

## YOLOv4와 라즈베리파이 기반 쓰레기 분리배출 자동화 시스템

김기현<sup>1</sup>·유호연<sup>1</sup>·이효진<sup>2</sup>·변혜원<sup>3\*</sup>

<sup>1</sup>성신여자대학교 정보시스템공학과 학사과정

<sup>2</sup>성신여자대학교 통계학과 학사과정

<sup>3\*</sup>성신여자대학교 정보시스템공학과 교수

## Automatic Garbage Classification System based on YOLOv4 and Raspberry Pi

Ki-Hyeon Kim<sup>1</sup> · Ho-Yeon Yu<sup>1</sup> · Hyo-Jin Lee<sup>2</sup> · Hae-Won Byun<sup>3\*</sup>

<sup>1</sup>Bachelor's Course, Department of Information System Engineering, Sungshin Women's University, Seoul (02844), Korea

<sup>2</sup>Bachelor's Course, Department of Statistics, Sungshin Women's University, Seoul (02844), Korea

<sup>3\*</sup>Professor, Department of Information System Engineering, Sungshin Women's University, Seoul (02844), Korea

### [요약]

본 논문에서는 YOLO 모델과 라즈베리파이를 이용하여 재활용 쓰레기를 9종류의 클래스(캔, 페트병, 펜, 마우스, 종이박스, 클립, 열쇠, 비닐, 일회용 커피 포장지)로 분리배출하는 새로운 자동화 시스템을 제안한다. 우리 시스템은 (1) YOLOv4 기반 재활용 쓰레기 분류기, (2) 라즈베리파이 기반 쓰레기 분리 배출기, 그리고 (3) 재활용 쓰레기 관리 앱으로 구성된다. YOLOv4 모델을 이용한 쓰레기 검출과 분류에서는 자체 생성한 데이터 셋, 크롤링한 데이터 셋과 공개 데이터 셋으로 재활용 쓰레기 데이터 셋을 구축하고, 모델 파라미터 최적화를 통해 YOLOv4 모델의 실시간 성능을 유지하면서도 정확도를 90% 이상으로 향상시켰다. 또한, 파라미터 최적화, 데이터 증강, 그리고 모델 학습의 반복 횟수 등의 효과를 측정하는 의미있는 실험 결과를 제시하고 그 결과를 시스템에 적용하여 실제 구동 환경에서 실시간으로 작동하는 재활용 쓰레기 분리배출 시스템을 최적화시켰다.

### [Abstract]

This paper proposes an automatic segregation system that classifies waste into 9 classes (can, pet, pen, mouse, paper box, clip, key, vinyl, coffee wrapping plastic alike stick), using the YOLOv4 model and Raspberry Pi. Our system consists of three parts, (1) Waste Classification based on YOLOv4, (2) Waste Disposal based on Raspberry Pi, and (3) Waste Management Application for inspection and retrain. We construct a big waste dataset by crawling data, using public data and making self-product data and improve the model accuracy up to 90%, maintaining the real-time performance of the YOLOv4 model. Our system is optimized in a real environment through various experiments on parameter optimization, data augmentation, and effect on the iteration number of the model train.

**색인어** : 딥러닝, 라즈베리파이, 재활용 쓰레기, 옴로, 옴로 버전4

**Keyword** : Deep Learning, Raspberry Pi, Recycling Waste, YOLO, YOLOv4

<http://dx.doi.org/10.9728/dcs.2021.22.12.2111>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Received** 05 November 2021; **Revised** 30 November 2021

**Accepted** 30 November 2021

**\*Corresponding Author, Hae-Won Byun**

**Tel:** +82-2-920-7615

**E-mail:** hyewon@sungshin.ac.kr

## I. 서론

현재 한국에서는 코로나 19로 배달 음식의 수요가 증가하여 쓰레기 발생량이 폭발적으로 증가하고 있다. 이러한 쓰레기들을 분리 배출하지 않고 일반 쓰레기로 구분 없이 한꺼번에 배출하는 경우 쓰레기 처리를 위한 노동 비용 증가나 환경 파괴와 같은 사회적 비용을 가중시켜 불필요한 손실이 발생하게 된다. 사람들이 쓰레기를 분리 배출하지 않는 주요 원인으로서는 쓰레기를 분리해야 하는 번거로움과 재활용 쓰레기에 관한 정보 부족 등과 같은 요소들이 보고되고 있다[1],[2]. 따라서 사용자들에게 쓰레기 재활용에 관한 정보를 제공함과 동시에 사용자들이 편리하게 쓰레기를 분리 배출할 수 있도록 하여 분리 배출을 활성화할 필요가 있다.

쓰레기 분리 배출에서 중요한 점은 분류 기준에 따라 각 카테고리(캔, 플라스틱, 유리, 종이, 종이 박스, 비닐 등)별로 다양한 쓰레기를 정확하고 빠르게 분류하는 것이다. 최근에는 물체를 분류하는 딥러닝 모델이 빠르게 발전하여, 인간보다 더 높은 정확도를 보이기도 한다. 특히, 여러 객체 검출 모델 중 YOLO 모델은 객체의 검출과 객체의 분류를 하나의 회귀(regression) 문제로 해결함으로써 실시간 성능으로 향상시켜 최근에 가장 각광받고 있다[3],[4]. YOLO 모델은 버전 5까지 성능과 연산 속도에 초점을 맞추어 개선되어 왔으며, 이 중 YOLOv4는 같은 시간 대비 준수한 성능을 보이는 모델로 알려져 있다[5].

본 논문은 기존 쓰레기 분리배출의 단점을 극복하고자 객체 검출 딥러닝 모델 YOLOv4를 도입하고 모델이 전달하는 분류 결과에 따라 재활용 쓰레기를 분류해서 자동으로 해당 쓰레기 통에 배출하는 분리배출 자동화 시스템을 제안한다.

본 논문이 기여하는 핵심을 정리하면 다음과 같다.

1. 재활용 쓰레기 분리배출 과정을 자동화하는 라즈베리파이 기반의 실시간 시스템을 제작하였다.
2. YOLOv4를 사용하여 재활용 쓰레기 이미지에서 쓰레기의 종류를 판별하였으며, 90%의 정확도를 실험결과로 제시하였다.
3. YOLOv4 파라미터 최적화, 데이터 증강, 반복 횟수 변화에 따른 성능 비교 분석 등 다양한 의미있는 실험 데이터를 제시하고 분석하였다.
4. 크롤링, 캐글 데이터, 자체 생성 데이터로 구성된 약 46,000개의 재활용 쓰레기 데이터셋을 구축 및 공개하였다.

본 논문의 구성은 2장에서 관련 연구들을 소개하고, 3장에서 시스템 개요, 4장에서 딥러닝 모델, 5장에서 하드웨어 제작, 6장에서 재활용 쓰레기 관리 앱 그리고 7장에서 실험 결과와 8장에서 마지막으로 결론을 서술하였다.

## II. 관련연구

최근에 로봇 분야에서 로봇의 핵심 기능으로서 쓰레기를

검출하고 분류하는 방법론에 관심을 가지고 연구가 진행되어 왔다. Rad와 Zhang 등은 SVM(Support Vector Machine)이나 Haar cascade classifier 등과 같은 기존의 컴퓨터 비전 기술을 이용하여 거리에 있는 쓰레기를 검출하고 분리하는 기법을 개발하고 쓰레기를 치우는 거리 청소 로봇에 적용하였다[6],[7]. 이 연구에서는 거리를 촬영한 영상에서 쓰레기가 있는 영역을 검출하고 그 영역을 깨끗이 청소하는 로봇을 개발하는데 집중하고 있다. 또한, Ramalingam 등은 실내 청소기 로봇을 제안하였는데, 로봇이 거실에 떨어진 쓰레기를 발견하고 쓰레기의 종류를 판단하는 기능에 집중하였다[8].

컴퓨터 비전 연구 그룹에서는 CNN, Faster R-CNN이나 MLP(Multi-Layer Perceptron) 등을 사용하여 쓰레기를 촬영한 영상에서 쓰레기를 검출하고 종류를 판단하는 문제를 해결하고자 하였으나 처리 및 반응 속도가 느린 문제점 등을 보고해 오고 있다[9]-[12].

최근에는 객체의 검출과 분류를 통합하여 실시간 성능으로 향상시켜 주목받고 있는 딥러닝 모델인 YOLO 를 사용하여 쓰레기 검출과 분류 문제를 실시간 성능으로 향상시킨 연구들이 진행되기 시작하였다. Ye 등은 YOLOv1 모델에 VAE(Variational AutoEncoder)를 적용하여 모델 크기를 줄이고 정확도와 속도를 향상시키고자 하였고 YOLOv1, Fast R-CNN 모델들의 결과를 비교 분석하였다[13]. Kumar 등은 YOLOv3를 사용하여 6개 종류의 쓰레기 분류 방법을 제시하고 YOLOv3-tiny 알고리즘과 결과를 비교분석하는 실험을 진행하였다[14]. Chen 등은 YOLOv4 모델을 도입하여 주로 속도 개선을 위한 연구에 집중하여 임베디드 장비에서 실시간 성능을 보이게 하였고, 반면에 정확도는 다소 감소하였다[15].

이전 연구들에서는 주로 YOLOv1, YOLOv3, YOLOv3-tiny 기반의 쓰레기 분류 기법에 주로 관심을 가졌으나, 본 논문에서는 YOLOv4 최신 모델을 이용하여 정확도를 향상시키는 부분에 집중하고 재활용 쓰레기 분리배출 과정을 자동화하는 라즈베리파이 기반의 물리적 시스템을 제안한다. 재활용 쓰레기를 배출하는 전 과정을 사람으로 인지할 수 있도록 실시간 성능도 보장한다.

## III. 시스템 개요

본 논문에서 제안하는 시스템은 재활용 쓰레기 검출 부분, 쓰레기 분리배출 제어 부분, 그리고 관리 어플리케이션으로 구성된다. 쓰레기 검출 부분에서는 총 9개의 클래스로 라벨링한 데이터 셋을 이용해 YOLOv4 모델을 학습시켰다. 쓰레기 분리배출 제어 부분에서는 컨베이어 벨트 위에 놓인 쓰레기를 검출 결과에 따라 쓰레기통의 해당 칸에 집어넣는 작업을 제어한다. 관리 어플리케이션은 사용자 모드와 관리자 모드로 구성된 앱으로서, 사용자에게는 분리배출 목록과 포인트를 보여 주고, 관리자에게는 데이터 검증 및 라벨링을 편리하게 할

수 있는 기능을 제공한다.

전체적인 시스템 작동 순서는 그림 1과 같다. 사용자가 재활용 쓰레기를 컨베이어 벨트 위에 놓으면 쓰레기는 컨베이어 벨트를 따라 전진하다가 벨트 끝에 도달하면 끝 부분 하단에 놓인 쓰레기통으로 낙하한다. 쓰레기통은 내부가 칸막이로 구분되어 있고 각 칸에 서로 다른 종류의 재활용 쓰레기를 보관하는 구조이다. 재활용 쓰레기를 촬영한 카메라 영상으로부터 쓰레기의 종류를 판별하고 그 종류가 무엇인지에 따라 쓰레기통을 회전시켜 해당 칸으로 쓰레기가 낙하하도록 제어한다. 쓰레기통에 스텝 모터를 장착하고 라즈베리 파이와 연결하여 쓰레기통의 회전을 제어하였다.

#### IV. 딥러닝 모델

##### 4-1 데이터 수집

본 논문에서 학습 데이터셋은 웹에서 크롤링한 데이터 셋, 공개 데이터 셋, 그리고 직접 제작한 데이터셋으로 구성하였다. 표 1에서 크롤링한 사이트들과 공개 데이터를 수집한 리소스들을 표기하였다. 크롤링한 데이터 셋과 공개 데이터 셋 중 물체의 형태가 실제 분리배출 환경에서 나타나는 형태와 상이한 데이터는 데이터 셋에서 제외시켰다. 데이터 셋을 자체 생성하기 위해서, 실제 환경과 유사한 환경을 구축하고 다양한 재활용 쓰레기 실물을 수집하여 스마트폰으로 촬영하였다. 그림 2는 수집한 이미지 데이터의 일부를 보여 주고 있다. 각 이미지의 해상도는 다소 차이가 있으나 평균 너비 576, 높이 324이고 RGB 컬러 이미지이다.

표 1. 데이터 수집

Table 1. Data Collection

Method	Resources
rawling data	Google, Yahoo, Bing, Naver, Daum
open data	Kaggle Garbage Classification https://www.kaggle.com/mostafaabla/garbage-classification WaDaba Images Database http://wadaba.pcz.pl/#download
self-creation	Recording 9 classes of garbages



그림 2. 종류별 재활용 쓰레기 이미지들

Fig. 2. Images of waste categories

표 2. 9개 클래스의 데이터 개수

Table 2. Number of data in 9 classes

class	crawling data	open data	self-creation	total
can	2,000	795	2,322	5,117
pet	2,000	883	2,147	5,030
pen	2,000	738	2,343	5,081
mouse	2,000	602	2,279	4,881
paper box	2,000	904	2,116	5,020
clip	2,000	503	2,782	5,285
key	2,000	811	2,235	5,046
vinyl	2,000	751	2,367	5,118
stick vinyl	2,000	0	3,269	5,269
total	18,000	5,987	21,860	45,847

사용하는 클래스는 총 9가지로 캔, 페트, 펜, 마우스, 종이 박스, 클립, 열쇠, 비닐, 스틱형 비닐로 구성하였다. 표 2는 각 클래스에 해당하는 데이터 셋의 구성을 보여 준다. 수집된 데이터는 총 45,847개 이고 클래스별 평균 데이터 개수는 약 5천 개이다. 실제 환경에서 입력되는 다양한 카메라 이미지에 대응하기 위해서 데이터 증강을 진행하였다. 실험에서는 45,847개의 데이터 셋 중에 데이터 정제를 통해 각 클래스당 평균 3,000개의 데이터를 사용하였고, 데이터를 6배 증강시

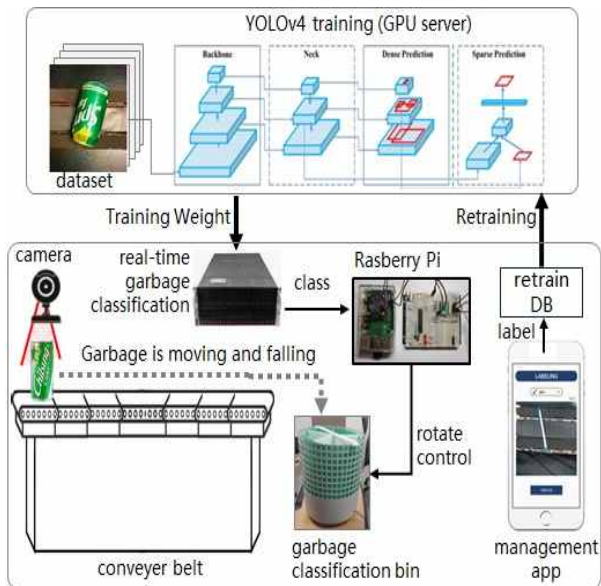


그림 1. 시스템 개요

Fig. 1. System overview

켜 클래스당 약 2만장, 전체 약 15만장의 데이터 셋을 사용하였다. 이미지 증강 기법은 좌우반전과 화질 조절 기법을 주로 이용하였다. 화질 조절 기법은 시스템이 설치되는 실제 환경에 있는 카메라 종류 및 품질에 대응하기 위해 사용되었다. 다음 그림 3은 이미지 데이터를 증강한 일부 결과를 보여주고 있다.

YOLO의 입력 형식에 따라 데이터 셋의 라벨링을 진행하였다. HSV 색상값 기반 이진화 방법으로 약 6천 장의 이미지를 자동으로 라벨링하는 SW를 자체 제작하여 진행하였으며, 나머지 약 2만 장은 오픈소스 도구 'BBox-Label-Tool' 을 이용하여 수작업으로 라벨링을 진행하였다. 라벨링을 위해서 재활용 쓰레기 모습이 포함되도록 경계 상자(bounding box)를 생성했다. GT(Ground Truth) 경계 상자의 정보는 상자의 중심 좌표와 가로, 세로 길이로 표현된다. 그림 4는 자체 제작한 자동 라벨링 SW를 보여주고 있다.

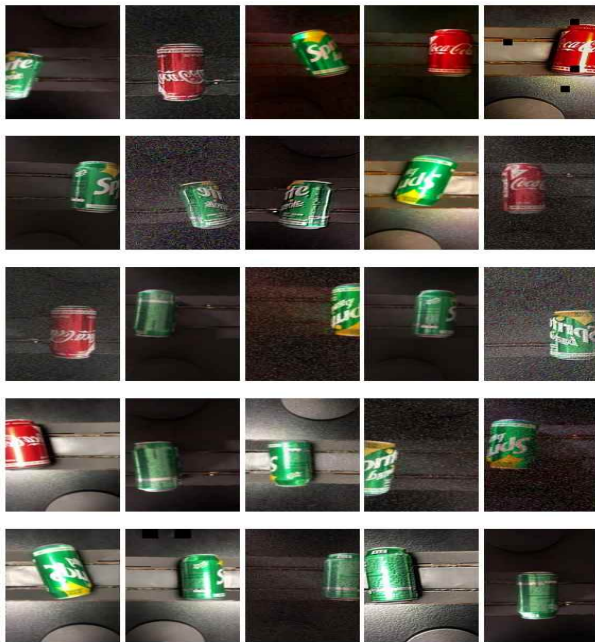


그림 3. 데이터 증강  
Fig. 3. Data Augmentation



그림 4. 자체 제작한 HSV 색상값 기반 자동 라벨링 SW  
Fig. 4. Automatic labeling SW based on HSV color model

#### 4-2 YOLO 모델 학습

YOLO는 물체의 영역을 검출하는 과정과 물체를 인식하는 과정을 하나의 네트워크 모델로 결합하여 기존 방법에 비해 월등한 성능을 보인다. 그림 5와 같이 입력 영상을 S x S의 그리드 셀로 분할하고 각 셀에서 경계 상자의 위치와 각 경계 상자에 객체가 존재하는지 여부를 나타내는 신뢰도(confidence), 그리고 어떤 클래스 종류인지를 나타내는 클래스 확률을 동시에 예측함으로써 빠른 추론 속도와 높은 정확도를 나타낸다. 따라서, YOLO 네트워크는 객체의 특징을 추출하는 서브 네트워크와 객체를 인지하는 서브 네트워크로 구성된다.

표 3. 학습 파라미터들  
Table 3. training parameters

parameter	value
data size	45,847 (7:2:1)
image width & height	416 x 416
classes	9
convolution filters	42
batch size	64 / mini batch size 8
max batches / steps	20,000 / 16,000, 18,000
learning rate	0.001
momentum	0.9
weight decay	0.0005

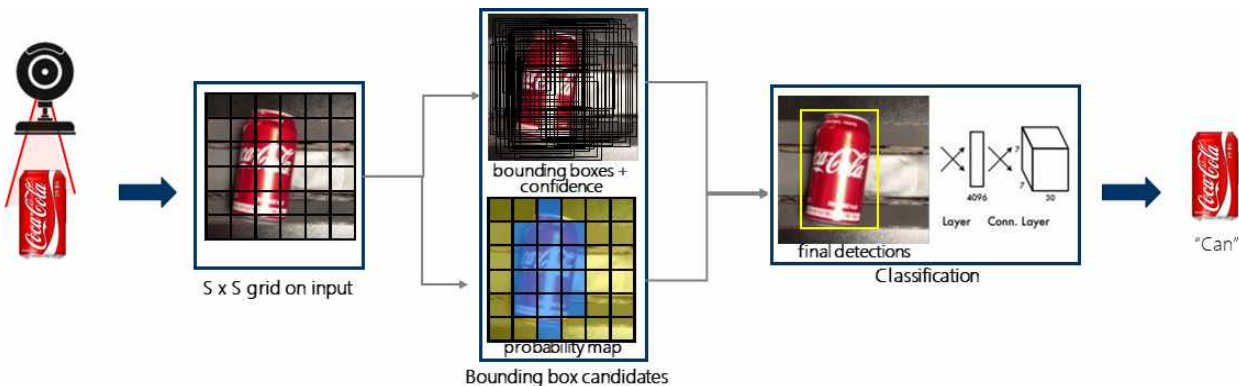


그림 5. YOLO 객체 검출 네트워크  
Fig. 5. YOLO object detection network



본 논문에서 사용하는 딥러닝 프레임워크는 오픈소스 기반의 YOLOv4 모델로서 53개의 CNN 계층으로 설계한 CSPDarknet-53 백본 네트워크를 사용하였다. 데이터 학습은 배치 크기를 64로 정의하고 컨벌루션 필터 42개, 최대 배치와 스텝은 각각 20,000과 16,000, 18,000으로 설정하였다. 훈련 데이터와 검증 데이터 및 실험 데이터는 각 7대 2대 1의 비율로 나누어 학습에 적용하였으며 학습에 총 36시간이 소요되었다. 표 3에서 실험에 사용된 파라미터들을 정리하였다.

## V. 하드웨어 제어

본 논문에서 제안하는 하드웨어 시스템은 재활용 쓰레기를 올려놓으면 자동으로 이동시키는 '컨베이어 벨트'와 쓰레기 종류에 따라 각각 독립적으로 구분하여 담은 '쓰레기 분류 통'으로 구성된다.

### 5-1 컨베이어 벨트 제작

컨베이어 벨트는 길이 120cm, 벨트 폭 3cm인 컨베이어 벨트 두 개를 이어 붙여서 제작하였고 벨트를 구동시키기 위해서 서보모터를 장착하였다. 서보모터의 속도는 588°/sec이며, 토크는 9.4Ncm 로서 그림 6과 같다.

### 5-2 쓰레기 분류 통 제작

쓰레기 분류 통은 딥러닝 모델에서 분류된 결과를 기준으로 쓰레기를 물리적으로 구분하는 역할을 하며, 그림 7과 같이 원통형의 외부 지지용 통과 내부 회전 통으로 구성되어 있다. 외부 지지용 통은 지름 40cm, 높이 40cm의 LDPE 재질의 통을 이용하였다. 내부 회전 통은 지름 36.5cm, 높이 25cm로 제작하였으며, 회전축은 지름 5mm, 높이 30cm의 스테인레스 샤프트를 이용하였다.



그림 6. 컨베이어 벨트  
Fig. 6. Conveyor belt

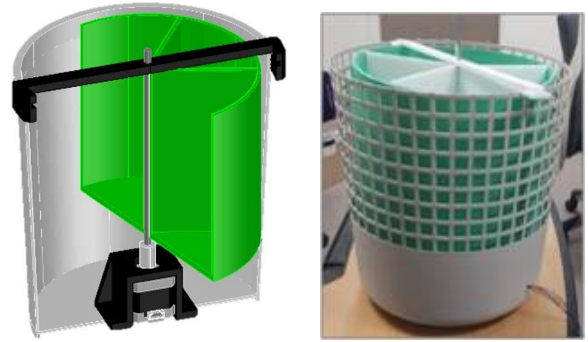


그림 7. 쓰레기 분류 통  
Fig. 7. Separate waste bin

9가지 종류의 쓰레기를 분류하기 위해 칸막이를 제작하여 원형 쓰레기통을 9개의 공간으로 구분하도록 쓰레기통 내부에 칸막이를 장착하였다. 분류 칸막이는 가로 18cm, 높이 25cm의 우드락 판넬로 제작하였으며, EVA 폼으로 테두리를 둘러 내부 회전 통을 제작하였다. 분류통을 회전시키기 위한 모터로 NEMA 17 스텝 모터를 이용하였으며 토크는 0.42Nm이다. 스텝 모터와 회전축은 커플링을 이용해 연결하였고, 스텝 모터를 외부 지지용 통에 고정하기 위해 3D 프린터로 브라켓을 출력하였다.

### 5-3 쓰레기 분리 제어

쓰레기 분류 통의 제어를 위한 장치로서 라즈베리파이 3B+를 사용하였으며 OS는 라즈비안 리눅스를 설치하였다. 스텝 모터의 스텝과 방향을 전기 신호로 변환하는 모터 드라이버는 A4988을 사용하였다. 또한, 딥러닝 모델에서 분류한 클래스를 라즈베리파이에 전달하기 위해 데이터베이스와 연동하였다. 그림 8은 칸막이가 설치된 쓰레기 통의 회전 제어를 위해 라즈베리파이를 도입하고 스텝 모터를 장착한 모습을 보여준다.

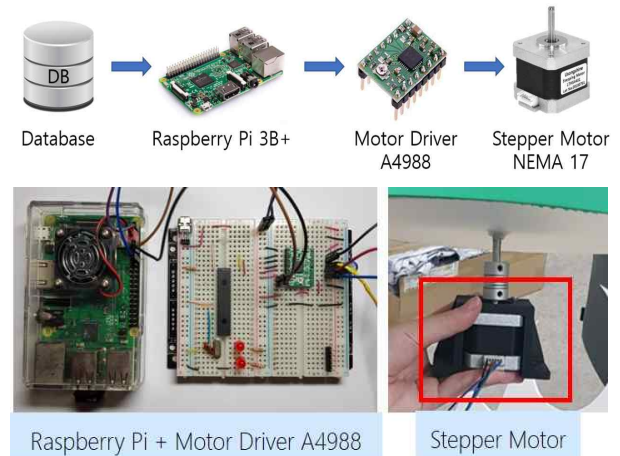


그림 8. 쓰레기 통 모터 제어  
Fig. 8. Motor control of waste bin

자체 제작한 쓰레기 분류 통의 모터 제어는 검출 테이블의 클래스가 이전 클래스와 차이가 발생할 때까지 대기하게 되며, 클래스 차이가 발생하면 모터를 제어한다. 클래스 차이에 따라 쓰레기 분류 통을 회전시키고 속도 개선을 위해 시계 방향 또는 반시계 방향 중 가까운 방향으로 회전하도록 하였다.

### VI. 관리 어플리케이션

재활용 쓰레기 분리배출 처리 상황을 실시간으로 편리하게 확인하고 관리를 용이하게 하기 위하여 모바일 앱을 제작하여 사용자와 관리자를 위한 서비스를 제공한다.

#### 6-1 사용자 앱

사용자 로그인을 통해 재활용 실적 확인과 포인트 확인과 같은 개인맞춤형 서비스를 제공한다. 재활용 실적 확인에서는 개인이 분리 배출한 목록을 쓰레기 종류별로 확인할 수 있다. 포인트 확인에서는 올바르게 작동한 분리배출 실적만큼 전환되는 누적 포인트를 확인할 수 있다. 그림 9의 왼쪽 첫 번째 그림에서 개인이 분리 배출한 쓰레기의 목록과 포인트 누적 현황을 확인할 수 있다. 향후, 이 포인트는 지역 단체와 협력하여 지역 화폐와 같은 개념으로 확장할 예정이다.

#### 6-2 관리자 앱

자동화 시스템에서 검출한 모든 분리배출 목록을 확인할 수 있으며, 신뢰성이 낮은 데이터에 한해 관리자가 앱에서 직접 검수하고 라벨링을 실시할 수 있다. 그림 9의 왼쪽 두 번째와 세 번째 그림에서 오검출이 발생한 쓰레기 항목을 검수하고 라벨링할 수 있는 기능을 지원하는 화면을 확인할 수 있다. 라벨링한 데이터는 서버에 전송되어 그림 10과 같이 매일 자정마다 자동으로 모델을 재학습하는데 이용된다.

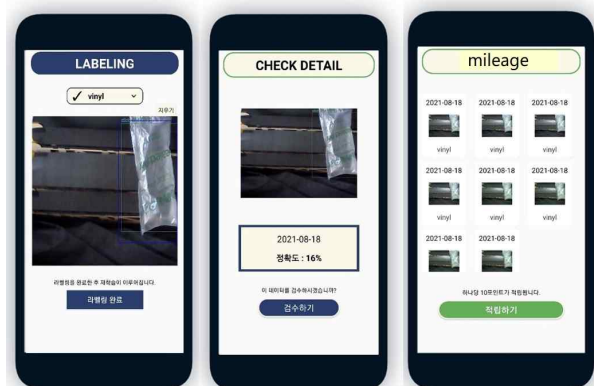


그림 9. 사용자 앱 & 관리자 앱  
Fig. 9. User app & administrator app



그림 10. 자동 재학습 로그 기록  
Fig. 10. Automatic Retraining log

### VII. 실험결과

YOLOv4를 이용하여 컨베이어 벨트에 놓인 재활용 쓰레기의 종류를 검출하고 이에 따라 내부가 칸막이로 구분된 쓰레기 통을 회전시켜 쓰레기를 실시간으로 자동 분리 배출하는 실험을 수행하고 성능을 확인하였다. 실험에 사용된 컴퓨터 사양은 Intel i7-7700, 4.2GHz CPU에 64GB 메모리, 그리고 NVIDIA GTX 1080 Ti GPU 4대, 그리고 Ubuntu 16.04가 사용되었다.

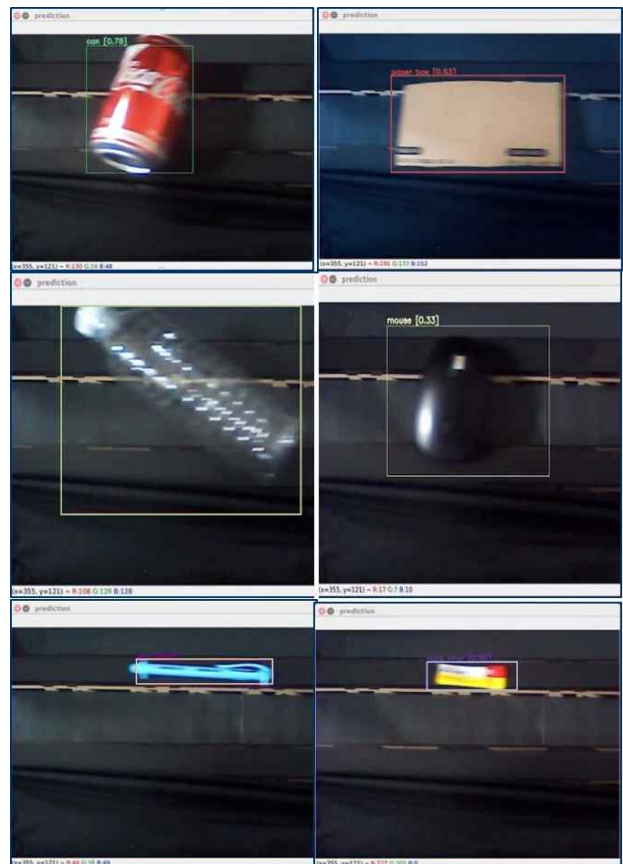


그림 11. YOLOv4 재활용 쓰레기 검출과 분류 결과  
Fig. 11. Waste detection and classification with YOLOv4

표 4. 모델 학습 결과 성능

Table 4. Model training performance

Model	mAP	Precision	Recall	F1 Score
YOLOv4	0.907	0.877	0.906	0.89

데이터 셋 약 30,000개 중 약 21,000개를 YOLO 모델 훈련에 사용하였고 나머지 9,000개를 성능 평가에 사용하였다. 그림 11은 쓰레기 분류 기능의 YOLOv4 인식 결과로서 예측된 경계 상자의 검출 예시를 보여주고 있다. 사전에 학습된 YOLO 모델을 사용하고 실제 상황과 유사하게 분리배출 실험 환경을 구축한 후 획득한 테스트 데이터를 입력하여 실시간으로 수행된 결과이다. YOLO 모델이 전반적으로 재활용 쓰레기의 검출과 인식에 성공하는 결과를 확인할 수 있다. 실제 환경에서 분리배출 자동화 시스템을 이용하여 100회 테스트한 결과 제대로 분류한 경우는 92회 도출되었으며 나머지 8회는 조명 반사로 인한 오검출, 제작한 쓰레기 분류 통에 비해 부피가 큰 물체의 분류로 인한 오류 등이 포함되었다. 자동화 시스템에서 발생하는 오검출은 관리자 앱에서 검수되고 재학습되어 시스템에 반영되므로 시간이 지남에 따라 오검출 비율이 현저하게 떨어지는 것을 확인하였다.

모델의 성능을 평가하기 위하여 객체 검출에서 널리 사용되는 평균 정밀도인 AP(Average Precision)를 사용하였다 [3]. 재현율(Recall)을 가로축으로 정확도(Precision)를 세로축으로 하는 그래프에서 AP는 그래프의 하단 면적에 해당되며 AP 수치가 높을수록 성능이 좋은 것을 의미한다. 클래스마다 AP를 계산할 수 있으며 각 클래스들의 AP를 모두 평균한 수치가 평균 AP, mAP(mean AP)이다. 본 논문에서 학습 결과 모델의 성능은 다음 표 4와 같이 평가되었다. 표에서 F1 Score는 재현율과 정확도의 조화평균으로 계산된다.

표 6. 모델 학습 결과 성능

Table 6. Model training performance

Iteration	Classes AP(%)										Precision	Recall	F1-score	Average IOU(%)	mAP (%)
	pet	pen	mouse	paper	key	clip	vinyl	stick vinyl	can						
4,000	73.89	58.37	72.66	71.55	47.09	22.56	53.75	45.76	87.77	47.09	22.56	53.75	45.76	87.77	
5,706	81.30	70.08	74.51	76.47	61.44	37.98	62.54	62.39	85.05	61.44	37.98	62.54	62.39	85.05	
7,412	88.98	81.88	95.41	87.55	70.28	51.14	64.93	71.05	99.01	70.28	51.14	64.93	71.05	99.01	
9,118	92.55	87.72	96.26	91.13	78.00	58.63	74.33	77.46	98.71	78.00	58.63	74.33	77.46	98.71	
10,824	94.89	90.66	94.44	91.41	79.90	61.38	76.77	79.62	99.83	79.90	61.38	76.77	79.62	99.83	
12,530	95.76	92.65	92.93	95.53	84.01	65.06	77.16	84.10	97.85	84.01	65.06	77.16	84.10	97.85	
14,236	96.84	94.58	97.36	93.58	90.29	70.77	82.58	84.78	99.35	90.29	70.77	82.58	84.78	99.35	
15,942	97.44	94.65	97.54	91.23	87.59	77.61	85.36	86.33	98.55	87.59	77.61	85.36	86.33	98.55	
17,648	97.10	94.21	97.98	95.96	90.92	72.31	79.62	85.92	99.80	90.92	72.31	79.62	85.92	99.80	
19,354	97.14	94.25	96.94	94.80	87.39	73.81	84.36	85.93	99.85	87.39	73.81	84.36	85.93	99.85	

표 5. 클래스당 평균 정밀도

Table 5. Class Average Precision

class	AP(%)	TP	FP
pet	97.44	14,283	943
pen	94.65	13,920	2,019
mouse	97.54	10,108	852
paper box	91.23	9,933	1,149
key	87.59	7,872	1,329
clip	77.61	6,949	2,502
vinyl	85.36	6,695	1,380
stick vinyl	86.33	6,512	1,241
can	98.55	9,744	270

표 5에서 클래스별 AP를 보여주고 있다. 클래스들 중에서 객체의 외곽선 형태와 색상이 비교적 명확한 페트와 캔은 각각 97.4%와 96.5%로 높은 정확도를 보였다. 그러나, 상대적으로 크기가 작은 클립이나 다양한 색상 및 투명도를 가진 비닐과 스틱 비닐의 AP는 다른 클래스에 비해 상대적으로 낮았다. 향후 연구에서 데이터 셋 확충과 투명도를 가진 객체 인식을 위한 추가 알고리즘이나 서버 네트워크 추가 등으로 해결할 필요가 있다.

표 6은 각 클래스마다 학습하는 과정에서 반복 횟수가 증가함에 따라 변화하는 모델의 성능을 보여준다. 10,000번의 반복 이후에 mAP는 거의 85%에 도달하고 15,000번의 반복 이후에는 90%에 도달하는 것을 확인할 수 있다. 표 7은 데이터 증강이 AP에 미치는 효과와 영향 정도를 실험한 결과를 보여주고 있다. 객체의 형태가 복잡하고 객체 내부에 빈 공간이 있는 열쇠나 클립과 같은 경우나 투명도 성질을 가지는 비닐 같은 경우에 다른 클래스와 비교하여 AP 자체도 낮은 편에 속하고, 데이터 증강 이전의 AP가 현격히 떨어지는 것을 발견할 수 있다.



향후 연구에서 특수한 형태나 성질을 가지는 객체의 정확도를 향상시키기 위한 서브 네트워크 추가 또는 다른 네트워크의 결합 등을 시도해 볼 필요가 있다. 그림 12에서는 모든 클래스에 대해서 학습이 진행됨에 따라 손실이 감소하는 형태를 보여주고 있다. 그래프에서 파란색은 손실을 나타내고 빨간색은 mAP를 나타낸다.

그리고, 속도에 관한 실험을 진행한 결과, 객체 검출과 인식의 처리 속도는 평균적으로 약 55 fps였으며 하드웨어 동작을 포함한 전체 반응속도는 평균적으로 1초 이내의 성능을 보여 실시간 자동화 시스템을 구현하는데 매우 적합한 것으로 결론을 낼 수 있다.

표 7. 클래스당 평균 정밀도 (데이터 증강 없는 경우 vs. 데이터 증강 있는 경우)

Table 7. AP without data augmentation vs. AP with data augmentation

class	without data Augmentation			with data augmentation		
	AP	TP	FP	AP	TP	FP
pet	90.71	1,423	199	97.44	14,283	943
pen	76.90	1,281	552	94.65	13,920	2,019
mouse	89.86	835	118	97.54	10,108	852
paper box	85.66	775	143	91.23	9,933	1,449
key	45.00	244	205	87.59	7,872	1,329
clip	33.28	264	354	77.61	6,949	2,502
vinyl	74.38	194	53	85.36	6,695	1,380
stick vinyl	64.66	323	152	86.33	6,512	1,241
can	97.25	964	27	98.55	9,774	270

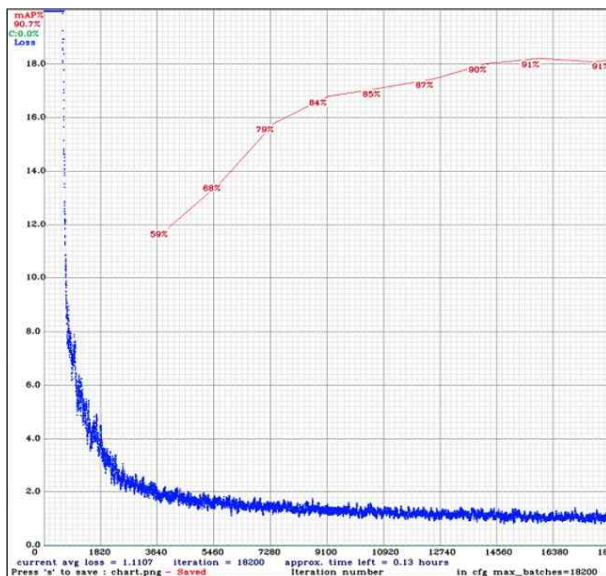


그림 12. 반복 횟수에 따른 손실 값과 mAP

Fig. 12. Loss value and mAP value w.r.t number of iterations

## VIII. 결 론

본 논문에서는 YOLOv4를 이용하여 9개 종류의 재활용 쓰레기를 분류하고 라즈베리파이 기반으로 재활용 쓰레기를 자동 배출하는 새로운 자동화 시스템을 제안하였다. 크롤링 데이터, 공개 데이터 그리고 자체 생성한 쓰레기 데이터로 구성된 45,847개의 데이터 셋을 구축하고 파라미터를 조절하여 9가지 클래스에 대해 강건한 쓰레기 분류에서 우수한 결과를 입증하였다. 또한, 파라미터 최적화, 데이터 증강, 훈련 반복 횟수 등과 관련 있는 다양한 실험을 통해 의미있는 데이터를 제시하였다. 쓰레기를 분리 배출하는 전 과정이 실시간을 유지하면서 자동화 시스템에서 유용하게 사용할 정도의 정확도를 약 90%까지 향상시켰다. 비록 정확도 90%가 만족할 만한 성능은 아니지만, 시스템에서 관리자 앱을 통해 매일 재학습하는 기능을 제공하여 시간이 지남에 따라 정확도가 향상되도록 시스템을 구축하였다. 이와 같이 우리가 제안한 시스템은 사용자와 관리자를 위한 앱을 전체 시스템에 결합함으로써 통합 시스템으로서의 유용성도 입증하고 있다.

향후, 아파트나 스마트 시티 등 시스템이 외부 환경에 설치되는 경우에 대비하기 위하여 다양한 날씨를 반영하도록 적절한 추가 학습을 통해 쓰레기 분류 모델을 더욱 강건하게 만들 필요가 있다. 클립과 같이 객체 내부에 빈 공간이 있는 복잡한 형태의 객체 인식 문제를 해결할 필요가 있고 특히 투명한 물체 인식의 정확도 개선은 흥미롭고 도전적인 문제이다. 또한, 현재 9가지 클래스 이외의 재활용 쓰레기나 재활용할 수 없는 쓰레기에 대한 문제, 클래스의 확장에 따른 쓰레기 분류 통의 한계에 대한 대안이 필요하다. 이와 같은 연구를 통하여 사용자들의 편의성을 재고하고 사회적 비용을 낮추며 더 나아가 쓰레기 분리배출의 활성화로 환경 개선까지 일부 해결할 수 있다고 예상된다.

## 감사의 글

이 논문은 2019년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음.

## 참고문헌

[1] Hoornweg, Daniel and Bhada-Tata, Perinaz. *What a Waste : A Global Review of Solid Waste Management*, World Bank, Mar. 2012.

[2] Tomas Ekvall, Getachew Assefa, Anna Björklund, Ola Eriksson, and Göran Finnveden, “What life-cycle assessment does and does not do in assessments of waste management”, *Waste Management*, Vol. 27, Issue 8, pp 989-996, Apr. 2007.  
<https://doi.org/10.1016/j.wasman.2007.02.015>



- [3] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *CoRR* abs/1506.02640, June 2015. <https://doi.org/10.1109/CVPR.2016.91>
- [4] Redmon, Joseph, and Ali Farhadi, "YOLOv3: An Incremental Improvement." *ArXiv* abs/1804.02767, Apr. 2018.
- [5] Bochkovskiy and Alexey, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *ArXiv* abs/2004.10934, Apr. 2020.
- [6] M. S. Rad, A. von Kaenel, A. Droux, F. Tiede, N. Ouerhani, H. K. Ekenel, and J.-P. Thiran, "A Computer Vision System to Localize and Classify Wastes on the Streets," *International Conference on Computer Vision Systems*. Springer, pp. 195–204, Oct. 2017. [https://doi.org/10.1007/978-3-319-68345-4\\_18](https://doi.org/10.1007/978-3-319-68345-4_18)
- [7] P. Zhang, Q. Zhao, J. Gao, W. Li, and J. Lu, "Urban Street Cleanliness Assessment Using Mobile Edge Computing and Deep Learning," *IEEE Access*, Vol. 7, pp. 550–563, May 2019. <https://doi.org/10.1109/ACCESS.2019.2914270>
- [8] B. Ramalingam, A. Lakshmanan, M. Ilyas, A. Le, and M. Elara, "Cascaded Machine-Learning Technique for Debris Classification in Floor Cleaning Robot Application," *Applied Sciences*, Vol. 8, No. 12, pp. 2649, Dec. 2018. <https://doi.org/10.3390/app8122649>
- [9] Gary White, Christian Cabrera, Andrei Palade, Fan Li, and Siobhan Clarke, "WasteNet: Waste Classification at the Edge for Smart Bins," *arXiv preprint at arXiv:2006.05873*, June 2020.
- [10] Devi, R.S.S., Vijaykumar, V.R., and Muthumeena, M., "Waste Segregation using Deep Learning Algorithm," *International Journal of Innovative Technology and Exploring Engineering*, Vol. 8, pp. 401–403, Dec. 2018.
- [11] Adedeji, O. and Wang, Z., "Intelligent Waste Classification System using Deep Learning Convolutional Neural Network," *In Proceedings of the Procedia Manufacturing*, Vol. 35, pp. 607–612. Jan. 2019. <https://doi.org/10.1016/j.promfg.2019.05.086>
- [12] Chu, Y., Huang, C., Xie, X., Tan, B., Kamal, S., and Xiong, X., "Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling," *Compu. Intell. Neurosci.*, Nov. 2018, 5060857. <https://doi.org/10.1155/2018/5060857>
- [13] Anbang Ye, Bo Pang, Yucheng Jin, and Jiahuan Cui, "A YOLO-based Neural Network with VAE for Intelligent Garbage Detection and Classification," *3rd International Conference on Algorithms, Computing and Artificial Intelligence(ACAI)*, No. 73, pp. 1–7, Dec. 2020, <https://doi.org/10.1145/3446132.3446400>
- [14] Saurav Kumar, Drishti Yadav, Himanshu Gupta, Om

Prakash Verma, Irshad Ahmad Ansari, and Chang Wook Ahn, "A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards SmartWaste Management," *Electronics*, No. 10, Vol. 14, Dec. 2021, <https://doi.org/10.3390/electronics10010014>

- [15] Qingqiang Chen and Qianghua Xiong, "Garbage Classification Detection Based on Improved YOLOv4," *Journal of Computer and Communications*, No. 8, pp. 285–294, Dec. 2020, <https://doi.org/10.4236/jcc.2020.812023>.



**김기현(Ki-Hyeon Kim)**

2018년~현 재: 성신여자대학교 정보시스템공학과 학사과정  
 ※ 관심분야 : 컴퓨터 그래픽스(Computer Graphics), 컴퓨터  
 비전(Computer Vision), 딥러닝(Dep Learning) 등



**유호연(Ho-Yeon Yu)**

2019년~현 재: 성신여자대학교 정보시스템공학과 학사과정  
 ※ 관심분야 : 컴퓨터 그래픽스(Computer Graphics), 컴퓨터  
 비전(Computer Vision), 딥러닝(Dep Learning) 등



**이효진(Hyo-Jin Lee)**

2018년~현 재: 성신여자대학교 통계학과 학사과정  
 ※ 관심분야 : 컴퓨터 그래픽스(Computer Graphics), 컴퓨터  
 비전(Computer Vision), 딥러닝(Dep Learning) 등



**변혜원(Hae-Won Byun)**

1990년 : 연세대학교 컴퓨터과학과 (이학사)  
 1992년 : KAIST 컴퓨터공학과 (공학석사)  
 2004년 : KAIST 컴퓨터공학과 (공학박사-컴퓨터 그래픽스)  
 2004년~2006년: KBS 기술연구소 연구원  
 2006년~현 재: 성신여자대학교 정보시스템공학과 교수  
 ※ 관심분야 : 컴퓨터 그래픽스(Computer Graphics), 컴퓨터  
 비전(Computer Vision), 딥러닝(Dep Learning) 등