

안면인식 CCTV와 홈 IOT의 통합 플랫폼 구현

안은모·김동회*

강원대학교 IT대학 전기전자공학부

Implementation of Integrated Platform of Face Recognition CCTV and Home IOT

Eun-Mo Ahn· Dong-Hoi Kim *

Kangwon National University, IT College, Electronic Electronics Engineering

[요 약]

기존의 안면인식 CCTV(Closed-circuit Television) 와 홈 IOT(Internet of Things)는 개별적인 각자의 하드웨어 요소를 가지고 있어 그들의 센서들과 CCTV의 측정 결과값을 동시에 한 화면에서 볼 수 없다는 단점을 가지고 있다. 본 논문에서는 기존 CCTV와 홈 IOT의 상기 단점을 극복하기 위하여 라즈베리파이를 이용하여 CCTV와 홈 IOT를 하나의 하드웨어 요소로 구성하고 그들의 측정 결과를 스마트폰 애플리케이션을 통해 한 화면에서 확인하는 하나의 통합 플랫폼을 제안한다. 제안하는 통합 플랫폼 CCTV와 홈 IOT 시스템은 애플리케이션을 스마트폰으로 실행하여 라즈베리파이에서 측정된 센서값과 파이카메라를 통해 촬영한 화면을 확인 할 수 있는 시스템이다. 구현된 시스템은 온도, 습도, 가스, 먼지를 측정하고 파이카메라를 통해 촬영한 화면에 안면인식 기술을 접목하여 스마트폰으로 한눈에 확인할 수 있게 하였다.

[Abstract]

As the existing face recognition CCTV and home IOT have each individual hardware component, they have a disadvantage that the measured results of their sensors and the CCTV can not be viewed on one screen at a time. In order to overcome the above disadvantages of existing CCTV and home IOT, this paper proposes an integrated platform which constitutes the CCTV and home IOT as one hardware component using Raspberry Pi and shows each result on one screen through Smartphone application. The proposed integrated platform CCTV and home IOT system is a system which can run the application as a Smartphone and check the sensor value measured by Raspberry Pi and the picture taken through the Pi camera. The implemented system measures temperature, humidity, gas, and dust, and implements face recognition technology on a screen shot through a Pi camera, allowing it to be seen at a glance with a Smartphone.

색인어 : 라즈베리파이, 안면인식, 파이카메라, CCTV, 홈 IOT, 애플리케이션

Key word : LoRa, Raspberry PI, Face Recognition, Pi Camera, CCTV, Home IOT Application

<http://dx.doi.org/10.9728/dcs.2018.19.2.393>



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 24 January 2018; Revised 12 February 2018

Accepted 26 February 2018

*Corresponding Author; Dong-Hoi Kim

Tel: +82-33-250-6349

E-mail: donghk@kangwon.ac.kr

I. 서론

CCTV와 홈 IOT 시스템을 탑재한 기기가 시중에 판매되고 있다. 시중 대부분의 CCTV와 홈 IOT는 독립적 구성되어 CCTV 따로 홈 IOT 따로 시스템을 구성하는 방식을 사용한다. 이러한 CCTV와 홈 IOT의 플랫폼이 있는데, LG 유플러스에서 출시한 ‘맘카’를 예로 들 수 있다. 맘카는 LG 유플러스에서 제공하는 서비스로 CCTV기기를 이용하여 사용자에서 영상을 확인할 수 있게 하며 녹화기능을 제공한다[1]. 또한, 홈 IOT 플랫폼 ‘IOT 움직임 알리미[2]’, ‘IOT 공기질 알리미’ 등으로 별개의 제품을 판매하고 단독 CCTV 플랫폼[3][4]과 홈 IOT 플랫폼 [5][6]에 관한 연구들도 있었다.

기존의 CCTV와 홈 IOT는 각각의 다른 기기를 사용하여 독립적으로 사용해야 하고 기존의 CCTV와 홈 IOT의 결과를 스마트폰을 통하여 볼 때 각각 다른 화면에서 개별적으로 확인을 해야하는 불편함이 있었다. 이러한 불편함을 극복하기 위해서 본 논문에서는 라즈베리파이 3를 이용하여 하드웨어적으로 하나의 기기로 CCTV와 홈 IOT를 하나의 통합 플랫폼으로 구성하여 한 화면에서 안면 인식한 CCTV와 각종 센서의 측정 값을 한눈에 확인 할 수 있게 애플리케이션을 구현하였다.

제안된 방법을 사용할 경우에 다른 여러 기기를 추가할 필요 없이 하나의 기기를 이용하여 CCTV와 홈 IOT를 구성했다는 장점을 가진다. 구체적으로 제안하는 CCTV와 홈 IOT의 통합 플랫폼은 라즈베리파이를 통해 온도, 습도, 먼지, 가스를 측정하고 피카메라를 통해 영상을 촬영한다. 촬영한 영상은 OpenCV의 Haar Cascade를 사용하여 얼굴을 검출한다. 스마트폰 애플리케이션을 통해 각각의 측정값과 영상을 한눈에 파악하고자 한다.

본 논문의 II장에서는 기존에 시중에 출시된 CCTV와 홈 IOT를 설명하고, III장에서는 제안하는 CCTV와 홈 IOT를 소개하며, IV장에서는 본 논문에서 제안된 방식으로 구현결과를 제시하고, 마지막 V장에서 결론을 맺는다.

II. 기존의 CCTV와 홈 IOT 플랫폼

2-1 기존 CCTV 플랫폼

LG 유플러스에서는 홈 CCTV로 ‘맘카’를 출시하였다. 맘카는 LG 유플러스의 홈 CCTV 서비스로 실시간 회선모니터링 및 야간에는 적외선 LED 모니터링 등을 제공하는 홈 CCTV 서비스이다. 실시간 모니터링, 상하좌우 회전카메라, 양방향 음성통화, 200만화소 화질, 적외선 LED 야간모니터링, 최대 25일 영상저장, 침입 알림, 겹쳐 및 간편녹화, 경보음 및 긴급통화, 외부 보기 차단, 동시접속, 간편 로그인, 최대 4대 동시 시청이 가능하다는 장점이 있다. 하지만 임대료 및 별도의 서비스 이용료를 요구한다. 맘카 이외에도 SK 브로드밴드의 Smart [Home] 안심캠 등이 존재하며 맘카와 같이 별도의 서비스 이용료 및 장비

값을 요구한다[7].

2-2 기존 홈 IOT 플랫폼

기존 홈 IOT는 여러 가지 기기들을 각각 독립적으로 판매하거나 패키지를 통해 필요한 기능들만 구성하여 판매하고 있다. ‘IOT 공기질 알리미’는 공기중의 미세먼지 수치와 습도 및 온도를 실시간으로 측정한다. 측정된 공기질 상태를 점수화 하여 좋음, 보통, 나쁨, 매우나쁨으로 나눠서 표기합니다. 표 1은 ‘IOT 공기질 알리미’의 기기 사양이다. ‘IOT 공기질 알리미’는 온도와 습도 먼지를 측정하여 사용자에게 알려준다[8].

표 1. 기존의 플랫폼 ‘IoT 공기질 알리미’ 사양
Table. 1. Existing platform ‘IoT air quality notification’ specification

Device name	IoT air quality notification
Model name	IAQ-LS100
Size	Width 106 X vertical 107 X height 60(mm)
Measurement sensor	Fine particulate matter (PM2.5), fine dust(PM10), temperature/humidity
Power	5V,2A
Communication	Wifi b/g/n 2.4Ghz

III. 제안하는 안면인식 CCTV와 홈 IOT의 통합 플랫폼

3-1 제안하는 통합 플랫폼의 하드웨어

홈 IOT의 하나인 ‘IOT 공기질 알리미’를 구성하기 위해 하나의 라즈베리파이에 온습도 센서, ADC, 가스센서, 먼지센서, 릴레이, Cooler fan을 이용하여 홈 IOT를 구성하고 안면인식 CCTV를 위해 그림 1와 같이 피카메라를 통해 CCTV를 구성하여 하나의 하드웨어로 CCTV와 홈 IOT를 같이 제공하는 새로운 안면인식 CCTV와 홈 IOT 통합 플랫폼을 제안하여 기존의 CCTV와 홈 IOT의 단점들을 해결하고 있다. 제안할 안면인식 CCTV와 홈 IOT이 기존 CCTV와 홈 IOT들과의 가장 큰 하드웨어 차이점은 각각 따로 존재하였던 CCTV와 홈 IOT를 라즈베리파이3에 통합하여 하나의 통합 플랫폼으로 구성했다는 장점이 있다.

3-2 제안하는 통합 플랫폼의 소프트웨어

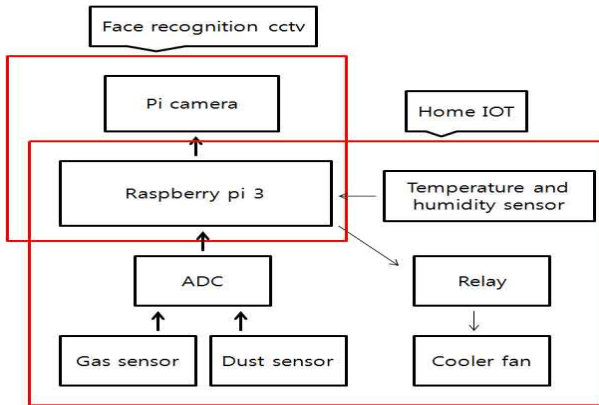


그림 1. 제안하는 통합 플랫폼의 하드웨어 구성도
 Fig. 1. Hardware configuration of the proposed Integrated platform

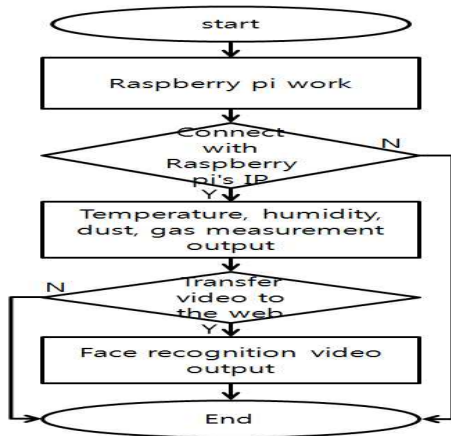


그림 2. 제안하는 통합 플랫폼의 동작 순서도
 Fig. 2. Operation flowchart of the proposed Integrated platform

그림 2와 같은 통합 플랫폼의 동작 순서도에 의해서 동작하는 새로운 안면인식 CCTV와 홈 IOT의 통합 플랫폼을 제안한다. SSH 원격제어를 통해 라즈베리파이를 작동시키고 스마트폰의 애플리케이션에 해당 라즈베리파이의 IP로 접속하여 해당 라즈베리파이가 측정한 온도, 습도, 먼지, 가스 측정값을 실시간으로 스마트폰 애플리케이션으로 출력하고 한번 더 SSH 원격제어를 통해 라즈베리파이가 웹으로 영상을 전송시키면 해당 웹페이지를 통해 안면인식 영상을 스마트폰 애플리케이션에 출력해준다. 스마트폰 애플리케이션은 하나의 화면에 온도와 습도, 먼지, 가스 측정값과 안면인식 영상을 한눈에 볼 수 있게 만들 것이다.

IV. 제안하는 통합 플랫폼의 구현결과 및 분석

표 2는 제안하는 통합 플랫폼의 구현 환경이다. 하드웨어는 라즈베리파이 3와 해당 센서들을 사용하였다. 라즈베리파이 3의 내부 언어로 Python[9]을 사용하였다. 스마트폰의 애플리케이션은 Android Studio[10]로 구현하였다.

표 2. 제안하는 통합 플랫폼의 구현 환경

Table. 2. Implementation environment of the proposed Integrated platform

Environment	Tool
Hardware	Raspberry Pi 3
Raspberry Pi internal language	Python
application	Android Studio

4-1 제안하는 통합 플랫폼의 하드웨어 구현

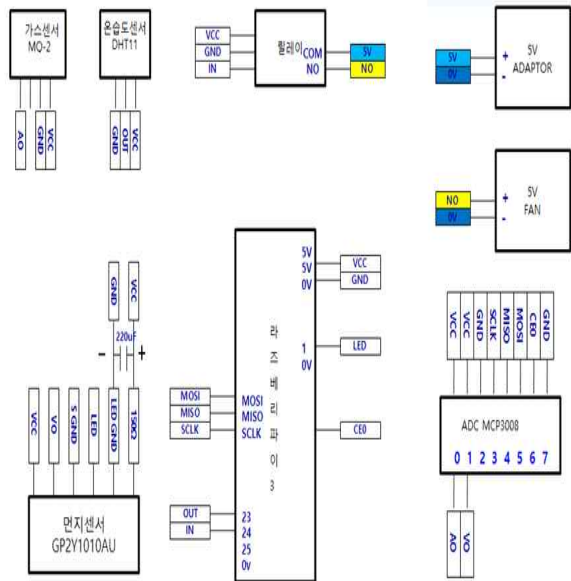


그림 3. 제안하는 통합 플랫폼의 하드웨어 연결

Fig. 3. Hardware connection of the proposed Integrated platform

그림 3과 같이 라즈베리파이 3를 중심으로 피카메라, 온도 센서(DHT11), 가스센서(MQ-2), 먼지센서(GP2Y1010AU), ADC(MCP3008), 릴레이, Cooler fan으로 구성된다.

습도 센서(DHT11)을 이용하여 온도와 습도를 측정하여 라즈베리파이로 바로 전달한다. 가스센서(MQ-2)를 이용하여 가스를 측정하고 먼지센서(GP2Y1010AU)를 통해 먼지를 측정하여 ADC(MCP3008)를 이용하여 아날로그 신호를 디지털 신호로 변환하여 라즈베리파이로 전달한다. 본 논문에서 ADC와 통

신하기 위해 SPI통신을 사용한다. 라즈베리파이는 3가지 타입의 시리얼 인터페이스를 GPIO 헤더를 통해 제공한다. I2C와 SPI, UART이다. SPI는 Serial Peripheral interface의 약자로 컴퓨터간의 데이터 통신을 위해 사용하는 것으로 센서, ADC, 다른 MCU와의 통신에 이용되며, 마스터 및 슬레이브로 나누어 상호간 통신을 할 수 있다. SPI 버스는 4가지 논리 신호를 가진다. SCLK는 직렬 클럭, MOSI는 마스터 출력과 슬레이브 입력, MISO는 마스터 입력과 슬레이브 출력, SS는 슬레이브 선택트이다. 라즈베리파이의 19번 PIN은 SPI의 MOSI, 21번 PIN은 SPI의 MISO, 23번 PIN은 SPI의 SCLK, 24번 PIN은 SPI의 SS이다. 이렇게 SPI통신으로 전달받은 먼지값이 일정값(0.5)이상일시 릴레이를 작동하여 Cooler Fan이 작동하게 하였다. 사용자는 안드로이드 스마트폰을 통해 애플리케이션으로 실시간 측정값을 확인할 수 있다. CCTV는 피카메라를 이용하여 구성하였다. 사전에 설정한 1초 간격으로 측정값을 수신하면 라즈베리파이에서 데이터를 통합 플랫폼의 애플리케이션으로 전달한다.

4-2 제안하는 통합 플랫폼의 소프트웨어 구현

1) 라즈베리파이 소프트웨어 구현

라즈베리파이의 소프트웨어를 구현하기 위해 5년 간 가장 폭발적으로 관심이 증가한 언어인 Python으로 구현하였다. 웹으로 영상을 보내기 위해 Mjpg-streamer을 이용하였다.

Mjpg-streamer는 라즈베리파이 카메라를 통해 jpg 촬영을 연속으로 수행해서 마치 영상처럼 보이도록 해주는 방법이다. 예를 들어 여러 가지 연속되는 그림을 그리고 그림들을 빠른 속도로 넘기면 정지된 이미지들이 연속적으로 넘어가면서 마치 영상물을 보는 것 같은 원리로 구현된다. 이 영상은 "http://RaspberryPi IP:8080/?action=stream"파일로 저장되는데 mjpg-streamer를 이용해 여러 장의 스틸샷을 웹으로 전송하여 사용자들이 영상처럼 볼 수 있도록 도움을 주는 툴이다[11].

안면인식 알고리즘으로 Haar feature-based Cascade 사용하였다. Haar feature-based Cascade 분류기를 이용한 객체 탐지는 폴로 비올라(Paul Viola)와 마이클존스(Michael Jones)의 논문 ("Rapid Object Detection using a Boosted Cascade of Simple Features" - 2001)에서 제안한 효과적인 객체 탐지 방법이다. 많은 수의 옹고 틀린 이미지들로 학습한 머신러닝이다[12].

그림 4은 안면인식 Python 소스로 먼저 import cv2를 로드한다. cap=cv2.VideoCapture(/dev/ video0)를 통해 카메라를 열고 해상도 320X240으로 설정한다. faceCascade=cv2.CascadeClassifier()를 이용하여 얼굴인식을 하기 위한 XML파일(harcascade_frontalf ace_default.xml)을 불러온다. cap.read()를 통해 영상을 읽고 cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)를 흑백으로 변경한다.

faceCascade.detectMultiScale(gray)로 얼굴을 검색하고cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)로 얼굴에 사각형 표시를 하고 cv2.imwrite("/home/pi/project/pic/asdf.jpg",frame)로

해당 위치에 저장한다. 그림 5은 제안하는 통합 플랫폼에 의해 안면인식전의 영상화면이고 그림 6은 Haar feature-based Cascade을 이용하여 안면 인식후의 영상 화면이다.

```
cap = cv2.VideoCapture('/dev/video0') #비디오장치 불러오기
cap.set(cv2.CAP_PROP_FRAME_WIDTH,320) #해상도 조정
cap.set(cv2.CAP_PROP_FRAME_HEIGHT,240)
faceCascade = cv2.CascadeClassifier('/home/pi/project/open
cv-3.2.0/data/haarcascades/haarcascade_frontalface_default
.xml') #얼굴인식을하기위한 xml파일읽어오기
while True:
    ret, frame = cap.read() #카메라읽어옴
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #흑백으로 변경
    faces = faceCascade.detectMultiScale(gray) #얼굴검색
    bigface = [0,0,0,0]
    for (x, y, w, h) in faces: #검출된 얼굴에 사각형 표기
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.imwrite("/home/pi/project/pic/asdf.jpg", frame)
```

그림 4. 라즈베리파이 안면인식 소스코드
Fig. 4. Source code of Raspberry pi face recognition



그림 5. 제안하는 통합 플랫폼에 의한 안면 인식전의 영상화면
Fig. 5. The image screen before face recognition by the proposed Integrated platform



그림 6. 제안하는 통합 플랫폼에 의한 안면 인식후의 영상화면
Fig. 6. The image screen after face recognition by the proposed Integrated platform

2) 애플리케이션 구현

애플리케이션은 Android Studio로 개발하였다. 구글이 안드로이드 앱 개발을 위해 JetBrains 사의 IntelliJ IDEA를 기반으로 만든 통합 개발 환경이다. 2014년 10월부터 이클립스를 대체하여 안드로이드 공식 IDE가 되었다. 안드로이드용 스마트폰 앱을 개발할 수 있어 본 작품에 사용하였다. 그림 7는 Activity가 실행될 때 호출되는 onCreate 소스이다. findViewById를 통해 뷰 인스턴스들을 초기화 해준다. NetworkUDPrev()로 수신객체를 선언하고 수신스레드를 선언하고 수신스레드를 시작해준다. connectBtn을 통해 “http://+ipaddress:8080/?action=stream”를 받아서 안면인식된 CCTV화면을 애플리케이션에 출력한다. 그림 8은 수신클래스 소스로 9000번의 포트번호를 이용하여 라즈베리파이로부터 패킷을 수신한다. 수신한 패킷을 통해 가스, 먼지, 온도, 습도를 저장하여 애플리케이션에 출력한다.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    humidity = (TextView)findViewById(R.id.humidity);
    gas = (TextView)findViewById(R.id.gas);
    temp = (TextView)findViewById(R.id.temp);
    dust = (TextView)findViewById(R.id.dust);

    webView = (WebView)findViewById(R.id.webView);
    webView.setWebViewClient(new WebViewClient()); // 이걸 안해주면 새창이 뜬다
    // webView.loadUrl("http://192.168.0.97:8080/?action=stream");
    ipAddress = (EditText)findViewById(R.id.ipAddress);
    connectBtn = (Button)findViewById(R.id.connectBtn);

    try{
        socket = new DatagramSocket(9000); //소켓통신을 위한 소켓 생성 포트 9000설정
        Log.e("sock1", "sock1");
    }
    catch (Exception ex) {Log.e("fail", "sock1");}

    udprev = new NetworkUDPrev(); //수신객체 선언
    cThreadrev = new Thread(udprev); //수신스레드 선언
    cThreadrev.start(); //수신스레드 시작

    connectBtn.setOnTouchListener((v, event) -> {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            (new Thread(new NetworkUDPrev("o"))).start();
            webView.loadUrl("http://"+ipAddress.getText().toString()+"*8080/?action=stream");
            return true;
        }
        return false;
    });
}
```

그림 7. onCreate 소스
Fig. 7. onCreate source code

```
public class NetworkUDPrev implements Runnable { //수신 클래스
    int flag = 0;
    public void run() {
        try {
            int port = 9000;
            while (true) {
                byte[] buf = new byte[1024];
                //ods.setText("");
                DatagramPacket packet = new DatagramPacket(buf, buf.length, serverAddrrev, port);
                //(MainActivity) getActivity().socket.receive(packet); //서버와 포트로 만들어진 패킷을 수신한다
                socket.receive(packet); //서버와 포트로 만들어진 패킷을 수신한다
                msg = new String(packet.getData(), 0, packet.getLength());
                Log.e(msg, msg);
                switch(msg.toCharArray()[0]) {
                    case 'g':
                        gas = msg.substring(1, msg.length());
                        break;
                    case 'h':
                        humidity = msg.substring(1, msg.length());
                        break;
                    case 't':
                        temp = msg.substring(1, msg.length());
                        break;
                    case 'd':
                        dust = msg.substring(1, msg.length());
                        dustTxt.post() -> {
                            dustTxt.setText(dust);
                            gasTxt.setText(gas);
                            tempTxt.setText(temp);
                            humidity.setText(humidity);
                        };
                        break;
                }
            }
        }
    }
}
```

그림 8. 수신클래스 소스
Fig. 8. Receive Class source code

4-3 구현 결과 및 분석

그림 9는 제안하는 안면인식 CCTV와 홈 IOT의 통합 플랫폼을 위에서 바라본 모습이다. 사용자는 컴퓨터의 SSH를 통해 라즈베리파이를 작동시키면 온도 센서와 가스센서, 먼지센서가 작동하여 측정을 시작하고 스마트폰 애플리케이션으로 접속하길 대기한다. 만약 스마트폰으로 접속을 한다면 측정된 온도, 습도, 가스, 먼지를 1초 간격으로 스마트폰으로 전달해준다. 또한 피카메라로 촬영한 영상을 웹스트리밍으로 전달하고 해당 웹페이지를 통해 애플리케이션으로 해당 영상을 실시간으로 출력해준다. 그림 10은 기존의 플랫폼인 ‘맘카’의 실행 화면이고 그림 11은 기존의 플랫폼인 ‘IOT 공기질 알리미’의 실행 화면이다.

그림 12는 제안하는 안면인식 CCTV와 홈 IOT 통합 플랫폼의 애플리케이션을 스마트폰에서 실행한 화면이다. 표3은 구현된 안면인식 CCTV와 홈 IOT의 기능과 기존의 ‘맘카’, ‘IOT 공기질 알리미’의 장단점을 비교 분석한 표를 보여주고 있다. 기존의 CCTV와 홈 IOT가 별개로 사용해야 했지만 본 논문에서는 CCTV와 홈 IOT를 하나의 하드웨어와 소프트웨어가진 통합 플랫폼으로 구현시킴으로써 그 단점을 탈피하였고, 기존 ‘IOT 공기질 알리미’로 측정되지 않은 가스를 가스센서를 통해 측정할 수 있게 하였다.

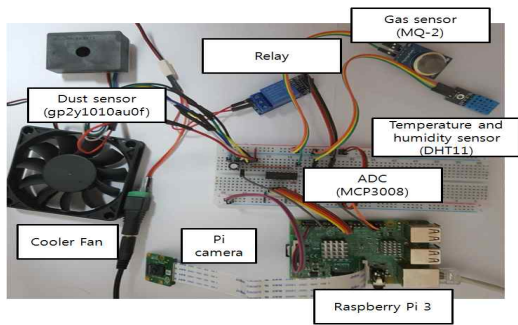


그림 9. 제안한 CCTV와 홈 IOT 통합 플랫폼의 전체 구현 사진
Fig. 9. Full Implementation picture of the proposed CCTV and Home IOT Integrated platform

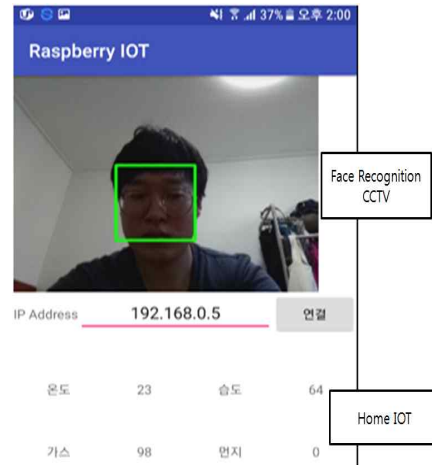


그림 12. 제안한 안면인식 CCTV와 홈 IOT 통합 플랫폼의 스마트폰 애플리케이션에 의한 사용화면
Fig. 12. Use screen by smartphone application of the proposed face recognition CCTV and home IOT Integrated platform



그림 10. 기존의 플랫폼인 '맘카' 사용화면
Fig. 10. 'Mamca' use screen which is the existing platform

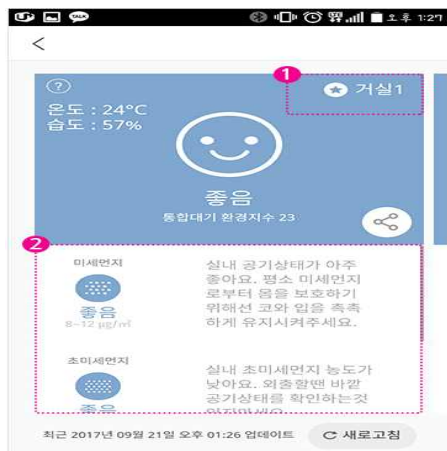


그림 11. 기존의 플랫폼인 'IOT 공기질 알리미' 사용화면
Fig. 11. 'IOT air quality notification' use screen which is the existing platform

표 3. 기존 플랫폼과 제안된 통합 플랫폼의 장단점
Table. 3. Pros and cons of the existing platform and proposed integrated platform

	Existing Platform 'Mamca'	Existing Platform 'IoT air quality notification'	The proposed integrated platform
Video	O	X	O
Alarm	O	O	X
Face recognition	X	X	O
Temperature, humidity, dust measurement	X	O	O
Gas measurement	X	X	O

V. 결 론

기존의 CCTV와 홈 IOT는 각각의 다른 기기를 사용하여 독립적으로 사용하였다. 또 기존의 CCTV와 홈 IOT를 스마트폰으로 볼 때 각각 다른 화면에서 확인을 하였다. 이러한 단점을 극복하기 위해서 본 논문에서는 라즈베리파이 3를 이용하여 하드웨어적으로 하나의 기기로 CCTV와 홈 IOT를 하나의 통합 플랫폼으로 구성하였고 또한, 한 화면에서 안면 인식한 CCTV와 각종 센서의 측정 값을 한눈에 확인 할 수 있게 애플리케이션을 구현하였다. 제안된 방법은 다른 여러 기기를 추가할 필요 없이 하나의 기기를 이용하여 CCTV와 홈 IOT를 구성했다는

장점을 가진다. 본 논문을 구현하기 위해서 하드웨어는 라즈베리파이를 사용하여 홈 IOT를 구현하였고, 파이카메라를 통해 CCTV를 구현하였다. 소프트웨어는 각종 센서 제어 및 OpenCV를 사용하기 위해 Python과 애플리케이션을 프로그래밍하기 위한 Android Studio를 사용하였다. 향후에는 CCTV를 안면 인식하여 인식한 방향으로 따라가고 다른 센서들을 추가하여 측정할 값을 확장할 수도 있다.

참고문헌

[1] Existing Platform ‘맘카’ [Internet]. <https://www.uplus.co.kr/soho/info/pack/DefaultPrdtcctv.hpi>.

[2] Existing Platform ‘IOT 움직임 알리미’ [Internet]. <https://www.uplus.co.kr/ent/iot/IotMovementInfo.hpi>.

[3] J-H Lee, G-I Park, C-H Shin, and S-O Yoo, “An Extensible Smart Home IoT System Based on Open Hardware Platforms”, *Journal of IEMEK*, Vol.11, No.6, pp. 369-377, 2016.

[4] Jeong-Rae Cho, Hye-Suk Kim, Doo-Keol Chae, and Suk-Ja Lim, “Smart CCTV Security Service in IoT(Internet of Things) Environment”, *The Journal of Digital Contents Society*, Vol. 18, No. 6, pp. 1135-1142, Oct. 2017

[5] Hye-Suk Kim, Byeong-Ju Park, and Young-Ju Cho, Smart Outlet System for Single-person Household based on IoT (Internet of Things)”, *The Journal of Digital Contents Society*, Vol. 18, No. 5, pp. 895-904, Aug. 2017

[6] Victor Bautista Saiz and Fernan Gallego, “GPU: Application for CCTV systems”, 2014 International Carnahan Conference on Security Technology (ICCST), 13-16 Oct. 2014.

[7] Existing Platform ‘Smart[Home]안심캠’ [Internet]. <http://www.skbroadband.com/product/home/Page.do?retUrl=/product/home/Cctv2>.

[8] Existing Platform ‘IOT 공기질 알리미’ [Internet]. <https://www.uplus.co.kr/ent/iot/IotAirQuality.hpi>.

[9] Python [Internet]. Available: <https://www.python.org/>

[10] Android Studio [Internet]. Available: <https://developer.android.com/studio/index.html?hl=ko>

[11] Using Raspberry Pi's picamera: Web Streaming (Mjpg-stream) [Internet]. <http://www.rasplay.org/?p=7174>.

[12] P. Viola, M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", CVPR 2001.



안은모(Eun-Mo Ahn)

2012년~현재: 강원대학교 IT대학
전기전자공학부 전자통신학과 학사과정
2018년 2월 ~ 현재: KT 근무

※관심분야 : 무선네트워크 및 사물인터넷(IOT) 등



김동회(Dong-Hoi Kim)

2005년 : 고려대학교 전파공학과
(공학박사)

1989년 1월 ~ 1997년 1월 : 삼성전자 전임연구원
2000년 8월 ~ 2005년 8월 : 한국전자통신연구원 선임연구원
2006년 3월 ~ 현재 : 강원대학교 IT대학 전기전자공학부
전자통신학과 교수

※관심분야 : 이동통신 및 무선 네트워크 등