

學 士 學 位 論 文

Raspberry Pi를 이용한 얼굴인식 스마트 도어락 설계 및 구현

Implementation of Face Recognition Smart Door Lock Using Raspberry Pi

이재희, 전해원

安 養 大 學 校
情 報 通 信 工 學 科
2 0 1 9 年

Raspberry Pi를 이용한 얼굴인식 스마트 도어락 설계 및 구현

Implementation of Face Recognition Smart Door Lock Using Raspberry Pi

이재희, 전해원

위 논문은 안양대학교 학생學位 논문으로
學位 논문 審査委員會에서 審査 通過하였음.

2019年 11月 28日

審査 委員長 : (印)

審 査 委 員 : (印)

審 査 委 員 : (印)

Raspberry Pi를 이용한 얼굴인식 스마트 도어락 설계 및 구현

Implementation of Face Recognition Smart Door Lock Using Raspberry Pi

이재희, 전해원

위 논문은 안양대학교 학생學位 논문으로
學位 논문 審査委員會에서 審査 通過하였음.

2019年 11月 28日

指導教授： 최선완 (印)

국 문 요 약

거의 모든 가정에서 디지털 도어락을 사용하고 있다. 그 중 키패드로 된 도어락이 대중화 되어있는데 키패드이기 때문에 비밀번호에 따라 버튼이 닿거나 지문이 묻어 노출되는 우려가 있다. 따라서 보급률이 높은 스마트폰의 애플리케이션과 생체인식인 얼굴 인식으로 잠금을 제어할 수 있는 도어락의 구현은 유용하게 사용될 것이다.

본 논문에서는 소켓 통신을 기반으로 한 얼굴인식 도어락을 구현한다. 도어락의 기본 기능은 키패드 잠금제어에 얼굴인식 잠금 제어, 애플리케이션 잠금제어 기능을 추가할 것이며 사용자는 애플리케이션으로 사용자의 얼굴을 등록할 수 있으며, 등록된 사용자의 얼굴과 인식이 안된 방문자의 얼굴을 확인할 수 있게 구현할 것이다.

목 차

제 1 장 서론	5
제 2 장 관련 연구	
제 1 절 관련 연구 및 분석	6
제 3 장 요구사항	9
제 4 장 모델	
제 1 절 설계 환경	10
제 2 절 설계 모델	12
제 5 장 설계	
제 1 절 설계 고려사항	13
제 2 절 설계	14
제 6 장 구현	
제 1 절 구현 환경	15
제 2 절 Firebase	16
제 3 절 구현 화면	17
제 7 장 결론 및 향후 계획	24
참고문헌	25
부록1. API(Application Programming Inter	26

그림 목차

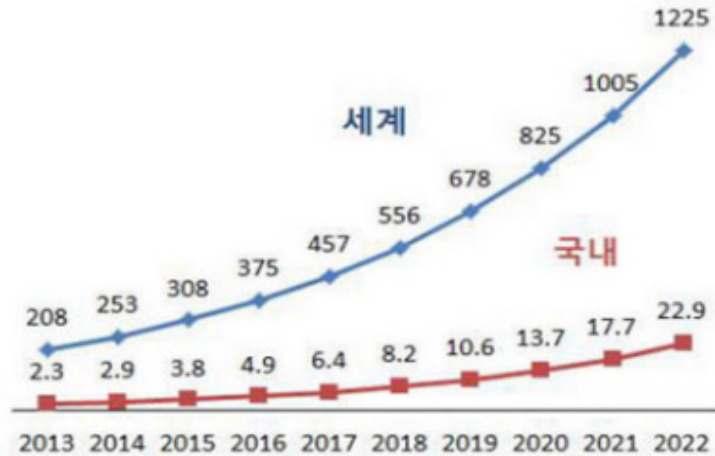
<그림 1-1> 사물인터넷 시장 규모	-----	5
<그림 4-1> 얼굴인식 도어락 구성요소	-----	10
<그림 4-2> 어플 UI 구성도	-----	11
<그림 4-3> 데이터베이스 구조	-----	12
<그림 5-1> 얼굴인식 도어락 설계 구조	-----	14
<그림 6-1> 안드로이드 어플 메인 화면	-----	17
<그림 6-2> 도어락 제어 구성 화면	-----	18
<그림 6-3> 방문자 기록 구성 화면	-----	19
<그림 6-4> 사용자 등록 구성 화면	-----	20
<그림 6-5> 사용자 목록 구성 화면	-----	21
<그림 6-6> 사용자 목록에 등록된 얼굴의 인식률	-----	22
<그림 6-7> 등록되지 않은 얼굴의 인식률	-----	23

표 목차

<표 2-1> 얼굴인식 도어락 모델 비교	-----	6
<표 3-1> 얼굴인식 도어락 기능 요구사항	-----	9
<표 6-1> 어플리케이션 구현 환경	-----	15

제 1 장 서론

사물인터넷 시장 규모(단위: 조원)



<그림 1-1 사물인터넷 시장 규모 [1] >

4차 산업시대가 도래하면서 사물인터넷이 주목을 받고 발전을 하고 있다. 그에 따라 사물 인터넷 기술은 실생활에 접목되어 스마트 홈 시스템이 점차 증가하는 추세이며, 사물 인터넷이 떠오르면서 보안 방식에 대해 많은 이슈가 떠오르고 있다. 그 중 생체인식 방식을 선택하였고, 이 기능을 접목시킬 사물로 도어락을 선정하였다. 현재 시중에 스마트 도어락 제품들이 나온 것들이 있지만, 가격이 비싸고 스마트 홈 연동 등의 번거로움 때문에 많이 사용되고 있지 않다. 그래서 본 논문에서는 사용자들이 어플을 통해 보다 쉽고 빠르게 제어할 수 있는 얼굴인식 도어락을 구현할 것이다. [2]

따라서 본 논문에서는 '2장'에서 다뤄질 시중에 나와있는 얼굴인식 도어락에 대한 관련 연구를 통해 사물인터넷(IoT)를 이용하여 어플리케이션으로 제어할 수 있는 새로운 얼굴인식 도어락 모델에 대해 기술한다. 이에 대한 구성요소와 설계 및 고려사항은 '4장'에서 상세히 기술하며 '3장'에서는 라즈베리파이와 안드로이드 어플리케이션의 통신에 사용되는 firebase에 대해 상세히 기술한다.

제 2 장 관련연구

제 1절 관련연구 및 분석

표 <2 - 1> 은 현재 보급되고 있는 다른 얼굴인식 도어락들에 대한 연구를 표로 작성한 것이다.

<표 2-1> 얼굴인식 도어락 모델 비교

기능 \ 이름	안면인식 도어락 (아이리시스) [3]	Canera Smart Lock (어스이) [4]	안면인식 도어락 (온페이스) [5]	얼굴인식 도어락
얼굴 인식	O	O	O	O
어플로 잠금 해제	X	O	X	O
등록 인원 수	100	100	100	10
사용 환경	X	ios	X	android
카메라 방식	적외선	적외선	적외선	일반 카메라

1. 안면인식 도어락 (아이리시스) [3]

1.1. 얼굴 인식

얼굴 인식 기능을 갖추고 있다.

1.2. 어플로 잠금 해제

IoT 홈 서비스를 지원하나, 별개의 어플로 제어가 불가능하다.

1.3. 등록 인원 수

최대 100명을 등록할 수 있다.

1.4. 사용 환경

어플리케이션 제어가 불가능 하기 때문에 해당없다.

1.5. 카메라 방식

적외선 카메라를 사용한다.

2. Camera Smart Lock (어스이) [4]

2.1. 얼굴 인식

얼굴 인식 기능을 갖추고 있다.

2.2. 어플로 잠금 해제

전용 어플을 지원하여 어플을 통해 제어가 가능하다.

2.3. 등록 인원 수

최대 100명을 등록할 수 있다.

2.4. 사용 환경

어플리케이션은 ios환경에서만 사용 가능하다.

2.5. 카메라 방식

적외선 카메라를 사용한다.

3. 안면인식 도어락 (온페이스) [5]

3.1. 얼굴 인식

얼굴 인식 기능을 갖추고 있다.

3.2. 어플로 잠금 해제

어플로 제어가 불가능하다.

3.3. 등록 인원 수

최대 100명을 등록할 수 있다.

3.4. 사용 환경

어플리케이션 제어가 불가능 하기 때문에 해당없다.

3.5. 카메라 방식

적외선 카메라를 사용한다.

4. 얼굴인식 도어락

4.1. 얼굴 인식

얼굴 인식 기능을 갖추고 있다.

4.2. 어플로 잠금 해제

전용 어플을 지원하여 어플을 통해 제어가 가능하다.

4.3. 등록 인원 수

최대 10명을 등록할 수 있다.

4.4. 사용 환경

어플리케이션은 android환경에서만 사용 가능하다.

4.5. 카메라 방식

라즈베리파이의 일반 카메라모듈을 사용한다.

본 논문에서 구현하는 스마트 얼굴인식 도어락은 소켓 통신을 기반으로 하여, 사용자 얼굴을 등록하고, 등록된 얼굴을 바탕으로 도어락에 있는 카메라를 이용해서 얼굴 인식을 하여 도어락 제어 하는 것에 초점을 맞추고자 한다.

제 3 장 요구사항

본 논문에서는 android 환경의 어플리케이션과 Raspberry를 이용하여 얼굴인식 도어락을 구현할 것이며, 각 구성요소의 기능 요구사항을 <표 3 - 1>에 정리하였다.

<표 3 - 1> 얼굴인식 도어락 기능 요구사항

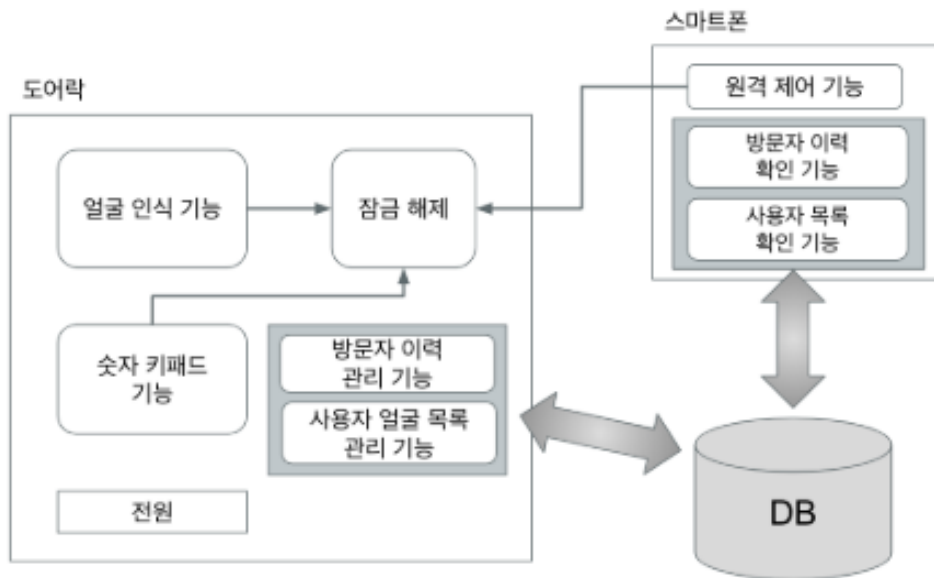
스마트폰	사용자 얼굴 등록 기능	
	사용자 목록 기능	
	원격 제어 기능	
	방문자 확인 기능	
도어락	얼굴 인식 기능	카메라 제어 신호
	얼굴 감지 후 신호 발생 기능	도어락 제어 신호
	키패드 기능	
	잠금해제 기능	

스마트폰의 어플리케이션은 도어락 제어, 방문자 확인, 사용자 등록, 사용자 목록 확인 이렇게 네 가지의 기능을 가지고 있으며 도어락은 <표 3 - 1>에 나와있는 것처럼 사용자나 방문자를 촬영 및 감지하고 신호 발생과 사진 저장 기능을 가지고 있다.

제 4 장 모델

제 1 절 구성 요소

얼굴인식 도어락의 구성요소는 다음 아래의 <그림 4 - 1>과 같다.



<그림 4 - 1>얼굴인식 도어락 구성 요소

얼굴인식 도어락은 기존의 번호 키패드 뿐만 아니라 얼굴 인식과 어플 원격제어로 잠금 해제를 할 수 있다. 얼굴 인식 기능은 사용자 얼굴과 방문자의 타임스탬프를 데이터베이스와 연동하여 인식하고 잠금해제를 한다.

얼굴인식 도어락의 어플 UI 는 다음 아래의 <그림 4 - 2>과 같다.



<그림 4 - 2> 어플 UI 구성도

어플에서는 도어락과 연동하여 얼굴인식을 할 수 있게 하는 사용자 등록을 시키고, 목록도 볼 수 있고, 도어락 잠금도 제어가 가능하고, 방문이력도 조회 할 수 있다.

사용자 등록 버튼을 누르고 도어락 앞에 있는 카메라에 가서 사진을 찍으면 사용자를 등록 할 수 있다. 그리고 사용자를 등록하게 된다면, 사용자 목록에 사진이 올라온다. 사용자를 등록한 이후에는 사용자의 얼굴이 얼굴인식 도어락에 찍힌다면, 자동으로 도어락의 잠금이 제어 된다. 만약 사용자의 얼굴이 등록되어있지 않은 경우, 33 번 이상 인식 실패시 방문 기록 조회에 사진이 전송된다. 사용자는 방문 기록 조회 버튼을 누르고 방문자의 얼굴을 볼 수 있고, 문을 열고 싶다면, 도어락 잠금 제어의 버튼을 열어서 도어락을 잠금 해제 할 수 있다.

제 2 절 데이터베이스 구성 요소

얼굴인식 도어락의 데이터베이스는 다음 아래의 <그림 4 - 3>과 같다.



<그림 4-3> 데이터베이스 구조

얼굴 인식 도어락의 데이터베이스는 사용자 등록과 사용자 목록 확인 기능에 사용되는 사용자 관련 스키마와 방문 목록 확인 기능에 사용되는 방문내역 관련 스키마로 구성되어 있다.

제 5 장 설계

5 장에서는 3 장의 요구사항과 4 장의 구성요소를 바탕으로 그에 알맞은 얼굴인식 도어락의 플랫폼에 대해 상세히 기술하고자 한다.

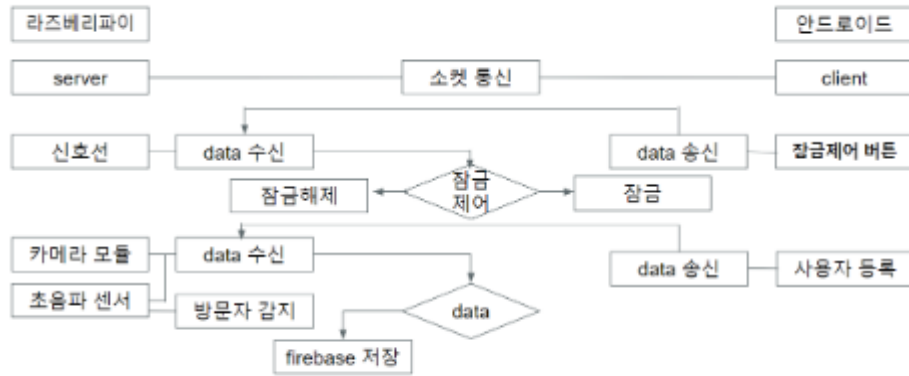
제 1 절 설계 고려사항

본 논문은 얼굴인식 도어락의 플랫폼 구현을 위하여 아래와 같은 고려사항을 제한하고 있다.

1. 기존에 나와있던 번호키 도어락 제품을 개조한다.
2. 어플리케이션은 사용자 점유율이 높은 Android 버전으로 구현한다.
3. 사용자위주로 직관적인 GUI 를 구현한다.
4. 모든 사용자의 정보는 firebase 실시간 DB 에 저장되어있다.
5. 얼굴 이미지는 firebase storage 에 저장되어 있으며 파일명으로 접근 가능하다.

제 2 절 설계

얼굴인식 도어락은 <그림 5 - 1>의 모델처럼 실행된다.



<그림 5-1> 얼굴인식 도어락 설계 구조

본 논문에서는 실시간 데이터베이스를 구현하기 위해 firebase 의 실시간 데이터베이스를 사용하였다. <그림 5 - 1>은 얼굴인식 도어락의 설계 구조를 표현한 그림이다. 얼굴인식 도어락은 안드로이드 어플리케이션과 도어락에 달린 라즈베리파이 사이에 Firebase 가 가운데에서 데이터베이스의 역할을 하고 있다. 어플리케이션을 통해 사용자의 얼굴을 등록할 수 있으며, Firebase 는 어플리케이션의 요구를 Raspberry 로 전달해주어 카메라 모듈로 촬영이 이루어지게 한다. Raspberry 는 카메라를 통해 찍은 사진을 OpenCV 소스를 통해 학습하고 최종 사진을 다시 Firebase 를 통해 어플리케이션으로 전송하여 사용자 목록에 추가시킨다. 사용자 목록에 등록되지 않은 방문자가 카메라 앞에서 3 번 이상 반복 감지될 경우, 도어락에서 신호음과 함께 사용자의 어플리케이션으로 방문자의 사진을 전송하여 범죄를 예방하는 효과를 기대할 수 있다.

제 6 장 구현

제1절 구현 환경

얼굴인식도어락이 개발된 환경은 다음 <표 6 - 1>와 같다.

<표 6 - 1> 애플리케이션 구현 환경

구분	구현 환경
OS	Android
Language	python, java
Tools	firebase
H/W	RaspberryPi

제 2 절 FireBase[6]

본 논문에서는 얼굴인식 도어락에서 사용될 데이터 베이스인 firebase에 대해서 기술한다. firebase란 클라우드 서비스 제공자이자 동시에 백엔드 기능을 가지고 있다. 그리고 원격 클라우드 서버를 제공하며 우리는 Firebase를 이용해 손쉽게 그 곳에 데이터를 저장하고 불러올 수 있다.

얼굴인식 도어락에서는 위의 firebase 기능 중에서 storage와 Realtime database 기능을 사용하려고 한다.

firebase storage는 이미지, 동영상 및 기타 대용량 파일을 저장하는 기능이다. 여기서는 개발자들이 쉽고 안전하게 이런 파일들을 업로드하고 다운로드할 수 있다. Google 클라우드 저장소가 제공하는 Firebase 저장소는 대규모의 확장성을 제공하며, Google 클라우드 프로젝트에서 저장된 파일에 손쉽게 액세스할 수 있게 해준다.

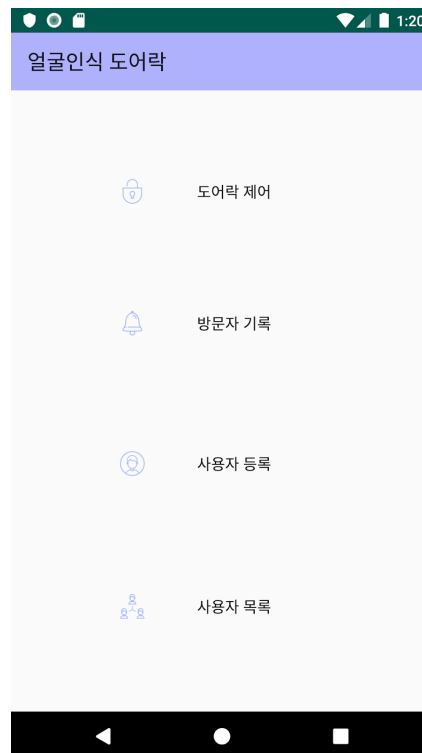
Realtime database는 Firebase에서 실시간 데이터베이스와 백엔드 서비스를 제공하는데, 이 서비스는 어플리케이션 개발자에게 하나의 API를 제공하는데 이 API는 어플리케이션 데이터가 고객 그리고 저장된 Firebase 클라우드와 자동 동기화되도록 해준다.

제 3 절 구현 화면

1. 메인화면

가. 어플의 메인 화면 구성은 <그림 6 - 1>과 같다.

나. 사용자가 원하는 기능을 클릭하면 다음 화면으로 넘어간다.

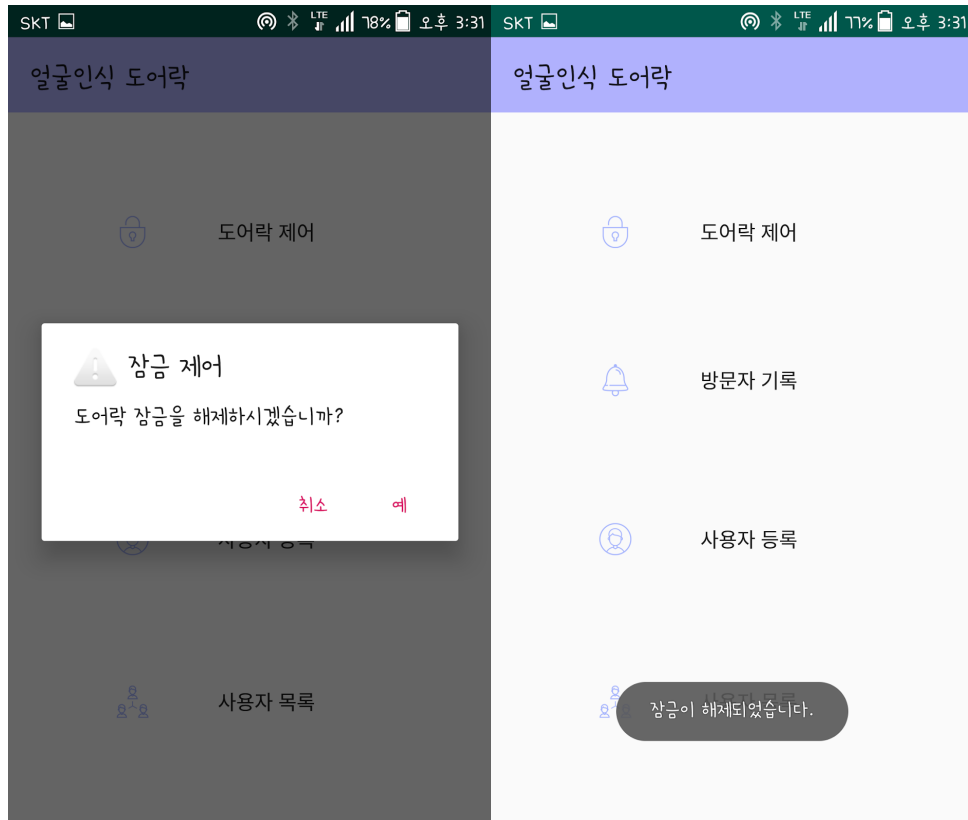


<그림 6 - 1> 안드로이드 어플 메인 화면

2. 도어락 제어 화면

가. 도어락 제어 버튼을 누른 화면은 <그림 6 - 2>와 같다.

나. '예' 버튼을 통해 잠금이 해제된다.

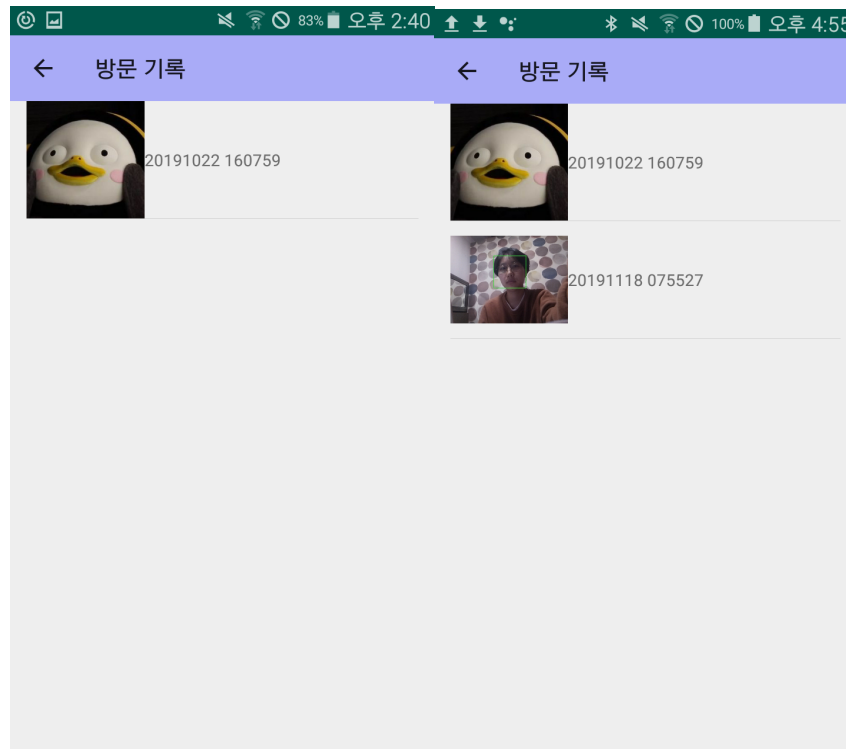


<그림 6 - 2> 도어락 제어 구성 화면

3. 방문자 기록 화면

가. 방문자 목록과 기록을 보여주는 화면은 <그림 6 - 3>과 같이 구성된다.

나. 왼쪽 화면에서 인식되지 않은 방문자 발생시 오른쪽 화면처럼 방문자가 추가된다.

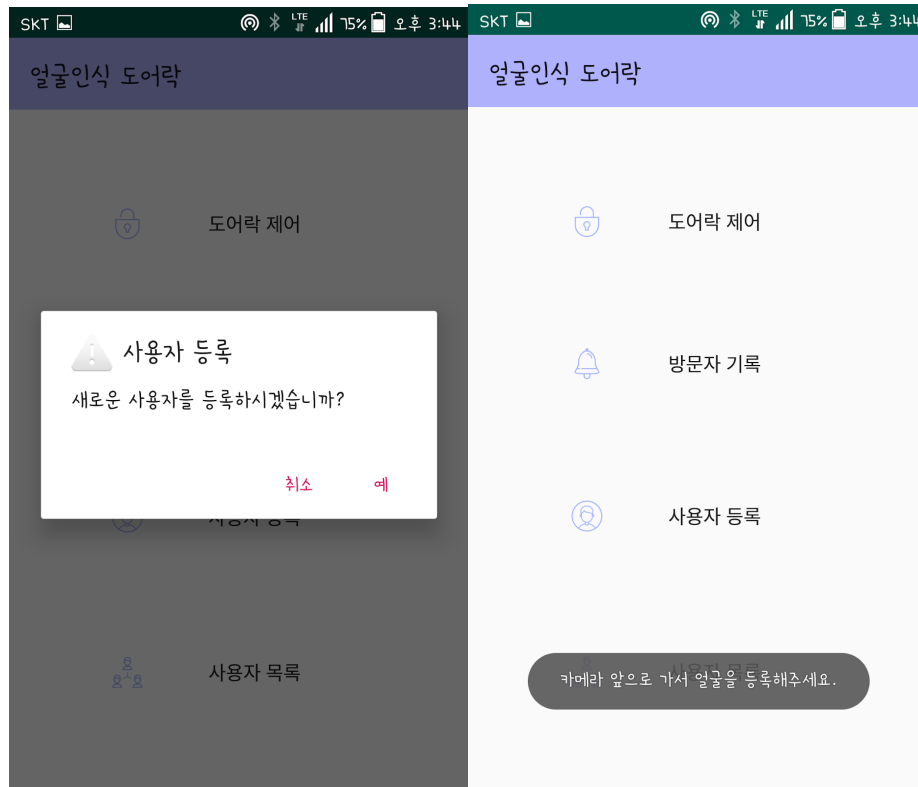


<그림 6 - 3> 방문자 기록 구성 화면

4. 사용자 등록 화면

가. 사용자를 등록하는 화면은 <그림 6 - 4>와 같이 구성된다.

나. '예' 버튼을 누르면 도어락 위의 카메라가 작동한다.

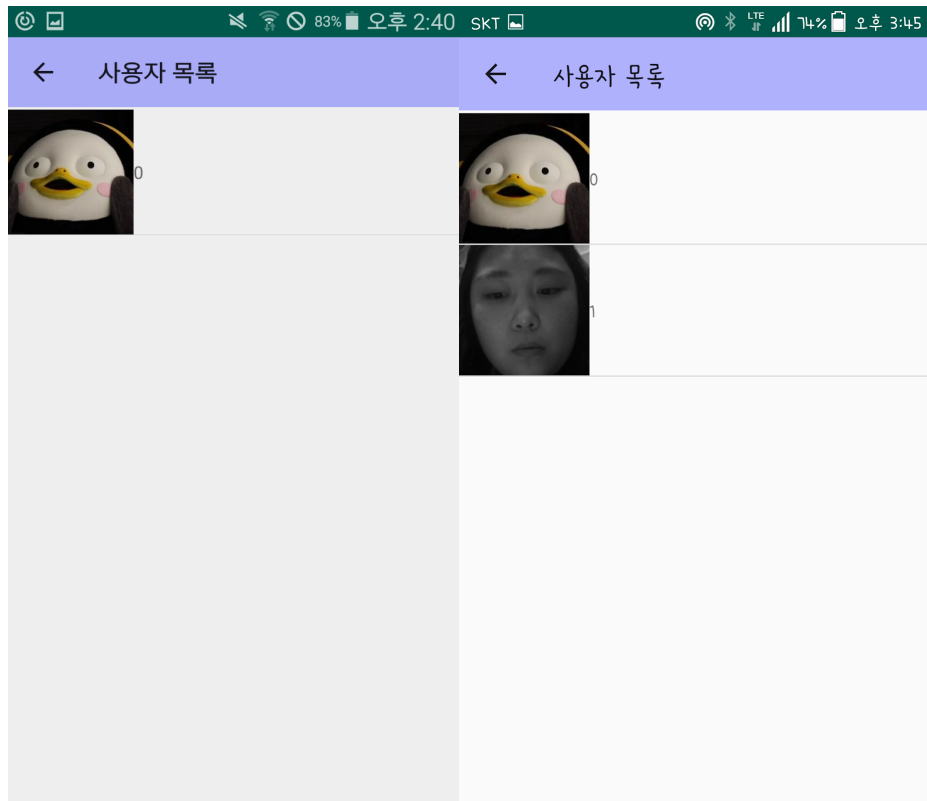


<그림 6 - 4> 사용자 등록 구성 화면

5. 사용자 목록 화면

가. 사용자 목록은 <그림 6 - 5>와 같이 구현된다.

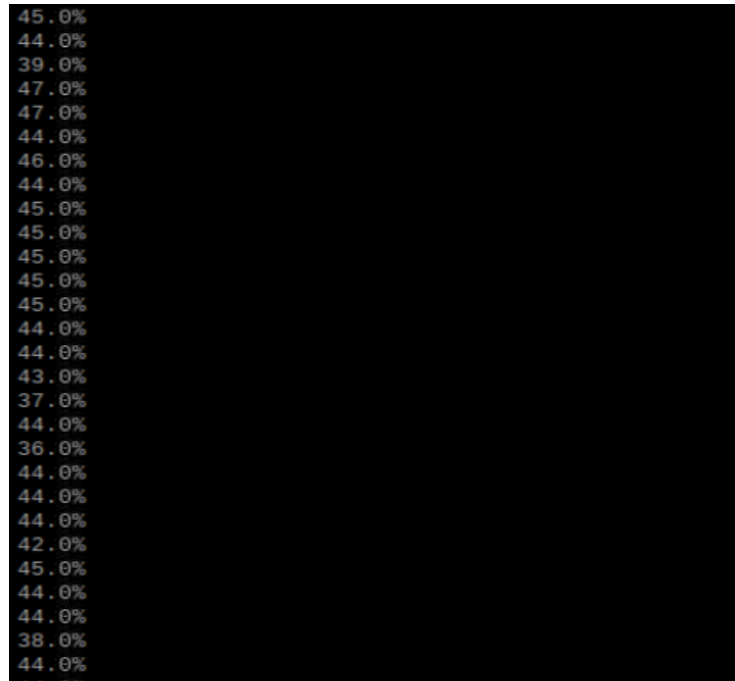
나. 왼쪽 화면에서 사용자 추가시 오른쪽 화면처럼 구현된다.



<그림 6 - 5> 사용자 목록 구성 화면

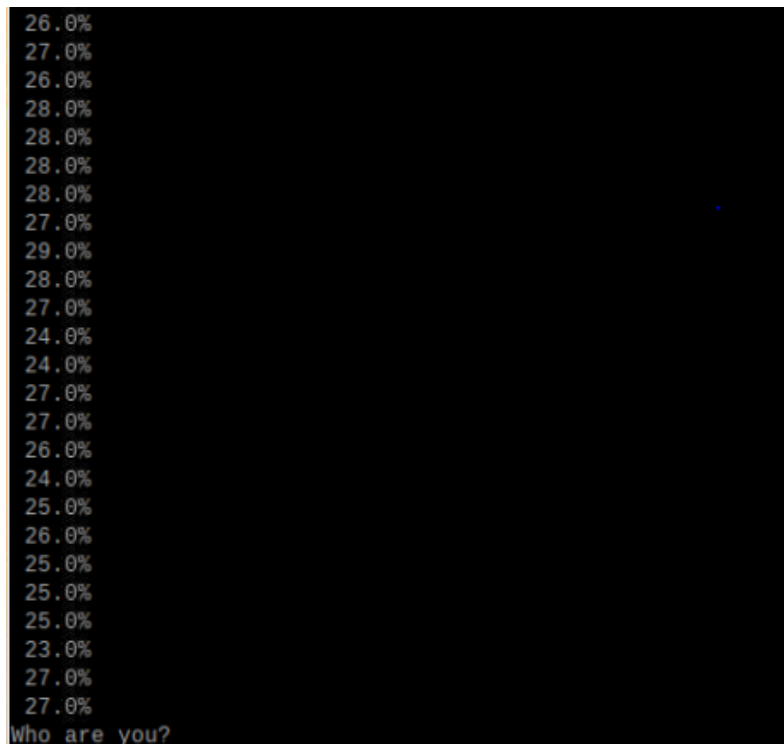
6. 얼굴 인식률

가. 사용자 목록에 등록된 얼굴에 대한 인식률은 <그림 6 - 6> 과 같다.



<그림 6 - 6> 사용자 목록에 등록된 얼굴의 인식률

나. 사용자 목록에 등록 되어있지 않은 얼굴에 대한 인식률은 <그림 6 - 7>과 같다.



<그림 6 - 7> 등록되지 않은 얼굴의 인식률

따라서 얼굴인식 도어락은 카메라를 통해 비춰진 얼굴이 43%이상의 인식률이 보인다면, 등록된 사용자라고 판단하고 도어락을 UNLOCK시킨다. 하지만 33번의 인식에서 연속으로 43%의 인식률에 도달하지 않는다면, 라즈베리파이를 통해 사진을 촬영하고, firebase storage에 사진을 저장하여 <그림 6 - 3> 리스트에 사진이 추가된다.

제 7 장 결론 및 향후 계획

본 논문에서는 라즈베리파이와 Firebase, 소켓 통신을 이용한 얼굴인식 도어락을 구현하였다. 본 논문에서는 비밀번호 대신 얼굴인식과 Android 기반의 애플리케이션으로 사용자가 편리하게 잠금제어를 할 수 있게 구현하였다.

본 논문에서는 보급률이 높은 기존 키패드 도어락을 사용함에 따라 비밀번호 노출에 대한 이슈가 떠오르는 것을 확인할 수 있었다. 본 논문에서 구현한 생체인식을 이용한 얼굴인식도어락으로 비밀번호 노출에 대한 위험성이 감소될 것으로 예상된다. 또한 애플리케이션을 이용하여 손님 방문시, 비밀번호를 따로 알려주지 않아도 쉽게 잠금 해제가 가능하기 때문에 스마트홈 연동을 해야만 하는 다른 스마트 도어락의 번거로움을 보완할 수 있다.

본 논문에서 구현한 얼굴인식 도어락은 인식되지 않는 방문자를 사진으로 찍어 데이터베이스를 통해 애플리케이션으로 확인할 수 있지만 향후에는 실시간으로 방문자를 확인할 수 있는 스트리밍 기능이 적용되어야 할 것으로 보인다. 또한 등록되지 않은 방문자가 중복하여 얼굴인식을 시도했을 경우, 사용자에게 위험인물로 알림을 보내는 기능을 추가하면 범죄에 대해서도 예방할 수 있을 것이다.

참고 문헌

[1] 사물인터넷 시장 규모

<https://www.kiet.re.kr>

[2] 안현, 윤영환, 이용민 , “OPENCV 라이브러리를 이용한 얼굴인식도어락 시스템의 구현”, 대한전자공학회 학술대회, 940-942(3pages), 2018.6

[3] IRISYS 얼굴인식도어락

<https://www.irisys.co.kr/>

[4] UE:E 얼굴인식도어락

<https://www.kickstarter.com/projects/1624790698/us-e-camera-equipped-smart-lock-with-facial-recogn>

[5] 온페이스

<http://www.onface.kr/sub/index.php>

[6] firebase

<https://firebase.google.com/>

부록 1. API

Face Recognition Smart Door Lock

-Variable

face_id : 새로 등록할 사용자의 번호
data : 클라이언트로부터 받을 메시지
res : 클라이언트에게 보낼 메시지
trig : 초음파센서 trig에 연결할 gpio핀 번호를 넣는 변수
echo : 초음파센서 echo에 연결할 gpio핀 번호를 넣는 변수
start : 처음 초음파의 echo input을 받는 시간
end : echo input이 끝나는 시간
distance : 초음파 센서와의 거리
count : 특정 거리 내에 도달한 횟수
cred : 비공개 키를 받을 변수
cap : 실행중인 비디오패치를 캡처 할 변수
ref : firebase database에 접근하여 새로운 사용자를 넣을 변수
now : 현재시간을 측정해서 넣을 변수
door : 도어락과 연결되어있는 릴레이를 제어해서 도어락을 열어주는 변수
confidence : 인식률을 저장하는 변수
g : 저장된 방문자의 목록 중 다음에 들어갈 번호를 String 형으로 바꾸는 변수
path : 훈련시킬 얼굴이 있는 폴더 주소를 저장하는 변수
detector : haarcascade 파일을 불러오는 변수
imagepath : path에서 사진을 불러와 저장하는 변수
faceSamples.append : 불러온 사진에서 사각형으로 얼굴 지정하여 요소 추가
faceSamples[] : 사각형으로 된 얼굴 사진을 저장하는 배열
ids[] : faceSample[]의 id를 받는 배열
recognizer.write : 훈련시킨 얼굴들에 대한 내용을 담은 trainer.yml 파일 저장

-Function

android.py : 라즈베리파이에서 서버역할을 하는 함수
do_some(input_button) : 안드로이드에서 버튼을 누르게 되면 특정 문구를 보내게 되는데, 그 문구에 맞게 명령을 처리해주고 완료 문구를 리턴해주는 함수
choumpa.py : 초음파 센서를 수행하고 특정 거리 이내에 도달하면 서버에게 사용자 인식코드를 실행하게끔하는 클라이언트역할을 하는 함수
newFace.py : 새로운 사용자를 등록하는 함수
DBInsert.py : firebase database에 새로운 사용자를 추가하는 함수

dbGuest.py : firebase database에 새로운 게스트를 추가하는 함수
door.py : 도어락을 제어하는 함수
final.py : 얼굴을 인식하여 인식률에 따라 잠금해제하거나 인식되지 않은 방문자의 사진을 찍어 저장하는 함수
training.py : 촬영된 인물의 사진 30장을 트레이닝하여 각 얼굴과 아이디를 training.yml파일로 저장하는 함수
def getImageAndLabels(path) : path에서 훈련시킬 얼굴 30장을 불러와 인식 훈련시키는 함수. faceSamples[]와 ids[]를 반환함.