

## 개방형 API를 사용한 챗봇과 영상 기반 비대면 출입자 모니터링 서비스

김태희 박구만

서울과학기술대학교 전자IT미디어공학과

ldos96@naver.com

## Non-Contact Monitoring Service based on Chatbot and Video using Open API

Kim, Tae-Hee Park, Goo-Man

Seoul National University Of Science And Technology Department of Electronic and IT Media Engineering

## 요약

코로나19로 인해 출입 관련 시스템도 비대면으로 변화하고 있다. 변화에 맞추어 비대면으로 출입자를 관리할 수 있는 프로그램을 개발하여 접촉 위험을 줄이고 출입자 모니터링에 실용성을 제공하고자 한다. 본 연구에서는 Raspberry Pi 카메라에 Alchera Face Authentication API를 적용하여 얼굴인식을 실시하며 정보를 AWS 클라우드에서 저장·관리 하는 시스템을 개발하였다. 챗봇 서비스를 통해 출입자를 확인할 수 있으며 메신저에서 쉽게 클라우드에 접근하여 정보를 확인할 수 있게 하였다. 이를 통해, 특정 장소를 비대면으로 관리하며 간편하게 출입자를 모니터링할 수 있을 것으로 기대한다.

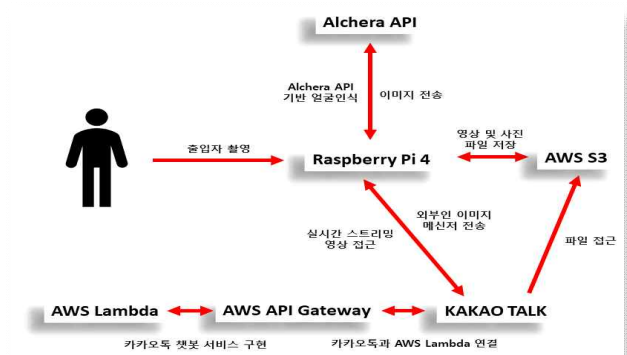
## 1. 작품의 제작 동기

코로나19가 발생한 후 비대면의 중요성이 대두되자 다양한 무인화 서비스가 출시되고 있다. 하지만 무인화 서비스에 대한 보안 시스템은 아직 부족한 것으로 사료된다. 특히 규모가 작은 무인상점에서 절도 및 무단 취식 범죄가 빈번히 발생하고 있다. 계속해서 무인화 서비스가 출시되는 만큼 이러한 무인화 서비스에 관한 적절한 보안 서비스가 필요해 보인다. 이러한 문제를 해결하기 위해 비대면 출입자 모니터링 시스템 제작을 고민하게 되었다.

비대면 출입자 모니터링 시스템은 안정적인 API를 활용하여 편리하고 실용적인 감시 서비스를 목표로 한다. 모바일 앱 데이터 플랫폼 기업 IGAWorks의 조사에 따르면 카카오톡은 월간 사용자 수가 4566만으로 대한민국의 90% 이상이 사용하는 앱이다. 사용빈도가 높은 open API를 활용할 수 있는 카카오톡 서비스를 사용하여 출입자를 모니터링할 수 있다면 매우 편리한 서비스가 될 수 있을 것이라 생각한다.

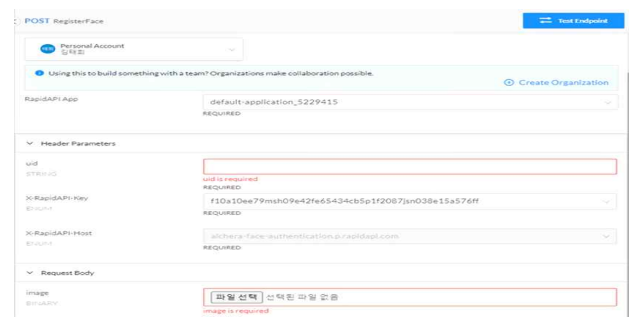
## 2. 작품의 설계 및 구현

본 서비스는 Raspberry Pi 4 기반으로 시스템이 동작한다. Raspberry Pi 4에 연결된 카메라를 Python으로 제어하여 출입자의 얼굴을 인식하는 순간 촬영한다. 그리고 촬영한 이미지 파일을 Alchera에서 제공하는 Face Authentication API의 데이터베이스로 보내 얼굴인식을 진행한다. API는 얼굴인식과 함께 데이터베이스를 제공하며 암호화된 바이너리 데이터를 저장하여 보안성이 뛰어나다[1]. 이 데이터베이스에 인물의 안면 정보를 등록할 수 있다.



&lt;그림 1&gt; 비대면 출입자 모니터링 전체 블록도

Face Authentication API에서 제공하는 API Key를 Python에서 requests 형식으로 사용하여 API의 데이터베이스와 정보를 주고받는 다.



&lt;그림 2&gt; Face Authentication API 데이터 등록화면

```
200
{"result":{"code":0,"message":"OK"},"uid":"th_kim","match":true}

200
{"result":{"code":0,"message":"OK"},"uid":"","match":false}
```

&lt;그림 3&gt; 인식 결과를 텍스트 형태로 출력한 결과

위의 결과에서 아래 텍스트가 등록되지 않은 출입자로 인식된 것이다. 이러한 결과가 출력되었을 때, 이미지 파일과 경고 문구를 카카오톡 메신저를 통해 사용자에게 보낸다. 이를 위해 카카오톡 채널을 개설하였다. 그 후, 카카오톡에서 제공하는 챗봇 서비스를 구현하여 개설한 채널에 적용하였다. 카카오톡에서 제공하는 오픈빌더는 챗봇의 기능을 개발할 수 있는 환경을 제공하여 사용자가 원하는 기능을 채널에 추가할 수 있다 [2].

챗봇 서비스를 통해 Amazon S3 웹 서비스 접근하고, 실시간으로 카메라 스트리밍 영상을 확인할 수 있는 URL을 받아오는 기능을 구현하였다. 기능 구현을 위해 AWS Lambda 서비스에서 챗봇 서비스 구현 코드를 작성하였다. 그 후, Lambda와 카카오톡 채널을 AWS API Gateway로 연동시킴으로써 작성한 챗봇 기능 구현 코드를 카카오톡 채널에 적용하였다. 카카오톡 채널에서 [화면 보여줘, 화면] 등의 단어를 입력하면 실시간 스트리밍 URL을 보내주고, [영상, 영상 보여줘] 등의 단어를 입력하면 Amazon S3에 저장된 영상을 다운로드 받을 수 있는 웹사이트 URL을 받아 영상 파일에 접근할 수 있다. 자세한 동작은 구현 결과에서 설명하겠다.



&lt;그림 3&gt; 영상 파일에 접근하는 단어 목록 (엔티티)

Amazon S3은 boto를 통해 Raspberry Pi와 연동된다. 카메라를 통해 녹화되는 파일이 Amazon S3에 저장된다. 또한 촬영된 사진도 AWS S3에 함께 저장된다.

### 3. 작품의 구현 결과

본 서비스의 카카오톡 채널은 카카오톡 어플리케이션에서 검색하면 쉽게 찾아볼 수 있다. 채널을 친구추가하면 서비스를 이용할 준비가 된다.



&lt;그림 4&gt; 카카오톡 채널 검색 화면

Raspberry Pi에서 프로그램을 실행시키면 자동으로 카카오톡 웹사이트에 접속하여 관리자 채널에 로그인한다. 로그인된 관리자 채널을 통해 채널을 추가한 사용자에게 메시지로 인식 결과를 보낸다.

OK  
등록된 사람입니다.

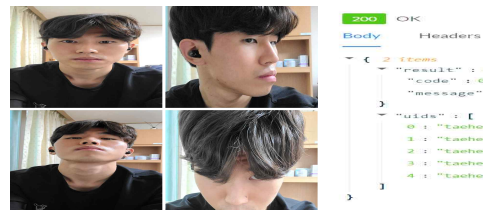
&lt;그림 5&gt; 등록 인물 인식 결과

이제 출입자가 인식되었을 때의 동작을 살펴보겠다. 위 그림은 등록된 인물이 인식되었을 때의 결과이다.



&lt;그림 6&gt; 좌:데이터베이스, 우:인식 장면

처음 Alchera 데이터베이스에 등록한 사진은 본인의 증명사진 한 장이었다. 그때의 얼굴 인식 결과는 10번 시도했을 때 4번 올바르게 인식되었고 나머지는 등록되지 않은 인물로 판단되어 메시지가 전송되었다. 인식률을 높이기 위해 여러 각도에서 찍은 사진을 데이터베이스에 추가한 뒤 다시 인식을 시도하였다.



&lt;그림 7&gt; 다양한 각도에서의 사진 저장 &lt;그림 8&gt; 데이터베이스 화면

얼굴의 앞, 옆, 위, 밑에서 사진을 찍어 등록하였으며 이때의 인식률은 10번 시도했을 때 10번 모두 정확하게 인식하였다. 추가로 안경을 쓴 사진을 등록하여 인식하였는데 처음에는 인식되지 않던 모습이 정확히 인식되었다. 즉, 데이터베이스에 등록하는 사진이 다양할수록 더 높은 인식률을 얻을 수 있었다. 하지만 같은 사진을 올릴 경우 데이터베이스에서 오류가 발생하며 등록되지 않는다.



&lt;그림 9&gt; 같은 사진을 등록할 때 발생하는 결과

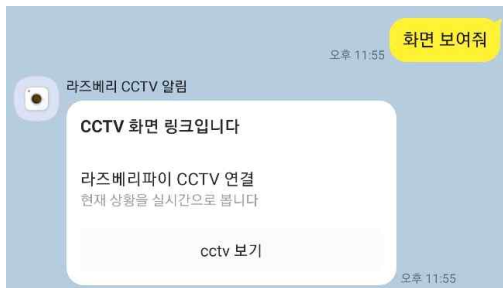
또한, 실험 결과 카메라와 사람의 거리가 60cm일 때 인식이 가장 활발하다는 것을 확인할 수 있었다.

이제 등록되지 않은 인물이 인식되었을 때의 결과를 살펴보겠다. 등록되지 않은 인물의 경우 [등록되지 않은 사람이 감지되었습니다]라는 문구와 함께 인물의 사진을 채널 사용자에게 전송한다.



<그림 10> 등록되지 않은 인물이 인식되었을 때의 결과

미등록 출입자에 대한 정보를 제공받은 사용자는 그 상황을 실시간으로 확인할 수 있다. 카카오톡 채널에서 [화면 보여줘] 라고 입력하여 전송하면 실시간 스트리밍 화면을 볼 수 있는 URL을 제공해준다.



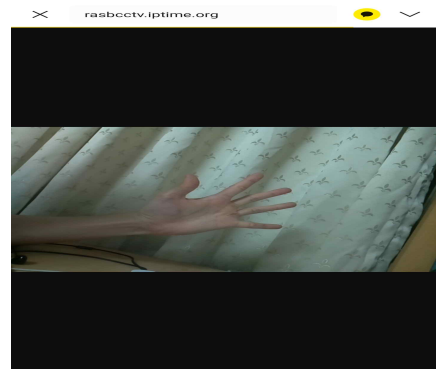
<그림 11> [화면 보여줘]에 대한 결과 (모바일 버전)

챗봇 구현 기능 중 나만의 엔티티를 만들어 사용할 수 있는 기능이 있다. 엔티티는 단어의 묶음이라 표현할 수 있는데 유사한 단어를 하나로 묶어 어떠한 기능에 적용시키면 유사 단어를 입력해도 같은 결과를 출력해준다 [3].



<그림 12> 유사 단어에 대해 같은 결과 출력 (PC 버전)

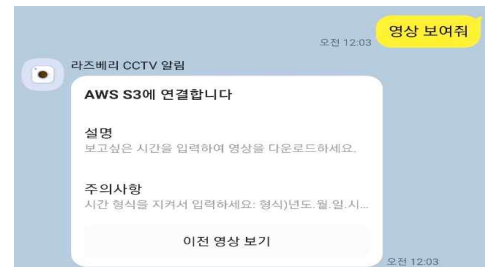
[cctv 보기] 버튼을 클릭하면 실시간 스트리밍 화면을 볼 수 있다. 실시간 스트리밍은 mjpg streamer를 사용하였다 [4]. 위의 그림을 통해 알 수 있듯이 실시간 스트리밍은 모바일 버전에서만 가능하다. 실시간 스트리밍 url은 Raspberry Pi가 연결되어 있는 공유기로 연결된다.



<그림 13> 실시간 스트리밍 영상

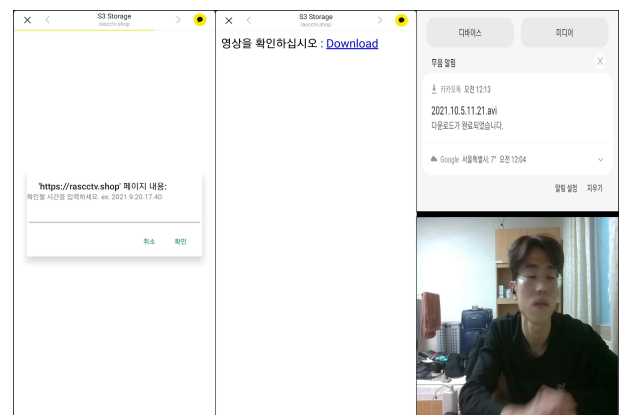
Amazon S3에 저장되어 있는 파일에 접근하고 싶을 땐 [영상 보여줘]를 입력하면 영상에 접근할 수 있다. [이전 영상 보기] 버튼을 클릭하면 웹 사이트로 접속하게 된다. 연결되는 웹 사이트는 도메인을 구입하여 구현한 웹 사이트이다. AWS Route53에서 이 도메인을 등록하여 Amazon S3에서 정적 웹 사이트를 호스팅하도록 하였다 [5].

웹 사이트에 접속하여 보고 싶은 시간대를 입력하면 입력한 시간부터 이후 1분까지 녹화된 영상 파일을 다운로드할 수 있다.



<그림 14> [영상 보여줘]에 대한 결과

주의사항은 시간의 입력 형식을 지켜야하는 것이다. 예시로 [2021.9.20.17.20]처럼 원하는 시간을 yyyy.m.d.h.m분으로 적어야 한다.



<그림 15> Amazon S3에 접근하여 영상 다운로드

객체 (56)

객체는 Amazon S3에 저장되어 있는 기본 엔티티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. [자세히 알아보기](#)

S3 URL 복사 URL 복사 다운로드 열기 삭제 작업 볼더 만들기

필요도

Q 객체로 검색하기

| 이름                   | 유형  | 마지막 수정                          | 크기      | 스토리지 클래스 |
|----------------------|-----|---------------------------------|---------|----------|
| 2021.10.28.16.16.avi | avi | 2021. 10. 28. pm 4:17:02 PM KST | 1.0MB   | Standard |
| 2021.10.28.16.20.avi | avi | 2021. 10. 28. pm 4:21:02 PM KST | 697.1KB | Standard |
| 2021.10.28.16.21.avi | avi | 2021. 10. 28. pm 4:22:02 PM KST | 665.4KB | Standard |
| 2021.10.28.16.24.avi | avi | 2021. 10. 28. pm 4:25:02 PM KST | 401.0KB | Standard |
| 2021.10.28.16.25.avi | avi | 2021. 10. 28. pm 4:26:02 PM KST | 763.3KB | Standard |
| 2021.10.28.16.26.avi | avi | 2021. 10. 28. pm 4:27:02 PM KST | 875.1KB | Standard |
| 2021.10.28.16.29.avi | avi | 2021. 10. 28. pm 4:30:03 PM KST | 205.0KB | Standard |
| 2021.10.28.16.30.avi | avi | 2021. 10. 28. pm 4:31:02 PM KST | 710.3KB | Standard |

&lt;그림 16&gt; Amazon S3에 저장된 영상 파일

#### 4. 작품의 기대효과

본 서비스는 카카오톡 메신저로 출입자의 정보를 받기 때문에 모니터링을 할 때 편리함을 느낄 수 있다. 사용자가 어디에 있던 카메라를 통해 현 상황을 확인할 수 있고 이전의 상황도 녹화된 영상 파일을 통해 간편하게 확인할 수 있다. 따라서 카메라를 장소의 제약 없이 설치하여 실용적으로 출입자를 관리함으로써 손쉽게 보안을 강화할 수 있을 것으로 사료된다. 본 서비스를 제어 시스템을 결합한다면 더 큰 시너지를 낼 수 있을 것으로 기대한다. 보안 출입문과 같은 시스템에 본 서비스를 적용시키면 얼굴인식을 활용하여 잠금을 해제하고 출입문 앞 상황을 간편하게 모니터링하면서 원격으로 출입을 제어가 가능해질 수 있을 것이다.

또한, 본 서비스는 다른 분야에도 적용될 수 있을 것으로 보인다. 카카오톡 채널을 사용하여 누구나 쉽게 접근이 가능하기 때문에 다수에게 메시지를 전달하는 서비스에도 활용이 가능할 것이다. 예를 들어, 등산객에게 야생동물의 출현을 알려주는 서비스로 활용할 수 있을 것이다.

현재는 다양한 개방용 API를 사용하였지만 본 서비스와 같은 목적의 프로세스를 구현하기 위해 자체 알고리즘을 구현하여 인식률을 높이고 서비스의 성능을 더욱 향상시킬 계획이다. 이를 통해 사용자에게 한층 더 뛰어난 비대면 모니터링 서비스를 제공하여 높은 실용성과 편리함을 제공할 수 있을 것으로 기대한다.

#### 5. 참고문헌

[1] Rapid API,

Link:

<https://rapidapi.com/alchera-alchera-default/api/alchera-face-authentication/details>

[2] 카카오 i open builder, “카카오 I 오픈빌더 도움말”,

Link:

<https://i.kakao.com/docs/getting-started-overview#%EC%98%A4%ED%94%88%EB%B9%8C%EB%8D%94-%EC%86%8C%EA%B0%9C>

[3] 카카오 i open builder, “카카오 i 오픈빌더 도움말-엔티티”,

Link:

<https://i.kakao.com/docs/key-concepts-entity>

[4] jacksonliam, Github-mjpg-streamer

Link:

<https://github.com/jacksonliam/mjpg-streamer>

[5] Amazon Web Services, “Route 53에 등록된 사용자 지정 도메인

을 사용하여 정적 웹 사이트 구성”,

Link:

[https://docs.aws.amazon.com/ko\\_kr/AmazonS3/latest/userguide/website-hosting-custom-domain-walkthrough.html](https://docs.aws.amazon.com/ko_kr/AmazonS3/latest/userguide/website-hosting-custom-domain-walkthrough.html)