



(나)반 13팀

20221791 박수현

20221794 변진영

20221795 서민경



목차

Step 01

기존 문제점과 사례



Step 02

Idea



Step 03

기존 아이디어와의 차별성

Step 04

인공지능 모델 설정



Step 05

System Architecture



Step 06

구현

기존문제점 - 사례



사고를 낸 중학생 2명은 전동 킥보드를 타기 위해 필요한 원동기 면허를 소지하고 있지 않았으며 '2인 이상 탑승 금지' '인도 주행 금지' 규정 모두를 위반한 것으로 드러났다.

단속은 안전모 미착용과 2인 이상 탑승 등 위반 사항을 중심으로 매주 한 차례 이상 제주경찰청과 합동 단속에 나설 계획입니다.

네 번째, 안전을 위해 하나의 기기에는 한 명만 탑승해야 한다. 전동킥보드는 1인용으로 만들어졌다. 2인 이상 탑승 시 균형잡기가 어려워지는 것은 물론 위험성이 높아진다.
(2인 이상 탑승 시 범칙금 4만 원)

국민의힘 송석준(이천·사진) 의원은 14일 최근 폭증하는 전동킥보드 관련 사고를 예방하기 위한 제정법이 발의됐다. 이번 법 제

=> 단속
=> 범칙금 부과
=> 법 제정

**인력 낭비, 적은 범칙금,
원인 원천 차단 불가능**

기존문제점 - 사례



헬멧을 쓰지 않을 경우 범칙금 2만원이다. 2인 이상 탑승하면 범칙금 4만원이다. 면허를 소지하지 않은 채 탑승하면 과태료 10만원이다. 다만 모두 행정처분으로 형사입건 대상은 아니다. 헬

이어 "두바퀴 차는 신체가 노출돼 있어 차량과 충돌할 경우나 도로에 넘어졌을 때 생명이나 신체에 큰 피해를 입을 수 있다"며 "각종 교통법규를 준수하고 안전 장구를 반드시 착용하고 운행하는 것이 바람직하다"고 강조했다.

=> 헬멧 미착용을 적발하는 것이 아니라,
탑승 때부터 헬멧 미착용자를 차단하는 방법은 없을까?

기존문제점 - 사례



창원 시내 한 도로변 횡단보도에 전동 킥보드가 쓰러진 채 방치돼 있다.



◇27일 오전 원주시외버스터미널 앞 인도에는 주차공간에 있어야 할 공유형 전동킥보드 및 전기자전거가 널브러져 있었다.

신고 방법 미흡, 견인 조치 과정 불편, 주차구역 미흡

=> 주차 가능 구역을 인지해서 알려주는 기능, 통행에 방해가 되면 주차가 불가능하게 하는 기능을 만들 수는 없을까?



실제 사례 및 실행 가능성



- 주차 문제 AR 해결 사례
- 구글 AR코어 API 활용 증강현실 앱을 통해, 주변 환경 스캔하여 올바르게 주차되었는지 파악
- 주변 시설물을 기준으로 삼아 올바른 주차를 할 수 있도록 증강현실 어플 적용

- 헬멧 착용 감지 사례
- 애플리케이션이 설치된 단말기의 카메라를 이용해 헬멧 착용 여부 인식
- 사전에 학습시킨 딥러닝 모델 mobileNet v2 SSD 사용

- 발바닥 압력 센서
- 발바닥 압력 센서를 통해 감지된 발바닥 압력 값과 사용자를 매칭시키는 시스템
- 발바닥 압력 센서를 이용해 개인 사용자의 움직임까지 감지

- 자이로 센서를 이용한 기울어짐 감지
- 유모차에 아두이노를 기반으로 초음파센서를 이용해 기울어짐 감지
- 기준점 이상 기울어지면, 자동으로 제동을 걸게끔 구성됨

Ideas



1. 전동킥보드 탑승 인원 제어

- 전동킥보드 탑승 적정 인원을
압력센서 활용하여 판단



2. 전동킥보드 주차 개선

- AI의 판단으로 주차가능구역 세밀조정
- 자이로센서 활용 주차 개선



3. 전동킥보드 안전모 착용감지

- 안전모 착용 이미지 학습을
통한 헬멧 착용감지 시스템





기존 제품 및 아이디어와의 차별성

전동킥보드 정원 초과 탑승 방지를 위한 자동 제동 시스템

PDF 원문보기

- 국가/구분 한국(KR)/공개특허
- 국제특허분류 (IPC8판) B62L-003/00 IPC 상세 B62J-099/00 IPC 상세 B62K-003/00 IPC 상세
- 출원번호 10-2019-0027942 (2019-03-12)
- 공개번호 10-2020-0109009 (2020-09-22)
- DOI <http://doi.org/10.8080/1020190027942>
- 발명자 / 주소 이원석 / 경기 고양시 일산서구 가좌*로 **, (가좌동, 가좌마을*단지아파트)
- 출원인 / 주소 이원석 / 경기 고양시 일산서구 가좌*로 **, (가좌동, 가좌마을*단지아파트)
- 심사청구여부 있음 (2019-03-12)
- 심사진행상태 거절결정(일반)
- 법적상태 거절



크게보기

거절된 특허

- 무게 감지 센서를 이용하여
발판부에 감지되는 무게의 수 감지
- 접촉 인지 센서를 이용하여
발판부에 접촉되는 발의 수 인식



기존 제품 및 아이디어와의 차별성

보완 및 차별성

- 압전 센서의 거리가 짧아 발의 개수가 제대로 세어지지 않을 수 있음
- 일정 무게와 접촉상태인 센서의 개수만을 인식하므로 정확성이 떨어짐

➡ 발바닥 이미지를 학습시켜 탑승 인원 인지에 대한 정밀도 높임



기존 제품 및 아이디어와의 차별성

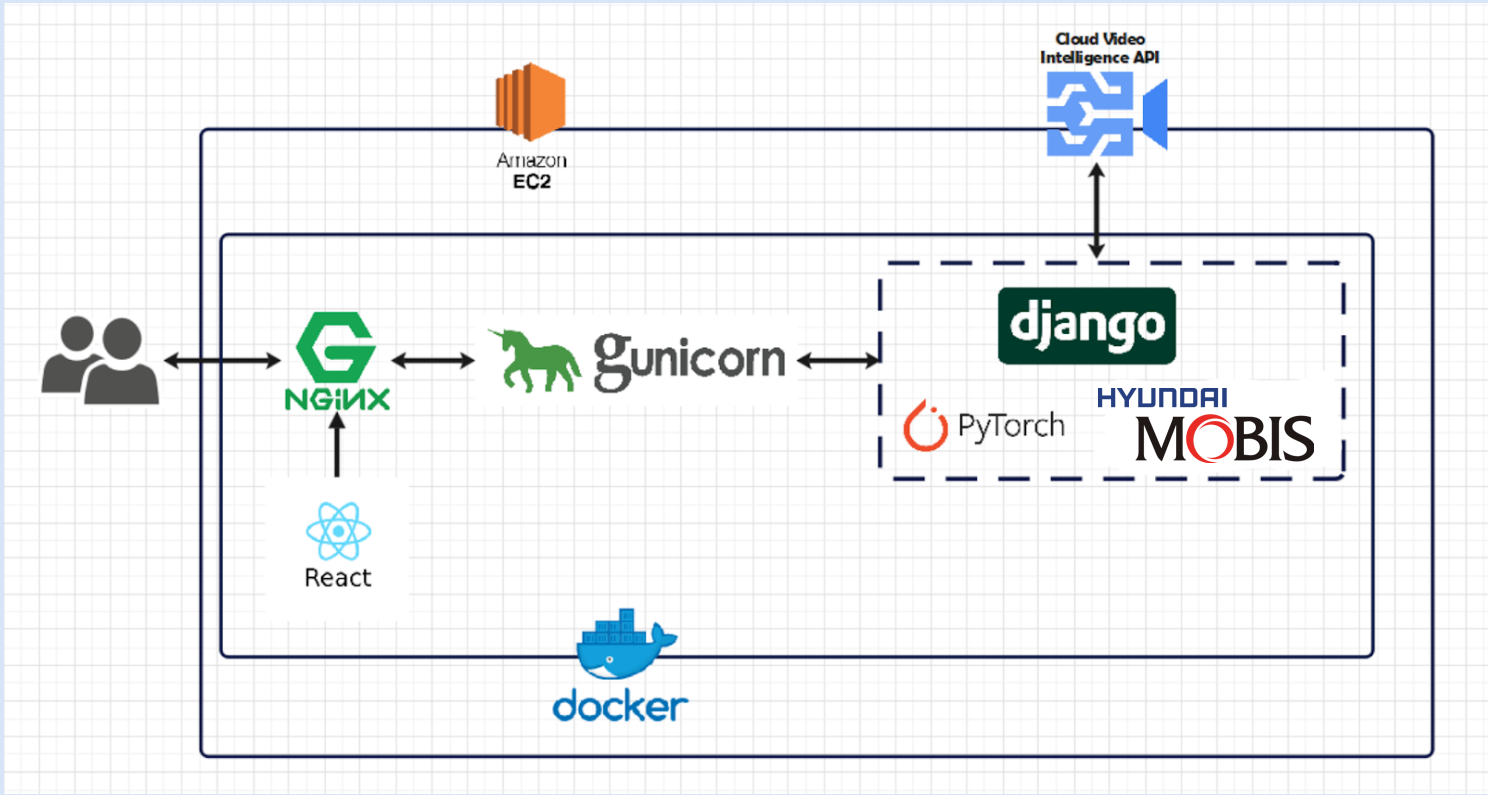
보완 및 차별성

- 전동킥보드 전용 주차구역을 설정해놓은 장소도 존재하지만
이는 단순히 구역만 설정해놓았을 뿐, 구역에 주차되었는지는 확인 X

➡ 주차 구역 안에 들어올 때만 주차가 완료되도록 발전시킴

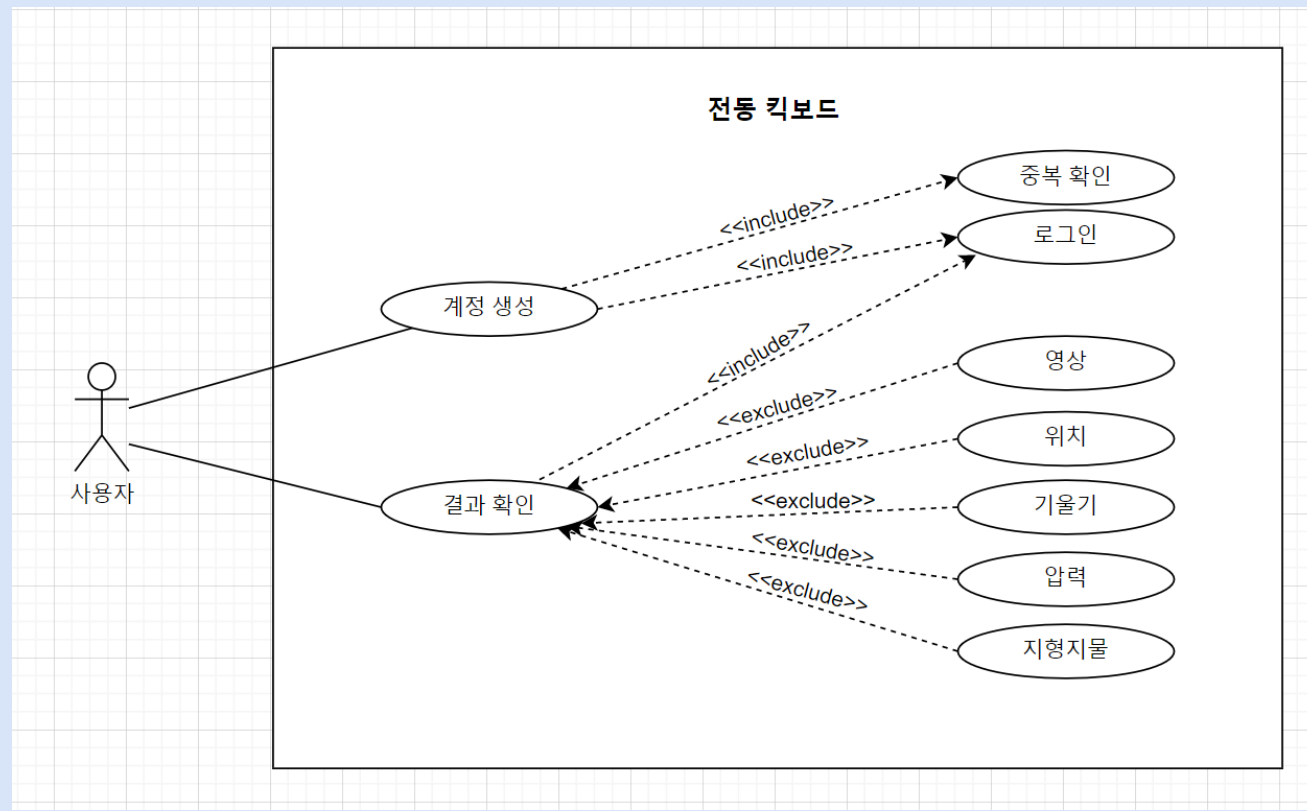


System Architecture

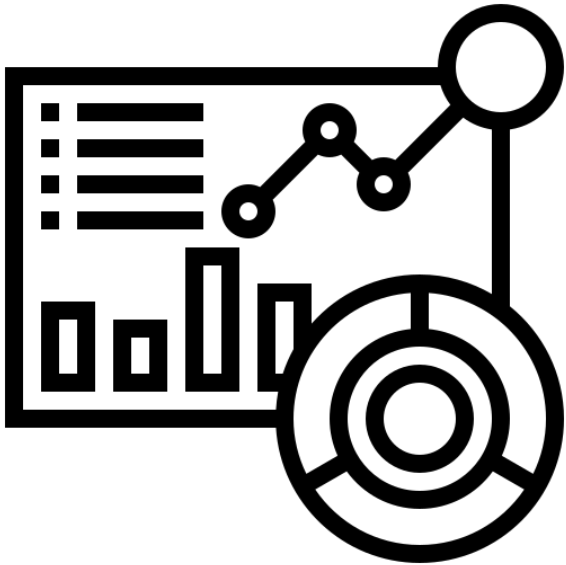




Use Case Diagram



데이터 수집



■ Labelling

- 그래픽 이미지 주석 도구
- 다양한 발바닥의 모양 및 헬멧을 쓴 형태의 많은 이미지들을 라벨링

■ Deep Residual Networks

- 이미지 인식을 위한 심층 잔여 학습
- 라벨링된 이미지들을 학습시키는 딥러닝 진행

인공지능 모델 설정

■ Google 얼굴 감지 AI (Cloud Vision Face Detection API)

- 이미지 내의 여러 얼굴의 감정 상태와 같은 주요 얼굴 속성을 함께 인식
- 특정 개인의 얼굴 인식은 지원되지 않아 개인정보 유출의 위험이 적음
- 얼굴 감지 방식: 얼굴 인식 요청 -> GCP(Google Cloud Platform) 프로젝트 인증 및 설정 -> 로컬 이미지에서 얼굴 인식 -> 원격 이미지에서 얼굴 인식

■ Google 동영상 AI (Cloud Video Intelligence API)

- 동영상에서 20,000개가 넘는 객체, 장소, 동작을 인식하는 동영상 분석
- 동영상, 샷 또는 프레임 수준의 다양한 메타데이터를 추출
- Video Intelligence API에는 이미 저장된 동영상과 스트리밍 동영상에서 수많은 객체, 장소 동작을 자동 인식하는 선행 학습된 머신러닝 모델

인공지능 모델 설정



■ 현대모비스 딥러닝 영상인식 기술

- 자동화 기법으로 영상데이터를 학습시킨 영상인식 인공지능
- 영상인식 기술의 핵심인 객체 인식 성능을 글로벌 선도업체와 대등한 수준으로 구현
- 차량, 보행자, 도로지형지물 등을 인식



■ Tensorflow

- 구글에서 공개된 딥러닝과 머신러닝 기술인 오픈소스 소프트웨어
- 발바닥 등 여러가지 이미지를 적용하여 딥러닝



```

1 int SensorPin0 = A0; //analog pin 0
2 int SensorPin1 = A1;
3 int SensorPin2 = A2;
4 int SensorPin3 = A3;
5 int SensorPin4 = A4;
6
7 void setup() {
8   // put your setup code here, to run once:
9   Serial.begin(9600);
10 }
11
12 void loop() {
13   // put your main code here, to run repeatedly:
14   int SensorReading0 = analogRead(SensorPin0);
15   int SensorReading1 = analogRead(SensorPin1);
16   int SensorReading2 = analogRead(SensorPin2);
17   int SensorReading3 = analogRead(SensorPin3);
18   int SensorReading4 = analogRead(SensorPin4);
19
20   int state_0;
21   int state_1;
22   int state_2;
23   int state_3;
24   int state_4;
25
26   int mfsr_r18 = map(SensorReading0, 0, 1024, 0, 255);
27   Serial.print(mfsr_r18);
28   Serial.print("\t");
29
30   int mfsr_r19 = map(SensorReading1, 0, 1024, 0, 255);
31   Serial.print(mfsr_r19);
32   Serial.print("\t");
33
34   int mfsr_r20 = map(SensorReading2, 0, 1024, 0, 255);
35   Serial.print(mfsr_r20);
36   Serial.print("\t");
37
38   int mfsr_r21 = map(SensorReading3, 0, 1024, 0, 255);
39   Serial.print(mfsr_r21);
40   Serial.print("\t");
41
42   int mfsr_r22 = map(SensorReading4, 0, 1024, 0, 255);
43   Serial.print(mfsr_r22);
44   Serial.print("\t");
45   Serial.print("\t");
46
47   if(mfsr_r18 > 100) state_0 = 1;
48   else state_0 = 0;
49   Serial.print(state_0);
50   Serial.print("\t");
51
52   if(mfsr_r19 > 100) state_1 = 1;
53   else state_1 = 0;
54   Serial.print(state_1);
55   Serial.print("\t");
56
57   if(mfsr_r20 > 100) state_2 = 1;
58   else state_2 = 0;
59   Serial.print(state_2);
60   Serial.print("\t");
61
62   if(mfsr_r21 > 100) state_3 = 1;
63   else state_3 = 0;

```

Output Serial Monitor X

Morsecode (Enter to send message to 'Arduino Uno' on 'COM7')

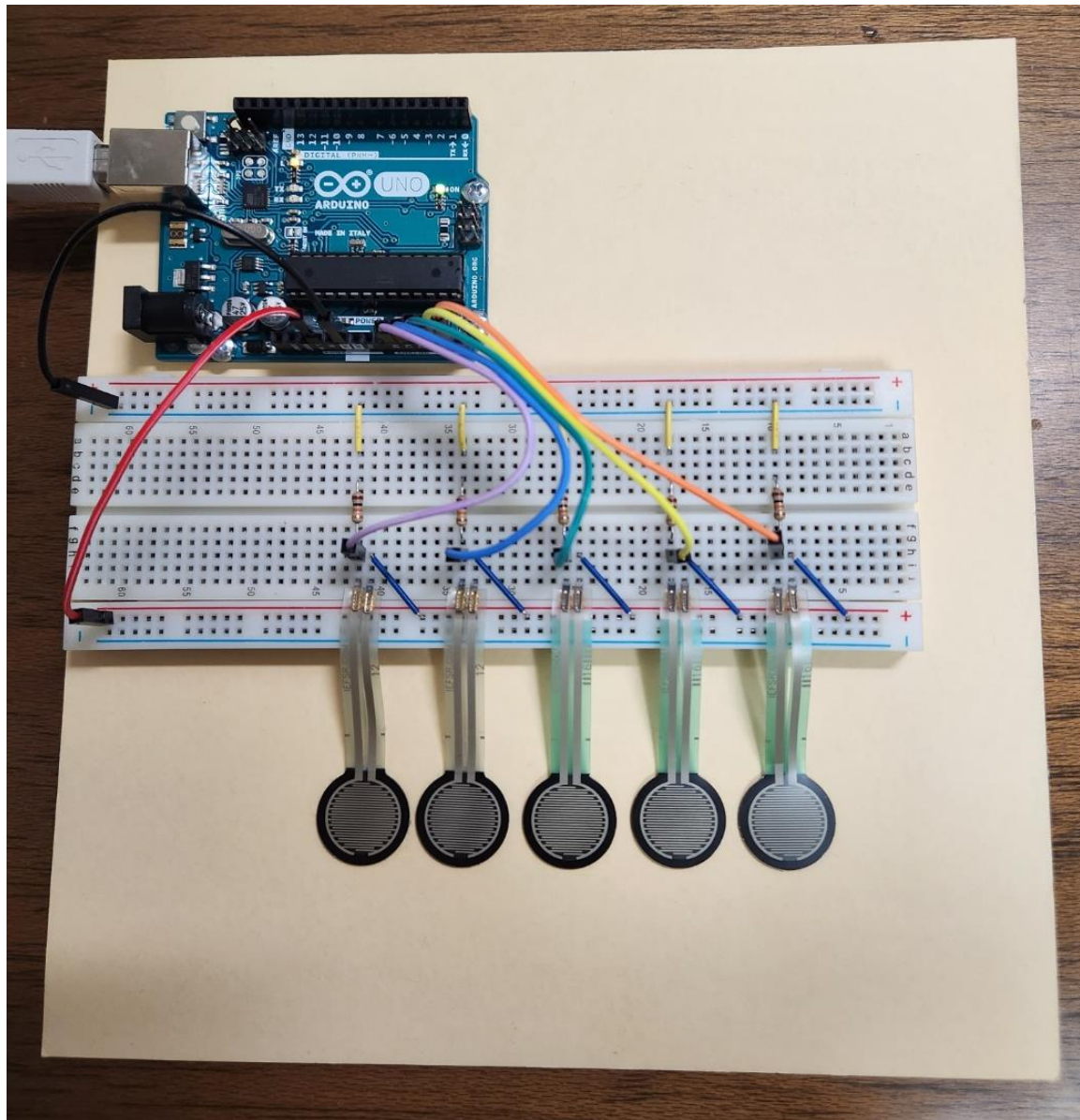
```

13 // put your main code here, to run repeatedly:
14 int SensorReading0 = analogRead(SensorPin0);
15 int SensorReading1 = analogRead(SensorPin1);
16 int SensorReading2 = analogRead(SensorPin2);
17 int SensorReading3 = analogRead(SensorPin3);
18 int SensorReading4 = analogRead(SensorPin4);
19
20 int state_0;
21 int state_1;
22 int state_2;
23 int state_3;
24 int state_4;
25
26 int mfsr_r18 = map(SensorReading0, 0, 1024, 0, 255);
27 Serial.print(mfsr_r18);
28 Serial.print("\t");
29
30 int mfsr_r19 = map(SensorReading1, 0, 1024, 0, 255);
31 Serial.print(mfsr_r19);
32 Serial.print("\t");
33
34 int mfsr_r20 = map(SensorReading2, 0, 1024, 0, 255);
35 Serial.print(mfsr_r20);
36 Serial.print("\t");
37
38 int mfsr_r21 = map(SensorReading3, 0, 1024, 0, 255);
39 Serial.print(mfsr_r21);
40 Serial.print("\t");
41
42 int mfsr_r22 = map(SensorReading4, 0, 1024, 0, 255);
43 Serial.print(mfsr_r22);
44 Serial.print("\t");
45 Serial.print("\t");
46
47 if(mfsr_r18 > 100) state_0 = 1;
48 else state_0 = 0;
49 Serial.print(state_0);
50 Serial.print("\t");
51
52 if(mfsr_r19 > 100) state_1 = 1;
53 else state_1 = 0;
54 Serial.print(state_1);
55 Serial.print("\t");
56
57 if(mfsr_r20 > 100) state_2 = 1;
58 else state_2 = 0;
59 Serial.print(state_2);
60 Serial.print("\t");
61
62 if(mfsr_r21 > 100) state_3 = 1;
63 else state_3 = 0;
64 Serial.print(state_3);
65 Serial.print("\t");
66
67 if(mfsr_r22 > 100) state_4 = 1;
68 else state_4 = 0;
69 Serial.print(state_4);
70
71 Serial.print("\r\n");
72
73 delay(100);
74 }
75

```

Output Serial Monitor X

Morsecode (Enter to send message to 'Arduino Uno' on 'COM7')



Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM7')

0	137	154	153	0	0	1	1	1	0
0	136	152	150	0	0	1	1	1	0
0	134	152	149	0	0	1	1	1	0
0	134	153	150	0	0	1	1	1	0
0	132	154	152	0	0	1	1	1	0
20	137	156	155	131	0	1	1	1	1
90	145	156	156	149	0	1	1	1	1
99	147	157	156	151	0	1	1	1	1
101	148	158	156	151	1	1	1	1	1
102	148	158	156	151	1	1	1	1	1
105	149	158	156	151	1	1	1	1	1
104	149	158	156	151	1	1	1	1	1
106	149	159	156	151	1	1	1	1	1
106	150	159	157	151	1	1	1	1	1
106	150	159	157	151	1	1	1	1	1
107	150	159	157	151	1	1	1	1	1
106	150	159	156	151	1	1	1	1	1
107	150	159	156	151	1	1	1	1	1
105	150	159	156	150	1	1	1	1	1
106	150	159	156	150	1	1	1	1	1
105	150	159	156	150	1	1	1	1	1
103	150	159	156	150	1	1	1	1	1
98	151	159	156	149	0	1	1	1	1
98	150	159	156	147	0	1	1	1	1
96	151	159	156	147	0	1	1	1	1
96	151	159	156	145	0	1	1	1	1
90	151	159	157	143	0	1	1	1	1
69	151	159	156	132	0	1	1	1	1
0	153	160	156	0	0	1	1	1	0
0	154	159	155	0	0	1	1	1	0
0	154	159	156	0	0	1	1	1	0
0	154	160	156	0	0	1	1	1	0
0	155	160	156	0	0	1	1	1	0
0	155	159	155	0	0	1	1	1	0
0	143	151	133	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	26	0	0	0	0	0	0	0	0
5	104	0	50	0	0	1	0	0	0
115	140	123	144	131	1	1	1	1	1
132	149	148	152	141	1	1	1	1	1
138	150	156	155	142	1	1	1	1	1
140	152	158	155	143	1	1	1	1	1
139	153	158	155	143	1	1	1	1	1
138	153	159	155	143	1	1	1	1	1
137	153	159	156	142	1	1	1	1	1
134	154	159	156	139	1	1	1	1	1
132	154	159	156	129	1	1	1	1	1
132	155	159	156	129	1	1	1	1	1
128	156	159	155	122	1	1	1	1	1
0	149	142	137	0	0	1	1	1	0



Arduino IDE 2.0.2 interface showing a sketch for an Arduino Uno. The sketch defines four sensor pins (A0, A1, A2, A3, A4) and reads their values in the loop function, mapping them to a range of 0 to 255. The Serial Monitor displays the output of the sketch, showing the mapped values for each sensor pin.

```
1 int SensorPin0 = A0; //analog pin 0
2 int SensorPin1 = A1;
3 int SensorPin2 = A2;
4 int SensorPin3 = A3;
5 int SensorPin4 = A4;
6
7 void setup() {
8   // put your setup code here, to run once:
9   Serial.begin(9600);
10 }
11
12 void loop() {
13   // put your main code here, to run repeatedly:
14   int SensorReading0 = analogRead(SensorPin0);
15   int SensorReading1 = analogRead(SensorPin1);
16   int SensorReading2 = analogRead(SensorPin2);
17   int SensorReading3 = analogRead(SensorPin3);
18   int SensorReading4 = analogRead(SensorPin4);
19
20   int state_0;
21   int state_1;
22   int state_2;
23   int state_3;
24   int state_4;
25
26   int mfsr_r18 = map(SensorReading0, 0, 1024, 0, 255);
27   Serial.print(mfsr_r18);
28   Serial.print("\t");
29
30   int mfsr_r19 = map(SensorReading1, 0, 1024, 0, 255);
31   Serial.print(mfsr_r19);
32   Serial.print("\t");
33
34   int mfsr_r20 = map(SensorReading2, 0, 1024, 0, 255);
35   Serial.print(mfsr_r20);
36   Serial.print("\t");
37
38   int mfsr_r21 = map(SensorReading3, 0, 1024, 0, 255);
39   Serial.print(mfsr_r21);
40   Serial.print("\t");
41
42   int mfsr_r22 = map(SensorReading4, 0, 1024, 0, 255);
43   Serial.print(mfsr_r22);
```

Serial Monitor Output:

Line	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10
0	137	154	153	0	0	1	1	1	0	
0	136	152	150	0	0	1	1	1	0	
0	134	152	149	0	0	1	1	1	0	
0	134	153	150	0	0	1	1	1	0	
0	132	154	152	0	0	1	1	1	0	
20	137	156	155	131	0	1	1	1	1	
90	145	156	156	149	0	1	1	1	1	
99	147	157	156	151	0	1	1	1	1	
101	148	158	156	151	1	1	1	1	1	
102	148	158	156	151	1	1	1	1	1	
105	149	159	156	151	1	1	1	1	1	
104	149	159	156	151	1	1	1	1	1	
106	149	159	156	151	1	1	1	1	1	
106	150	159	157	151	1	1	1	1	1	
106	150	159	157	151	1	1	1	1	1	
107	150	159	157	151	1	1	1	1	1	
106	150	159	156	151	1	1	1	1	1	
107	150	159	156	151	1	1	1	1	1	
105	150	158	156	150	1	1	1	1	1	
106	150	158	156	150	1	1	1	1	1	

```

int SensorPin0 = A0; //analog pin 0
int SensorPin1 = A1;
int SensorPin2 = A2;
int SensorPin3 = A3;
int SensorPin4 = A4;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  void loop() {
    // put your main code here, to run repeatedly:
    int SensorReading0 = analogRead(SensorPin0);
    int SensorReading1 = analogRead(SensorPin1);
    int SensorReading2 = analogRead(SensorPin2);
    int SensorReading3 = analogRead(SensorPin3);
    int SensorReading4 = analogRead(SensorPin4);

    int state_0;
    int state_1;
    int state_2;
    int state_3;
    int state_4;

    int mfsr_r18 = map(SensorReading0, 0, 1024, 0, 255);
    Serial.print(mfsr_r18);
    Serial.print("\t");

    int mfsr_r19 = map(SensorReading1, 0, 1024, 0, 255);
    Serial.print(mfsr_r19);
    Serial.print("\t");

    int mfsr_r20 = map(SensorReading2, 0, 1024, 0, 255);
    Serial.print(mfsr_r20);
    Serial.print("\t");

    int mfsr_r21 = map(SensorReading3, 0, 1024, 0, 255);
    Serial.print(mfsr_r21);
    Serial.print("\t");

    int mfsr_r22 = map(SensorReading4, 0, 1024, 0, 255);
    Serial.print(mfsr_r22);
    Serial.print("\t");
    Serial.print("\t");
  }
}

```

```

if(mfsr_r18 > 100) state_0 = 1;
else state_0 = 0;
Serial.print(state_0);
Serial.print("\t");

if(mfsr_r19 > 100) state_1 = 1;
else state_1 = 0;
Serial.print(state_1);
Serial.print("\t");

if(mfsr_r20 > 100) state_2 = 1;
else state_2 = 0;
Serial.print(state_2);
Serial.print("\t");

if(mfsr_r21 > 100) state_3 = 1;
else state_3 = 0;
Serial.print(state_3);
Serial.print("\t");

if(mfsr_r22 > 100) state_4 = 1;
else state_4 = 0;
Serial.print(state_4);

Serial.print("\r\n");

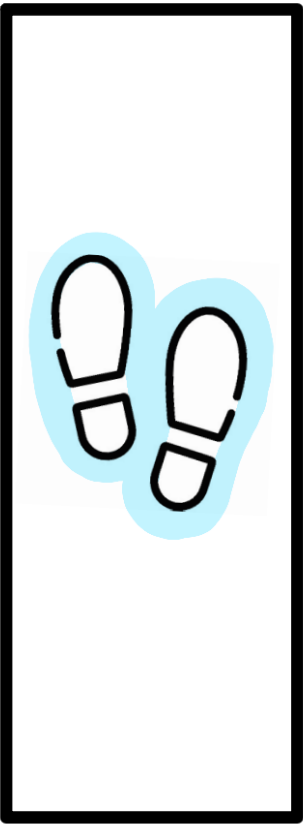
delay(100);
}

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM7')

0	137	154	152	0	0	1	1	0
0	136	152	150	0	0	1	1	0
0	134	152	149	0	0	1	1	0
0	134	153	150	0	0	1	1	0
0	132	154	152	0	0	1	1	0
20	137	156	155	13	0	1	1	1
90	145	156	156	14	0	1	1	1
99	147	157	156	15	0	1	1	1
101	148	158	156	15	1	1	1	1
102	148	158	156	15	1	1	1	1
105	149	158	156	15	1	1	1	1
104	149	158	156	15	1	1	1	1
106	149	159	156	15	1	1	1	1
106	150	159	157	15	1	1	1	1
106	150	159	157	15	1	1	1	1
107	150	159	157	15	1	1	1	1
106	150	159	156	15	1	1	1	1
107	150	159	156	15	1	1	1	1
105	150	159	156	15	1	1	1	1
106	150	159	156	15	1	1	1	1
105	150	159	156	15	1	1	1	1
103	150	159	156	15	1	1	1	1
98	151	159	156	149	1	1	1	1
98	150	159	156	147	0	1	1	1
96	151	159	156	147	0	1	1	1
96	151	159	156	145	0	1	1	1
90	151	159	157	143	0	1	1	1
69	151	159	156	132	0	1	1	1
0	153	160	156	0	0	1	1	0
0	154	159	155	0	0	1	1	0
0	154	159	156	0	0	1	1	0
0	154	160	156	0	0	1	1	0
0	155	160	156	0	0	1	1	0
0	155	159	155	0	0	1	1	0
0	143	151	132	0	0	1	1	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	26	0	0	0	0	0	0	0
5	104	0	50	0	0	1	0	0
115	140	123	144	13	1	1	1	1
132	149	148	152	141	1	1	1	1
138	150	156	155	142	1	1	1	1
140	152	158	155	143	1	1	1	1
139	153	158	155	143	1	1	1	1
138	153	159	155	143	1	1	1	1
137	153	159	156	142	1	1	1	1
134	154	159	156	139	1	1	1	1
132	154	159	156	129	1	1	1	1
132	155	159	156	129	1	1	1	1
128	156	159	155	122	1	1	1	1
0	149	142	137	0	0	1	1	0



0000000000
0000000000
0000000000
0000000000
0110000000
1111000110
1111001111
0111001110
0111001110
0110000110
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000



```
1  #!/usr/bin/env python
2
3  import argparse
4
5
6  # [START vision_sole_detection]
7  def detect_soles(path):
8      """Detects soles in an image."""
9      from google.cloud import vision
10     import io
11     client = vision.ImageAnnotatorClient()
12
13     # [START vision_python_migration_sole_detection]
14     with io.open(path, 'rb') as image_file:
15         content = image_file.read()
16
17     image = vision.Image(content=content)
18
19     response = client.s_detection(image=image)
20     soles = response.s_annotations
21     -----
22     likelihood_name = ('UNKNOWN', 'VERY_UNLIKELY', 'UNLIKELY', 'POSSIBLE',
23         |               | 'LIKELY', 'VERY_LIKELY')
24     print('soles:')
25
26     if response.error.message:
27         raise Exception(
28             '{}\nFor more info on error messages, check: '
29             'https://cloud.google.com/apis/design/errors'.format(
30                 response.error.message))
31     # [END vision_python_migration_s_detection]
32 # [END vision_s_detection]
33
34
35 # [START vision_s_detection_gcs]
36 def detect_soles_uri(uri):
37     """Detects soles in the file located in Google Cloud Storage or the web."""
38     from google.cloud import vision
39     client = vision.ImageAnnotatorClient()
40     # [START vision_python_migration_image_uri]
```

```
61 # [START vision_label_detection]
62 def detect_labels(path):
63     """Detects labels in the file."""
64     from google.cloud import vision
65     import io
66     client = vision.ImageAnnotatorClient()
67
68     # [START vision_python_migration_label_detection]
69     with io.open(path, 'rb') as image_file:
70         content = image_file.read()
71
72     image = vision.Image(content=content)
73
74     response = client.label_detection(image=image)
75     labels = response.label_annotations
76     print('Labels:')
77
78     for label in labels:
79         print(label.description)
80
81     if response.error.message:
82         raise Exception(
83             '{}\nFor more info on error messages, check: '
84             'https://cloud.google.com/apis/design/errors'.format(
85                 response.error.message))
86     # [END vision_python_migration_label_detection]
87 # [END vision_label_detection]
88
89
90 # [START vision_label_detection_gcs]
91 def detect_labels_uri(uri):
92     """Detects labels in the file located in Google Cloud Storage or on the
93     Web."""
94     from google.cloud import vision
95     client = vision.ImageAnnotatorClient()
96     image = vision.Image()
97     image.source.image_uri = uri
98
99     response = client.label_detection(image=image)
100    labels = response.label_annotations
```



UI / UX Design



 KickBoard

<https://bit.ly/3iivbIE>

출처



인터넷 자료

- 전동 킥보드 주차문제 해결방안 제시: <https://www.haevol.com/1>
- 전동 킥보드 관련 법안(주차구역 등): <https://blog.naver.com/cys0523/222680596165>
- 현대 모비스 딥러닝 영상인식 기술: <https://www.sisain.co.kr/news/articleView.html?idxno=34371>
- 전동 킥보드 관련 법안: <https://blog.naver.com/cys0523/222680596165>
- 발바닥 압력 센서 활용: KR20180024833A - 발바닥 압력 센서 및 그 제어 방법 - Google Patents
- 유모차 자이로센서 활용 기울어짐 방지 시스템: <https://scienceon.kisti.re.kr/srch/selectPORSrchArticle.do?cn=JAKO201724963131701>
- AR 활용 주차문제 해결 아이디어: <https://www.etnews.com/20220513000123>

논문

- 헬멧 착용 감지 논문”, 박준호, 황지민, 고유정, 김세하, 이현서. (2022). 딥러닝 기반의 전동킥보드 헬멧착용 인식시스템 개발, 한국컴퓨터정보학회 학술발표논문집, 30(1), 281-282, <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11037750>

출처



오픈소스

- 동영상 얼굴 감지 google API python 코드:
[python-videointelligence/video_detect_faces_gcs.py at6b8aec968ba874c6bdbe70737c3dd99adaef8a34 · googleapis/python-videointelligence · GitHub](#)
- 동영상 장면 변환 google API python 코드:
<https://github.com/googleapis/python-videointelligence/blob/HEAD/samples/analyze/analyze.py>
- Deep Residual Networks: [KaimingHe/deep-residual-networks: Deep Residual Learning for Image Recognition \(github.com\)](#)
- Labellmg: [heartexlabs/labellmg: Labellmg is now part of the Label Studio community. The popular image annotation tool created by Tzutalin is no longer actively being deveoped, but you can check out Label Studio, the open source data labeling tool for images, text, hypertext, audio, video and time-series data. \(github.com\)](#)

감사합니다