

AI+X 고급 (9주차)

산출물범위 확정 및 과제점검

Schedule: Onward to Christmas

11/3(8주차)	중간발표(KT 교수님 방문)
11/10(9주차)	모델링 개선 지속, 산출물 범위 확정
11/17(10주차)	과제점검 및 피드백(Stakeholder Map, Service Billboard)
11/24(11주차)	중간점검(KT 교수님 방문)
12/1(12주차)	과제점검 및 피드백
12/8(13주차)	과제점검 및 피드백
12/15(14주차)	최종발표 점검
12/22(15주차)	최종발표(KT 교수님 방문)

학점

Ideation 15점(P/F)

중간발표 20점

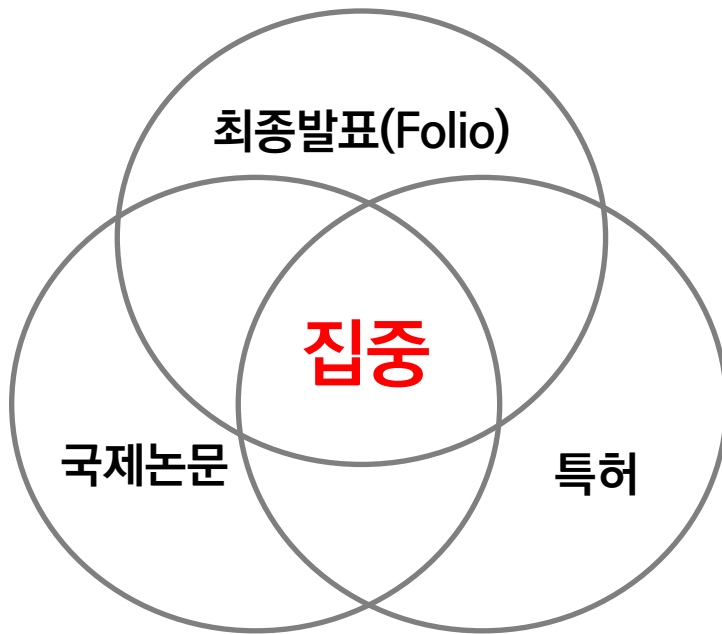
중간점검 15점(P/F) 11월 24일

최종발표 30점

참여도(태도, Peer Eval) 10점

출석 10점

Working Smart



서비스 디자인 어워드
추후 고민

최종발표

- 지금도 상당히 쉼이 올라옴.
- Stakeholder Map은 개선 필요.
 - . 돈의 흐름을 좀 더 선명하고 확실하게 드러나게끔! (벌금, 보상금, 보증금 등)
 - . 최종발표 시점에서는 “그럴 것 같다”를 지양, 확실하게!
 - . 목표: 직관적 Visualization, 설명 없더라도 한 눈에!
- 최종발표 데모 시나리오를 정할 것.
 - . 시나리오를 정하고 역순으로 개발범위 정할 것.
 1. 데모할 것(데모 범위)
 2. 개발했지만 데모하기 부끄(최종발표 개발주장 범위)
 3. 개발 안 했지만 시간만 있으면 개발 가능(특히 주장 범위)

특허

현 최근접 특허 (Nearest Patent)

공개 퍼스널 모빌리티 공유 서비스 시스템의 머신러닝을 이용한 불량 주차 검출 방법 및 장치 (Method and apparatus for detecting bad parking using machine learning of personal mobility sharing service system)

유사특허 N

공보

IPC : G06Q 50/30 G06Q 50/26 ...

출원인 : 주식회사 매스아시아

출원번호 : 1020220106614

출원일자 : 2022.08.25

등록번호 :

등록일자 :

공개번호 : 1020230032944

공개일자 : 2023.03.07

대리인 : 특허법인 신우

발명자 : 임현규 오승윤

열기

특허

기존 청구항 대비 현 발명 청구항 적기 시작합니다.

번호	청구항
1	다량의 주차사진을 수집하기 위한 주차사진 수집부;
	수집된 주차사진을 관리하여 학습데이터를 생성하는 학습데이터 생성부;
	생성된 학습 데이터를 이용하여 불량주차 분석모델을 생성하는 모델 학습부;
2	반납과정에서 주차사진을 수신하기 위한 주차사진 수신부;
	생성된 불량주차 분석모델에 의해 주차사진 수신부의 주차사진을 판독하여 불량주차 여부를 판별하는 불량주차 판별부;
	불량주차로 판단되면 사용자에게 주차 정정을 요청하거나 운영자에게 불량주차를 신고하는 자동신고부; 및 불량주차 분석 결과를 평가한 후 재학습을 통해 평가하기 위한 모델 갱신부를 포함하는 퍼스널 모빌리티 공유 서비스 시스템의 머신러닝을 이용한 불량 주차 검출 장치 .
3	제1항에 있어서, 상기 모델 학습부는
	스마트폰으로부터 수신한 퍼스널 모빌리티 주차 사진에 대해서, 상기 주차 사진을 기계학습의 입력으로 하고 불량 주차 여부를 판단하기 위한 머신러닝 의 학습 및 추론에 필요한, 퍼스널 모빌리티 공유 서비스 운영사의 자사 퍼스널 모빌리티, 퍼스널 모빌리티 공유 서비스 운영사의 자사 퍼스널 모빌리티 가 아닌 타사의 퍼스널 모빌리티 , 인도, 자전거 도로, 차도, 횡단보도, 시각 장애인을 위한 점자 보도블록, 엘리베이터 문, 소방시설, 육교 및 계단, 난간, 티널 내부, 교량, 버스 정류장, 택시 승차장 중 적어도 1개 이상의 객체에 대해, Object Detection이나 Instance Segmentation 중 적어도 1개 이상을 수행하도록 기존의 주차 사진에 대해 상기 객체의 Bounding box 혹은 Instance Segmentation 라벨링된 학습데이터를 입력하거나, 자동 라벨링(Auto-Labeling)을 수행해 인공지능 인공 신경망을 학습하도록 하는 객체 기계학습부;
	상기 주차사진만을 기계학습의 입력으로 하거나, 상기 주차 사진과 객체 정보를 기계학습의 입력으로 하고 불량 주차 여부 혹은 인도 중앙 주차, 자전거 도로 중앙 주차, 횡단보도 앞 주차, 시각 장애인용 점자 블록 위 주차, 엘리베이터 문 앞 주차, 소방시설 근처주차, 육교 및 계단 위 주차, 난간 위 주차, 티널 내부 주차, 교량 위 주차, 버스 정류장 앞주차, 택시 승차장 앞 주차 중 적어도 1개 이상의 불량 항목에 대해 각 항목에 해당하는지를 출력으로 하도록 인공지능망을 학습하도록 하는 불량 주차 기계학습부를 포함하는 퍼스널 모빌리티 공유 서비스 시스템의 머신러닝을 이용한 불량 주차 검출 장치 .

4	제1항에 있어서, 상기 자동 신고부는
	스마트폰에서 송신한 상기 주차 사진에 대해, 불량 주차 추론부에서 불량 주차로 추론한 경우거나, 상기 불량 주차 항목 각각에 해당하는지 추론했을 때 상기 항목 중 1개 이상에 해당할 경우에 사용되거나 퍼스널 모빌리티 공유 서비스 운영자에게 이를 즉시 알리고 대처해야 할 경우, 퍼스널 모빌리티 공유 서비스 운영자의 컴퓨터 및 스마트폰 등 스마트 기기에 불량 주차로 추론되거나 상기 불량 주차 항목 중 적어도 1개 이상에 해당하는 것으로 추론되는 퍼스널 모빌리티 의 고유 번호, 상기 퍼스널 모빌리티 의 GPS 좌표 정보, 마지막 사용자의 주차 위치, 수신한 상기 주차 사진, 불량 주차 추론부에서의 추론 결과 중 적어도 1개 이상을 송신하는 것을 특징으로 하는 퍼스널 모빌리티 공유 서비스 시스템의 머신러닝을 이용한 불량 주차 검출 장치 .
	퍼스널 모빌리티 공유 서비스 운영 시스템 에서 머신러닝 을 이용하여 불량 주차를 검출하기 위한 방법에 있어서,
5	사용자 스마트폰이 서비스 이용 후 반납을 위해 퍼스널 모빌리티 를 주차하고 스마트폰 애플리케이션을 통해 퍼스널 모빌리티 와 주변 환경을 함께 촬영해 전송하는 단계;
	상기 사용자 스마트폰으로부터 수신한 퍼스널 모빌리티 주차 사진에 대해서, 상기 주차 사진을 기계학습의 입력으로 하고 불량 주차 여부 판단하기 위한 머신러닝 의 학습 및 추론에 필요한, 퍼스널 모빌리티 공유 서비스 운영사의 자사 퍼스널 모빌리티, 퍼스널 모빌리티 공유 서비스 운영사의 자사 퍼스널 모빌리티 가 아닌 타사의 퍼스널 모빌리티 , 인도, 자전거 도로, 차도, 횡단보도, 시각 장애인용 점자 보도블록, 엘리베이터 문, 소방시설, 육교 및 계단, 난간, 티널 내부, 교량, 버스 정류장, 택시 승차장 중 적어도 1개 이상의 객체에 대해, Object Detection이나 Instance Segmentation 중 적어도 1개 이상을 수행하도록 기존의 주차 사진에 대해 상기 객체의 Bounding box 혹은 Instance Segmentation 라벨링된 학습데이터를 입력하거나, 자동 라벨링(Auto-Labeling)을 수행해 인공지능 인공 신경망을 학습하는 객체 기계학습 단계;
	상기 객체 기계학습 단계에 의해 학습된 상기 인공지능망을 이용해 사용자의 상기 주차 사진을 입력으로 하고, 상기 1개 이상의 객체에 대한 Object Detection 혹은 Instance Segmentation 추론을 진행하는 객체 추론 단계;
6	상기 주차 사진만을 기계학습의 입력으로 하거나, 상기 주차 사진과 객체 정보를 기계학습의 입력으로 하고 불량 주차 여부 혹은 인도 중앙 주차, 자전거 도로 중앙 주차, 횡단보도 앞 주차, 시각 장애인용 점자 블록 위 주차, 엘리베이터 문 앞 주차, 소방시설 근처주차, 육교 및 계단 위 주차, 난간 위 주차, 티널 내부 주차, 교량 위 주차, 버스 정류장 앞주차, 택시 승차장 앞 주차 중 적어도 1개 이상의 불량 항목에 대해 각 항목에 해당하는지를 출력으로 하도록 인공지능망을 학습하도록 하는 불량 주차 기계학습 단계;
	상기 불량 주차 기계학습부에 의해 학습된 상기 인공지능망을 이용해, 상기 주차 사진 혹은 상기 주차 사진과 객체 추론부에서 수신한 객체 정보를 입력으로 하여 불량 주차 여부 혹은 상기 불량 주차 항목을 중 1개 이상의 항목들에 대해 각 항목에 대한 불량 항목에 해당하는지를 추론하는 불량 주차 추론 단계; 및
	상기 사용자 스마트폰에서 송신한 상기 주차 사진에 대해, 불량 주차 추론부에서 불량 주차로 추론한 경우거나, 상기 불량 주차 항목 각각에 해당하는지 추론했을 때 상기 항목 중 1개 이상에 해당할 경우에 퍼스널 모빌리티 공유 서비스 운영자에게 이를 즉시 알리고 대처해야 할 경우, 퍼스널 모빌리티 공유 서비스 운영자의 컴퓨터 및 스마트폰 등 스마트 기기에 불량 주차로 추론되거나 상기 불량 주차 항목 중 적어도 1개 이상에 해당하는 것으로 추론되는 퍼스널 모빌리티 의 고유 번호, 상기 퍼스널 모빌리티 의 GPS 좌표 정보, 마지막 사용자의 주차 위치, 수신한 상기 주차 사진, 불량 주차 추론부에서의 추론 결과 중 적어도 1개 이상을 송신하는 자동 신고 단계를 포함하는 퍼스널 모빌리티 공유 서비스 시스템의 머신러닝을 이용한 불량 주차 검출 방법 .

특허

개발한 알고리즘
vs 다른 알고리즘

- 현 발명의 핵심 차별화? Distance 기반 정교한 탐지
- 가능한한 학기전 변리사님께
 1. 기존 최근접 대비 차별적 청구항
 2. 선행기술조사서(특허, 논문) (**복수개 필요**)
 3. 도면 기반 발명의 설명
- 차순위 특허? 해외특허?
- 담당자도 정합시다.

특허

- 주의사항

- . 특허는 차별적 아이디어를 주장하는 것이지만 반드시 개발을 요구하지는 X
- . 단, 아이디어가 **개발개연성**은 가지고 있어야함
; 일반적으로 될 수 있는걸 증명한 것(POC)을 개발하면서 씀

Development Plausibility

논문

- AI 서비스 논문으로 살짝 부족한 부분이 있음.
- AI 기술 논문으로도 경쟁력 있는 것이 아님.
- 주저자 정하고, 주도적으로 근접 논문 찾아야함.

. 해외논문 中心 다수 찾아야!

그런데 AI 성능평가만으로 X

. 핵심: “아무도 안 한 AI 서비스를 우리가 개발했는데 이렇게 의미(평가, 검증)가 있다.”

논문

- 지난 Plus One 저널 논문 대비 보완 필요함.
 - . 소비자/사용자는 없더라도 Stakeholder Interview는 필요해보임.
(Stakeholder Map은 당연히 필요)
 - . 학기내 할 수 있으면 좋고,
학기 종료 이후라도 필요해보임.
- 주저자(1저자) 선정합시다.

과제 현황

코드 리뷰

▼ 1108 - 11/10일 피드백

1. 피드백 공유

2. 해야할 일

a. 가성비

i. 3가지(최종 발표,특허,논문) 다 하는 것을 고려

b. 거리에 대한 생각

i. 예외의 경우

1. 키포드가 여러 대 있을 때의 경우

2. 같은 장소인데 사진이 여러 대 찍혀서 다른 장소라고 인식할 경우

- 신고 여러 번 되는 경우 방지해야 함

3. 키포드가 누워있는 경우

4. 거리 계산 구현 중 → 논문의 경우 현재 위치에서부터의 거리 측정

- 진행상황공유 → 조수환 → 11/9 토글

- 진행상황공유 → 노유진 → 1109거리계산 토글

c. 탐지의 범위를 넓히는것

i. 박스로 처리 애매, 횡단보도 근처 점자블록 필수 → 횡단보도는 제외

ii. 점자블록(segmentation)이나 지하철역(object detection)

- 점자블록의 경우 로보 플로우의 데이터로 학습, 결과 첨부

Exception Case는
List Up하면 됨

?

할 수 있으면 Good

```
# 킥보드 좌표 저장 및 이미지 내의 길이 비율 계산
```

```
if cls[0] == 0.0:
```

```
    scooter_cm = 120 높이 1.2m
```

```
    scooter_x_min = int(xyxy_list[0][0])
```

```
    scooter_x_max = int(xyxy_list[0][2])
```

```
    scooter_y_min = int(xyxy_list[0][1])
```

```
    scooter_y_max = int(xyxy_list[0][3])
```

```
    scooter_height = int(box.xywh.cpu().detach().numpy().tolist()[0][3])
```

```
    #scooter_half_height = int(scooter_height / 2)
```

```
    pixel_to_cm = scooter_height / scooter_cm
```

```
    pixel_ratio_array.append(pixel_to_cm) Pixel to cm Ratio 계산
```

```
else:
```

```
    # 킥보드가 아닌 객체의 좌표 저장
```

```
    non_scooter_boxes.append((int(xyxy_list[0][0]), int(xyxy_list[0][2])))
```

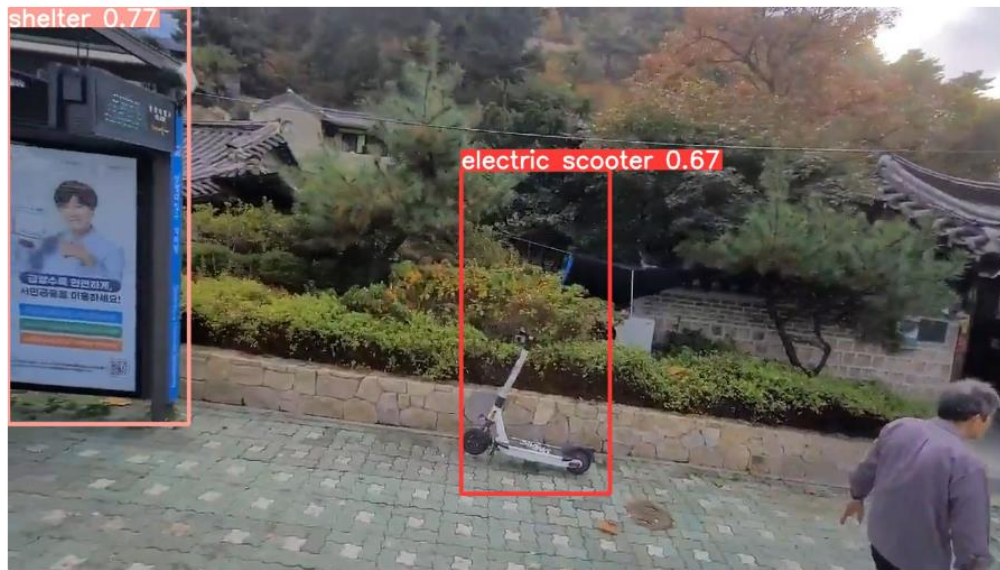
```
try:
    if scooter_height:
        # 동영상일 경우 평균 사용
        pixel_to_cm = int(np.mean(pixel_ratio_array))

        for idx, box_coords in enumerate(non_scooter_boxes):
            # 킥보드가 아닌 객체의 x좌표와 킥보드의 x좌표 비교
            x_calc_1 = abs(scooter_x_min - box_coords[0])
            x_calc_2 = abs(scooter_x_max - box_coords[1])
            x_calc_3 = abs(scooter_x_min - box_coords[1])
            x_calc_4 = abs(scooter_x_max - box_coords[0])

            min_pixel_distance.append(min(x_calc_1, x_calc_2, x_calc_3, x_calc_4))

        estimated_distance_cm = round(min(min_pixel_distance) / pixel_to_cm, 2)
        print(estimated_distance_cm) 최종 거리 추정

except:
    pass
```

이 이미지로 거리 계산 했음. 일단 이 이미지는 높이가 잘못 잡히긴 했지만 잘못 잡혔을 때를 기준으로 측정했을 때 거리가 116이 나온거면 대충 맞는듯? 킥보드 박스 높이랑 버정 사이의 거리랑 얼추 맞는 것 같은데..

◦ 추가 고려 사항

- 킥보드가 여러 대일 경우
- 킥보드가 누워 있을 경우 → if문 하나 추가
- 거리 직선 그리기
- 코드 최적화
- 버정과 킥보드가 겹쳐 있을 경우 → 어차피 거리 계산 결과가 작게 나올 것이므로 무관

개발 집중 우선순위 정할 필요

- 중요도(얼마나 빈번하게 발생하느냐)
- 최종데모 중심

```
In [27]: ### 각 객체의 좌표 추출 후 정류장과 킥보드 좌표 리스트 따로 생성 ###
def extract_coord(results):
    coord = results[0].boxes.xywh.numpy().tolist() # 좌표
    label = results[0].boxes.cls.numpy().tolist() # 클래스값

    for i in range(len(label)): # 좌표 리스트(x,y,w,h) 뒤에 클래스값 추가
        coord[i].append(label[i])

    # 정류장과 스쿠터 리스트 따로 만들
    busstop_coord = list()
    scooter_coord = list()

    # 정류장 / 스쿠터 리스트 분리
    for i in range(len(coord)):
        left_coord = list()
        right_coord = list()
        left_right_label = list()

        left_coord.append(coord[i][0] - (coord[i][2]/2)) #  $x - (w/2)$ 
        left_coord.append(coord[i][1] + (coord[i][3]/2)) #  $y + (h/2)$ 
        right_coord.append(coord[i][0] + (coord[i][2]/2)) #  $x + (w/2)$ 
        right_coord.append(coord[i][1] + (coord[i][3]/2)) #  $y + (h/2)$ 

        left_right_label.append(left_coord) # 바운딩 박스 왼쪽 아래 좌표
        left_right_label.append(right_coord) # 바운딩 박스 오른쪽 아래 좌표
        left_right_label.append(coord[i][4]) # 클래스 값
```

개인면담

- Career
- 이메일 주시기 바랍니다.
- 차주 화~목 상대적으로 좋습니다.