv4/samples/python/video.py
```

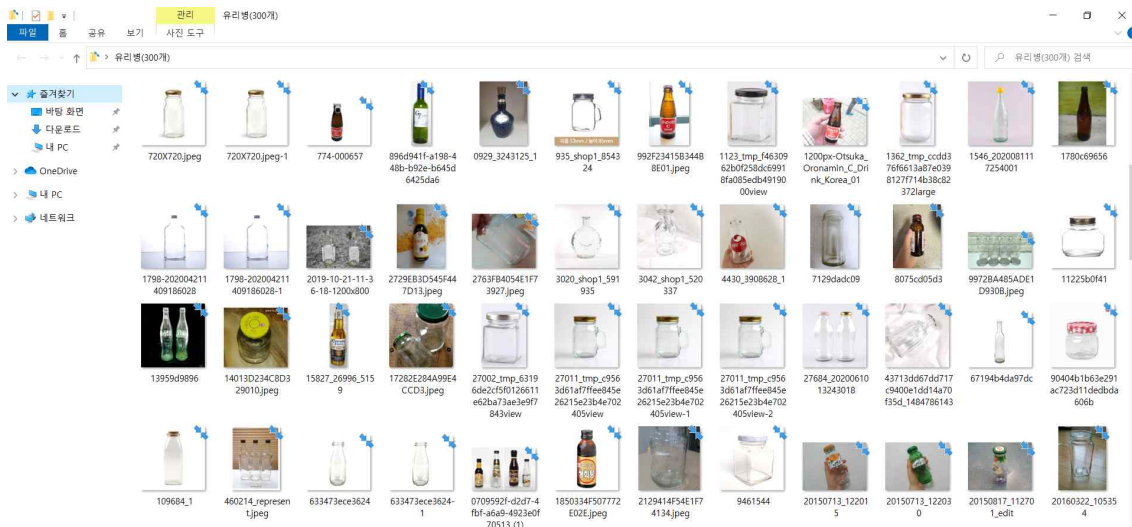
촬영 중단하고 싶을 때는 터미널 창을 누르고 ctrl + c를 누르면 중단

<OpneCV 설치>

<얼굴인식 Test 코드 - OpenCV> : <https://stickode.tistory.com/197>

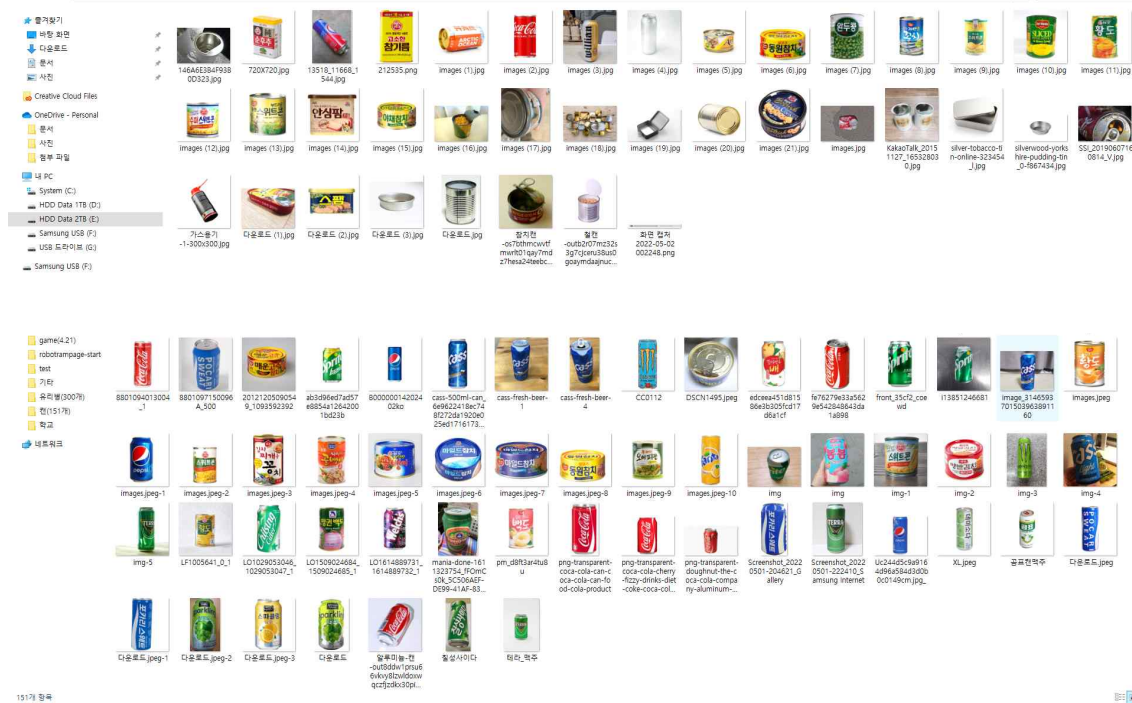
<코드 + 결과>

### 카메라가 인식하기 위한 이미지 찾기 유리병 이미지 사진





## 캔 이미지 사진



## 플라스틱 이미지 사진

## 라즈베리파이 환경 구축

터치스크린에 연동(PC로는 아직 안해봄)

-터치스크린은 친구의 도움을 받음



라즈베리파이로 사물인식 가능

\* 라즈베리파이에 카메라와 Tensorflow 설치 : 라즈베리파이의 환경을 구축한 후 카메라와 Tensorflow를 설치하였다.

<https://seo-dh-elec.tistory.com/32>

:: 코드

```
~ $ git clone https://github.com/EdgeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi.git
```

\* 라즈베리파이 사물인식 : 카메라가 잘 작동하는지 테스트하기 위해 사물을 인식할 수 있는 모델을 불러와 테스트해 보았다.

:: TFLite\_webcam 실행시키기

1. 사용환경 만들어주기

-source project\_env/bin/activate

(raspberrypi% -> (project\_env)rasberryip%로 변한 것을 확인하면 됨)

2. python으로 된 webcam을 미리 다운로드 한 모델을 이용하여 작동

-python TFLite\_detection\_webcam.py --modeldir=test\_model

(해석: python으로 작성된 웹캠 파일을 "test\_model(임시파일명)"을 사용하여 작동시킬 것이다.)

:: 실행 결과



컨베이어 벨트 제작

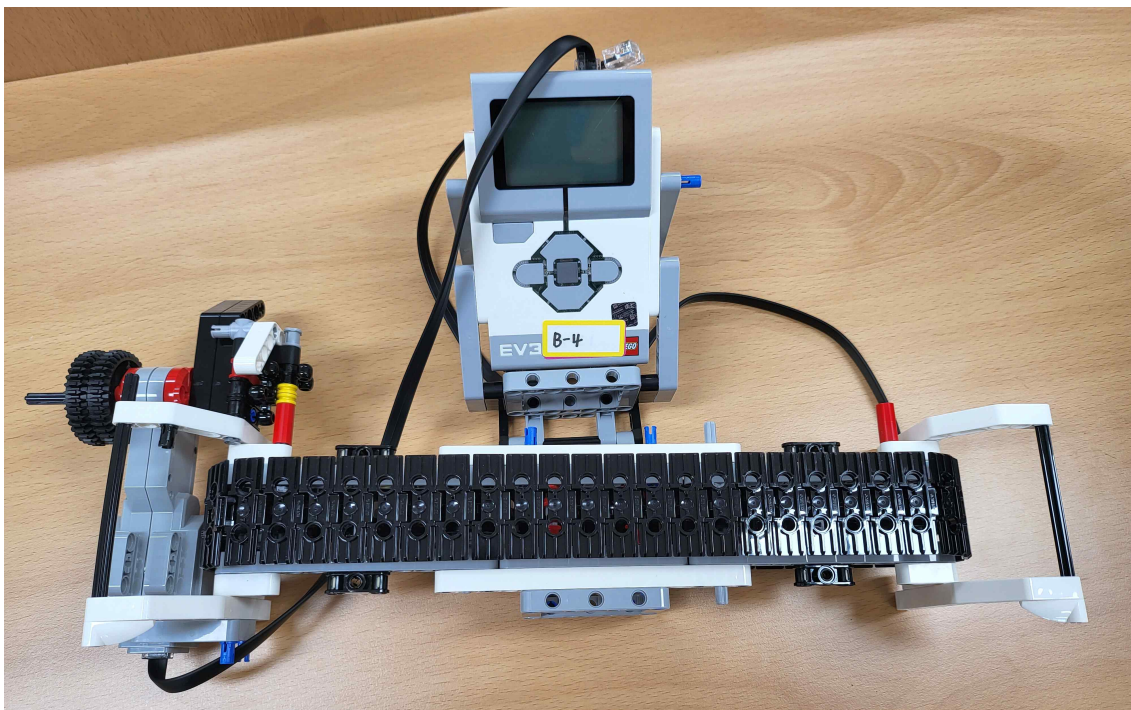
: 레고마인드스툼 EV3로 'LEGO education' 사이트에서 제공해주는 컨베이어 벨트 조립 설명서를 다  
운받아 제작하였다.

<https://education.lego.com/ko-kr/lessons/ev3-cim/make-a-sorting-machine#%EC%A1%B0%EB%A6%BD-%ED%8C%81>

- 조립 과정

설명서의 2번~59번까지의 과정으로 만들었다. 색상을 인식하는 센서 부분(36번)을 제외하고 만들었다.

- 완성된 모습



- MINDSTORMS 소프트웨어 다운 : 로봇을 작동시키기 위해서 레고마인드스톰 소프트웨어를 다운받았다.

<https://education.lego.com/ko-kr/downloads/mindstorms-ev3/software>

- 코드 작성 및 실행 : 모터를 돌아가게 하기 위해 코드를 작성하였다. 그리고 로봇과 컴퓨터를 선으로 연결하여 코드를 직접 실행해보았다.

:: 무한 루프에서 A포트로 연결된 모터에서 40%의 속도로 수행

\* 이미지 분류 : 구글 코랩에서 오픈소스를 활용해 코딩

<https://velog.io/@0hye/Colab-x-PyTorch-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EB%B6%84%EB%A5%98>

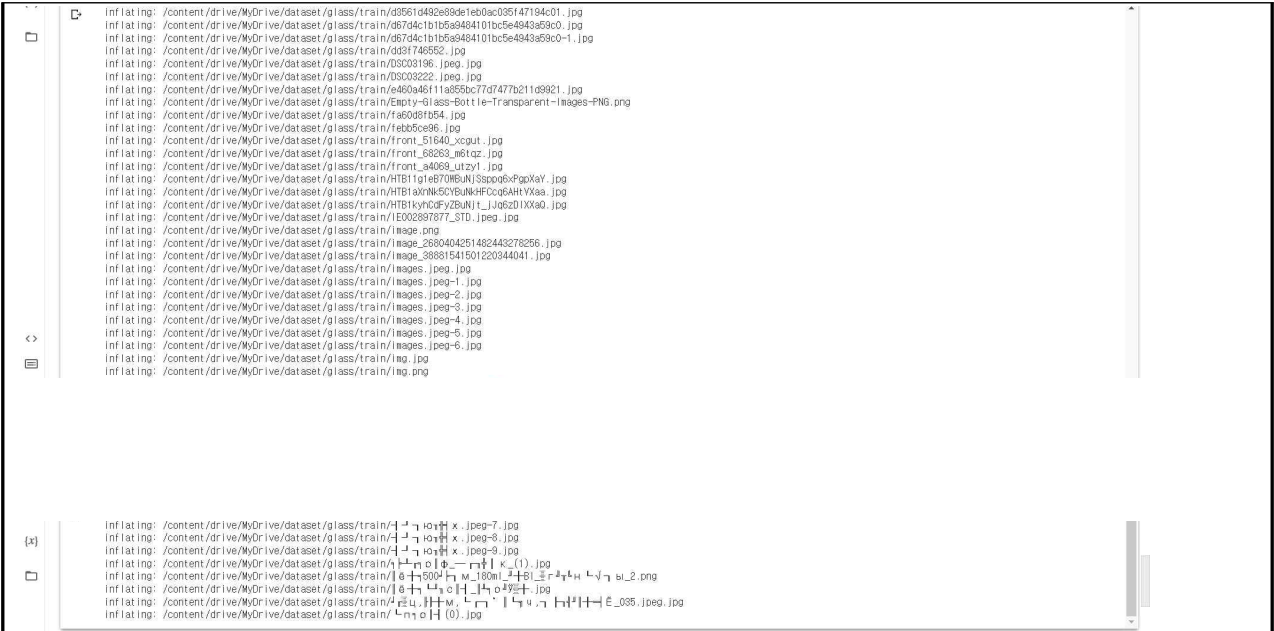
[Google Colab 사용 배경]

원래 'Jupyter notebook'을 사용하려 했는데 여러 자료를 검색하다가 'Google Colab'을 알게 되었다. Jupyter notebook은 Local이라서 이 프로그램이 깔려 있는 컴퓨터의 IP만을 가지고 로그인을 하기 때문에 다른 컴퓨터에서 사용할 수 없다는 단점이 있는데, Google Colab은 그런 단점이 존재하지 않는다. Google Colab은 아무 컴퓨터에서나 다 사용할 수 있다. 또한 Google Colab은 Goolge drive에서 파일을 사용하기 용이하고, GPU도 제공해준다.

이러한 이유로 Google Colab을 사용하기로 하였다.

Google Colab 사이트 : <https://colab.research.google.com/notebooks/intro.ipynb>

:: 구글 드라이브에서 '유리병' 이미지 폴더의 압축 풀기



:: Model

:: Hyperparameters & Device Setting

:: Dataset & Dataloader Setting(오류가 난 부분)



오류의 내용은 'train'이라는 폴더를 못 읽어 온다는 것이다. 이를 해결하기 위해 검색을 해봤지만, 아직 해결하지 못하고 있다.

:: Model Initialization



:: Loss & Optimizer

Loss & Optimizer

```
loss = m.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=lr)
```

## :: Training & Evaluation

```
images = input.to(device)
labels = target.to(device)

# Forward pass
outputs = model(images)
_, predicted = torch.max(outputs.data, 1)
total += len(labels)
correct += (predicted == labels).sum().item()

print('Epoch [{}]/[{}], Test Accuracy of the model on the {} test images: {}%'.format(epoch, total_epoch, total, 100 * correct / total))
```

더블클릭 또는 Enter 키를 눌러 수정

## 계속해서 구글코랩에서 학습

The screenshot shows a Google Colab notebook with the following code and output:

```
transforms.ToTensor(),
transforms.Normalize([0.485, 0.456, 0.406],
                    [0.229, 0.224, 0.225]))

DATASET_PATH = "/content/drive/MyDrive/dataset/SeperateCollection"
# Pass transforms in here, then run the next cell to see how the transforms look
train_data = datasets.ImageFolder(DATASET_PATH + '/train', transform=train_transforms)
test_data = datasets.ImageFolder(DATASET_PATH + '/test', transform=test_transforms)

train_loader = torch.utils.data.DataLoader(train_data, batch_size=BATCH_SIZE, shuffle=True, num_workers=1)
test_loader = torch.utils.data.DataLoader(test_data, batch_size=BATCH_SIZE, shuffle=True, num_workers=1)

print("Preparing dataset done!")
Preparing dataset done!

pretrained_model = models.resnet18(pretrained=True)
num_ftrs = pretrained_model.fc.in_features
pretrained_model.fc = nn.Linear(num_ftrs, 3)

pretrained_model = pretrained_model.to(device)

Download: "https://download.pytorch.org/models/resnet18-33702f4c.pth" to /root/.cache/torch/hub/checkpoints/resnet18-33702f4c.pth
100% 44.7M/44.7M [00:00<00:00, 108MB/s]

num_ftrs
512

criterion = m.CrossEntropyLoss()
optimizer = optim.Adam(pretrained_model.parameters(), lr=LEARNING_RATE)

total_params = 0
for param_name, param in pretrained_model.named_parameters():
    if param.requires_grad:
        total_params += len(param.reshape(-1))
print("Number of Total Parameters: {}".format(total_params))
Number of Total Parameters: 11,178,061

test_accuracy: 0.750
Predict: 1, Label: 2
Predict: 1, Label: 1
Predict: 1, Label: 1
Predict: 0, Label: 0
Predict: 1, Label: 1
Predict: 1, Label: 1
Predict: 1, Label: 1
Predict: 0, Label: 0
```

The evaluation results show a test accuracy of 0.750. Below the code, there are eight images of various bottles and containers with their predicted labels and ground truth labels:

- Image 1: A blue plastic bottle. Predict: 1, Label: 2.
- Image 2: A clear glass bottle. Predict: 1, Label: 1.
- Image 3: A clear glass bottle. Predict: 1, Label: 1.
- Image 4: A yellow and orange plastic bottle. Predict: 0, Label: 0.
- Image 5: A red and yellow plastic bottle. Predict: 1, Label: 1.
- Image 6: A brown glass bottle. Predict: 1, Label: 1.
- Image 7: A clear glass bottle. Predict: 1, Label: 1.
- Image 8: A blue and white plastic bottle. Predict: 0, Label: 0.



각각 구분은 하지만 플라스틱과 유리병은 투명하고 비슷해 구분을 잘 못하는 것 같다.  
라즈베리파이에는 적용은 되지 않는다.

Teachable Machine 사이트 : <https://teachablemachine.withgoogle.com/>

사진을 업로드 하면 자동으로 학습이 된다.

[이미지 업로드 및 학습 결과]

[Tensorflow로 파일 다운받는 창]

[TensorflowLite로 파일 다운받는 창]

<Teachable Machine 라즈베리 파이>

TensorflowLite로 다운받고 라즈베리파이 적용하는 법 :

<https://gpiocc.github.io/learn/raspberrypi/ml/2020/06/20/martin-ku-custom-tensorflow-image-classification-with-teachable-machine.html>

Tensorflow로 다운받고 라즈베리파이 적용하는 법 :

<https://www.digikey.kr/en/maker/projects/raspberry-pi-teachable-machine-teachable-pi/88c52e75f3d247fc9035480e6a1c17b1>

분리수거 학습 모델 서칭 및 적용시도

[TACO]

-길거리에 버려진 쓰레기들을 판별하는 학습 모델이다.

깃허브: <https://github.com/pedropro/TACO>

- TACO 학습 모델 예시

TACO라는 학습코드를 사용하려 했지만 라즈베리파이에서 지정한 크기보다 커서 라즈베리파이에서 학습되지 않는다.

- TACO 학습코드 라즈베리파이에서 실행한 내용

[Redacted]

[Underwater-Plastic-Detection-and-Cleaning-using-Computer-Vision-and-Deep-Learning]

- 수중에서 발견되는 쓰레기를 검출하는 학습모델이다.

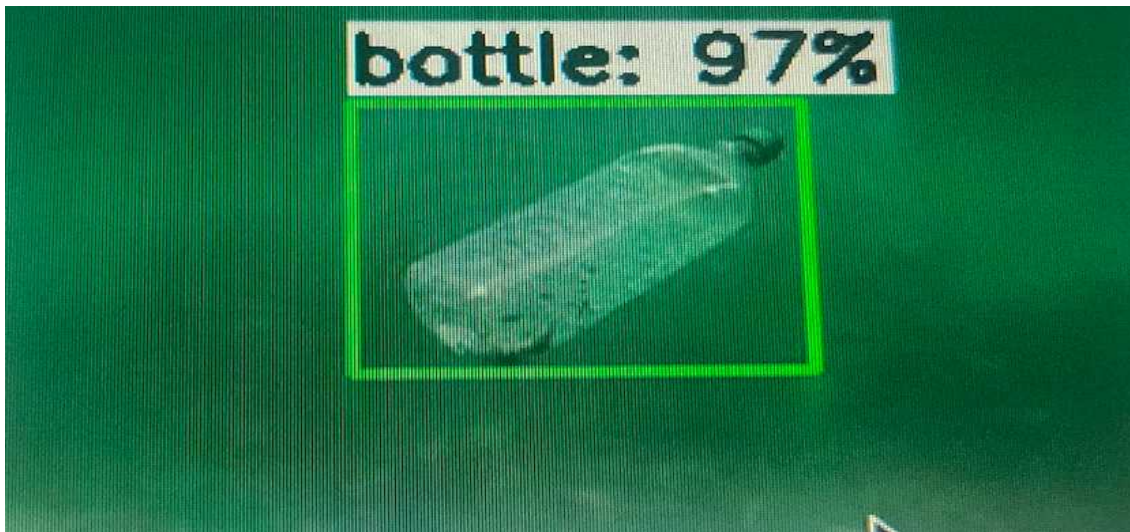
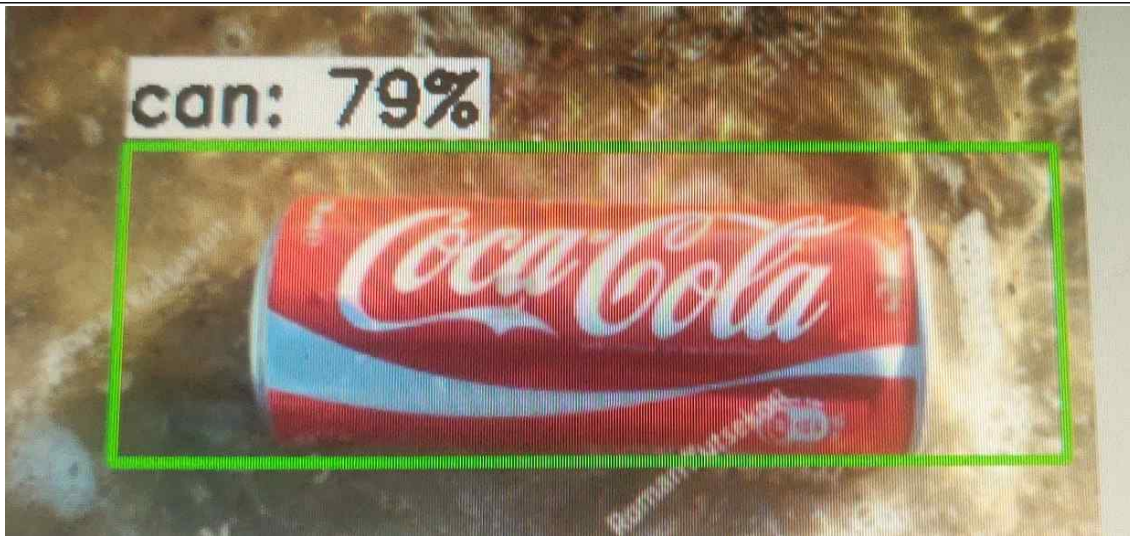
빨대, 병, 캔, 가방, 컵 등 5개의 쓰레기를 검출한다.

깃허브 :

<https://github.com/Vaishnvi/Underwater-Plastic-Detection-and-Cleaning-using-Computer-Vision-and-Deep-Learning>

- Underwater-Plastic-Detection-and-Cleaning-using-Computer-Vision-and-Deep-Learning 학습모델 예시





텐서플로우 학습모델을 쓰려면 .tflite 과 labelmap 파일이 있어야 한다.

-labelmap에는 분류된 이미지 목록이 있다.

```
5 lines (5 sioc) | 25 Bytes
1 bottle
2 can
3 straw
4 cup
5 bag
```

### Recycle-Item-Detection

-골판지, 금속 등 6가지 재활용 아이템 검출하는 코드이다.

깃허브 : <https://github.com/Aida81/Recycle-Item-Detection>



```
INFO: tensorflow.global_step/sec: 0
2021-01-29 04:03:51.216474: W tensorflow/core/framework/allocator.cc:124] Allocation of 207860000 exceeds 10% of system memory.
2021-01-29 04:08:51.417082: W tensorflow/core/framework/allocator.cc:124] Allocation of 207860000 exceeds 10% of system memory.
2021-01-29 04:08:53.055711: W tensorflow/core/framework/allocator.cc:124] Allocation of 127783760 exceeds 10% of system memory.
INFO: tensorflow:Recording summary at step 0.
INFO: tensorflow/global_step: 1: loss = 31.1770 (52.634 sec/step)
INFO: tensorflow/global_step: 2: loss = 28.5280 (4.404 sec/step)
INFO: tensorflow/global_step: 3: loss = 25.7936 (4.407 sec/step)
INFO: tensorflow/global_step: 4: loss = 24.5088 (4.351 sec/step)
INFO: tensorflow/global_step: 5: loss = 22.8618 (4.385 sec/step)
INFO: tensorflow/global_step: 6: loss = 22.0500 (4.379 sec/step)
INFO: tensorflow/global_step: 7: loss = 20.7771 (4.388 sec/step)
INFO: tensorflow/global_step: 8: loss = 19.5354 (4.343 sec/step)
INFO: tensorflow/global_step: 9: loss = 18.9881 (4.389 sec/step)
INFO: tensorflow/global_step: 10: loss = 17.5021 (4.340 sec/step)
INFO: tensorflow/global_step: 11: loss = 16.7320 (4.375 sec/step)
INFO: tensorflow/global_step: 12: loss = 15.6354 (4.311 sec/step)
INFO: tensorflow/global_step: 13: loss = 15.0864 (4.489 sec/step)
INFO: tensorflow/global_step: 14: loss = 13.8846 (4.410 sec/step)
INFO: tensorflow/global_step: 15: loss = 13.1002 (4.356 sec/step)
INFO: tensorflow/global_step: 16: loss = 12.5327 (4.336 sec/step)
INFO: tensorflow:Recording summary at step 16.
INFO: tensorflow/global_step: 17: loss = 12.8482 (7.546 sec/step)
INFO: tensorflow/global_step/sec: 0.191526
INFO: tensorflow/global_step: 18: loss = 11.8935 (4.373 sec/step)
INFO: tensorflow/global_step: 19: loss = 11.2214 (4.432 sec/step)
INFO: tensorflow/global_step: 20: loss = 11.2359 (4.393 sec/step)
INFO: tensorflow/global_step: 21: loss = 10.8061 (4.380 sec/step)
INFO: tensorflow/global_step: 22: loss = 10.4622 (4.369 sec/step)
INFO: tensorflow/global_step: 23: loss = 10.2453 (4.364 sec/step)
INFO: tensorflow/global_step: 24: loss = 10.2590 (4.375 sec/step)
INFO: tensorflow/global_step: 25: loss = 9.5733 (4.358 sec/step)
INFO: tensorflow/global_step: 26: loss = 9.8959 (4.456 sec/step)
INFO: tensorflow/global_step: 27: loss = 9.2940 (4.342 sec/step)
INFO: tensorflow/global_step: 28: loss = 9.2073 (4.388 sec/step)
INFO: tensorflow/global_step: 29: loss = 9.1035 (4.403 sec/step)
INFO: tensorflow/global_step: 30: loss = 8.8653 (4.435 sec/step)
INFO: tensorflow/global_step: 31: loss = 9.0215 (4.397 sec/step)
```

-사진이 4696개가 있는데 파일을 트레이닝시키는데 너무 많은 시간이 걸린다.  
(약 6시간 소요)

-주피터에서는 코드가 달라서 오류가 발생한다.

레고마인트스톱과 라즈베리파이 연동

구글 검색어 : Ev3 라즈베리파이  
<http://www.ntrexgo.com/archives/34454>

하드웨어적 구성도

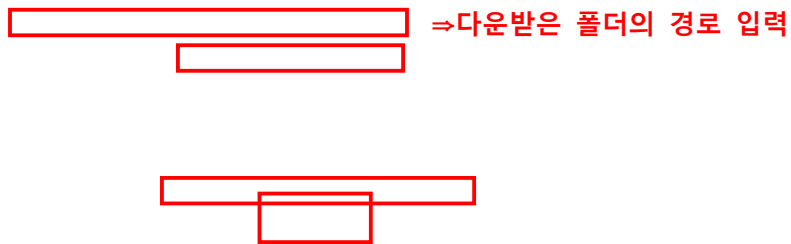
<이미지 xml 파일 만들기>

'LabelImg' 프로그램을 깃허브에서 다운받아 각 이미지에 대한 xml 파일을 만든다.

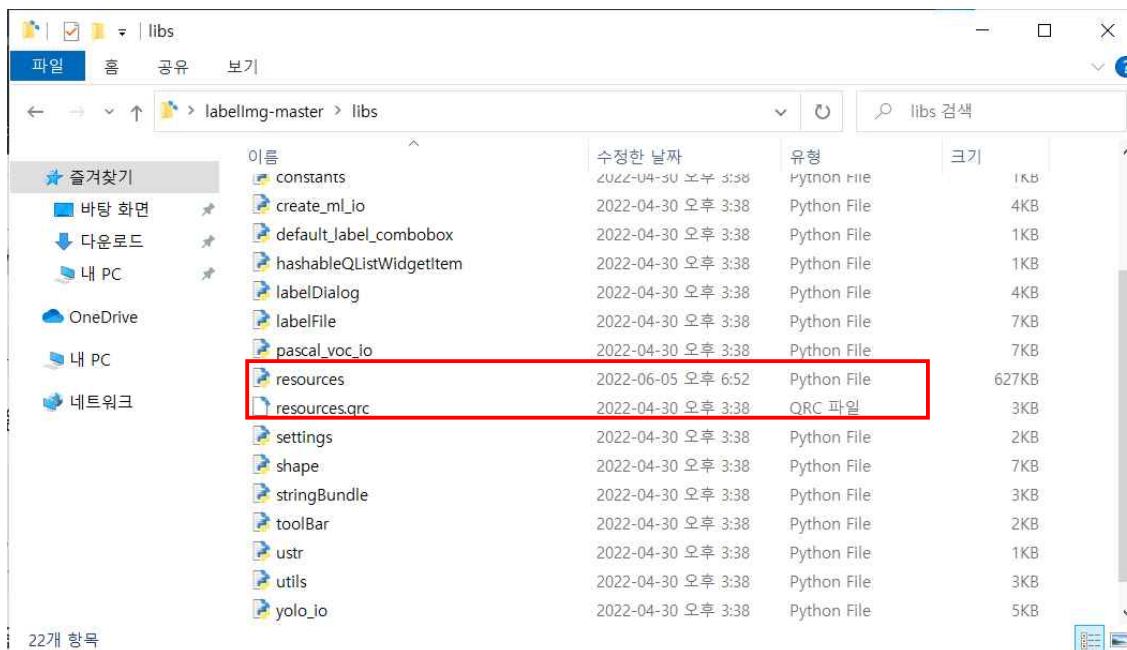
LabelImg 깃허브 : <https://github.com/tzutalin/labelImg>

① 깃허브에서 LabelImg를 다운받아 압축을 푼다.

② Anaconda Prompt를 관리자 권한으로 실행시킨 후 명령을 입력한다.



③ resources.qrc과 resource.py 파일을 labelImg-master\libs로 옮겨준다.



④ labelImg.py를 실행시켜준다.

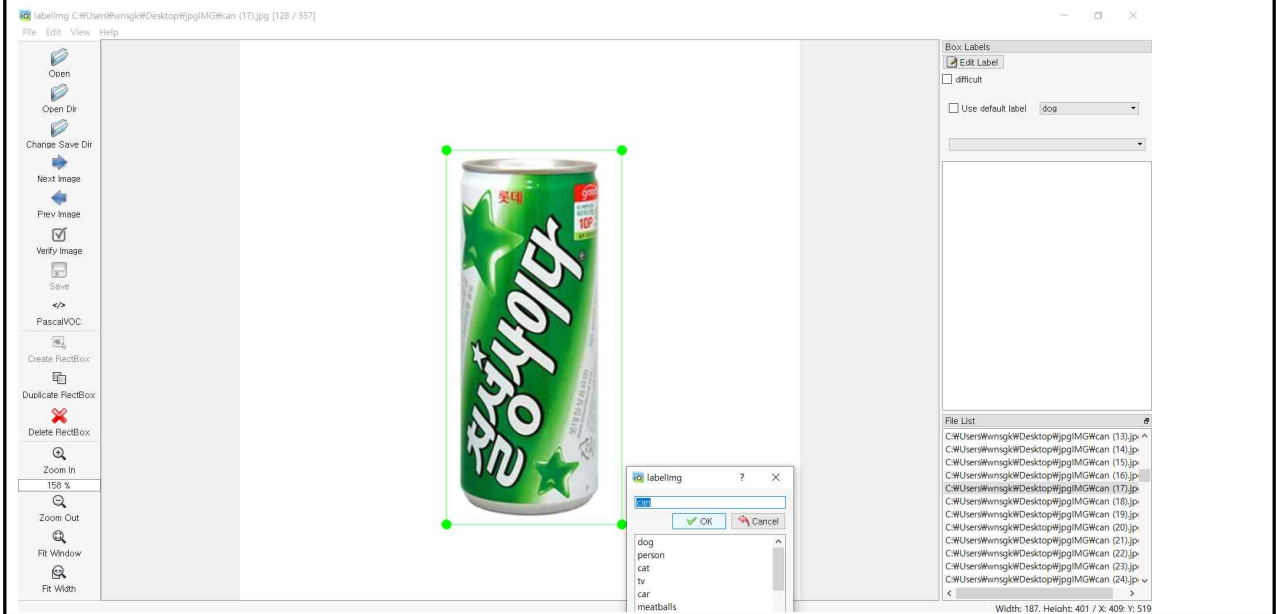


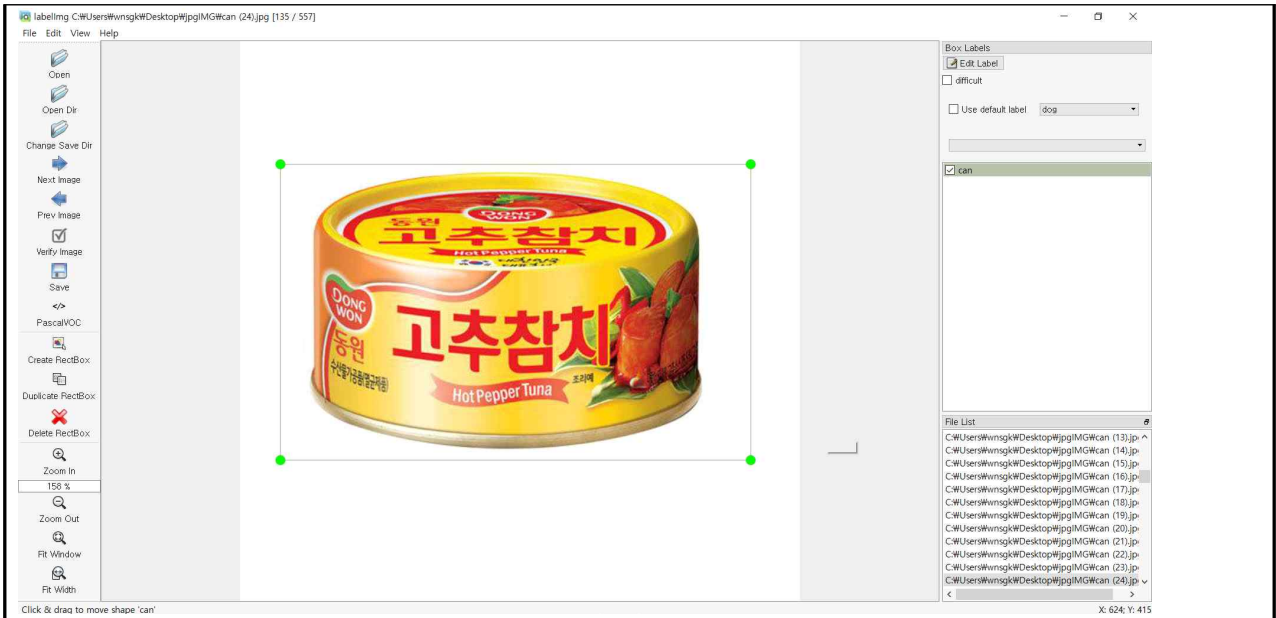
⑤ 다음 화면이 실행되면 성공이다.

'Open Dir'로 폴더를 불러온다. 불러온 폴더에서 이미지를 선택해서 작업을 하면 된다.  
W키를 누르면 경계상자를 그릴 수 있고, Ctrl+S를 누르면 xml 파일로 저장된다.

:: 플라스틱(111개)

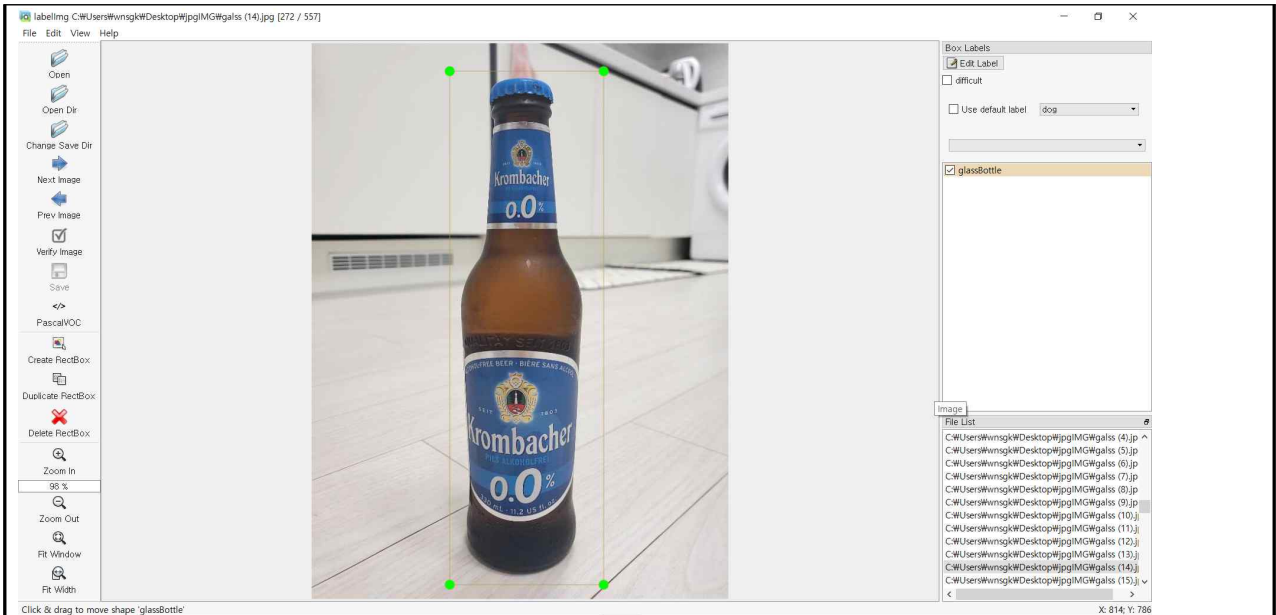
:: 캔(147개)





:: 유리병(299개)





<직접 수집한 이미지를 이용하여 학습시키고 .flite 파일 다운받기>

:: 구글 코랩 코드 실행 및 결과

Model\_Maker\_Object\_Detection.ipynb

Copyright 2021 The TensorFlow Authors.

```

[ ] Licensed under the Apache License, Version 2.0 (the "License");
  # you may not use this file except in compliance with the License.
  # You may obtain a copy of the License at
  # https://www.apache.org/licenses/LICENSE-2.0
  # Unless required by applicable law or agreed to in writing, software
  # distributed under the License is distributed on an "AS IS" BASIS,
  # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  # See the License for the specific language governing permissions and
  # limitations under the License.

```

Licensed under the Apache License, Version 2.0 (the "License");

- Train a custom object detection model with TensorFlow Lite Model Maker

In this colab notebook, you'll learn how to use the [TensorFlow Lite Model Maker](#) to train a custom object detection model to detect Android figurines and how to put the model on a Raspberry Pi.

The Model Maker library uses *transfer learning* to simplify the process of training a TensorFlow Lite model using a custom dataset. Retraining a TensorFlow Lite model with your own custom dataset reduces the amount of training data required and will shorten the training time.

- Preparation

Install the required packages

Start by installing the required packages, including the Model Maker package from the [GitHub repo](#) and the pycocotools library you'll use for evaluation.

Step 6: Evaluate the TensorFlow Lite model.

Several factors can affect the model accuracy when exporting to TFLite:

- Quantization helps shrinking the model size by 4 times at the expense of some accuracy drop.
- The original TensorFlow model uses per-class non-max suppression (NMS) for post-processing, while the TFLite model uses global NMS that's much faster but less accurate. Keras outputs maximum 100 detections while tflite outputs maximum 25 detections.

Therefore you'll have to evaluate the exported TFLite model and compare its accuracy with the original TensorFlow model.

Before compiling the .tflite file for the Edge TPU, it's important to consider whether your model will fit into the Edge TPU memory. The Edge TPU has approximately 8 MB of SRAM for caching model parameters, so any model close to or over 8 MB will not fit onto the Edge TPU memory. That means the inference times are longer, because some model parameters must be fetched from the host system memory.

One way to eliminate the extra latency is to use model pipelining, which splits the model into segments that can run on separate Edge TPUs in series. This can significantly reduce the latency for big models.

The following table provides recommendations for the number of Edge TPUs to use with each EfficientDet-Lite model.

Model architecture	Minimum TPUs	Recommended TPUs
EfficientDet-Lite0	1	1
EfficientDet-Lite1	1	1
EfficientDet-Lite2	1	2
EfficientDet-Lite3	2	2
EfficientDet-Lite4	2	3

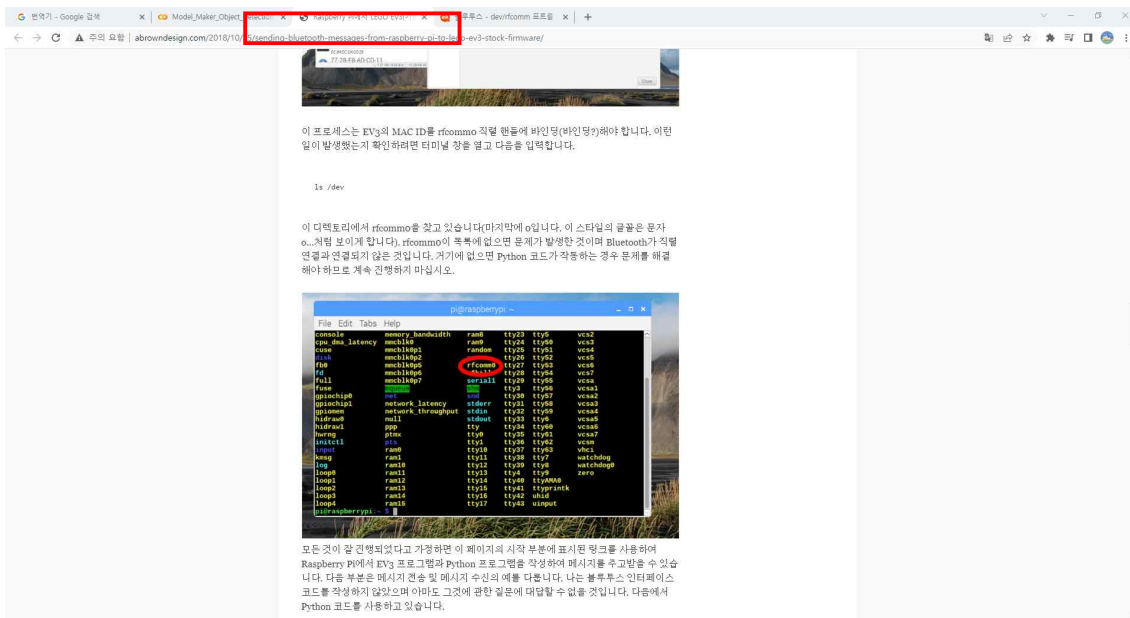
If you need extra Edge TPUs for your model, then update NUMBER\_OF\_TPUS here:

Finally, we'll copy the metadata, including the label file, from the original TensorFlow Lite model to the Edge TPU model.

:: 결과

:: 라즈베리파이 레고마인드스탐 블루투스 연결

<http://www.abrowndesign.com/2018/10/25/sending-bluetooth-messages-from-raspberry-pi-to-lego-ev3-stock-firmware/>



:: 여기 단계까지 진행

- 다음 단계는 실행이 되지 않는다.

- 빨간색 동그라미가 원래는 라즈베리파이에서 자동으로 잡혀야 하지만 잡히지 않는다.

:: 빨간색 동그라미 해결

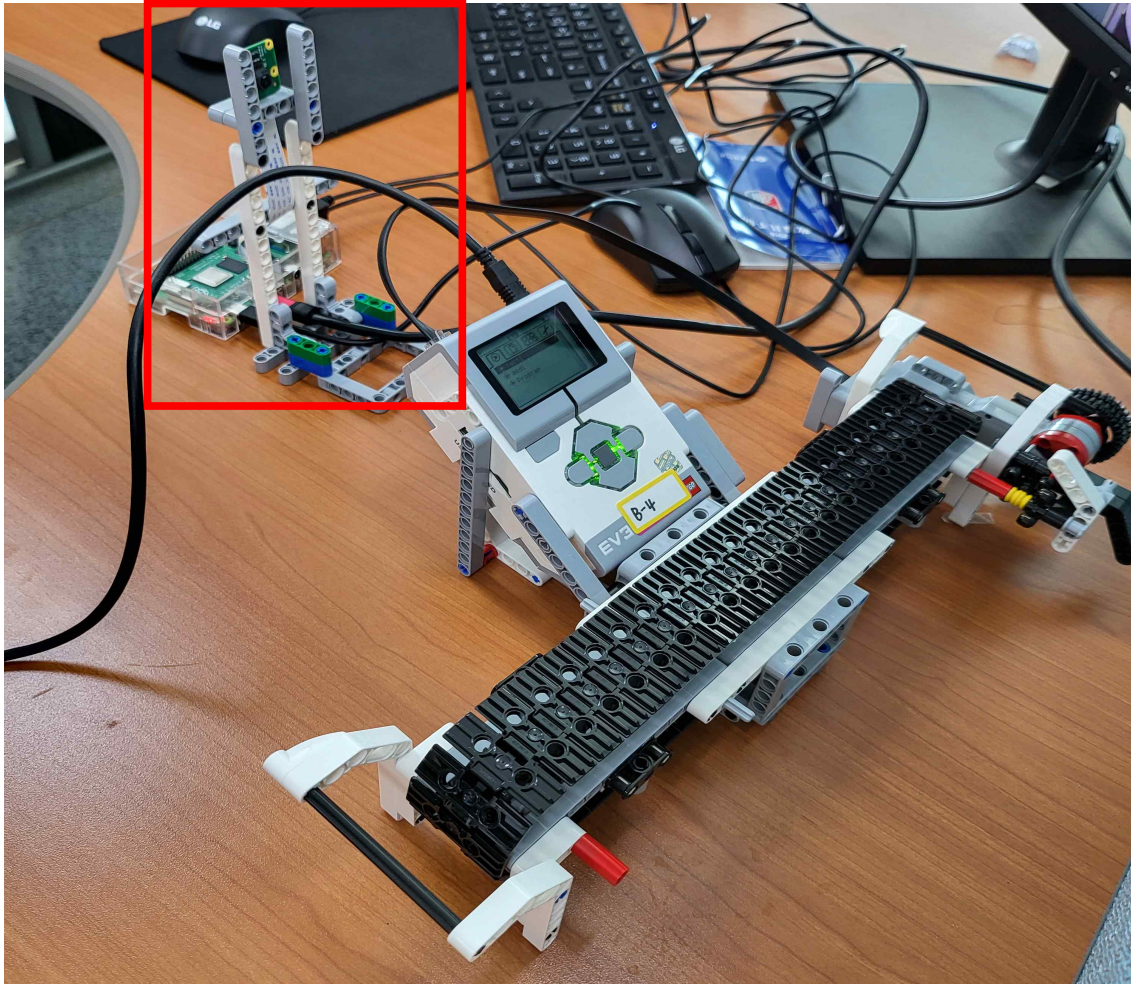
-빨간색 네모칸 추가



### 3-1 설계보완점 및 목표구현 정도

- 기존 예제코드를 사용하지 않고 직접 만들어서 구현함.
- 이미지로 학습해서 성분이 아닌 이미지로 구분함
- 라즈베리파이와 레고마인트스툼을 블루투스로 연결해 라즈베리파이 값을 레고마인트스툼이 인식해 각각 구분하고 원하는 위치에 멈출 수 있게 구현하지 못함.

### 3-2 완성작품 사진



-카메라 고정대 추가

시연 결과

응용사례

시각장애인을 인식하는 모델 만들기

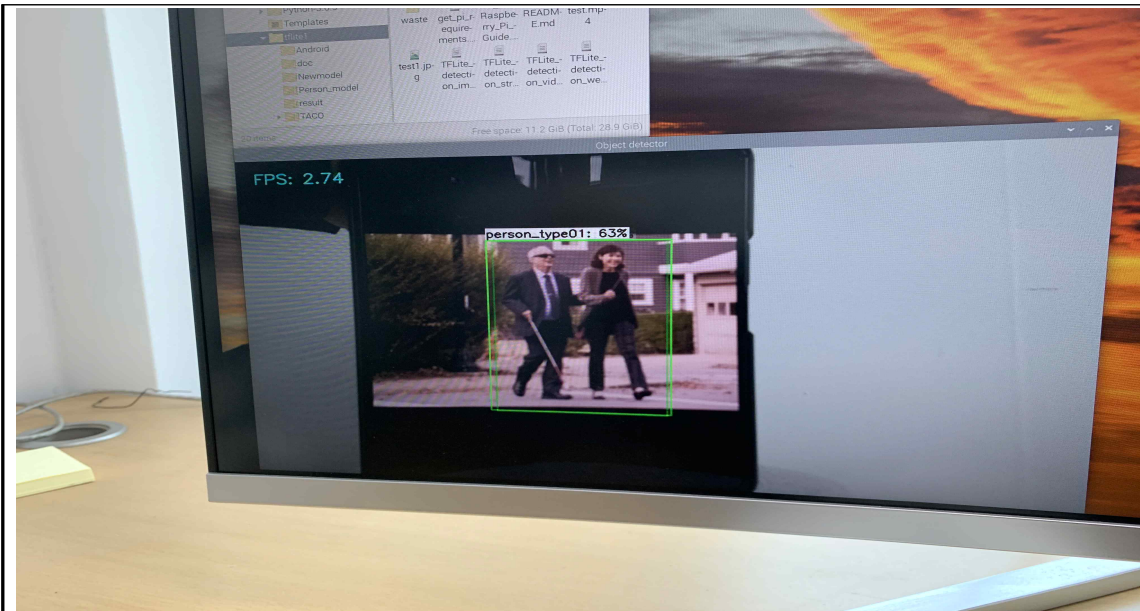
- 지팡이 이미지(100)
  
- 휠체어 이미지(100)
  
- xml 파일 만들기는 전과 동일하다.

확인 결과

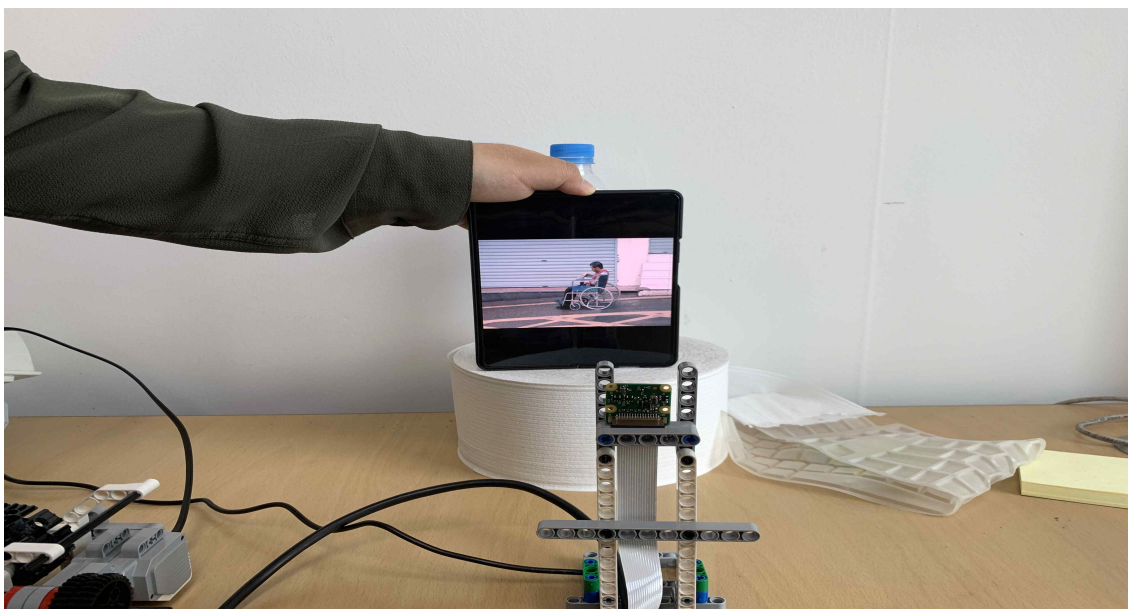
- 지팡이

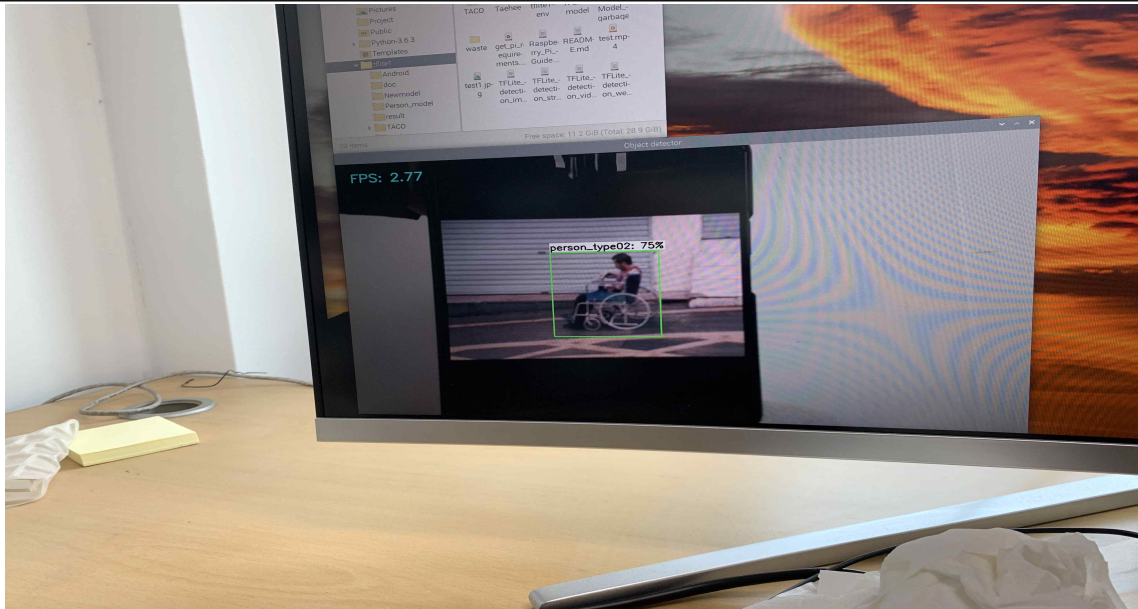






■ 휠체어





■ 활용방안

- 횡단보도에서 시각장애이용 신호버튼을 눌러주는 게 아닌 카메라로 시각장애인(지팡이, 휠체어)을 인식해 신호를 바꿔준다.

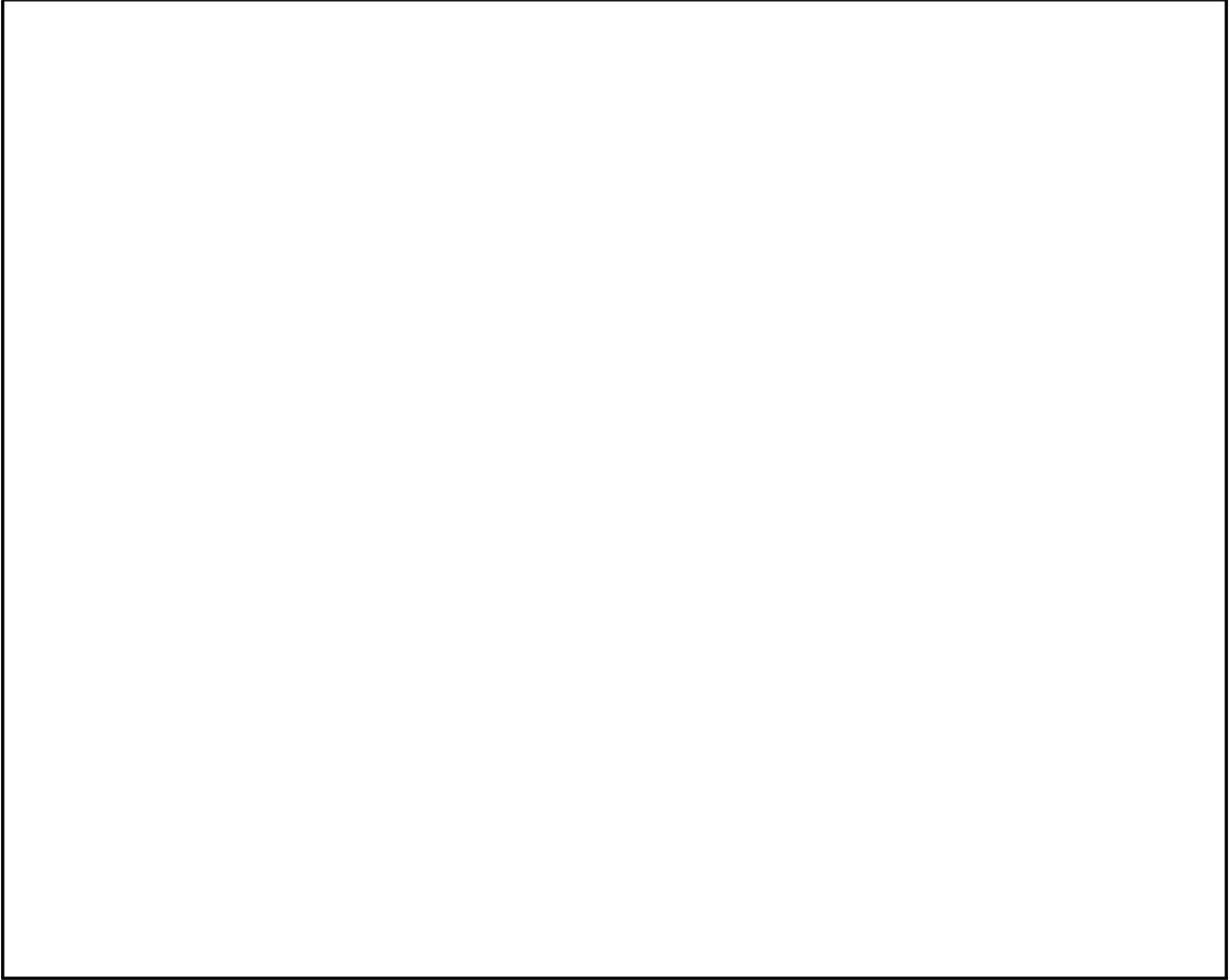
3-3 향후 개선사항

- 이미지가 아닌 성분으로 구분할 수 있게 구현하려함.
- 라즈베리파이와 레고마인트스탐을 연결하려함.

※ 예산 집행현황(기술이전형 과제수행의 경우 기업연계 재료비 사용 내역도 추가하여 작성)

구분	일자	사용 내역	금액
없음			0
합계			0

※ 최종 결과보고서에는 반드시 개발 작품의 사진이 포함되어야 함



## 캡스톤디자인 산학연계 교육 실적보고서

캡스톤디자인 교과목명 (교과목코드)	캡스톤디자인2(기업연계프로젝트 374120(01))			
캡스톤디자인 과제명	자동분리수거시스템			
교육기간	2022년 03월 02일 ~ 2022년 06월 06일			
교육개요	AI관련 지식 제공 및 공부 지도			
기업체전문가	소속	유)이티컴퍼니	성명	정영호
교육내용	AI관련 지식 제공 및 공부 지도			
교육운영결과	지식전달 및 교육 지도			

위와 같이 캡스톤디자인(과제명) 산학연계 교육 실적보고서를 제출합니다.

2022년 06월 20일

기업체전문가 : 정영호



원광대학교 LINC 3.0 사업단장 귀하

## 캡스톤디자인 지도 실적 보고서(지도교수용)

캡스톤디자인 교과목명 (교과목코드)	캡스톤디자인2(기업연계프로젝트 374120(01))			
캡스톤디자인 과제명	자동분리수거시스템			
지도학생	디지털콘텐츠공학과 20192875 임소정 디지털콘텐츠공학과 20192857 박준하 디지털콘텐츠공학과 20162925 이환규 디지털콘텐츠공학과 20183139 임은서			
지도개요	AI관련 지식 제공 및 공부 지도			
지도교수	소속	디지털콘텐츠공학과	성명	이현창
세부 지도내용	<ul style="list-style-type: none"> <li>• 과제 방향 제시해주기</li> <li>• 지식 전달해주기</li> </ul>			
수행기간	2022년 03월 02일 ~ 2022년 06월 06일			

위와 같이 캡스톤디자인(과제명)의 실적 보고서를 제출합니다.

2022년 06월 20일

지도교수 : 이현창

(인) 

원광대학교 LINC 3.0 사업단장 귀하