

“IoT 기술을 이용한 스마트 주거환경 서비스”

- 풀커버 (Full Cover) -

32151648 박동학

32150781 김승준

32152057 방승환

32155068 홍승기

2020. 04. 14

Index

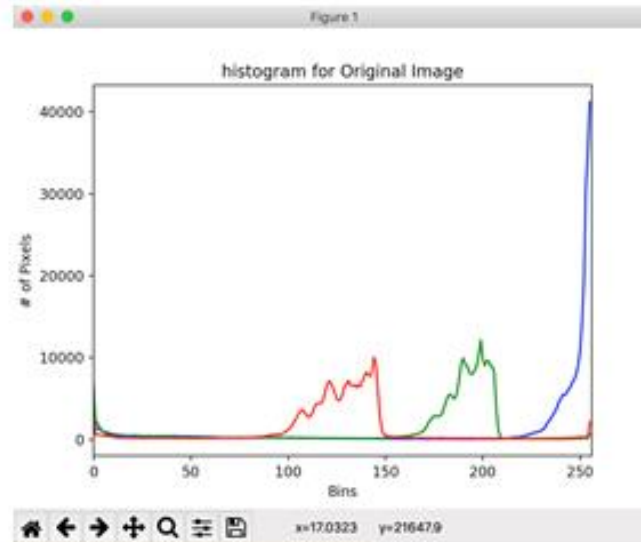
1. 구현 현황
2. Image Processing 기술 개요
3. Jetson nano 조립 및 OS 설치
4. 안면 인식 구현
5. 다음주 목표

1. 구현 현황

1. Jetson nano 조립 및 OS 설치
2. Open CV와 카메라 모듈을 이용한 안면인식
3. 프로젝트에 요구되는 추가 물품 발주

2. Image Processing 기술 개요

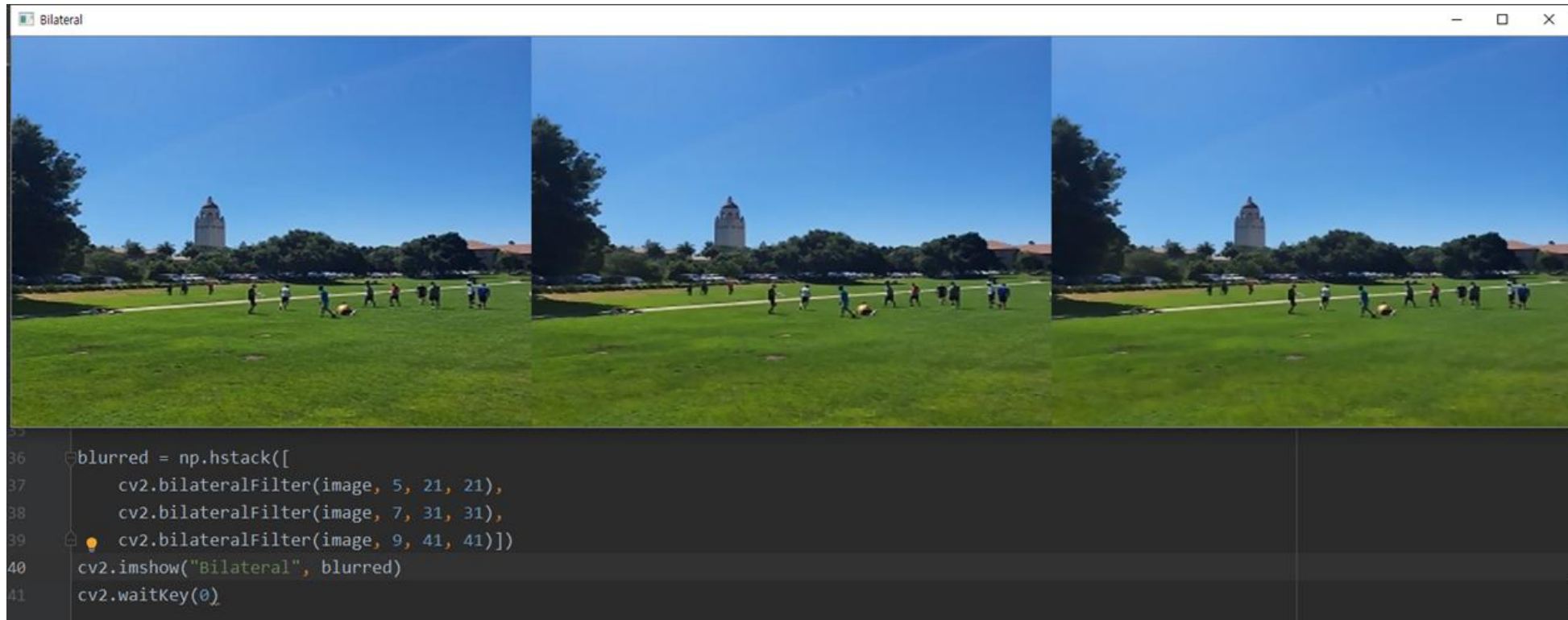
Histogram



```
1 from matplotlib import pylab as plt
2 import numpy as np
3 import argparse
4 import cv2
5
6 def plot_histogram(image, title, mask = None):
7     chans = cv2.split(image)
8     colors = ("b", "g", "r")
9     plt.figure()
10    plt.title(title)
11    plt.xlabel("Bins")
12    plt.ylabel("# of Pixels")
13    for (chan, color) in zip(chans, colors):
14        hist = cv2.calcHist([chan], [0], mask, [256], [0, 256])
15        plt.plot(hist, color = color)
16        plt.xlim([0, 256])
17
18    ap = argparse.ArgumentParser()
19    ap.add_argument("-i", "--image", required=True, help="Path to the image")
20    args = vars(ap.parse_args())
21    image = cv2.imread(args["image"])
22    cv2.imshow("Original", image)
23    plot_histogram(image, "histogram for Original Image")
24    mask = np.zeros(image.shape[:2], dtype="uint8")
25    cv2.rectangle(mask, (15, 15), (130, 100), 255, -1)
26    cv2.imshow("Mask", mask)
27    masked = cv2.bitwise_and(image, image, mask = mask)
28    cv2.imshow("Applying the Mask", masked)
29    plot_histogram(image, "Histogram for Masked Image", mask = mask)
30    plt.show()
```

2. Image Processing 기술 개요

Blur and Smoothing



2. Image Processing 기술 개요

Edge Detection

```
32 image = cv2.imread("./image1.jpg")
33 image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
34 image = cv2.GaussianBlur(image, (5, 5), 0)
35 cv2.imshow("Blurred", image)
36
37 canny = cv2.Canny(image, 30, 150)
38 cv2.imshow("Canny", canny)
39 cv2.waitKey(0)
```



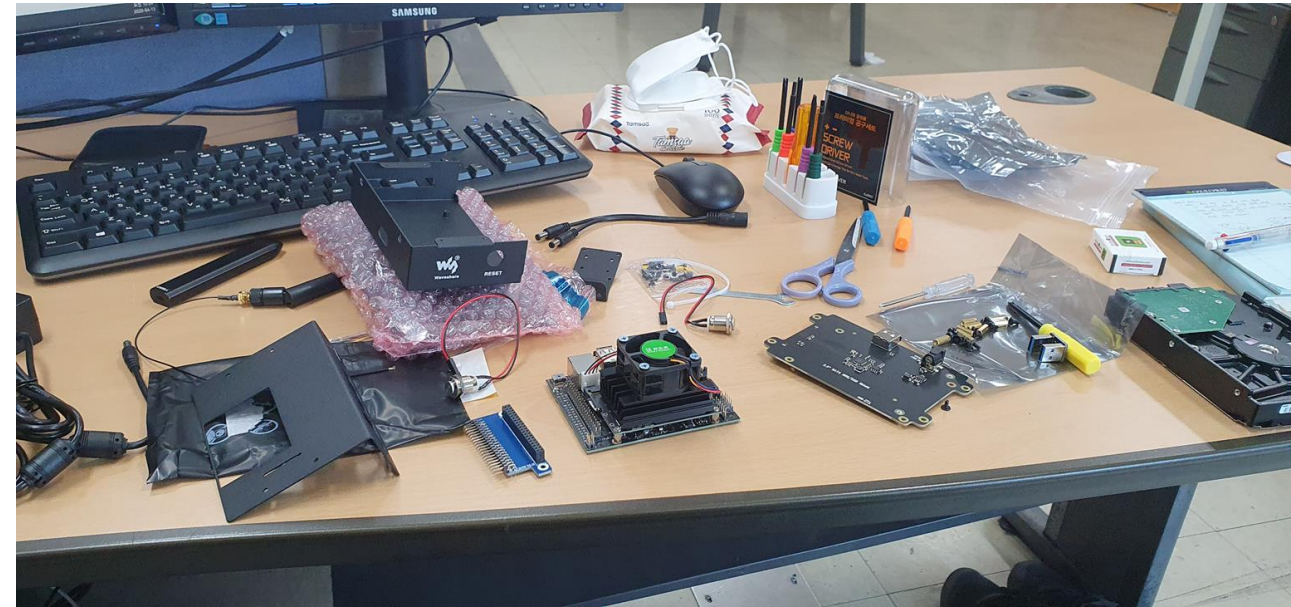
2. Image Processing 기술 개요

Contours

```
for (i, c) in enumerate(cnts): # enumerate 인덱스 번호와 리스트의 원소를 tuple형태로 반환
    (x, y, w, h) = cv2.boundingRect(c) # 컨투어를 인자로 받고 컨투어를 감싸는 직사각형을 반환
    print("Coin #%d" %(i+1))
    coin = image[y:y + h, x:x + w]
    cv2.imshow("Coin", coin)
    mask = np.zeros(image.shape[:2], dtype="uint8")
    ((centerX, centerY), radius) = cv2.minEnclosingCircle(c) # 컨투어를 인자로 받고 최소 외접원을 반환
    cv2.circle(mask, (int(centerX), int(centerY)), int(radius), 255, -1)
    mask = mask[y:y + h, x:x + w]
    cv2.imshow("Masked Coin", cv2.bitwise_and(coin, coin, mask=mask))
cv2.waitKey(0)]
```



3. Jetson nano 조립 및 OS설치



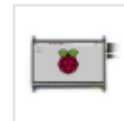
주문상품



무선 랜카드 안테나 + 케이블 [SZH-IWA013]

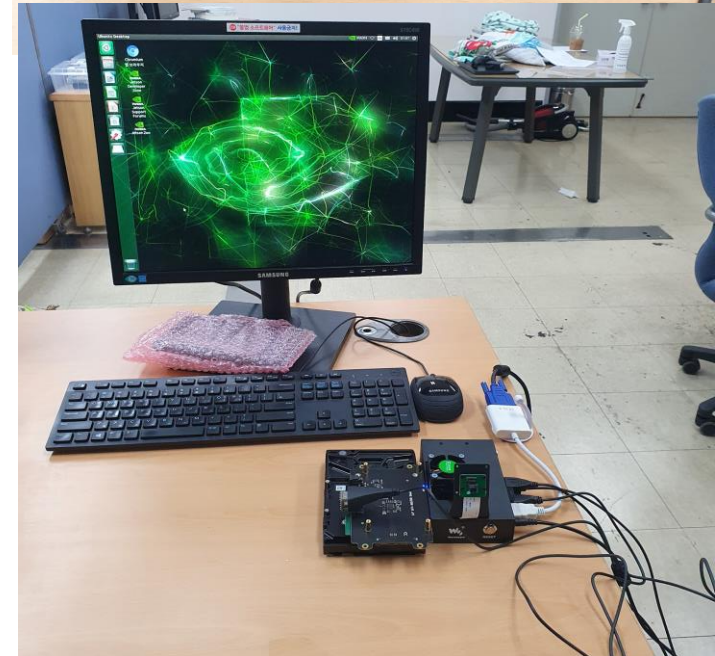


무선 랜 카드 intel 8260AC 867Mbps [SZH-IWA006]

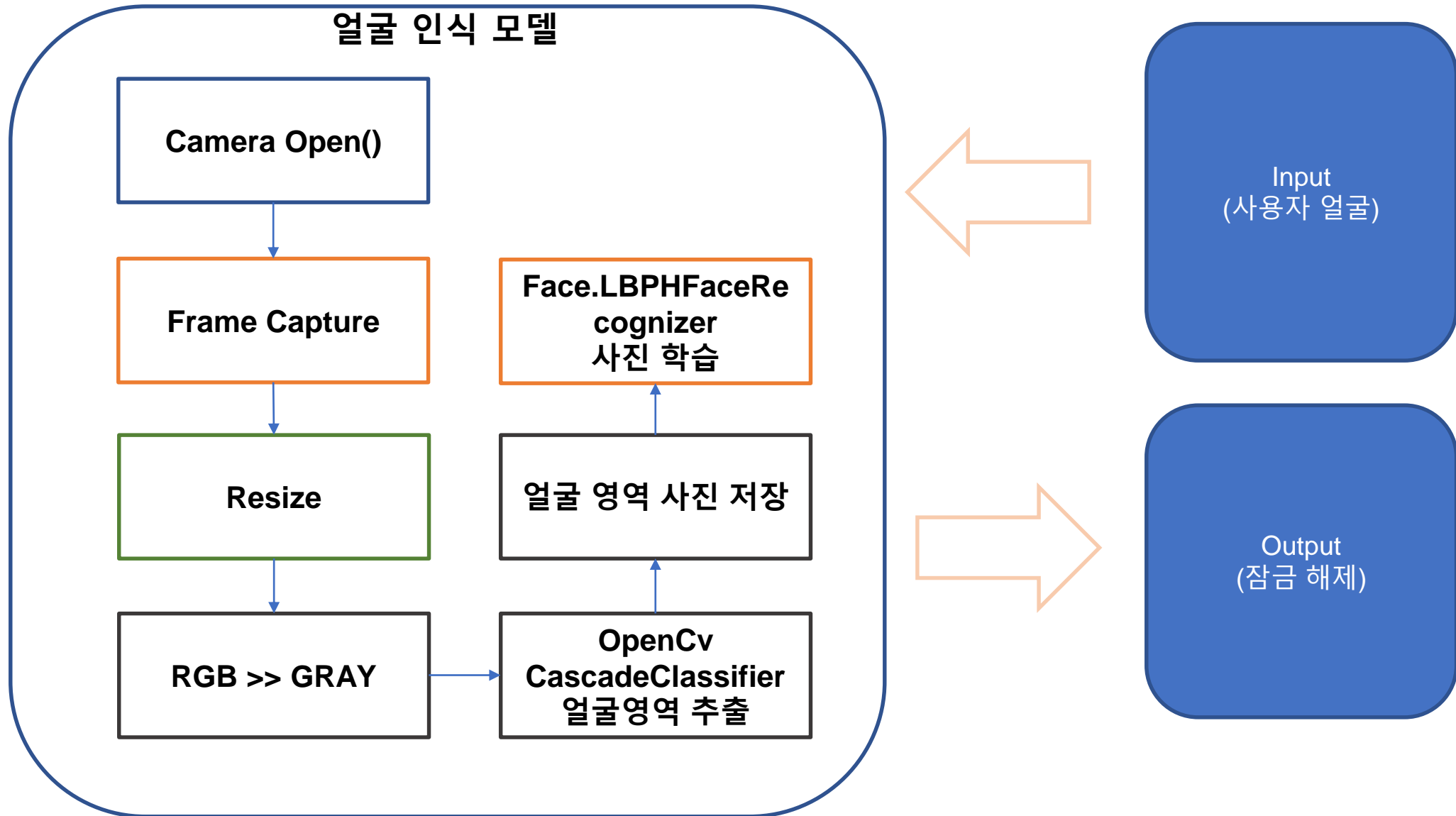


7인치 라즈베리파이 1024x600 HDMI 터치스크린 LCD / Raspberry Pi 7inch HDMI LCD (C)

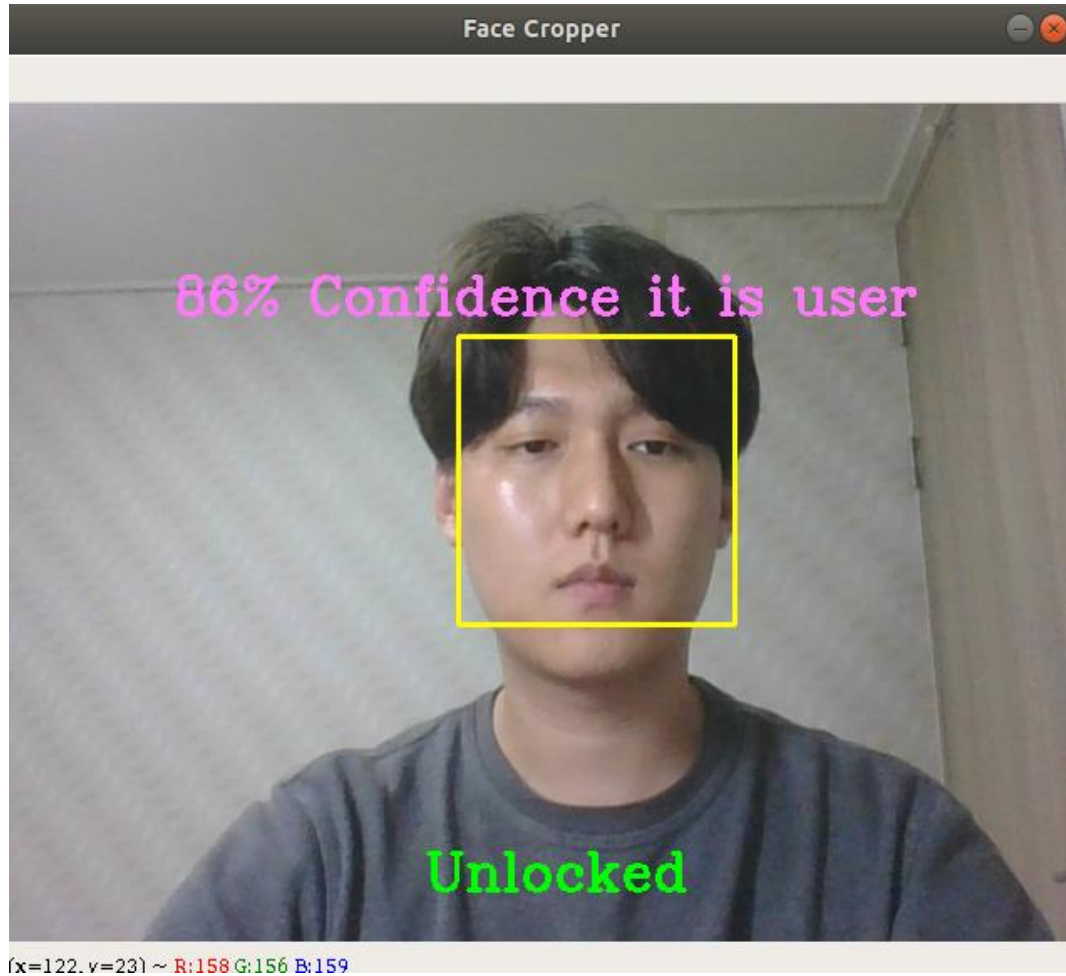
3. Jetson nano 조립 및 OS설치



4. 얼굴 인식 구현



4. 얼굴 인식 구현



- ❑ 리더 '박동학'의 얼굴 인식 과정
- ❑ Python opencv와 카메라 모듈을 활용하여 얼굴 인식을 구현

5. 다음주 목표

1. 안면 인식 정확도 개선 및 고도화
2. 번호판 인식 코드 Jetson Nano 탑재 후 테스트
3. 라즈베리 파이, 아두이노 연동 테스트

