

스마트 출입관리 시스템

학부생 : 김영호 지도교수 : 유국열
영남대학교 기계IT대학 정보통신공학과
(우) 38541 경북 경산시 대학로 280
21912254@yu.ac.kr, kyoo@yu.ac.kr

요 약

우리가 일상생활에 사용하는 공동현관 도어락은 비밀번호를 입력하여 출입하는 경우가 대다수이며, 지정된 공동현관 비밀번호가 길거나 복잡하다면 출입에 있어 불편함을 느끼게 된다. 또한 배달이나 택배 배송을 받는 과정에서 공동현관 비밀번호를 제공해야 한다는 사실에 외부인의 출입을 꺼리는 경우가 발생한다. 이러한 문제점들을 보완하기 위하여 얼굴인식을 통해 출입이 가능한 도어락을 설계하여 편의성을 개선하고자 한다.

1. 서 론

스토킹 범죄가 강력범죄로 이어지는 사건이 전국적으로 연이어 발생, 사회적 문제로 대두되고 있는 가운데, 일부 공동주택의 공동현관 비밀번호가 노출돼 있어 범죄의 표적이 될 수 있다는 지적이다. 외부인을 일차적으로 막기 위해 설치한 보안장치가 제 기능을 하지 못하고 있기 때문이다.

전북 전주시 완산구 효자동의 한 원룸촌을 예시로 대부분 건물 1층에는 잠금장치가 있는 공동현관이 설치돼 있었다. 모두 비밀번호를 입력해 문을 여는 방식이었다. 하지만 잠금장치 윗부분의 빈 곳이나 벽면에 비밀번호로 보이는 4자리의 숫자가 적혀 있었고, 인근에는 잠금장치를 사용하는 방법이 적힌 안내문까지 붙어 있었다. 실제 기자가 안내문의 설명대로 숫자를 잠금장치에 입력하니 잠금장치가 열렸다. 외부인의 출입을 막아 거주민의 안전을 위해 설치한 잠금장치가 사실상 무용지물인 셈이다.

이러한 문제점을 보완하기 위하여 거주자의 정보를 관리함과 동시에 허가받지 않은 외부인의 출입을 줄일 수 있으며, 애플리케이션을 통해 초대받은 손님의 출입을 원격으로 제어하는 도어락을 설계하였다. 예시로 등록된 거주자일 경우, 간편하게 얼굴인식을 통해 간편한 출입이 가능하다. 하지만 배달 혹은 택배기사가 방문하여 307호의 초인종을 눌렀을 경우, 307호의 애플리케이션으로 알림이 전송되며, 애플리케이션을 통해 방문자의 모습을 확인할 수 있다. 이때 의심스럽거나 사전에 약속된 택배인지 검토하여 공동현관 개방 여부를 선택할 수 있다.

또한 아두이노에 미세먼지 센서, 온도-습도 센서, 불꽃 감지 센서와 같이 다양한 센서를 사용하여 거주지의 실내 환경을 측정한다. 애플리케이션을 통해 실내 환경을 확인할 수 있으며, 미세먼지 농도가 나빠지거나 화재가 감지되었을 경우, 알림을 통해 경보를 울린다. 이를 통해 더 나아가다면 다양한 IoT를 제어할 수 있도록 가능성을 열어둘 수 있다.

이 논문은 다음과 같이 구성되어 있다. 2절에서는 얼굴인식 출입관리 시스템과 원격 제어가 가능한 도어락을 포함한 관련 연구를 간략하게 소개하며, 이들 기존 방식의 문제점과 한계점에 대하여 설명한다. 3절에서는 본 논문에서 제안하는 스마트 출입관리 시스템에 대하여 상세하게 설명한다. 4절에서는 본 논문에서 제안하는 기술/기능의 기능 구현에 대하여 설명하며, 5절에서는 제안된 기술/기능의 성능을 측정하고, 측정 결과를 분석한다. 6절에서 결론을 맺으며 향후 연구개발 계획을 간략히 소개한다.

2. 관련 연구

2.1 얼굴인식 출입관리 시스템

1990년부터 범죄예방과 관련한 방어공간에 대한 실증연구가 도시계획 분야에 적용되어왔다. 범죄예방을 위한 기준은 건축물의 외부 환경의 구조(도로, 보도, 조명, 가로등, 담장 등)로 구성되어 있다. 하지만 외부 및 공용공간에서 현관문을 통하여 건축물 내부로 진입하여 단위세대(주거시설)에서 이루어지는 범죄예방은 고려되어 있지 못한 실정이다.[1]

스토킹 범죄가 증가하고 있는 가운데, 일부 공동주택의 공동현관 비밀번호가 노출돼 있어 범죄의 대상이 될 수 있다는 문제가 제기되고 있다. 이에 따라 설치된 보안장치가 기능을 제대로 수행하지 못하고 있다. 또한 배달이나 택배 배송을 받을 때 외부인에게 공동현관 비밀번호를 제공해야 하는 점으로 인해 출입을 꺼리는 경우가 발생한다. 이러한 문제를 해결하기 위해 OpenCV 기술을 통해 얼굴인식 도어락 개발을 목표로 설정하였다.

2.2 애플리케이션을 이용한 출입 관리 및 모니터링

일반 가정집의 인터폰은 주거자가 외부 방문자의 신원을 확인하기 위해 반드시 직접 현관으로 가야 하며, 특정 시간에만 방문자의 출입을 허용할 수 있어 유연성이 부족하다. 또한, 특정 시점에 택배나 배달을 받으려면 집에 있어야 하거나, 외출 중인 경우에는 택배를 받지 못할 수 있는 불편함이 있다.

이러한 한계점을 극복하고자 스마트폰 애플리케이션을 통해 직관적이고 간편한 사용자 인터페이스를 제공하여, 사용자가 손쉽게 출입 관리를 제어하고 실내 환경을 모니터링할 수 있는 기능을 제공하고자 한다. 이를 통해 등록된 주거자는 특정 장소에 있지 않아도 애플리케이션을 통해 외부 방문자의 출입을 간편하게 제어할 수 있으며, 방문자의 모습을 확인하고 현관문 개방 여부를 선택할 수 있다.

3. 제안된 기술/기능의 제목

3.1 제안된 기능 구조 (기능 블록도)

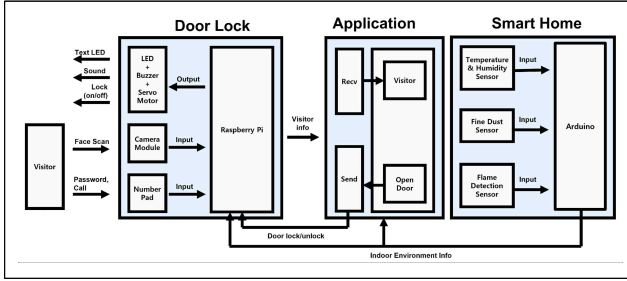


그림 1. 기능 블록도 (Functional Block Diagram)

도어락을 작동하기 전, 회원 정보(학번, 이름, 전화번호, 생일, 주소)를 등록한 뒤, 얼굴 검출을 통해 얼굴 정보를 저장하고 학습시킨다. 이를 통해 회원 정보 검색과 삭제 기능을 통해 회원 정보를 관리할 수 있다.

정보 저장을 마친 이후, 도어락을 작동하여 비밀번호, 얼굴인식, 호출 기능을 통해 상황에 따라 도어락의 잠금상태, 소리, LCD가 출력된다. 방문자가 거주자를 호출할 경우는 방문하고자 하는 호수(ex. 307호)를 입력하여 애플리케이션에게 요청메시지를 전송한다. 이때 애플리케이션(client)을 도어락(server)과 통신을 연결한 뒤, 도어락의 호출 수신을 대기하고 있는 상태여야 한다. 요청 메시지를 수신한 애플리케이션은 도어락의 카메라로 찍고 있는 영상을 캡처하여 사진을 수신하여 도어락의 개방 여부를 선택할 수 있다.

또한 아두이노를 통해 미세먼지 센서, 불꽃감지 센서, 온습도 센서 측정값으로 실내 환경 수치와 수치에 따른 경고를 애플리케이션으로 전송하여 현재 실내 환경 수치와 위험도를 확인할 수 있다.

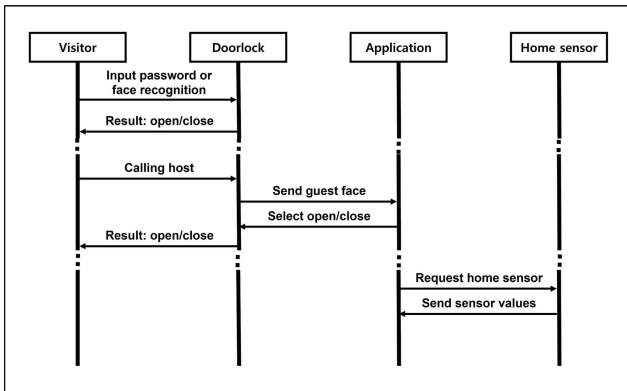


그림 2. 기능 순서도 (Sequence Diagram)

스마트 출입관리 시스템의 기능을 살펴보면 거주자의 출입, 방문자의 출입, 거주지의 실내 환경 측정 세 종류로 나뉜다.

첫 번째 순서도의 경우, 방문자가 비밀번호나 얼굴인식을 통해 공동현관의 도어락을 개방하고자 한다. 이때 입력된 정보의 일치 여부에 따라 도어락의 개방 여부가 결정된다. 즉, 비밀번호를 알고 있으면서 얼굴 정보가 저장된 거주자에 해당한다.

두 번째 순서도의 경우, 방문자가 거주자 호출을 통해 공동현관의 도어락을 개방하고자 한다. 이후 거주자의 애플리케이션으로 호출 알림이 전송되고, 방문자의 모습을 확인할 수 있다. 이때 거주자는 해당 방문자가 초대받은 손님인지, 수상한 사람인지 확인하여 도어락의 개방 여부를 선택할 수 있다. 즉, 손님, 배달, 택배와 같은 방문자에 해당한다.

세 번째 순서도의 경우, 거주자는 애플리케이션을 통해 거주지의 실내 환경 수치를 확인할 수 있다. 또한 실내 환경 수치가 나쁜 경우, 애플리케이션에 경고 알림과 함께 위험도를 나타낸다.

3.2 제안된 기능 구조의 알고리즘 (Pseudo code)

3.2.1 스마트 출입관리 시스템

라즈베리파이에서 스마트 출입관리 시스템을 실행한다.

```
def main():
    # Create and Run server thread
    server_thread = threading.Thread(target=server.start, args=(doorlock,))
    server_thread.start()
    time.sleep(1)

    while True:
        # Print menu
        print("=====")
        print("1. Start Doorlock System")
        print("2. User Data Management")
        print("3. Exit")
        print("=====")

        # Select menu
        choice = input("Enter your choice (1-3): ")
        print()

        # 1. Start Doorlock System
        if choice == "1":
            # ...

        # 2. User Data Management
        elif choice == "2":
            # ...

        # 3. Exit
        elif choice == "3":
            # ...

        # Invalid choice
        else:
            # ...
```

그림 3. main.py

스레드를 사용하여 서버를 작동한다. 그리고 ‘도어락 작동’, ‘거주자 정보 관리’, ‘스마트 출입관리 시스템 종료’ 중 선택이 가능하다.

3.2.2 임베디드 시스템

도어락의 상태에 알맞게 임베디드 시스템이 작동한다.

```
# Servo motor class
class Servo:
    # Servo motor setting
    def __init__(self, pin):
        # ...

    # Servo motor rotation
    def turn(self, angle):
        # ...

    # Servo motor stop
    def stop(self):
        # ...

# Piezo buzzer class
class Buzzer:
    # Piezo buzzer setting
    def __init__(self, pin):
        # ...

    # Piezo buzzer sound output
    def buzz(self, freq_list, duration, repeat):
        # ...

    # Piezo buzzer stop
    def stop(self):
        # ...

# Photo resistor class
class Photoreistor:
    # Photo resistor setting
    def __init__(self, pin):
        # ...

    # Photo resistor measures
    def rc_time(self):
        # ...

    # Photo resistor stop
    def stop(self):
        # ...
```

그림 4. embedded.py

도어락의 잠금 여부를 제어하는 서보모터 클래스, 소리를 재생하는 피에조 부저 클래스, 공동현관 닫힘 여부를 검사하는 조도 센서 클래스가 있으며, 라즈베리파이의 GPIO 핀을 지정하여 부품들을 제어할 수 있다.

3.2.3 도어락 상태 출력

16x2 LCD를 통해 도어락의 상태를 문자로 출력한다.

```
# Print waiting (before doorlock start)
def print_waiting():...

# Print recognizing (face recognizing)
def print_recognizing():...

# Print doorlock open (match face information)
def print_face_unlock_correct(name):...

# Print doorlock incorrect (mismatch face information)
def print_face_unlock_incorrect():...

# Print unrecognizable (unrecognizable face)
def print_face_unlock_unrecognizable():...

# Print password field
def print_password():...

# Print password incorrect (mismatch password)
def print_password_incorrect():...

# Print warning (error occurred 3 times)
def print_warning():...

# Print call field
def print_call():...

# Print calling host (resident call)
def print_calling_host():...

# Print call error (wrong resident information)
def print_call_error():...

# Print doorlock open
def print_doorlock_open():...

# Print doorlock close
def print_doorlock_close():...

# Print main (select menu)
def print_main():...

# Password function (1. password button)
def password(doorlock):...

# Face Unlock function (2. face button)
def face_unlock(face, doorlock):...

# Call function (3. call button)
def call(server, face, doorlock):...

# Read the key value
def read_lcd_buttons():...

# Main LCD function
def start_doorlock(server, face_module, doorlock):...
```

그림 5. LCD.py

도어락 버튼을 통해 비밀번호 입력, 얼굴인식, 호출 기능 중 선택이 가능하다. 그리고 도어락의 개방 여부, 비밀번호 일치 여부, 얼굴인식 일치 여부, 오류 및 경고, 거주자 호출과 같이 다양한 경우를 도어락 LCD에 문자로 출력하여 현재 도어락의 상태를 나타낸다.

3.2.4 도어락 숫자패드 제어

공동현관 도어락에 연결된 3x4 숫자패드의 입력과 LCD 출력의 일부를 제어한다.

```
# Check & Print password
def password(LCD_module, doorlock):...

# Check & Print call room
def call(LCD_module, doorlock, user_data):...

# Press "*" to main screen
def main_button(wait_time):...
```

그림 6. keypad.py

도어락의 비밀번호 입력과 거주자 호출 기능을 실행한다.

3.2.5 얼굴인식 및 학습

거주자의 얼굴 정보를 인식하고 학습하여 공동현관 도어락의 얼굴인식 기능을 제어한다.

```
# Face dataset (capture face to model training)
def face_dataset(face_id):...

# Function to get the images and label data
def get_images_and_labels(path):...

# Face training
def face_training():...

# Face recognition (doorlock)
def face_recognition(doorlock):...

# LCD.py -> call(server, face, doorlock)
def guest_face():...
```

그림 7. face.py

라즈베리파이에 연결된 카메라 모듈을 통해 얼굴

을 인식하며, dataset에 거주자의 얼굴을 저장하여 이를 학습한다. 도어락의 얼굴인식 기능을 통해 거주자의 얼굴 일치율이 60% 이상일 경우, 도어락을 개방한다. 또한 방문자가 호출 기능을 사용할 경우, 방문자의 얼굴을 저장하여 애플리케이션에서 확인할 수 있다.

3.2.6 거주자 정보 관리

거주자의 ID, 이름, 전화번호, 생일, 호수, 얼굴 정보를 관리한다.

```
# Save user information (save user info + face)
def save_user_info(face):...

# Search user information by ID
def search_user_info(id):...

# Search users information by (ID, name, phone, birthday, room)
def search_users_by_info(info_type, info):...

# Delete user information
def delete_user_info(id):...

# Print user information by ID
def print_user_info(id):...

# User data menu (data management)
def user_data(face_module):...
```

그림 8. user_data.py

거주자의 정보를 저장, 검색, 삭제, 출력이 가능하다. ID는 거주자 1명의 고유한 숫자를 나타내지만, 이름이나 호수는 중복될 수 있으므로 정보에 알맞게 거주자의 정보 검색이 가능하다.

3.2.7 도어락 상태 제어

상황에 알맞도록 공동현관 도어락의 상태를 제어한다.

```
# Doorlock class
class Doorlock:
    # Doorlock Setting (servo motor, photo resistor, buzzer)
    def __init__(self):...

    # Doorlock open
    def open(self, name=None):...

    # Doorlock close
    def close(self):...

    # Doorlock incorrect
    def incorrect(self, type):...

    # Doorlock warning
    def warning(self):...

    # Doorlock reset
    def reset(self):...
```

그림 9. doorlock.py

도어락을 제어하는 클래스이며, 도어락의 상태에 따라 서보모터, 조도 센서, 피에조 부저를 사용하여 잠금 여부, 소리 재생, 공동현관 닫힘 여부 검사를 실행한다.

3.2.8 서버 (멀티 소켓 통신)

서버(라즈베리파이)와 클라이언트(애플리케이션)가 통신하도록 한다.

```
# Start server
def start(doorlock):...

# Client connected
def handle_client(client_socket, doorlock):...

# Update client address with user id
def update_client_address(user_id, client_address):...

# Send image to client
def send_image(client_socket, image_path):...

# Knock Knock (Server -> Client)
def knock_knock(user):...

# Send home sensor values (Arduino -> Client)
def return_home_sensor():...
```

그림 10. server.py

다중 소켓 통신을 통해 클라이언트와 연결한다. 그리고 클라이언트 주소를 저장하여 방문자의 호출, 방문자의 모습, 실내 환경 측정값들을 전송할 수 있다.

3.2.9 실내 환경 측정값 전송

아두이노에서 측정한 실내 환경 측정값들을 입력 받고 전송한다.



그림 11. home.py

애플리케이션에 실내 환경을 출력하기 위해 측정값들을 서버로 전송한다.

3.2.10 실내 환경 측정 (아두이노)

아두이노에서 실내 환경 측정값들을 측정하여 라즈베리파이로 전송한다.

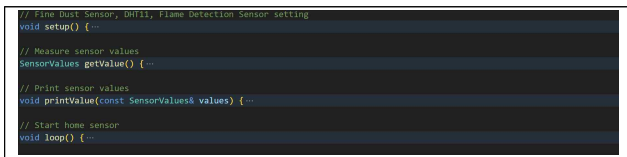


그림 12. home.ino

미세먼지 센서, 온도-습도 센서, 불꽃감지 센서를 사용하여 실내 환경을 측정한다.

4. 기능 구현

4.1 하드웨어 및 소프트웨어 플랫폼

표 1. 플랫폼 정보

플랫폼	주요 내용
하드웨어	Raspberry Pi 4 Model B: Servo Motor, Photo Resistor, Piezo Buzzer, Button, 3x4 Keypad, Camera Module, 16x2 LCD Arduino UNO R3: DHT11, Flame Detection Sensor, Fine Dust Sensor
소프트웨어, 웹웨어	Raspberry Pi: Raspberry Pi OS Arduino UNO: Arduino IDE Application: Android 13
응용 프로그램	Camera Module: OpenCV LCD: RGB1602 Fine Dust Sensor: PM2008 I2C DHT11: DHT11

본 논문에서 사용된 하드웨어 및 소프트웨어 플랫폼 정보는 다음과 같다.

4.2 임베디드 시스템

비밀번호 입력, 얼굴인식, 호출 기능을 통해 제어되며, LCD로 상태를 표시하고 서보모터, 피에조 부저, 조도 센서를 사용하여 잠금 상태, 소리 경고, 문 닫힘 여부를 관리한다.

비밀번호 입력 기능의 경우, 사용자는 도어락의 잠금을 해제하기 위해 설정된 비밀번호를 입력하여 잠금 상태를 해제할 수 있다.

얼굴인식 기능의 경우, 등록된 얼굴 정보를 인식하여 잠금 해제 여부를 결정할 수 있다. 저장된 거

주자의 얼굴 정보와 비교하여 얼굴 일치율이 60% 이상일 경우 도어락이 개방된다.

호출 기능의 경우, 방문자는 호출 기능을 사용하여 거주자에게 알림을 보낼 수 있으며, 거주자는 애플리케이션을 통해 방문자를 확인하고 도어락 개방 여부를 결정한다.



그림 14. 도어락 기능

도어락 상태는 16x2 LCD를 통해 출력되며, 출력 내용에는 잠금 상태, 비밀번호 일치 여부, 얼굴인식 일치 여부, 오류 및 경고, 거주자 호출 등이 포함된다. 또한, 도어락은 서보모터를 통해 잠금 상태를 제어하며, 피에조 부저를 사용하여 경고나 알림 소리를 제공하고, 조도 센서를 이용해 공동현관의 닫힘 여부를 검사한다.

4.3 사용자 정보 관리

사용자의 정보를 등록, 검색, 삭제할 수 있으며, 얼굴 정보를 검출하여 시스템에 저장하고 학습한다.

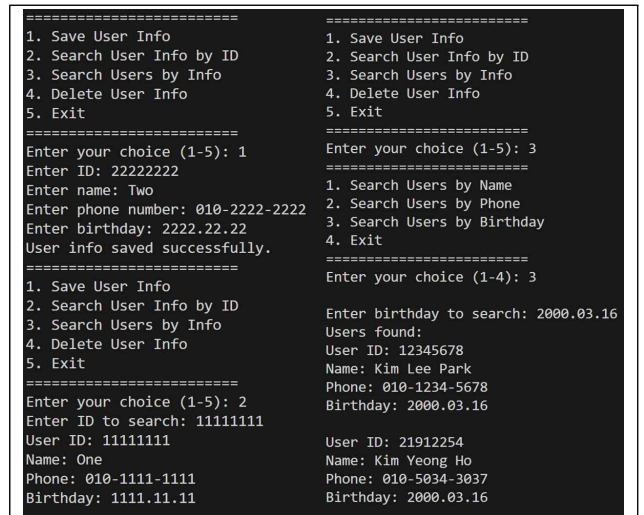


그림 15. 사용자 정보 관리

사용자의 정보를 등록하기 위해 학번, 이름, 전화번호, 생일, 주소 등의 회원 정보를 저장한 뒤, 얼굴 정보를 검출하여 시스템에 저장하고 학습시키는 과정을 거친다.

등록된 사용자 정보는 ID를 기준으로 검색할 수 있으며, 필요에 따라 정보 삭제도 가능하다. 또한 이름이나 호수는 중복될 수 있으므로, 고유한 ID를 통해 정확한 검색이 가능하다.

4.4 OpenCV 기반 얼굴인식

'Cascade Classifier Training'을 기반으로 얼굴 특징 추출하고, 실시간으로 얼굴을 감지하고 인식한다.



그림 16. 얼굴 학습을 위한 데이터 수집

사용자의 정보를 저장하는 과정에서 얼굴을 학습하기 위해 사용자의 얼굴을 여러 방향으로 각각 50장씩 학습 데이터셋에 저장한다. 이후 학습 데이터셋에서 얼굴이 포함된 이미지를 수집하고, 각 이미지에서 얼굴 영역을 지정한다.

각 이미지에서 지정된 얼굴 영역을 사용하여 특징 추출을 위한 기본 단위로 사용되는 얼굴 패치를 추출한 뒤, 각 얼굴 패치에서 밝기 차이를 나타내는 Haar-like 특징을 추출한다. 추출된 Haar-like 특징을 정규화하여 특징 값의 범위를 조정하고, 불필요한 특징을 제거한 뒤, 얼굴 특징 벡터를 생성한다. 마지막으로 생성된 얼굴 특징 벡터를 사용하여 Cascade Classifier를 학습한 뒤, 얼굴을 감지하고 인식할 수 있도록 한다.

이를 통해 추출된 얼굴 특징 벡터를 저장된 데이터베이스와 비교하여 가장 유사한 얼굴을 인식하고, 임계값을 기반으로 일치율에 따라 도어락의 개방 여부를 결정한다.

4.5 양방향 통신

서버인 스마트 출입관리 시스템과 클라이언트인 애플리케이션의 양방향 통신을 기반으로 방문자의 호출을 관리하며, 실내 환경 측정값을 제공한다.

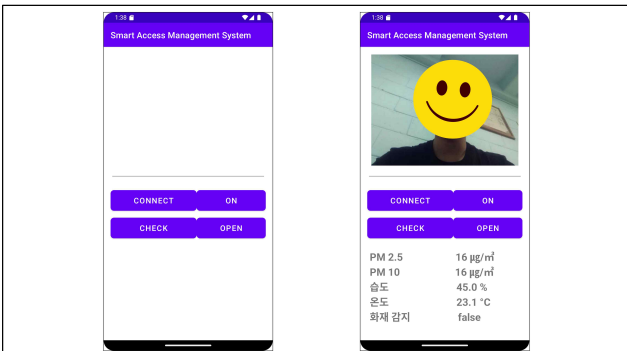


그림 17. 애플리케이션

서버 역할을 하는 라즈베리파이는 네트워크 연결을 설정하고, 다중 소켓 통신을 위해 특정 포트 번호를 할당하여 대기한다. 사용자는 클라이언트 역할인 애플리케이션을 실행하여 네트워크 연결을 초기화하고, 서버의 IP 주소와 포트 번호를 사용하여 서

버에 연결을 요청한다. 이때 클라이언트는 고유 ID를 전송하게 되며, 서버는 사용자 정보에 등록된 ID인지 검토한 후 연결 여부를 결정한다.

방문자가 도어락에서 특정 호수를 호출하면 서버는 해당 호수의 거주자 애플리케이션으로 방문자 호출 알림을 전송한다. 애플리케이션을 통해 알림을 확인한 뒤, 캡처된 방문자의 신원을 확인 후 도어락 개방 여부를 결정하여 응답 메시지를 서버로 보낸다. 서버는 이 응답 메시지를 처리하여 도어락을 제어한다.

서버는 아두이노에서 측정한 실내 환경 측정값을 수집하고, 이를 클라이언트 애플리케이션으로 전송한다. 애플리케이션은 수신한 센서 데이터를 바탕으로 실내 환경 수치를 사용자에게 보여주고, 위험 수치일 경우 경고 알림을 표시한다.

5. 성능 측정 및 분석

5.1 성능 측정 방법 및 기능 구성도

성능 측정 기능 구성도를 통해 양방향 통신 정보 전달 시간과 OpenCV 기반 얼굴인식 일치율을 측정하여 성능 측정을 진행한다.

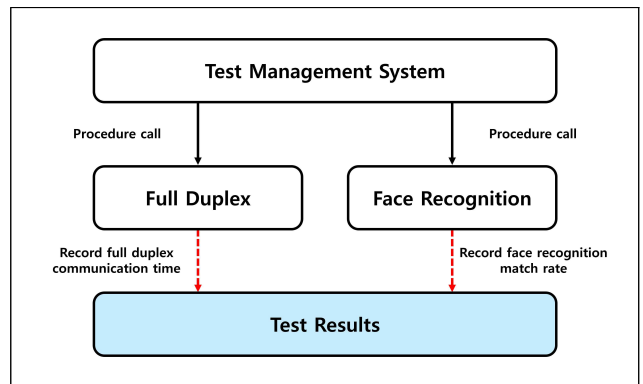


그림 18. 성능 측정 기능 구성도

양방향 통신 정보 전달 시간 측정은 서버에서 클라이언트로의 이미지 전송 시간을 측정한다. 측정 시스템은 서버에서 클라이언트로 이미지를 전송하고, 클라이언트가 수신 시간과 서버의 응답 시간을 기록하여 전송 시간을 계산한다. 목표는 정보 전달 시간을 1초 이내로 유지하는 것이며, 이를 위해 100회의 전송 시간을 측정한다.

OpenCV 기반 얼굴인식 일치율 측정은 얼굴인식 모델의 정확도를 평가한다. 정면만 학습한 경우와 정면 및 상하좌우의 측면을 학습한 경우, 각각 100회씩 10번의 테스트를 통해 모델의 일치율을 측정한다. 목표는 얼굴인식 일치율이 60% 이상이도록 하는 것이며, 이를 통해 표본 수를 늘려 성능이 개선되었는지 검토한다.

5.2 양방향 통신 측정 결과값 및 분석 내용

양방향 통신 측정은 서버에서 클라이언트로 이미지 전송 시간을 100회 측정하여 성능을 평가하였으며, 목표 성능은 정보 전달 시간 1초 이내로 설정하였다.

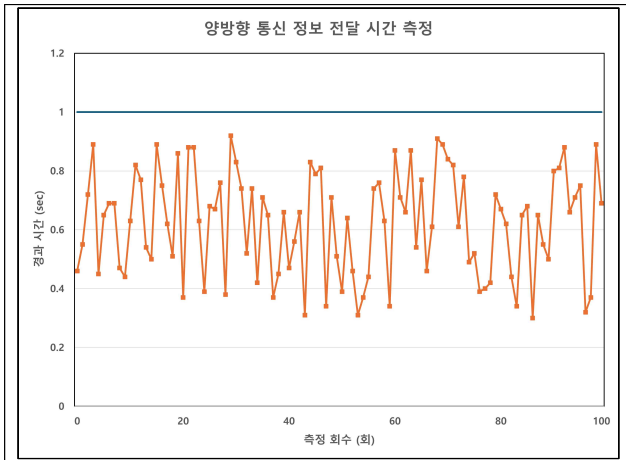


그림 19. 양방향 통신 정보 전달 시간 측정

양방향 통신 측정 결과, 100회의 측정값 평균은 0.62초로 성능 목표인 1초 이내를 성공적으로 충족하였다. 그래프를 통해 최댓값과 최솟값을 비교하였을 때, 극단적인 값이 나타나지 않아 전체적으로 일관된 성능을 보여준다. 선 그래프를 사용하여 각 측정값과 평균값을 비교한 결과, 대부분의 측정값이 평균값에 근접하게 분포하고 있다.

이를 통해 서버와 클라이언트 사이의 정보 전달 시간이 1초 이내로 성능 목표를 만족하는 것을 확인할 수 있다.

5.3 얼굴인식 측정 결과값 및 분석 내용

OpenCV 기반 얼굴인식의 일치율을 정면만 학습한 경우와 정면 및 상하좌우 측면을 학습한 경우로 각각 100회씩 10번 측정하여 성능을 평가하였으며, 목표 성능은 얼굴인식 평균 일치율 60% 이상으로 설정하였다.

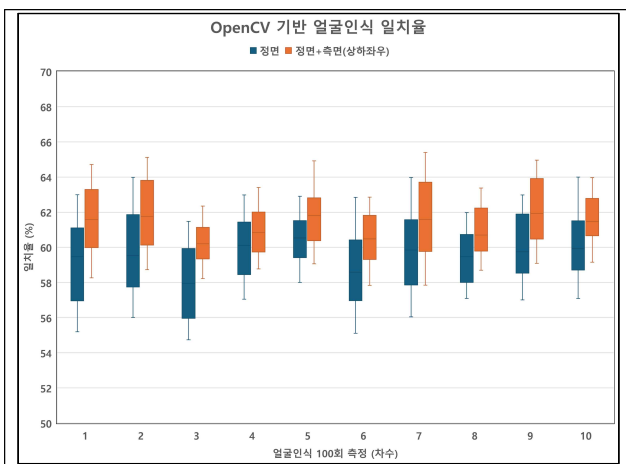


그림 20. OpenCV 기반 얼굴인식 일치율

정면만 학습한 경우, 평균 일치율이 약 59%로 목표 일치율에 약간 미치지 못하였다. 일치율은 55%에서 63% 사이에 분포하였으며, 측정된 일치율 중 일부는 목표를 초과하였지만 전체 평균에서는 목표를 달성하지 못하였다.

정면과 상하좌우 측면을 함께 학습한 경우, 평균 일치율이 약 61.5%로 목표 일치율을 충족하였다. 일치율은 58%에서 65% 사이에 분포하였으며, 정면만

학습한 경우보다 더 높은 일치율을 기록하였다.

이를 통해 다양한 각도의 표본을 증가시켜 얼굴인식의 정확도가 개선되었고, 얼굴인식 평균 일치율이 60% 이상으로 성능 목표를 만족하는 것을 확인할 수 있다.

6. 결 론

공동현관 도어락의 한계점 개선을 목표로, 기존 방식의 문제점을 해결하고 성능을 향상시키기 위한 새로운 접근 방안을 제시하였다. 스마트 도어락 시스템의 중요성은 보안과 편의성 측면에서 크며, 이 논문에서 제안한 해결방안은 이러한 시스템의 실용성을 더욱 높이는 데 필요하다.

본 논문에서는 라즈베리파이와 아두이노를 사용하여 양방향 통신과 OpenCV 기반 얼굴인식을 통한 스마트 도어락 시스템을 제안하였다. 양방향 통신은 서버인 라즈베리파이와 클라이언트인 애플리케이션의 양방향 통신 기능을 제공하였으며, OpenCV 기반 얼굴인식은 다양한 각도의 얼굴 데이터를 학습하여 얼굴인식 일치율을 높이는 방식을 사용하였다. 성능 측정 방법으로는 양방향 통신의 경우 100회의 이미지 전송 시간을 측정하였으며, OpenCV 기반 얼굴인식은 정면 및 다양한 각도의 얼굴 데이터를 각각 100회씩 10번 테스트하여 일치율을 평가하였다. 그 결과, 양방향 통신의 정보 전달 시간을 1초 이내로 성공적으로 유지하였고, 얼굴인식의 경우 다양한 각도의 데이터를 학습한 모델이 평균 60% 이상의 일치율을 기록하여 성능 목표를 충족하였다.

스마트 도어락 시스템을 통해 기존의 공동현관 도어락의 한계점을 개선하고, 보안성과 신뢰성을 높이는 데 기여하였다. 향후 계획으로는 시스템의 안정성을 더욱 향상시키기 위해 다양한 환경에서의 테스트를 추가하고, 얼굴인식 정확도를 높이기 위한 알고리즘 개선을 계획하고 있다.

참고문헌

- [1] 유희중, “범죄예방을 위한 단위세대의 현관문 인식에 관한 설문조사 연구”, Vol. 16, No. 3, pp 188, 2018.
- [2] “Real-Time Face Recognition: An End-to-End Project”, MJRoBot (Marcelo Rovai), February 23, 2018.