

<https://doi.org/10.7236/JIIBC.2022.22.1.171>

JIIBC 2022-1-24

실시간 코로나19 확진자 웹 모니터링 시스템에 대한 연구

Study on the Real-time COVID-19 Confirmed Case Web Monitoring System

유영균*, 조성욱*, 고동범**, 박정민***

Youngkyon You*, Seonguk Jo*, Dongbeom Ko**, Jeongmin Park***

요약 본 논문은 캠퍼스 내 건물별 출입구에 출입명부를 관리할 수 있는 장치를 설치하고, 수집된 데이터를 근거로 코로나19 확진자에 대한 감시 및 추적 시스템을 소개한다. 기존 QR 기반의 전자출입명부는 스마트폰으로 QR 코드를 인식해야 하는 불편함과 건물 내 출입자의 온도를 측정할 수 없다는 단점이 있다. 또한, 국가가 캠퍼스 내의 확진자와 접촉자 정보를 관리함에 있어, 구성원들에게 신속한 정보 공유와 추적이 쉽지 않다. 이는 확진자와 밀접 접촉한 이가 또 다른 확진자를 만드는 경우를 초래할 수 있다. 따라서, 본 논문에서는 이에 대응하기 위해 얼굴 인식 기능과 온도 센서를 포함한 장치를 캠퍼스 내 건물 출입구에 설치하고, 관리자가 실시간으로 건물별 구성원들의 출입현황을 모니터링 하여 신속하게 추적할 수 있는 기능을 소개한다.

Abstract This paper introduces a monitoring and tracking system for corona19 confirmed patients based on the collected data by installing a device that can manage the access list at the entrance to each building on the campus. The existing QR-based electronic access list can't measure the temperature of the person entering the building and it is inconvenient that members have to scan their QR codes with a smartphone. In addition, when the state manages information about confirmed patients and contacts on campus, it is not easy for members to quickly share and track information. These could lead to cases where a person is in close contact with an infected person developing another patient. Therefore, this paper introduces a device using face recognition library and a temperature sensor installed at the entrance of each building on the campus, enabling the administrator to monitor the access status and quickly track members of each building in real-time.

Key Words : Face Recognition, Real-Time Monitoring, Tracking, SMS Alert, Stable Server

*준회원, 한국산업기술대학교 컴퓨터공학부

**준회원, 한국전자통신연구원 인공지능연구소

***정회원, 한국산업기술대학교 컴퓨터공학과(교신저자)

접수일자 2021년 12월 1일, 수정완료 2022년 1월 2일
게재확정일자 2022년 2월 4일

Received: 1 December, 2021 / Revised: 2 January, 2022 /

Accepted: 4 February, 2022

*Corresponding Author: jmpark@kpu.ac.kr

Dept. of computer engineering, Korea Polytechnic University,
Korea

I. 서 론

코로나19 확진자가 증가함에 따라 정부는 확진자 발생 시 확진자의 동선을 파악하고 국민에게 알림 서비스를 제공하기 위해 매장이나 건물 출입구에 QR 코드 인증용 태블릿 PC를 설치해 QR 코드를 찍게 하거나 안심콜을 걸어서 출입명부를 기록하는 시스템을 갖추게 했다. 이는 국가가 대기업들과 협업하여 QR 코드와 안심콜로부터 생성된 많은 데이터를 관리하며 확진자 발생 시에 대응하기 위함이다.

미국의 경우 차세대 보안 기술로 주목받고 있는 블록체인 기술과 QR 코드를 결합하여 전자출입명부를 구성한 예가 있다^[1]. 이는 건강관리시스템의 탈중앙화를 통해 데이터의 보안과 투명성을 증가시키고 DLS(distributed ledger system)를 통해 중복 작업을 방지하기 위함이다. 즉 데이터가 탈중앙화된 데이터베이스로 수집, 공유, 접근되기 때문에 국가의 각 지역은 단순히 정보를 공유하는 것으로 일이 줄어드는 것이다. 그러나 일일이 핸드폰으로 QR 코드를 찍어야 하는 불편함은 변하지 않는다.

'Proximity-base COVID-19 contact tracing system'은 회사나 공장에서 직원들이 코로나에 감염될 경우 밀접접촉으로 검사 해야 할 직원을 판별하기 위해 고안된 시스템이다^[2]. BLE(Bluetooth Low Energy) 기술을 이용한 이 확진자 감시 시스템은 사람이 많은 회사나 공장에서 확진자가 발생하면 확진자와 밀접 접촉한 사람들을 판별해서 검사를 받게끔 한다. 직원들의 옷이나 바지 주머니에 작은 장치를 넣고 스마트폰과 동기화시킨 다음 확진자로 판명되는 직원과 밀접 접촉한 직원의 정보를 데이터베이스로부터 얻어 검사를 받게끔 한다. 그러나 직원이 가지고 있는 장치가 훼손되거나 장치를 소지하고 있지 않으면 부정확한 결과를 초래할 수 있다는 문제가 있다.

다른 확진자 관리 애플리케이션으로는 얼굴 인식 기술을 기반으로 한 확진자 관리 시스템이 있다.^[3] 이 시스템은 감기에 걸린 사람에 대해서 얼굴 인식을 하고 온도 측정을 한 다음 데이터를 저장한다. 이 시스템은 추적을 위한 별도의 알고리즘을 제공하지 않지만 추후 감기에 걸린 사람이 확진자로 판명되면 그 확진자의 최근 14일 동안의 얼굴 이미지와 타임 스탬프를 조회하고 밀접 접촉한 사람들에 대한 정보를 보건 당국에 보내 조치를 취하게 한다. 하지만, 보건 당국이 대응할 때까지 많은 시간이 걸릴 수 있으며 그동안 또 다른 확진자를 만들어 낼 수 있고 미리 구성원 등록을 해 놔야 한다는 단점을 가지

고 있다.

따라서, 본 논문에서는 기존 연구 및 시스템의 단점을 보완하여 기업이나 학교 등 자체적으로 비상대책본부를 설립하여 개인정보와 출입명부를 관리하고 확진자 발생 시 빠르게 대응하는 방법에 대해 논의하고자 한다. 웹 기반으로 설계된 애플리케이션은 얼굴 인식 장치로부터 구성원의 얼굴을 인식해 관리자가 실시간으로 건물별 구성원의 출입 현황을 모니터링할 수 있게 한다. 또한, 출입 시간뿐만 아니라 온도까지 같이 표시하며 온도별로 구성원들을 분류한다. 이는 위험군, 경고군, 정상군으로 나눠 관리자가 한눈에 관리하기 쉽게 구성한다.

모든 출입 로그는 데이터베이스로 투명하게 저장되며 이 데이터를 사용해 확진자 발생 시에는 밀접 접촉자에게 SMS 알림을 보내거나 웹 관리자가 여러 조건을 조합하여 검색을 진행한 다음 검색 결과에 해당하는 구성원들에게 알림을 보낼 수 있으며 확진자 동선의 시각화 서비스를 통해 한눈에 확진자의 동선과 방역이 필요한 장소를 파악할 수 있다. 얼굴 인식 장치는 한 건물에 종속되기보다는 재등록을 통해 어느 건물에서나 재사용이 가능하고 구성원으로 등록되어 있지 않은 사람은 기기로부터 QR 코드를 받아 빠르게 구성원 등록이 가능하다. 추가로 안정적인 서버 운영을 위해 AWS에서 제공하는 EKS(Elastic Kubernetes Service)를 사용하고 프로메테우스와 그라파나를 연동하여 리소스 모니터링을 가능하게 구성하고자 한다.

II. 관련 연구

본 장에서는 알림 서비스를 사용한 정보 공유 체계가 가져온 이점과 본 논문에서 확진자 감시를 위해 사용한 기술 및 전체적인 시스템 구조에 대해 알아본다.

1. 정보통신기술을 기반으로 한 국가 관리 시스템

국가 관리 시스템은 정보통신기술(ICT)의 발전과 메르스 사태로 인해 개정된 감염병 예방법을 기초로 한다^[4]. 현재 스마트폰과 이동통신사업의 발전으로 모든 국민의 핸드폰 보급률은 100%에 가깝다. 즉, 정부는 이를 고려해 자연재해나 사회재난 발생 시에 발생지역에 있는 국민의 휴대전화로 한 번에 메시지를 송신함으로써 긴급 알림을 해왔다. 정부는 국민과의 소통의 매개체로 휴대전화를 적극적으로 선택한 것이다^[5].

다양한 매개체로 국민과 실시간으로 소통을 했다는 점과 필요한 정보를 대중들에게 전달하는 소통 창구로 유튜브, 페이스북이나 긴급재난문자를 사용하여 정부 및 지자체가 일관되고 정확한 정보를 국민과 시민들에게 효율적으로 전파하였다.

2. 얼굴 인식 기술

얼굴 인식 장치는 얼굴 인식 기술로 얼굴의 특징점을 추출하여 개인을 인식하는 라이브러리를 사용한다^[6]. 특정 사람의 사진을 작업 폴더에 위치시키고 명령어를 입력하면 그 사진을 기반으로 저장해둔 다른 사진들과 비교하여 입력한 사진에 나온 사람을 식별한다. 해당 라이브러리는 이미지에서 얼굴의 위치(픽셀 좌표)를 찾는다. 얼굴은 색상이 필요하지 않기 때문에 흑백으로 바뀌게 되고 이미지의 모든 단일 픽셀과 해당 픽셀을 둘러싸고 있는 픽셀들을 살핍으로써 얼마나 어두운지 알아낸다. 이를 통해서 이미지가 어두워지는 방향을 나타내는 화살표를 그리게 된다. 이를 반복하게 되면 결국 모든 픽셀이 화살표로 바뀌고 이러한 화살표들을 그래디언트(gradients)라고 부르게 된다. 이를 통해 전체 이미지에서 밝은 부분으로부터 어두운 부분으로의 흐름을 알게 된다.

결과적으로 원본 이미지의 간단한 표현을 얻게 되고 이를 사용하여 카메라로 인식된 사람과 표현에 가장 가까운 사람을 찾게 되는 것이다. 이 라이브러리는 딥러닝으로 구축된 dlib의 최첨단 얼굴 인식을 사용하여 구축되었으며 모델의 정확도는 'Labeled Faces in the Wild' 벤치마크에서 99.38%의 정확도 평가를 받았다. 이를 사용하여 캠퍼스 내 구성원의 얼굴을 인식하여 개인을 식별하고 온도 데이터와 함께 데이터베이스에 저장한다.

3. Kubernetes Cluster

2014년 6월에 처음 구글로부터 공개된 쿠버네티스는 다수의 컨테이너 관리, 자동 배포 및 배포된 컨테이너에 대한 동작 보증과 부하에 따른 동적 확장 등의 기능을 제공한다^[7]. 본 논문에서는 웹 서버의 지속적이고 안정적인 운영을 위해 쿠버네티스를 사용한다. 이를 통해 서버 점검 시에도 지속적인 서버 운영이 가능해지고 다수의 구성원이 출입함으로써 발생하는 많은 요청 및 응답에 대해 부하분산이 가능해진다.

현재 퍼블릭 클라우드 업체에서 제공하는 관리형 쿠버네티스로는 EKS(Amazon Elastic Kubernetes Service),

AKS(Azure Kubernetes Services), GKE(Google Kubernetes Engine) 등이 있다. 본 논문에서는 노드들을 자동으로 관리해주는 EKS를 cloudformation으로 구성된 쿠버네티스 클러스터를 사용한다.

III. 시스템 구조 및 프로세스

본 장에서는 구성원의 동선 추적 및 관리를 위한 실시간 감시 시스템 구조의 전체적인 구조 및 설계를 논의한다.

1. 시스템 프로세스

본 절에서는 캠퍼스 구성원들이 건물에 출입 시 진행되는 프로세스 4단계를 설명한다.

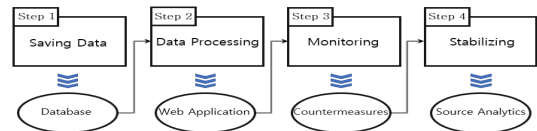


그림 1. 시스템 프로세스
 Fig. 1. System Architecture

Step 1: Saving Data

캠퍼스 내 건물에 출입하는 구성원들은 온도 측정 센서가 포함된 얼굴 인식 장치를 통해 인식을 진행한다. 장치로부터 얻어진 구성원의 간단한 신상, 체온, 건물 정보 및 출입 시간은 데이터베이스로 저장된다.

그림 2와 같이 마스크를 쓰고 있다면 마스크를 내리고 장치가 인식하게 해야 한다. 인식 후 장치의 스크린에는 인식한 구성원의 이름 및 온도가 표시된다. 장치로부터 얻은 데이터는 EKS 클러스터 내에서 컨테이너 형태로 실행되고 있는 데이터베이스에 저장된다.



그림 2. 장치를 통한 얼굴 인식과 온도 측정
 Fig. 2. Face Recognition and Temperature Measurement

Step 2: Data Processing

그림 3은 해당 시스템의 ERD를 나타낸다. 여러 테이블 중 status 테이블이 중요한 부분을 담당한다. status 테이블은 매핑 테이블의 역할을 하며 이는 member 테이블과 facility 테이블의 PK를 외래 키로 참조하는 테이블로 값 집합을 저장한다. status 테이블에는 출입 시간, 온도, 건물 정보 그리고 구성원에게 고유하게 할당된 번호가 기록된다. facility 테이블과 member 테이블이 1:N 관계를 구성하며 구성원은 여러 건물에 출입할 수 있음을 나타낸다.

sms 테이블은 구성원 개인적으로 자신의 동선을 알고 싶을 때 동선 정보를 제공하기 위해 만들어졌다. 특정 번호로 자신의 이름을 적어서 문자를 송신하게 되면 전화번호와 문자에 적힌 이름을 member 테이블을 통해 대조하는 간단한 인증 절차를 거치고 최근 일주일간의 동선을 문자로 보내주는 서비스를 제공한다. 구성원은 본인 외의 다른 구성원의 동선을 알릴 수 없으며 문자에 적힌 이름이 전화번호와 일치하지 않을 시에는 인증이 제대로 되지 않았다는 알림 문자를 받게 된다. 그 외 admin 테이블은 웹에 접근할 수 있는 관리자 정보를 저장한다.

데이터베이스로 저장되는 데이터는 여러 조건을 거친 질의를 통해 웹에 표시된다. 데이터베이스는 Spring에서 JPA를 사용하여 구현하고 동적 쿼리 작성을 위해 QueryDSL를 사용한다.

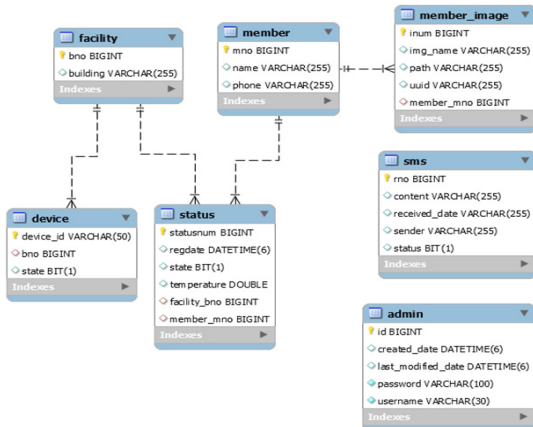


그림 3. ERD
Fig. 3. Entity-Relationship Diagram

Step 3: Monitoring

그림 4는 캠퍼스 내 코로나19 감시체계를 관리하는 관리자가 웹을 통해 실시간으로 구성원들의 상태를 대시

보드 형태로 확인할 수 있음을 보여준다. 건물별로 구성원의 출입 현황과 온도를 포함한 로그가 실시간으로 업데이트되며 이는 다양한 그래프로 구별되고 시각화된다. 온도는 정상군, 경고군, 위험군으로 나뉘게 되며 그룹마다 구성원을 모아볼 수 있는 기능을 제공한다. 확진자 발생 시에는 검색처리를 통한 알림이 가능해 빠르게 구성원들에게 확진자 동선 알림 및 검사 촉구를 할 수 있다.

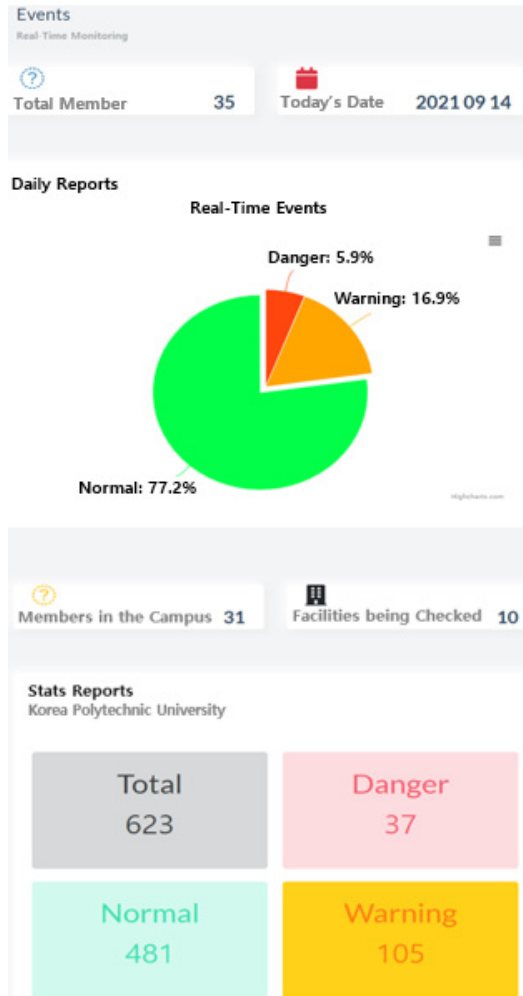


그림 4. 실시간 온도 현황판
Fig. 4. Real-Time Temperature Status Dashboard

2. 시스템 구조

구성원 출입 시 얼굴 인식 장치로부터 구성원을 인식하고 데이터베이스에 저장한 다음 실시간으로 웹에 표시하는 전체적인 시스템 구조는 그림 5와 같다.

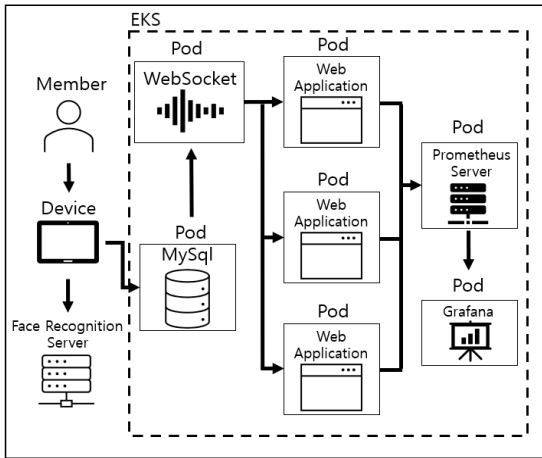


그림 5. 전체 시스템 구조
 Fig. 5. Whole System Architecture

가. 전체 시스템 구조

(1) Member

미리 등록된 구성원은 장치로부터 얼굴 인식과 온도 측정이 가능.

(2) Device

구성원의 얼굴을 인식하고 온도를 측정한다. 다음 데이터를 얼굴 인식 서버로 전송, 구성원이 아닐 경우 QR 코드를 발급해 등록시킴.

(3) Face Recognition Server

구성원 얼굴의 특징점을 통해 특정 표현을 얻은 후 저장

(4) MySql

데이터베이스로서 구성원 정보, 구성원 출입 현황, 관리자 정보 등 중요 데이터들이 저장됨.

(5) WebSocket

데이터베이스로부터 주기적으로 출입 현황 데이터를 가져와 배포된 웹으로 전송.

(6) Web Application

관리자가 모니터링할 수 있도록 websocket으로부터 전송받은 데이터를 실시간으로 웹에 시각화하며 확진자 발생 시 조치를 취할 수 있게 함^[8].

(7) Prometheus

웹 어플리케이션으로부터 각종 메트릭을 수집.

(8) Grafana

프로메테우스로부터 수집한 시스템 메트릭을 각종 그래프로 표시

Step 4: Stabilizing

웹 서버가 동작하는 중에 에러가 발생해 서버가 중단되어 복구할 때까지 모니터링이 불가능하거나 웹으로 한번에 많은 요청(request)이 발생함으로 인해 웹서버에 부하가 커질 수 있다. 이는 쿠버네티스를 사용함으로써 무중단 서비스 운영과 부하분산을 가능하게 할 수 있다. 또한, 서버에서 사용하는 리소스 모니터링을 통해서 문제가 될 가능성이 있는 정보를 한눈에 파악하고 이상 신호가 감지될 경우 slack 메시지를 보내게 만들 수 있다.

3. EKS 구조

본 시스템은 쿠버네티스 클러스터를 생성하기 위해 퍼블릭 클라우드 업체에서 제공하는 관리형 쿠버네티스인 EKS(Amazon Elastic Kubernetes Service)를 사용한다. EKS는 모든 노드를 AWS가 관리한다. 그림 6과 같이 가용 영역에서 쿠버네티스 컨트롤 플레인 인스턴스를 실행해서 고가용성을 보장한다.

그림 6은 현재 사용하는 EKS 노드 안의 파드의 구성을 나타낸다. 클러스터는 3개의 노드(Worker node)로 구성되어 있으며 각 노드에는 파드들이 scheduler에 의해 할당된다. 웹 애플리케이션은 deployment 오브젝트로 생성한 후 3개의 파드를 생성하여 노드마다 하나씩 할당된 상태다. 이는 하나의 노드가 갑자기 의도치 않게 종료되는 상황에서 클러스터가 자동으로 다른 가용 노드로 연결함으로써 지속적인 서비스가 가능하게 한다. 또한, 처리해야 할 요청들을 각 노드에 있는 파드로 부하분산 해준다.

prometheus 관련 파드는 각 노드로부터 메트릭을 수집하도록 도와주며 이는 프로메테우스 서버가 주기적으로 정해진 HTTP 엔드 포인트에 접근하여 수집한다. pushgateway는 프로메테우스가 주기적으로 데이터를 가져오기 위해 접근하는 곳이다. kube-state-metric은 단순한 서비스로 쿠버네티스 API 서버를 감시하고 오브젝트의 상태 메트릭을 생성한다.

Grafana와 관련된 파드는 수집된 메트릭을 시각화해주는 역할을 한다. 노드마다 있는 kube-proxy 파드는 네트워크 규칙을 유지하고 다른 파드의 네트워크 통신을 가능하게 한다. WebSocket 파드는 구성원들의 출입 현황 데이터를 각 노드의 web app 파드로 일정한 시간 간격으로 브로드캐스팅(broadcasting) 한다. sms-controller는 구성원 본인의 7일간 동선을 문자 메시지로 알림 받을 수 있는 기능을 제공한다.

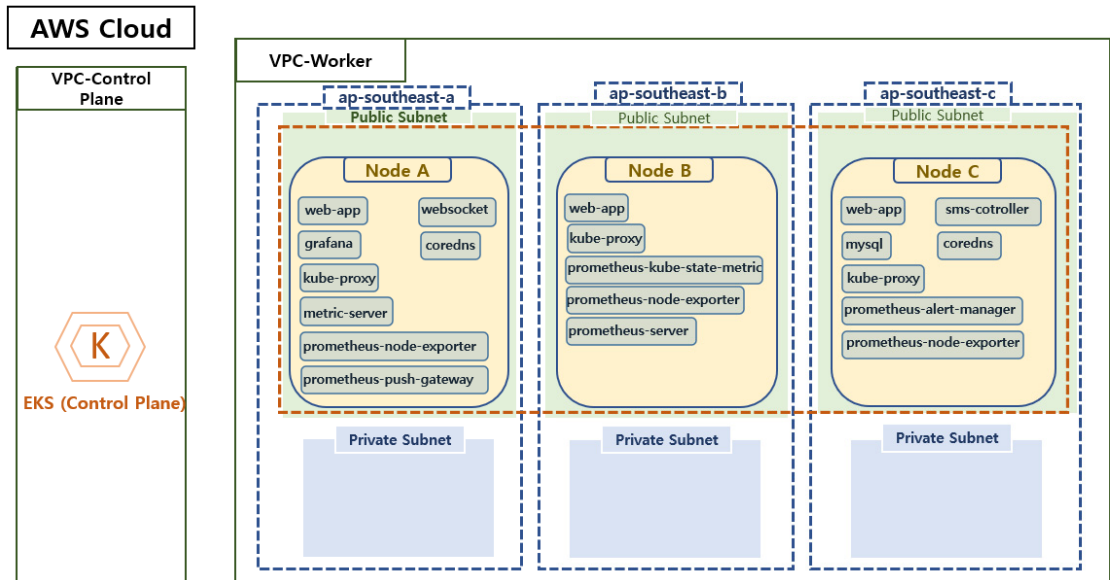


그림 6. 현재 Amazon EKS에 올려진 파드
Fig. 6. Pods on Amazon EKS

IV. 실험 및 결과

본 장에서는 얼굴 인식 기술의 정확도를 살펴보고 인식된 구성원이 웹에 실시간으로 잘 표시되는지 확인하며 건물에 많은 사람이 갑자기 몰릴 때 발생하는 트래픽을 EKS가 부하분산 하는 것을 실험한다.

1. 얼굴 인식 정확도 측정

웹에서 제대로 모니터링하고 정확한 결과를 나타내기 위해서는 얼굴 인식 기술의 높은 정확도와 웹에서 제대로 데이터를 표시하는 것이 중요하다. 얼굴 인식의 결과가 바르지 못하면 부정확한 결과를 도출할 수 있고 동선 추적이 어려울 수 있기 때문이다.

그림 7은 구성원의 사진을 등록하고 각 사진을 얼굴 인식 장치에서 얼굴 인식의 정확도를 백분율 그래프로 나타낸다. x 축은 기기에 등록된 구성원의 수이며 y 축은 각각 백번 인식 시도를 한 결과이다. 여기서는 다른 구성원으로 인식하거나 아예 인식하지 못한 경우엔 인식 실패라고 정의한다.

얼굴 인식 장치는 등록된 구성원의 데이터가 적을 때는 비교적 높은 정확도를 가지는 것을 볼 수 있다. 하지만 이는 사용하는 카메라의 품질과 인식하는 환경에 따라 달라질 수 있다. 얼굴을 인식하는 환경의 밝기, 카메

라와 구성원 사이의 거리 등 여러 변수에 의해 인식률은 달라질 수 있다. 본 실험은 인식 카메라가 구성원을 인식하는 환경과 카메라와 구성원 사이의 거리를 최대한 같게 하여 정확도를 측정했다.

얼굴 인식 장치의 인식 결과를 확인하고 해당 결과가 웹에 잘 표시되는지 그림 8을 통해 확인할 수 있다. 장치는 구성원을 인식할 때마다 웹으로 구성원에 대한 데이터를 전송한다.

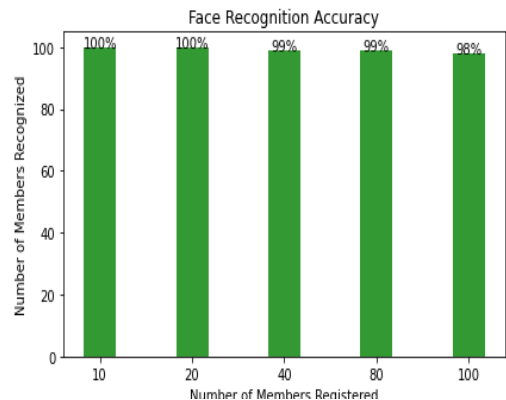


그림 7. 얼굴 인식 장치의 정확도
Fig. 7. Accuracy of Face Recognition Device

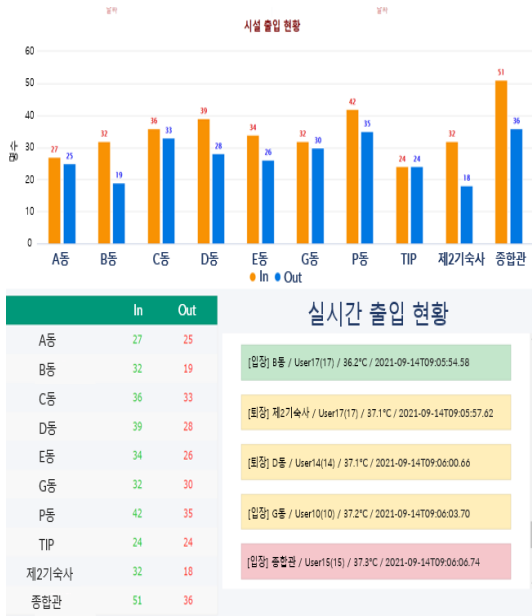


그림 8. 실시간 대시보드
 Fig. 8. Real-Time Dashboard

2. 부하분산

부하분산은 쿠버네티스가 기본적으로 제공하는 기능 중의 하나이다. 이는 웹에서 많은 요청(request)으로 인해 트래픽이 커지는 위험을 감소시킨다. 한번에 많은 사람이 건물에 몰리게 되면 웹 서버는 많은 요청을 처리해야 할 수 있다. EKS는 이를 대비해 트래픽을 각 노드로 분산시켜주는 작업을 해준다. 실험을 위해 서비스하고 있는 웹 애플리케이션을 각 노드에 replica로 만들어 지속적인 요청(request)을 보내고 grafana 대시보드를 통해 부하분산이 제대로 되는지 확인해 본다.

그림 9는 세 개의 노드에 각각 할당된 웹 애플리케이션이 구동되고 있는 파드의 네트워크 트래픽을 보여준다. 초기 상태로 세 노드는 일정한 범위 안에서 안정적인 트래픽을 나타낸다.

웹 서버에 많은 요청을 보내는 상황을 가정하기 위해 PowerShell을 이용해 파드로 접근할 수 있는 External IP 주소를 사용하여 지속해서 요청(request)을 보내고 결과를 grafana 대시보드로 확인한다.

그림 10을 통해 특정 IP 주소로만 요청을 보냈을 때 자동으로 세 개의 파드로 트래픽이 분산되는 것을 볼 수 있다. 또한, 초기 상태의 네트워크 in 그리고 out이 40 Bytes 이 내였으나 오랫동안의 요청(request)을 보낸 결과 각 노드의 트래픽이 부하분산이 되는 것을 볼 수 있다.

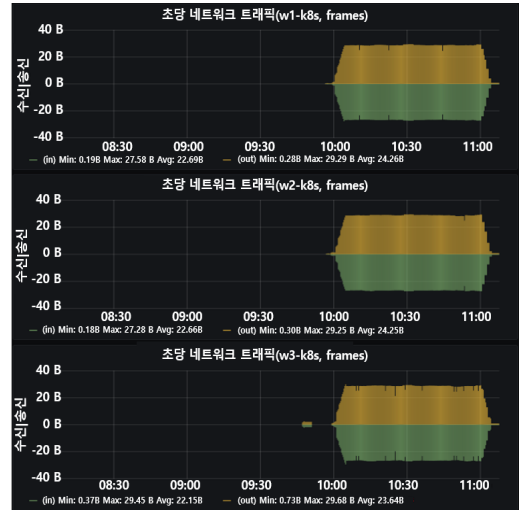


그림 9. 노드별 초기 네트워크 트래픽
 Fig. 9. Initial Network Traffic of each Node

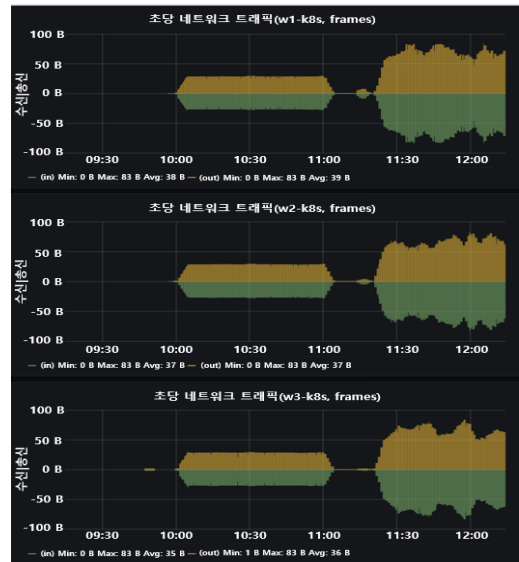


그림 10. 부하분산
 Fig. 10. Load-Balance

V. 결론

본 논문에서는 캠퍼스 내 코로나19 확진자 감시 및 빠른 대응 방안으로 얼굴 인식을 사용하여 데이터를 수집하고 웹으로 실시간으로 시각화하여 관리자가 유사시에 빠른 조치를 취할 수 있는 기능에 관해 이야기하였다. 또한, 시스템이 안정적으로 운영되기 위해서 쿠버네티스를 사용

하여 무중단 서버 운영 및 리소스 모니터링을 통해 좀 더 체계적인 운영을 가능하게 했다. 그러나 좀 더 정형화되고 안정적인 서버 운영을 위해서는 마이크로 서비스 형태의 웹 애플리케이션을 구성하는 것이 좋다. 이는 모든 기능을 하나의 파드 단위로 분리하여 구동하는 것이다. 그리고 코로나가 장기화되면서 마스크 착용 의무화가 강조되고 있는 상황에서 마스크를 착용한 상태에서 얼굴 인식을 할 수 없다는 것은 큰 단점이다. 이는 마스크를 착용한 상태에서도 사람을 인식할 수 있는 추가적인 알고리즘을 사용해 보완할 필요가 있다. 마지막으로 좀 더 확실한 확진자 모니터링을 위해서는 대량의 구성원 데이터가 있을 때도 얼굴 인식이 정확한 결과를 뽑아낼 수 있도록 라이브러리의 수정이 필요하다. 따라서 향후 연구로 대량의 얼굴 이미지 데이터가 있을 때도 얼굴 인식의 정확도를 높이기 위한 작업을 수행하고 마스크를 착용한 상태에서도 인식이 가능한 알고리즘을 추가할 뿐만 아니라 웹 애플리케이션의 마이크로 서비스화를 통해 좀 더 유연하고 독립적인 웹 애플리케이션을 개발하는 연구를 진행할 예정이다.

References

- [1] Cisneros, B., Ye, J., Park, C. H., & Kim, Y. "CoviReader: Using IOTA and QR Code Technology to Control Epidemic Diseases across the US," In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0610-0618. IEEE. 2021.
DOI: <https://10.1109/CCWC51732.2021.9376093>
- [2] Lubis, A., F., "Proximity-Based COVID-19 Contact Tracing System Devices for Locally Problems Solution," In 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 365-370. IEEE. 2020
DOI: <http://10.1109/ISRITI51436.2020.9315498>.
- [3] Tan, W., & Liu, J. "Application of Face Recognition in Tracing COVID-19 Fever Patients and Close Contacts," In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1112-1116. IEEE. 2020.
DOI: <http://10.1109/ICMLA51294.2020.00179>.
- [4] Yoon, S., C., Lee, J., U., Kim, I., Y., & No, S., U., "Korea's response to COVID-19 using ICT and its results." Korea International Cooperation Agency, Issue Report . pp. 1-64. 2020.
- [5] Ju, S. L., Kang, H., & Oh, S. H. "Correlation analysis between COVID-19 cases and emergency alerts service," The Journal of the Institute of Internet, Broadcasting and Communication, Vol. 21, No.5, 1-9. 2021.

- [6] Ha, Y., J., Yu, J., H., & Ohm, S., Y., "Bandal: Face Recognition Attendance Management System using OpenCV Library," In Proceedings of the Korea Information Processing Society Conference, pp. 255-257. Korea Information Processing Society. 2020.
- [7] Zhang, X., Li, L., Wang, Y., Chen, E., & Shou, L., "Zeus: Improving Resource Efficiency via Workload Colocation for Massive Kubernetes Clusters," in IEEE Access, Vol. 9, pp. 105192-105204, 2021.
DOI: <http://10.1109/ACCESS.2021.3100082>.
- [8] Zhang, L., & Shen, X., "Research and development of real-time monitoring system based on WebSocket technology," Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), 2013, pp. 1955-1958.
DOI: <http://10.1109/MEC.2013.6885373>.

저 자 소 개

유 영 균(준회원)



- 2016년 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 학사과정
- 주관심분야 : Web Application, Data Analysis, Cloud Computing

조 성 욱(준회원)



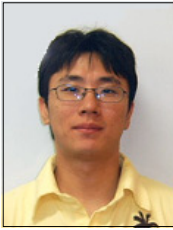
- 2017년 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 학사과정
- 주관심분야 : Artificial Intelligence Research, Artificial Intelligence Engineering

고 동 범(준회원)



- 2021년 ~ 현재 : 한국전자통신연구원 박사후 연구원
- 2021년 : 한국산업기술대학교 컴퓨터 공학과 박사
- 2018년 : 한국산업기술대학교 컴퓨터 공학과 석사
- 2016년 : 한국산업기술대학교 컴퓨터 공학과 학사
- 주관심분야 : CPS, Autonomic Computing, Artificial Intelligence

박 정 민(정회원)



- 2014년 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 부교수
 - 2012년 ~ 2014년: 한국전자통신연구원 선임연구원
 - 2008년 ~ 2011년 : 동양미래대 조교수
 - 2009년 : 성균관대학교 컴퓨터공학과 박사
- 2005년 : 성균관대학교 컴퓨터공학과 석사
 - 2003년 : 한국산업기술대학교 컴퓨터공학과 학사
- 주관심분야 : CPS, Autonomic Computing, Software Engineering

※ “이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기본연구지원사업임 (NRF-2021R1F1A1063634)”