

사물인터넷과 Amazon Web Services를 활용한 공간 혼잡도 측정 시스템

김태국*

국립부경대학교 컴퓨터·인공지능공학부 교수

Spatial Crowdedness Measurement System using IoT and Amazon Web Services

Tae-Kook Kim*

Professor, School of Computer and Artificial Intelligence Engineering, Pukyong National University

요약 본 논문에서는 사물인터넷과 Amazon Web Services(아마존 웹 서비스, AWS)를 활용한 공간 혼잡도 측정 시스템에 관해 연구하였다. 기존의 공간 혼잡도 측정 시스템은 고가의 서버를 필요로 하고, 공간 인식 성능 개선을 위한 고비용의 알고리즘 개발 및 업데이트가 필요한 문제가 있다. 제안한 시스템은 소형/저가형 싱글 보드 컴퓨터인 라즈베리 파이(Raspberry Pi)에서 OpenCV를 통해 웹카메라(or CCTV) 화면을 캡처하여 AWS로 정보를 전송하고 처리하여 적은 비용으로 구현이 가능하다. AWS에서는 수신된 공간 화면 이미지 정보를 Amazon S3(Amazon Simple Storage Services)에 저장하고, Amazon Lambda에서 Amazon Rekognition으로 전송하여 이미지를 통해 혼잡도를 분석한다. Amazon Rekognition 서비스는 이미지 처리 건당 0.001달러로 적은 비용으로 인공지능(AI) 기술을 사용할 수 있고, 사람 객체 인식을 통해 혼잡도를 분석할 수 있다. 분석된 혼잡도는 DB(Database)에 저장하고, 결과를 화면에 출력한다. 제안한 공간 혼잡도 측정 시스템은 저비용으로 공공장소 등에서의 공간 혼잡도 확인 등에 활용될 수 있을 것으로 기대한다.

주제어 : 사물인터넷, 빅데이터, 라즈베리 파이, 공간 혼잡도, 인공지능

Abstract In this paper, we conducted research on a spatial crowdedness measurement system using the Internet of Things(IoT) and Amazon Web Services(AWS). Current spatial congestion measurement systems require expensive servers and entail significant investment in algorithm development and updates to enhance spatial recognition performance. The proposed system can be implemented at low cost by capturing the screen of a web camera(or CCTV) through OpenCV on a small/low-cost single board computer, Raspberry Pi, and transmitting and processing the information to AWS. Within AWS, the received spatial image information is stored in Amazon S3(Amazon Simple Storage Service). Subsequently, Amazon Lambda transfers the images to Amazon Rekognition for congestion analysis based on the images. The Amazon Rekognition service offers the capability to utilize artificial intelligence(AI) technology for image processing at a low cost of 0.001 dollars per image. Through human object recognition, it enables congestion analysis. The analyzed congestion level is stored in DB(Database), and the result is displayed on the screen. The proposed system is expected to be used for checking crowdedness at low cost in public places.

Key Words : Internet of Things(IoT), big data, Raspberry Pi, spatial crowdedness, artificial intelligence

*이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2023-00242528).

*교신저자 : 김태국(king@pknu.ac.kr)

접수일 2023년 6월 1일 수정일 2023년 8월 5일 심사완료일 2023년 8월 10일

1. 서론

이태원 참사, 코로나 사태 등으로 공간의 혼잡도에 대한 관심이 늘어나고 있다. 혼잡도는 이미지 영상을 분석하거나, 센서를 이용한 방법이 있다. 전철, 버스 등의 교통 시스템에서의 혼잡도 측정 시스템에 관한 기존의 연구는 하중 센서 데이터를 기반으로 혼잡도를 측정한다. 하지만 이러한 방식은 하중 측정 장치가 탑재되어야 사용이 가능한 문제점이 있다[1-5]. 또한 이미지 영상을 통한 기존의 공간 혼잡도 측정 시스템은 고성능의 서버 구축, object detection(사람 객체 인지) 알고리즘 개발 등의 비용 증가 문제가 있다[5]. 이러한 문제를 해결하기 위해 사물인터넷과 Amazon Web Services(AWS)를 활용한 공간 혼잡도 측정 시스템을 제안하였다. 이 시스템은 사물인터넷 기술 등을 활용하여 웹카메라(or CCTV)의 영상을 AWS로 전달하고, Amazon Rekognition, Amazon S3, Amazon Lambda 등의 서비스를 활용하여 저비용으로 공간 혼잡도를 측정할 수 있다. AWS는 사용한 서비스의 금액만 지급하면 되므로 고가의 서버 구축 비용, 알고리즘 개발 비용 등을 줄일 수 있다[6-10].

제안한 AWS와 웹카메라를 활용한 공간 혼잡도 측정 시스템은 공공장소, 행사장, 다중이용시설 등에서 안전 확보를 위한 공간 혼잡도 측정에 활용될 수 있을 것으로 기대한다.

2. 배경 기술

2.1 Amazon Web Services

Amazon Web Services는 다양한 기능의 서비스를 제공하는 클라우드 플랫폼이다[6]. AWS는 스토리지, 데이터베이스와 같은 인프라 기술부터 기계 학습 및 인공지능 등의 기술을 제공한다. AWS의 서비스는 Representational State Transfer(REST) 및 Simple Object Access Protocol(SOAP)를 통해 접근하여 이용 가능하다. REST와 SOAP는 API를 구축하는 방법을 정의하는 온라인 데이터 전송 방식이다. 즉, AWS 서비스를 API로 제어할 수 있다.

2.2 Amazon Rekognition

Amazon Rekognition은 아마존에서 제공하는 클라우드 기반 SaaS(Software as a Service) 컴퓨터 비전

(computer vision) 플랫폼이다. 인공지능 기계 학습을 통해 이미지 인식 및 비디오 분석을 자동화하고 비용을 절감할 수 있다. Amazon Rekognition에서는 사용한 이미지 처리 건수 만큼만 비용을 지불하며, 이미지 1건 처리당 0.001달러(1.3원)의 비용이 발생한다[7].

2.3 Amazon S3

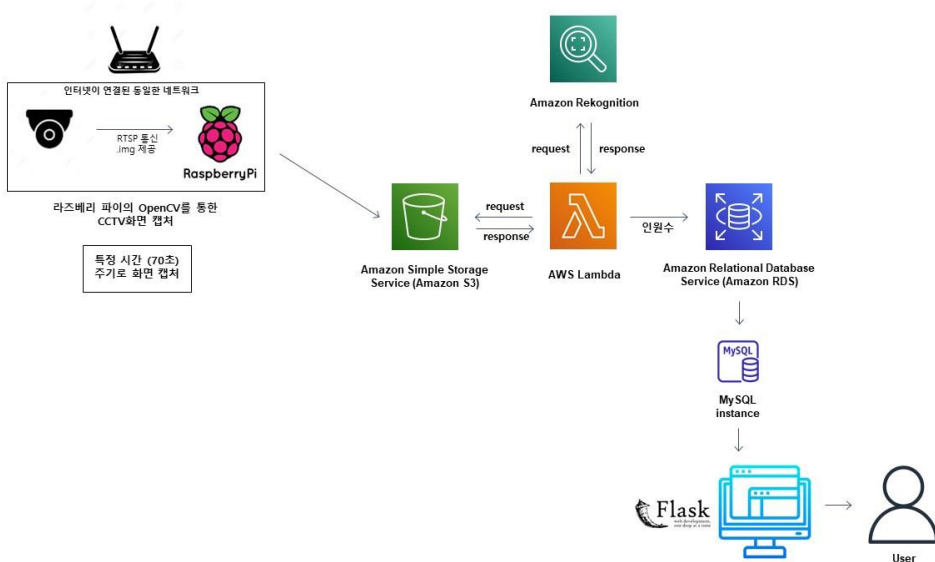
Amazon Simple Storage Service(Amazon S3)는 확장성과 보안 및 성능을 제공하는 객체 스토리지 서비스이다. 효율적인 스토리지와 쉬운 사용과 관리를 통해 비용을 최적화하고 데이터를 정리할 수 있다. 모든 클라우드 데이터에서 빅데이터 분석, 인공지능, 기계 학습을 실행할 수 있다. Amazon S3는 파일 데이터를 뜻하는 객체와 객체 저장 영역인 버킷이 있다. 각 객체에는 키가 존재하고, 버킷 내 객체에 대한 고유한 식별자이다. 버킷과 객체는 private이며 권한을 부여한 경우 접근이 가능하다. 객체를 Amazon S3로 저장한 후에는 다른 AWS 서비스와 사용할 수 있다[8].

2.4 Amazon Lambda

AWS Lambda는 서버를 관리하지 않고도 다양한 애플리케이션이나 백엔드 서비스에 대한 코드를 실행할 수 있는 서버리스 컴퓨팅 서비스이다[9]. Lambda가 지원하는 언어로 코드를 입력만 하면 요청에 따라 코드를 실행한다. Lambda API를 사용하여 Lambda 함수를 호출하거나, 다른 AWS 서비스의 이벤트에 응답하여 함수를 실행한다. 예를 들어, 사용자 지정 이벤트 트리거를 지원하기 때문에 Amazon S3를 사용하여 업로드 후에 실시간으로 AWS Lambda 데이터 처리를 할 수 있다. 이때, 트리거란 Lambda 함수를 실행하는 이벤트를 말한다.

2.5 Amazon RDS

Amazon Relational Database Service(Amazon RDS)는 클라우드에서 관계형 데이터베이스를 운영할 수 있는 웹 서비스이다[8]. MariaDB, Microsoft SQL Server, MySQL, Oracle, PostgreSQL과 같은 데이터베이스를 사용할 수 있다. AWS Command Line Interface(AWS CLI), Amazon RDS API 또는 AWS Management Console을 통해 데이터베이스 인스턴스를 생성할 수 있다[10].



[Fig. 1] Proposed system configuration

3. 공간 혼잡도 측정 시스템

3.1 시스템 구성

본 시스템은 라즈베리 파이(Raspberry Pi), 웹카메라, 네트워크, 이미지 등을 분석하기 위한 AWS 시스템으로 구성된다.

[Fig. 1]은 제한한 시스템 구성을 나타낸다. 저가/소형의 싱글 보드 컴퓨터인 라즈베리 파이[11,12]에 웹카메라(or CCTV)를 연결하고 OpenCV를 통해 화면을 캡처한다. OpenCV는 실시간으로 컴퓨터 비전을 목적으로 라이브러리를 제공한다[13-15]. 또한 OpenCV는 얼굴 인식 기능을 제공한다. 웹카메라가 Raspberry Pi와 통신하며, RTSP Protocol을 통해 영상을 송수신하며, Raspberry Pi의 OpenCV를 통해 실시간 이미지를 캡처하고 AWS S3에 저장하게 된다.

이미지를 분석하기 위한 AWS S3, AWS Lambda, AWS Rekognition, Amazon RDS 서비스를 사용하여 구성이다. AWS S3에 이미지가 저장되는 것을 트리거로 지정하여 AWS Lambda가 호출된다. 호출된 Lambda를 통해 S3에 저장된 이미지를 요청하고, 요청된 이미지는 Rekognition의 요청을 통해, 이미지를 객체 검출하며, 객체 중 사람을 라벨링 하여 인원수를 반환해 준다. 해당 반환 값은 Lambda로 반환되며 사진에 정해놓은 혼잡도 기준에 따라 혼잡도를 측정하여 시간, 인원수, 혼

잡도를 RDS로 전송한다.

마지막으로 RDS의 경우 분산 관계형 데이터베이스이므로, 실질적으로 사용자에게 데이터를 제공하기 위해 관계형 데이터베이스인 MySQL에 저장하고 사용자에게 정보를 제공한다.

3.2 이미지 추출

이미지를 추출하기 위해서 웹카메라와 Raspberry Pi의 통신이 가능해야 하며, S3에 저장하기 위해선 인터넷이 연결되어 있어야 한다. 라즈베리 파이는 무선과 유선 인터넷이 가능하고, 웹카메라와 라즈베리 파이는 RTSP Protocol을 통해 통신한다. 통신에 성공하면 라즈베리 파이의 OpenCV를 이용하여 이미지를 캡처하고, AWS와 연동되어 있는 S3의 정보를 입력하여 설정한 파일명으로 S3에 업로드 하며, 해당 코드는 70초 단위로 무한 반복된다. [Fig. 2]는 공공장소인 강의실 웹카메라를 통해 추출된 이미지를 나타낸다. 추출된 이미지는 라즈베리 파이를 통해 AWS S3에 전송된다.

3.3 얼굴 인식 및 인원수 검출

Lambda를 통해 S3에 .jpg 파일이 저장되면 트리거가 발생하여 <Table 1>의 Lambda 함수가 호출된다.



[Fig. 2] Web camera image

<Table 1> Lamda function source code

```

import json
import boto3
import time
import io
from datetime import datetime
import threading
import sys
import logging
import pymysql
import dbinfo

s3= boto3.client('s3')

def detect_labels(bucket,photo):
    rekognition = boto3.client("rekognition")
    response =
    rekognition.detect_labels(Image={'S3Object':{'Bucket':bucket,'Name':p
    hoto}},MaxLabels=10)

    return response['Labels']

def lambda_handler(event, context):

    now=datetime.now()
    realti=now.strftime("%Y%m%d%H%M")
    grade=0
    numberpeople=""

    try:
        for record in event['Records']:
            filename = record['s3']['object']['key']
            bucket = record['s3']['bucket']['name']
            source= detect_labels(bucket,filename)

            for i in source:
                if 'Person' in str(i):
                    numberpeople+=str(len(i['Instances']))
                    print(len(i['Instances']))
                    break
            else:
                numberpeople=str(0)

            realtxt=realti+".txt"
            # Spatial Crowdedness Grade
            if 0 <= int(numberpeople) <= 20:
                grade = 1 #less crowded
            elif 20 < int(numberpeople) <= 40:
                grade = 2 #moderately crowded
            elif 40 < int(numberpeople) <= 60:
                grade = 3 #crowded
            elif 60 < int(numberpeople):
                grade = 4 #very crowded

```

```

        with open('/tmp/'+realtxt,w) as f:
            f.write(f'{numberpeople} / {grade} / {realti}')

        s3.upload_file('/tmp/'+realtxt,'ras1',realtxt) #s3 .txt upload
        print("success")

    except Exception as e:
        print(e)

    result = updateDB(numberpeople, realti, grade) #DB upload

    if result:
        return "complete"
    else:
        return "fail"

def updateDB(numberpeople, realti, grade):
    connection = pymysql.connect(host=dbinfo.db_host,
    port=3306, user=dbinfo.db_username, passwd=dbinfo.db_password,
    db=dbinfo.db_name)
    #DB linkage

    try:
        with connection.cursor() as cursor:
            SQL = "insert into bus (num, busID,grade) values (%s,
            %s, %s)"
            cursor.execute(SQL, (numberpeople, realti, grade))
            connection.commit()

    except:
        print("data insert error")
        return False #Return 0 in result if data insertion fails

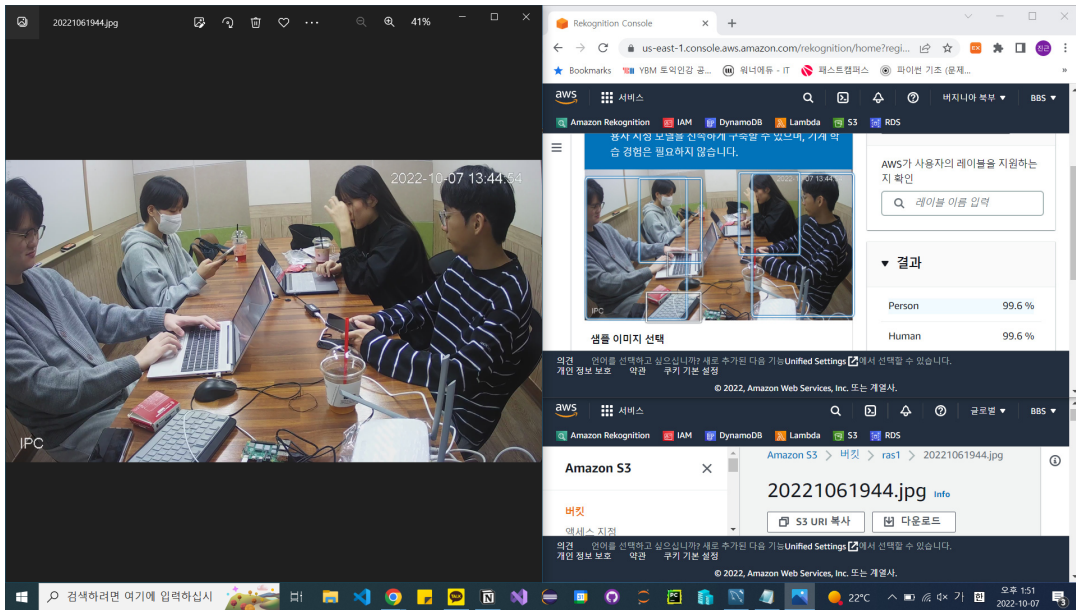
    else:
        print("data insert success")
        return True #Return 1 in result if data insertion success

    finally:
        connection.close()

```

[Fig. 3]은 공공장소인 강의실의 이미지를 통한 공간 혼잡도 분석 결과를 나타낸다. Lambda 함수에서 AWS Rekognition을 활용하여 클라우드 기반으로 컴퓨터, 사람 등의 객체를 인식하고, 사람이 4명이라고 하는 결과 값을 얻을 수 있다. AWS Rekognition을 통해 최신 객체인식 기계 학습 기술을 그대로 적용할 수 있어서 사용자는 추가적인 서버 구축 비용, 성능 개선 비용이 필요 없게 된다. 작성한 Lamda 함수 소스코드는 공간의 크기는 일정하다고 가정하여, 사람이 20명 이하일 때는 여유, 20~40명인 경우는 보통, 40~60명인 경우는 혼잡, 60명 이상인 경우에는 매우 혼잡으로 표시하였다.

본 연구를 통해 AWS를 활용하여 실제 시스템을 구현하여 공간 혼잡도 측정이 가능함을 확인하였다. 향후 연구에서는 공간 크기를 자동으로 인지하여 공간 크기와 인원수에 따라 공간 혼잡도를 인공지능으로 측정하는 시스템을 연구할 예정이다.



[Fig. 3] Spatial crowdedness image analysis result

4. 결론

논문에서는 사물인터넷과 AWS를 활용한 공간 혼잡도 측정에 관해 연구하였다. 고가의 서버 구매 없이 저비용으로 웹카메라로 공간 혼잡도를 측정하는 시스템을 제안하였다. 제안한 시스템은 추가적인 장치 없이 기존 웹카메라 영상을 활용하고, AWS의 인공지능 기능을 이용하여 공간 혼잡도를 측정할 수 있는 특징을 가지고 있다. 웹카메라를 활용한 공간 혼잡도 측정시스템은 라즈베리 파이, 웹카메라, AWS, OpenCV 등으로 구성된다. 웹카메라에서 촬영되는 영상 이미지를 RTSP 프로토콜을 통해 라즈베리 파이에서 OpenCV로 이미지를 처리하여 AWS 시스템으로 전송한다. Amazon Rekognition 등의 이미지 분석을 통해 혼잡도 분석값을 DB에 저장하여 사용자에게 정보를 제공한다. 본 연구를 통해 웹카메라와 AWS를 활용하여 저비용으로 공간 혼잡도 측정의 가능성을 확인하였다.

REFERENCES

[1] J.G.Kim, Y.J.Kim, K.H.Park, S.W.Jeon, T.K.Kim "A Study on the Congestion Measurement System of Public Transportation Using CCTV," 2022 Korea Multimedia Society Spring conference, Vol.25, No.1,

pp.25-27, 2022.

[2] S.Lee, H.Yoon, T.Kim, K.Hong, "Research on the Analysis of Changes in Train Congestion of Urban Railway according to COVID-19 Situation," Journal of Korean Society for Urban Railway, Vol.9, No.4, pp.1131-1141, 2021.

[3] Y.G.Kim, T.D.Kim, Y.S.Kim, S.H.Lee, K.I.Jung, N.J.Kim, J.A.Oh, S.M.Chang, "Urban rail passenger congestion measurement practice tests for data validation," 2016 Korean Society for Urban Railway Spring conference, pp.27-32, 2016.

[4] A.Bakht, H.Lee, "Deep Learning Framework for Spatial Crowdedness Estimation and Comparison Analysis with Machine Learning," Journal of Korean Institute of Intelligent Systems, Vol.32, No.1, pp.76-85, 2022.

[5] C.J.Jeong, K.Y.Park, G.Park, "Real Time Crowd Estimation System Using Embedded Hardware," Journal of Satellite, Information and Communications (kosst), Vol.8, No.4, pp.26-29, 2013.

[6] Amazon, Amazon Web Services[Internet], <https://aws.amazon.com>.

[7] Amazon, Amazon Web Services Rekognition[Internet], https://aws.amazon.com/ko/rekognition/?nc2=h_q_l_prod_ml_rek.

[8] Amazon, Amazon Web Services S3[Internet], https://aws.amazon.com/ko/s3/?nc2=h_q_l_prod_st_s3.

[9] Amazon, Amazon Web Services Lambda[Internet], https://aws.amazon.com/ko/lambda/?nc2=h_q_l_prod_fs_lbd.

- [10] Amazon, Amazon Web Services RDS[Internet], https://aws.amazon.com/ko/rds/?nc2=h_ql_prod_fs_rds.
- [11] J.H.Moon, B.Peng, J.H.Kwon, T.K.Kim, "Implementation of Smart Umbrella Stand Based on IoT," Journal of Internet of Things and Convergence, Vol.9, No.1, pp.57-64, 2023.
- [12] Raspberry Pi Foundation, Raspberry Pi[Internet], <https://www.raspberrypi.com>.
- [13] Intel, OpenCV[Internet], <https://opencv.org>.
- [14] D.H.Kim, S.Y.Kim, "A Study on Risk Situation Awareness Using OpenCV," Journal of The KIECS, Vol.16, No.2, pp.211-218, 2021.
- [15] D.J.Kim, W.S.Choi, S.P.Ju, S.M.Yoo, J.Y.Choi, H.K.Park, "Smart Streetlight based on Accident Recognition using Raspberry Pi Camera OpenCV," The Journal of The Korea Institute of Electronic Communication Sciences (Electronic Communication), Vol.17, No.6, pp.1229-1236, 2022.

김 태 국(Tae-Kook Kim)

[종신회원]



- 2004년 8월 : 고려대학교
전기전자전파공학부(공학사)
- 2006년 8월 : 고려대학교
메카트로닉스학과(공학석사)
- 2014년 8월 : 고려대학교
모바일솔루션학과(공학박사)

- 2016년 3월 ~ 2022년 2월 : 동명대학교 AI학부 교수
- 2022년 3월 ~ 현재 : 국립부경대학교 컴퓨터인공지능공학부 교수

<관심분야>

사물인터넷(IoT), 콘텐츠 전송 네트워크(CDN), 이동성, 인공지능(AI), 빅데이터, 모바일 서비스