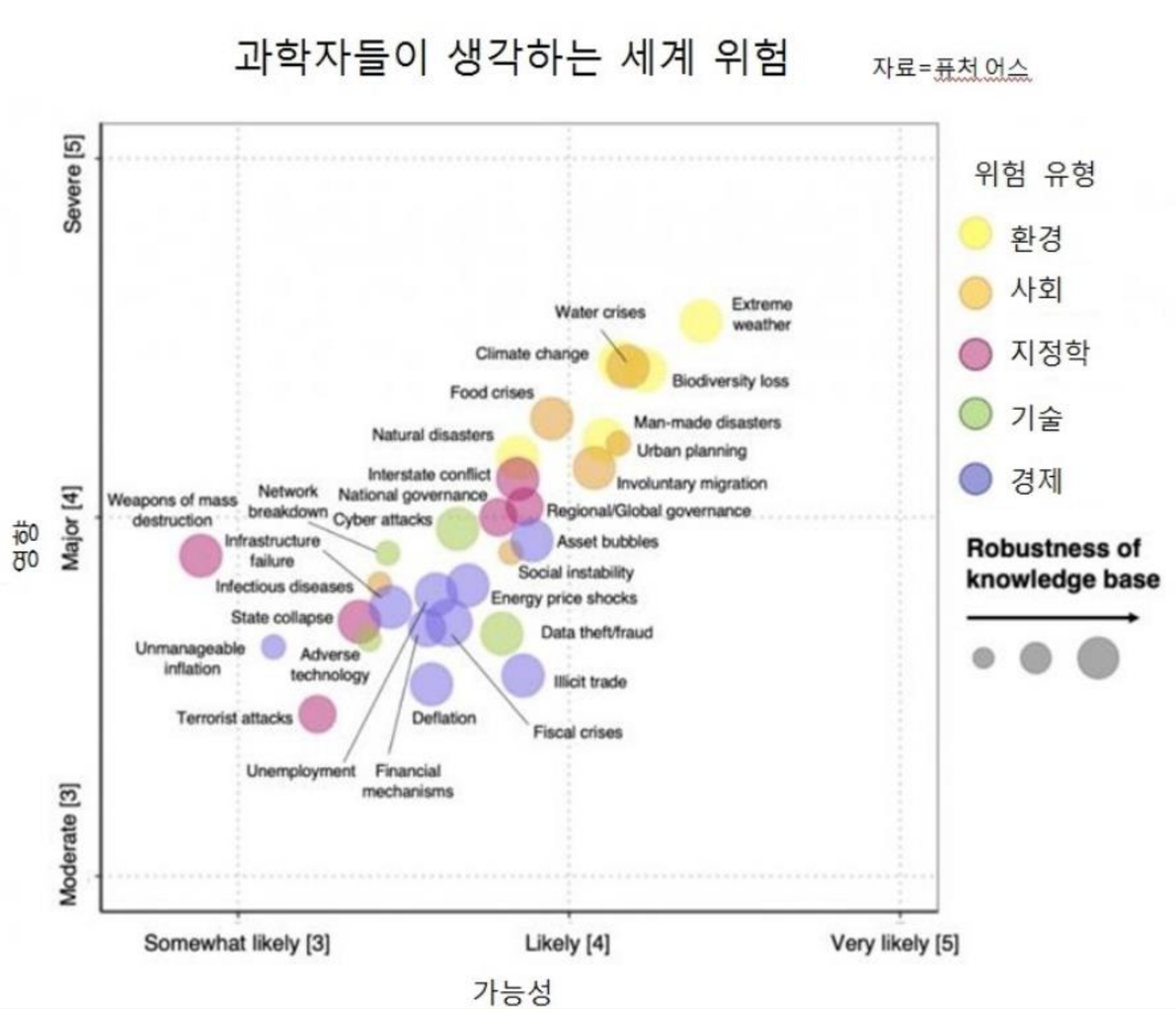
FOR EARTH, FOR US

나반 3팀

20211729, 20213062, 20213098

1. 문제 인식



“환경 문제의 심각성”

환경 문제 인식 부족

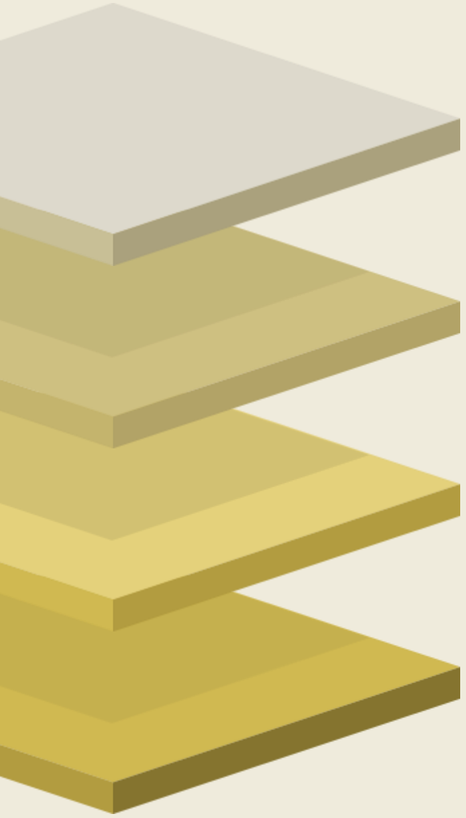
과학자들이 생각하는 세계 위험에 환경 문제가 높은 순위로 선정되었듯이, 환경 문제는 점점 심각해지고 있지만 사람들의 환경문제에 대한 인식은 여전히 미흡

개인의 실천성 부족

환경 문제에 해결을 위해 적극적 실천이 부족함. 해결해야 하는 문제라고 생각할지라도, 대중적인 환경 앱의 부재와 같이 동참 방법이 부족한 경우도 존재

2. 아이디어 소개

"누구나 쉽게 환경 문제 해결에 동참할 수 있는 앱 개발"



- 이미지 인식 기술로 길거리 쓰레기 수집 후 처리
- 환경 관련 활동 권장(공모전/봉사활동 참여 인증)
- 환경 관련 학습 콘텐츠 제공(영상/퀴즈)
- '따릉이' 같은 친환경 수단 이용 편리하게 연계

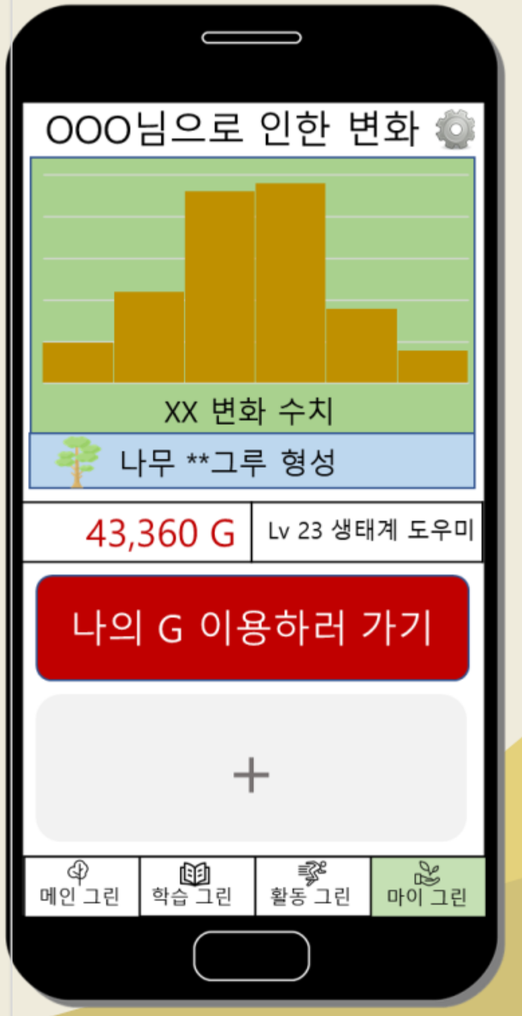
환경 포인트 도입
+
환경 제품 구매나
상품권에 활용가능

3. 구현 방법 - 디자인

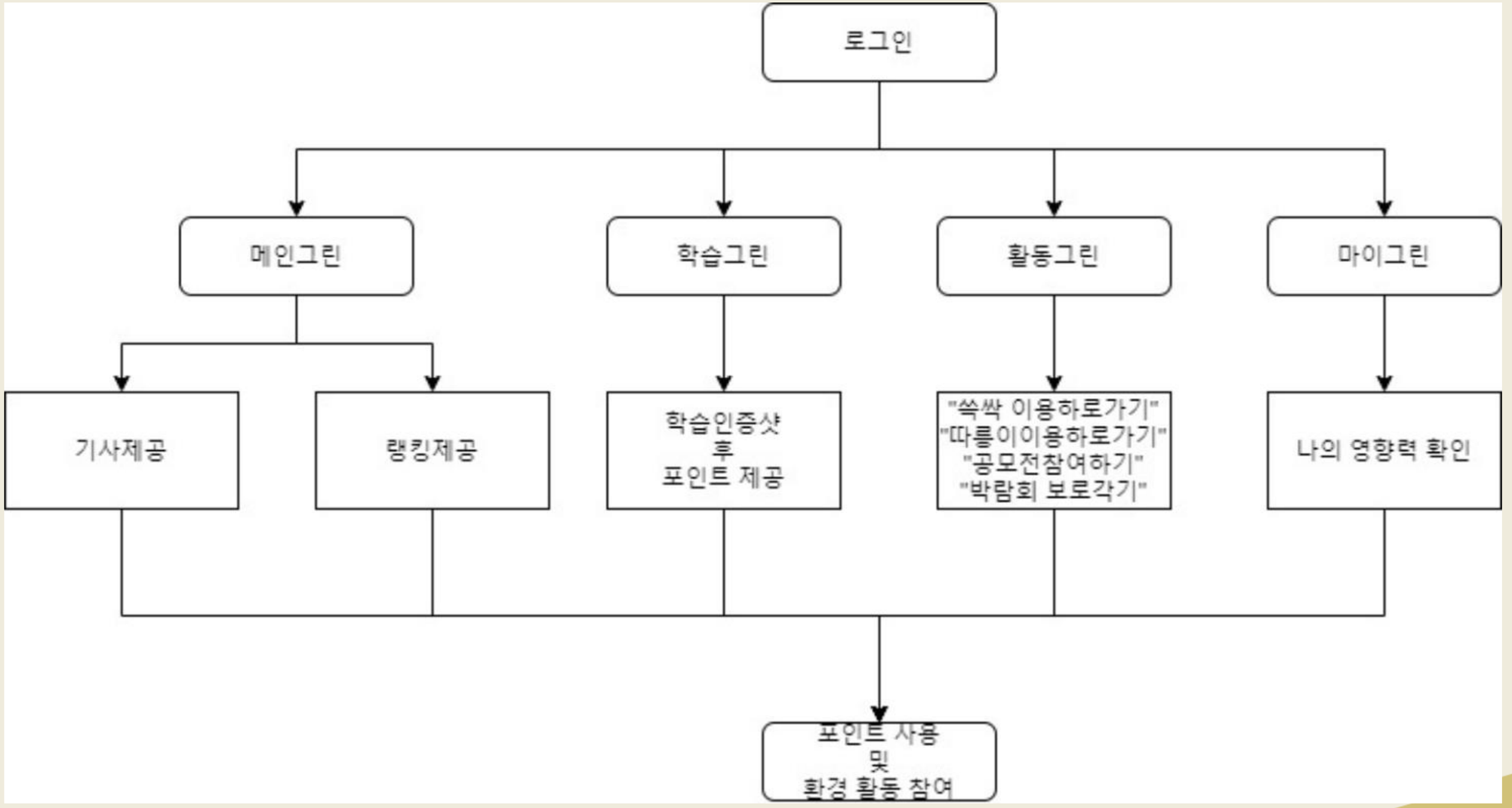

OOO님의 그린
43,360 G
전국 : 3600등
세계 : 133303033등
Lv 23 생태계 도우미
환경 관심가지기
'해수면 상승 가속화, 얼마 남지 않아 ....'
-OO일보-
'스마트 쓰레기통으로 바뀐 길거리...'
-xx속보-
'올해 발생한 이상기후, 쉽게 생각하면...'
-**일보-
+
메인 그린
학습 그린
활동 그린
마이 그린


43,360 G
Lv 23 생태계 도우미
주간
월간
연간
480G
분리수거 게임하기
학습하기
120G
환경 관련 유튜브 시청하기
학습완료
300G
환경 지식 퀴즈 풀기
학습 중
280G
환경 정보 공유하기
학습하기
370G
(심화) 환경 지식 퀴즈 풀기
학습하기
200G
환경 활동 OX 퀴즈
학습완료
+
메인 그린
학습 그린
활동 그린
마이 그린


43,360 G
Lv 23 생태계 도우미
쓱.싹.
이용하러가기
따릉이
이용하러가기
공모전
참여가기
박람회
보러가기
+
메인 그린
학습 그린
활동 그린
마이 그린


OOO님으로 인한 변화
XX 변화 수치
나무 **그루 형성
43,360 G
Lv 23 생태계 도우미
나의 G 이용하러 가기
+
메인 그린
학습 그린
활동 그린
마이 그린

3. 구현 방법 - 구조도


로그인
메인그린
학습그린
활동그린
마이그린
기사제공
랭킹제공
학습인증샷
후
포인트 제공
"쓱싹 이용하로가기"
"따릉이이용하로가기"
"공모전참여하기"
"박람회 보로각기"
나의 영향력 확인
포인트 사용
및
환경 활동 참여

4. 향후 계획

자가용 대체 수단 대상 확대

따릉이와 같은 일부 수단에 그치지 않고, 대중교통이나 전기차와 같이 자가용을 대체하거나 친환경적인 수단을 이용했을 때에도 포인트 대상에 포함

지역 환경 커뮤니티 기능 활성화

지역 환경과 관련된 커뮤니티 기능을 활성화하여 주변 쓰레기 발생 지역에 대한 문제 건의나 환경 관련 정보를 서로 공유할 수 있는 서비스를 제공

봉사시간 제공 서비스 도입

앱에서 제공하는 많은 활동에 참여하는학생들에게 봉사시간도 제공하여 환경적인 삶을 생활습관으로 얻을 수 있도록 유도

향후 등장하는 환경 서비스 추가

많은 환경 관련 콘텐츠와 활동을 쉽게 접할 수 있는 시스템이므로, 향후에 등장하는 참신한 환경 관련 서비스가 나왔을 때, 이를 연계하여 앱에서 사용하고 포인트를 제공받도록 추가

Reference

쓰레기 문제의 심각성
https://www.yna.co.kr/view/AKR20210719145400501

오픈소스 출처
https://deep-eye.tistory.com/18

https://thomapple.tistory.com/entry/YOLO-%EC%82%AC%EB%AC%BC%EC%9D%B8%EC%8B%9D-python

Appendix

https://github.com/kchsugo/opensource

<GitHub의 src/ 폴더를 실행>

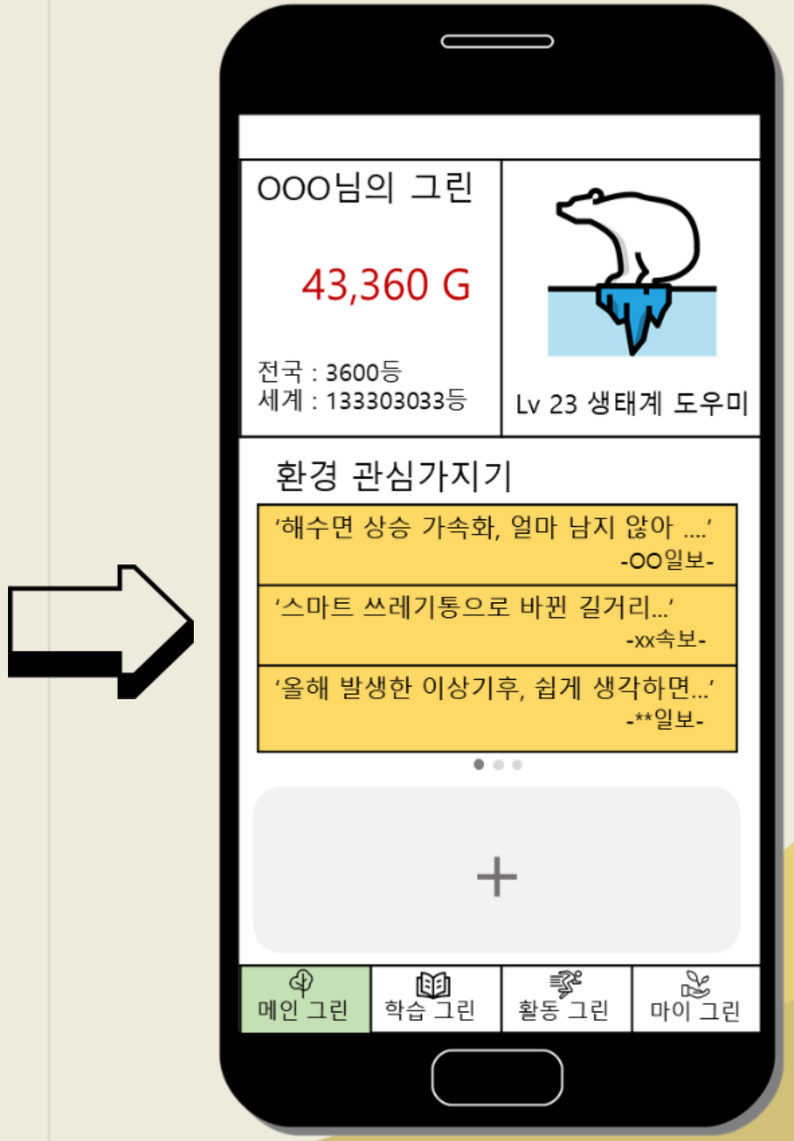
- naver_croll.ipynb파일은 네이버 기사를 크롤링하는 소스코드로 환경 관련 기사 추천에 활용
- person_open.ipynb파일은 스마트 쓰레기통에 들어갈 얼굴인식 소스코드
- recycle.ipynb파일은 스마트 쓰레기통에서 분리수거를 인식하는 소스코드

Appendix

<네이버 기사 크롤링 소스코드>

```python
import requests
from pandas import DataFrame
from bs4 import BeautifulSoup
import re
from datetime import datetime
import os

date = str(datetime.now())
date = date[:date.rfind(':')].replace(' ', '_')
date = date.replace(':','시') + '분'



query = input('검색 키워드를 입력하세요 : ')
news_num = int(input('총 필요한 뉴스기사 수를 입력해주세요(숫자만 입력) : '))
query = query.replace(' ', '+')


news_url = 'https://search.naver.com/search.naver?where=news&sm=tab_jum&query={}'

req = requests.get(news_url.format(query))
soup = BeautifulSoup(req.text, 'html.parser')


news_dict = {}
idx = 0
cur_page = 1

print()
print('크롤링 중...')
```

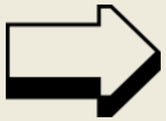
```python
while idx < news_num:
### 네이버 뉴스 웹페이지 구성이 바뀌어 태그명, class 속성 값 등을 수정함(20210126) ###

    table = soup.find('ul',{'class' : 'list_news'})
    li_list = table.find_all('li', {'id': re.compile('sp_nws.*')})
    area_list = [li.find('div', {'class' : 'news_area'}) for li in li_list]
    a_list = [area.find('a', {'class' : 'news_tit'}) for area in area_list]

    for n in a_list[:min(len(a_list), news_num-idx)]:
        news_dict[idx] = {'title' : n.get('title'),
                          'url' : n.get('href') }
        idx += 1

    cur_page += 1

    pages = soup.find('div', {'class' : 'sc_page_inner'})
    next_page_url = [p for p in pages.find_all('a') if p.text == str(cur_page)][0].get('href')

    req = requests.get('https://search.naver.com/search.naver' + next_page_url)
    soup = BeautifulSoup(req.text, 'html.parser')

print('크롤링 완료')

print('데이터프레임 변환')
news_df = DataFrame(news_dict).T
print(news_df)

folder_path = os.getcwd()
xlsx_file_name = '네이버뉴스_{}_{}.xlsx'.format(query, date)
news_df.to_excel(xlsx_file_name)

print('엑셀 저장 완료 | 경로 : {}₩₩{}'.format(folder_path, xlsx_file_name))
os.startfile(folder_path)
```



Appendix

<사람인식 소스코드>

```python
import numpy as np
import cv2

detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0)

while (True):
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

    cv2.imshow('frame', img)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```



앱 사용자가 직접 분리수거를 했는지 확인하기 위해서 쓰레기와 사람이 있는 사진에서 사람을 감지하여 앱을 악용하는 것을 방지할 수 있음

Appendix

<재활용 인식 소스코드>

```python
import numpy as np
import os
import six.moves.urllib as urllib
import sys
import tarfile
import tensorflow as tf
import cv2

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")

from utils import label_map_util
from utils import visualization_utils as vis_util

tf.reset_default_graph()
tf.get_default_graph()

# What model to download.
MODEL_NAME = 'ssd_mobilenet_v1_coco_11_06_2017'
MODEL_FILE = MODEL_NAME + '.tar.gz'
DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'

# Path to frozen detection graph. This is the actual model that is used for the object detection.
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'

# List of the strings that is used to add correct label for each box.
PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')

NUM_CLASSES = 90
```

```python
opener = urllib.request.URLopener()
opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)
tar_file = tarfile.open(MODEL_FILE)

for file in tar_file.getmembers():
    file_name = os.path.basename(file.name)
    if 'frozen_inference_graph.pb' in file_name:
        tar_file.extract(file, os.getcwd())

detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

def load_image_into_numpy_array(image):
  (im_width, im_height) = image.size
  return np.array(image.getdata()).reshape((im_height, im_width, 3)).astype(np.uint8)

# Size, in inches, of the output images.
IMAGE_SIZE = (12, 8)

with detection_graph.as_default():
    with tf.Session(graph=detection_graph) as sess:
        cam = cv2.VideoCapture(0)
```

```python
        if ret_val:
            # Expand dimensions since the model expects images to have shape: [1, None, None, 3]
            image_np_expanded = np.expand_dims(image, axis=0)
            image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

            # Each box represents a part of the image where a particular object was detected.
            boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

            # Each score represent how level of confidence for each of the objects.
            # Score is shown on the result image, together with the class label.
            scores = detection_graph.get_tensor_by_name('detection_scores:0')
            classes = detection_graph.get_tensor_by_name('detection_classes:0')
            num_detections = detection_graph.get_tensor_by_name('num_detections:0')

            # Actual detection.
            (boxes, scores, classes, num_detections) = sess.run(
                    [boxes, scores, classes, num_detections],
                    feed_dict={image_tensor: image_np_expanded})

            # Visualization of the results of a detection.
            vis_util.visualize_boxes_and_labels_on_image_array(
                    image,
                    np.squeeze(boxes),
                    np.squeeze(classes).astype(np.int32),
                    np.squeeze(scores),
                    category_index,
                    use_normalized_coordinates=True,
                    line_thickness=8)

            cv2.imshow('my webcam', image)

            if cv2.waitKey(1) == 27:
```

⇩

분리수거를 올바르게 했는지 판별함

Appendix

<환경 앱 구현 예시 - 프로토타입>

https://xd.adobe.com/view/2a2af5fe-5d1c-4a50-b32a-601fd292f6e8-7f79/?fullscreen