KIPS Trans. Softw. and Data Eng.
Vol.12, No.4 pp.189~198
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2023.12.4.189

Threat Situation Determination System Through AWS-Based Behavior and Object Recognition

Ye-Young Kim† · Su-Hyun Jeong†† · So-Hyun Park††† · Young-Ho Park††††

ABSTRACT

As crimes frequently occur on the street, the spread of CCTV is increasing. However, due to the shortcomings of passively operated CCTV, the need for intelligent CCTV is attracting attention. Due to the heavy system of such intelligent CCTV, high-performance devices are required, which has a problem in that it is expensive to replace the general CCTV. To solve this problem, an intelligent CCTV system that recognizes low-quality images and operates even on devices with low performance is required. Therefore, this paper proposes a Saying CCTV system that can detect threats in real time by using the AWS cloud platform to lighten the system and convert images into text. Based on the data extracted using YOLO v4 and OpenPose, it is implemented to determine the risk object, threat behavior, and threat situation, and calculate the risk using machine learning. Through this, the system can be operated anytime and anywhere as long as the network is connected, and the system can be used even with devices with minimal performance for video shooting and image upload. Furthermore, it is possible to quickly prevent crime by automating meaningful statistics on crime by analyzing the video and using the data stored as text.

Keywords : Danger Situation Determination, AWS-based, Object Recognition, Behavior Recognition

AWS 기반 행위와 객체 인식을 통한 위협 상황 판단 시스템

김 예 영† · 정 수 현†† · 박 소 현††† · 박 영 호††††

요 약

길거리에서 묻지마 범죄가 자주 발생함에 따라 CCTV의 보급이 증가하고 있다. 그러나 수동적으로 작동되는 CCTV의 단점 때문에 지능형 CCTV의 필요성이 주목 받고 있다. 이러한 지능형 CCTV의 무거운 시스템 때문에, 높은 성능의 기기들이 필요해 일반 CCTV를 대체하는데 비용적 측면에서 부담이 발생한다. 이 문제를 해결하기 위해 낮은 품질의 영상도 인식하며 높지 않은 성능의 기기에서도 시스템이 구동되는 지능형 CCTV 시스템이 필요하다. 따라서 본 논문은 AWS 기반 플랫폼을 활용하여 시스템을 경량화하고 영상을 텍스트화하여 실시간으로 위협을 감지할 수 있는 Saying CCTV 시스템을 제안한다. 이는 YOLO v4와 OpenPose를 사용해 추출한 데이터를 바탕으로 위험 객체와 위협 행동 그리고 위협 상황을 판단하며, 위험도를 머신러닝으로 계산하도록 구현하였다. 이를 통해, 언제 어디서나 네트워크만 연결되면 시스템을 동작시킬 수 있으며, 영상 촬영과 이미지 업로드가 최소한의 성능의 기기에서도 시스템 사용이 가능하다. 나아가 영상을 분석하여 텍스트로 저장되는 데이터들로 하여금 범죄의 유의미한 통계를 자동화하여 신속한 범죄 예방이 가능하다.

키워드 : 위협 상황 판단, AWS-based, 객체 인식, 행위 인식

1. 서 론

최근 길거리에서의 묻지마 범죄가 자주 발생하고 있다[1]. 묻지마 범죄의 동기는 충동적인 경우가 많기 때문에 범행 타깃, 시간, 위치 등을 예측하기 어렵다[2-4]. 범죄를 예측하여 미연에 방지하기 위한 여러 연구가 진행되고 있다.

첫째, 클라우드 연계형 지능형 에지 CCTV 서비스는 지능형 CCTV 서비스와 분산 에지 클라우드를 접목한 기술로 객체와 행위를 탐지한 요약 영상을 지능적으로 선정하여 저장하는 시스템이다. 하지만 기존 방법은 영상을 저장하는 것이 주된 기능이고 실시간 범죄 예측 및 대응 기술 연구가 부족한 실정이다. 둘째, 실시간 범죄 모니터링을 위한 CCTV 협업 추적시스템에서는 용의자가 CCTV에서 벗어나 이에 대한 정보를 전달하여 용의자를 추적하는 기술을 제안한다. 기존 연구에서는 용의자 추적이 가능하나 추적 이후 대응을 위한 관제사가 필요하다는 한계점이 존재한다. 셋째, 지능형 CCTV

※ 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2022R1F1A1074065)
† 비 회 원 : 숙명여자대학교 IT공학전공 학사과정
†† 비 회 원 : 숙명여자대학교 IT공학전공 학사
††† 정 회 원 : 동국대학교 WISE캠퍼스 컴퓨터공학과 조교수
†††† 종신회원 : 숙명여자대학교 인공지능공학부 교수
Manuscript Received : June 15, 2022
First Revision : November 24, 2022
Accepted : December 24, 2022
* Corresponding Author : Young-Ho Park(yhpark@sm.ac.kr)

서비스는 CCTV 영상을 즉각 판단하여 미리 설정된 조치를 하도록 학습된 CCTV 기술을 제안한다[5]. 하지만 기존 지능형 CCTV는 많은 대수의 CCTV 영상을 처리해야 하는데 이는 빅데이터 처리할 수 있는 고비용의 기반 시스템 구축을 필요로 한다[6]. 또한, 기존의 시스템은 위협 상황을 판단하는 알고리즘 연구가 주된 기능이고 End-to-End 제품 개발 연구가 부족한 실정이다.

본 논문의 공헌은 다음과 같다. 첫째, 본 논문에서는 아마존 웹 서비스(AWS, Amazon Web Services)의 클라우드 컴퓨팅 서비스를 기반으로 하여 위험 객체/행위 인식, 위험도 평가, 위험도에 따른 결과 처리 등 지능형 CCTV 구현에 필요한 전반적인 기능을 모두 포함하는 위협 상황 판단 시스템인 Saying CCTV를 제안한다. 제안하는 Saying CCTV는 AWS의 리소스를 활용하여 고비용을 필요로 하는 영상처리를 클라우드 상에서 진행한다. 또한, 실생활에서 사용 가능한 라즈베리파이 단말기에서 작동이 가능하도록 구현된 End-to-End 제품이다.

둘째, 본 논문에서는 영상에서 뽑아낸 이미지를 통해 유의미한 데이터를 추출하기 위한 이미지 텍스트화 방법을 제안한다. 이를 위해서 각 함수에서는 객체 인식, 행위 인식, 접근성 판별, 사람과 위험 객체의 연관성 판별을 진행한다.

셋째, 본 논문에서는 이미지 텍스트화 결과를 바탕으로 위험도를 평가하는 방법을 제안한다. 이를 위하여 공간 밝기, 위험 객체 유무, 접근성, 객체와 사람의 연관성의 5가지 요소를 척도로 위험도를 종합적으로 평가하는 의사 결정 트리 기반의 위험도 평가 알고리즘을 제안한다. 평가된 위험도에 따라 사용자한테 텍스트를 송출해주거나 영상 저장방식들을 결정하는 후처리 과정이 진행된다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 기술인 CCTV 통합관제센터와 지능형 영상 보안 기술을 소개한다. 3장에서는 Saying CCTV의 개요, 구조, 서버와 클라이언트의 구성을 설명한다. 4장에서는 위험도 측정을 위한 시스템의 주요 알고리즘을 설명하고, 5장에서는 시스템이 작동되는 구현 화면을 설명한다. 6장에서는 시스템 사용 실험 결과를 통해 시스템의 효용성에 대해 증명하고 마지막 7장에서 결론과 기대 효과를 설명한다.

2. 관련 연구

본 장에서는 클라우드 연계형 CCTV 서비스와 실시간 범죄 추적 CCTV 시스템 연구에 대해 설명한다.

2.1 클라우드 연계형 지능형 에지 CCTV 서비스

본 절에서는 클라우드 연계형 지능형 에지 CCTV 서비스를 소개한다. 클라우드 연계형 지능형 에지 CCTV 서비스는 분산 에지 클라우드를 지능형 CCTV 기술에 접목한 서비스이다. 방대한 CCTV 영상을 모두 저장하는 것이 아닌 객체와 행위를 탐지한 요약 영상을 지능적으로 선정하여 저장하는 시스템이다. 이는 상대적으로 컴퓨팅 성능이 낮은 CCTV 기기 대신 촬영된 영상을 에지 클러스터가 받아 영상을 요약하는 방식이다. 이 에지 컴퓨팅은 클라우드-네이티브 방식을 통해 전국에 분산된 클러스터들을 함께 관리할 수 있다[7].

CCTV의 컴퓨팅 성능에 대한 대안으로 클라우드 연계 방안이 제시되었다. 하지만 기존 방법은 영상을 저장하는 것이 주요 기능으로 실시간 범죄를 예측하고 대응하는 등 CCTV 기기의 독립적인 이용을 위한 기술 및 설계가 부족하다.

2.2 실시간 범죄 모니터링을 위한 CCTV 협업 추적시스템

본 절에서는 실시간 범죄 모니터링을 위한 CCTV 협업 추적시스템을 소개한다. CCTV 협업 추적시스템이란 용의자가 한 CCTV에서 벗어나 다른 CCTV로 이동했을 경우 관제사에게 이에 대한 정보를 전달하여 용의자를 추적하는 기술이다. [8]은 지자체 관제센터 서버의 시스템을 사용하며, 각 CCTV가 영상을 서버에 보내면 객체 감지, 특징 추출, 분류를 통해 용의자를 추적한다. 추적된 결과는 자동으로 관제사에게 전달이 된다. 기존에는 관제사가 직접 영상을 보고 위험 상황을 판단하였지만[8]에서는 이 과정이 자동화되면서 관제센터의 운영 효율성이 증가한다.

기존 연구에서는 용의자의 추적은 가능하였으나 추적 이후 대응을 위해서는 관제사가 필요하다[8]. 또한, 기존 연구는 CCTV 본 기기에서 범죄 예측 시스템 컴퓨팅을 지원하지 않고, 관제센터의 서버에 의존하며 CCTV의 목적을 달성하기 위해 기기가 완전한 독립을 하지 못했다는 한계점이 있다.

2.3 관계형 추론을 통한 위험 상황 판단

본 절에서는 관계형 추론을 통한 위험 상황 판단연구에 대해 소개한다. 관계형 추론을 통한 위험 상황 판단은 이미지 속 위험 요소들의 위치값과 거리값 등 관계를 분석하여 정확한 위험 상황을 인식할 수 있는 방법이다. [9]는 객체 검출 단계와 관계형 추론 단계를 거치며, 물체 감지 단계에서는 객체 인식 모델을 사용하여 위험한 요인들을 인식하고 부분 이미지를 얻는다. 얻은 이미지를 통해 특징 집합을 얻게 되고, 관계형 추론 단계에서 입력값으로 쓰인다. 관계형 추론 단계에서는 앞서 얻은 이미지들의 정보와 특징 집합을 통해 관계성을 유추하여 물체의 위치와 물체 사이의 거리의 관계를 통해 위험도를 판단한다[9].

이전 연구에서 위험 물체만 인식했다면[10] 물체와 사람 간의 위치와 거리를 통해 관계성을 파악하였다[9]. 하지만 위험 상황 판단에 있어 다양한 요소들의 관계성이 존재하며, 기존 연구에서는 주로 물체와 사람의 관계성만 파악했기 때문에 다양한 요소들의 관계성을 고려하는 추가 연구가 필요한 실정이다.

3. Saying CCTV

본 장에서는 Saying CCTV의 개요와 시스템 구성, 시스템

동작을 설명하고 이후 서버와 클라이언트의 주요 기능을 설명한다.

3.1 시스템 개요

본 절에서는 Saying CCTV의 개요를 설명한다. Saying CCTV는 위협 행위와 위험 객체 인식을 바탕으로 위협 상황을 판단하며, 이를 소리로 출력하고 텍스트로 저장하는 시스템이다. 과정에서 나온 결과들은 모두 데이터베이스에 텍스트로 저장되고, 위험하다고 판단된 경우에는 관련 영상을 클라우드에 저장한다. 저장소에 저장된 데이터베이스와 영상은 사용자가 원할 때 확인할 수 있다.

본 시스템의 구성은 다음과 같다. 첫 번째로, 결과를 음성으로 출력하는 기능의 카메라가 연결된 라즈베리파이 클라이언트이다. 두 번째로, 실시간으로 찍혀진 영상을 인식하고 위험도를 판단하는 아마존 웹 서비스 서버로 이루어진다.

아마존 웹 서비스 기반의 Saying CCTV는 다음과 같은 장점이 있다. 본 시스템은 최소한으로 촬영된 영상을 인터넷을 통해 전송할 수 있는 환경만 된다면 이미지를 분석하고, 결과를 도출해낼 수 있다.

3.2 시스템 구조

본 절에서는 시스템 구조를 설명한다. 클라이언트인 라즈베리파이는 영상을 촬영하는 기능인 카메라와 음성을 출력해주는 블루투스 이어폰을 서로 연결해주는 기능을 한다. 또한 클라우드를 통하여 데이터를 연동한다.

Saying CCTV는 아마존 웹 서비스 플랫폼의 리소스를 활용한다. 첫 번째 리소스는 이미지와 영상 그리고 출력 음성을 저장하는 저장소인 S3이다. 두 번째 리소스는 서버리스 컴퓨팅 서비스인 람다(Lambda) 함수들과 이를 연결하는 계단 함수(StepFunction)이다. 이미지 인식, 위험도 계산 그리고 라즈베리파이와의 통신을 위해 람다를 사용한다. 세 번째 리소스는 텍스트를 목소리 음성으로 출력하는 폴리(Polly)와 최종 결과와 결과 값을 수시로 저장하는 두 개의 다이나모디비(DynamoDB)이다. 마지막으로, 아마존 웹 서비스의 사물인터넷 엠큐티티(MQTT, Message Queuing Telemetry Transport)를 활용하여 클라우드와 라즈베리파이 간의 통신을 진행한다.

Saying CCTV의 동작 과정은 Fig. 1과 같다. 라즈베리파이가 사용자의 앞을 촬영하고, 촬영한 이미지를 S3에 업로드한다. 이미지가 Simple Storage Service(S3)에 업로드 되면, 람다 함수가 자동으로 트리거된다. 총 3개의 람다로 구성되어있는 계단 함수를 통해 이미지를 분석하고 위험도를 판별하며, 위험도에 따라 상황을 알려주는 문장을 폴리를 사용하여 음성 파일을 추출해 S3에 저장한다. 이후 계산된 위험도를 라즈베리파이에 전달하고, 라즈베리파이는 받은 결괏값을 바탕으로 촬영주기를 빠르게 변경하거나, 영상을 촬영하거나, 음성을 출력하는 등의 설정을 변경하여 동작한다. 여기서 촬영된 영상은 다시 S3로 전달되어 저장된다. 또한, 시스템의 모든 과정에서 얻은 데이터는 다이나모디비1에 저장된다. 이 중 날짜, 위험 단계 수치, 이미지 속 사람 별 판단 데이터(객체 인식, 접근성, 행위), 영상 저장 위치 등 특정 결괏값들은 다이나모디비2에 저장된다.

3.3 라즈베리파이 클라이언트

본 절에서는 라즈베리파이 클라이언트를 소개한다. 본 논문에서 사용한 라즈베리파이는 리눅스 환경에서 파이썬 코드를 통해 AWS와 연동되며 카메라를 이용하여 영상을 촬영해

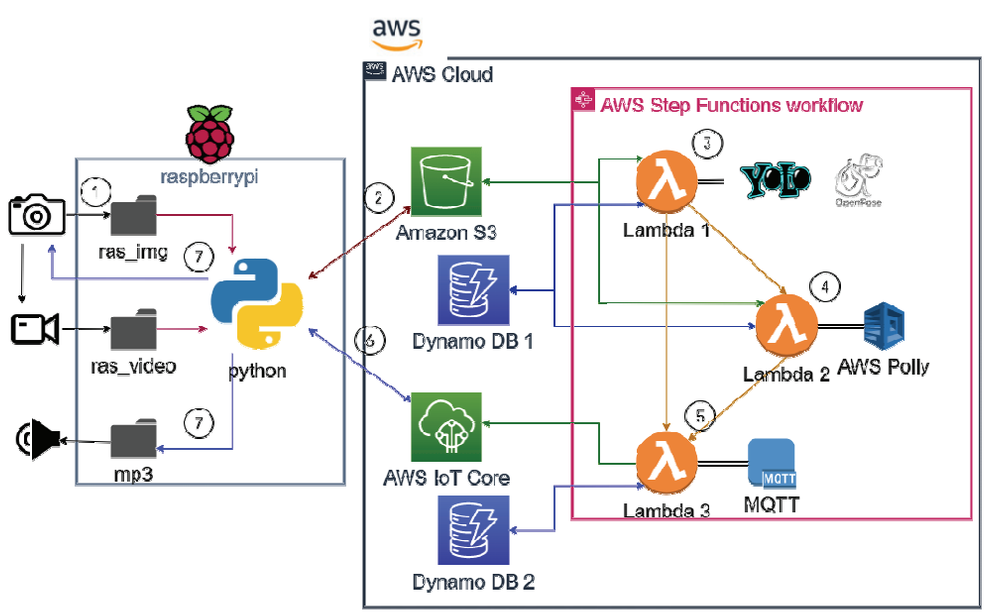

Fig. 1. Composition Diagram of Saying CCTV

전송한다. 실시간으로 처리되는 위험도 결괏값에 따라 음성 파일을 받는 등 데이터를 주고받으며 음성을 출력하거나 영상을 촬영한다.

위험도가 인식되지 않을 경우, 라즈베리파이는 주기적으로 촬영만 하는 대기 상태로 변경 및 유지한다. 서버는 아마존 웹 서비스 사물인터넷 엠큐티티를 통해 위험도 결과에 따른 간단한 텍스트 데이터를 전달받는데, 결괏값에 따라 상태가 대피 음성 출력 여부와 위험 상황 영상 저장 여부, 촬영 주기 감소 여부 등이 변동된다. 4.3.2절에서는 동작 변화에 대한 자세한 설명을 소개한다.

3.4 AWS 서버

본 절에서는 본 논문에서 사용하는 AWS 서버를 소개한다. 본 논문에서는 AWS의 서버리스 방식을 사용해 클라이언트가 보내는 영상을 분석하고 결괏값을 다시 클라이언트에 보내는 작업을 처리한다. 서버리스의 환경을 선택한 이유는 서버 관리 부담을 줄이고, 효율적인 코드 수정 및 활용을 위함이다.

본 논문에서는 서버리스 컴퓨팅 서비스인 세 개의 람다, S3, 두 개의 다이나모디비, 그 외 여러 리소스들을 사용한다. 본 논문에서는 위의 기능들이 한 애플리케이션으로 작동할 수 있도록 서버리스 애플리케이션 모델(AWS SAM, Serverless Application Model)을 사용한다. 또한, 본 논문에서 이미지를 처리하는 오픈포즈(OpenPose)와 YOLO v4는 텐서플로우 라이트(TensorflowLite) 프레임워크 상에서 동작한다. 마지막으로 이미지를 가공 및 처리하는 OpenCV 라이브러리를 사용한다.

본 논문의 AWS 기반 서버는 이미지가 촬영되어 저장되는 과정 동안 항상 프로비저닝(Provisioning)한 준비 상태를 유지한다. 또한, AWS 기반 서버는 람다의 특성상 지연 시간이 생길 때 동적 병렬 처리를 지원하기 때문에 많은 이미지를 동시에 처리 가능하다[11].

4. 위험도 측정 모델

본 장에서는 본 시스템에서 제안하는 위험도 측정 모델을 소개한다. 4.1절에서는 이미지 텍스트화 기능에 대해 설명한다. 소개하고 4.2절에서는 위험도 평가 기능에 대해 소개한다. 4.3절에서는 위험도에 따른 후처리 방안을 소개한다.

4.1 이미지 텍스트화

본 절에서는 영상에서 뽑아낸 이미지를 통해 유의미한 데이터를 추출해 내는 각 함수들을 설명한다.

1) 객체 인식

본 절에서는 객체 인식 방법을 소개한다. 본 논문에서는 위협 상황을 신속하게 처리하기 위해서 빠른 속도와 영상에서의 높은 정확도의 판단력을 가진 YOLO v4를 사용한다[12]. 위험 객체로는 칼, 방망이, 유리병, 가위, 우산을 선정하였다[13]. 객체 인식의 결괏값은 객체의 클래스, 인식 확률, 위치 좌푯값이며, 이는 4.1.3절의 접근성 판별과 4.1.4절의 사람과 위험 객체의 연관성 판별에서 사용한다.

YOLO v4는 코코 데이터셋(COCO Dataset)[14]으로 학습된 모델이며, 기존 모델을 활용하여 칼, 방망이, 가위, 우산 객체를 인식한다. 하지만 일부 클래스 중 낮은 정확도를 보이는 경우에 한에서는 추가 학습을 진행하여 정확도 고도화 작업을 진행하였다. 예를 들어서, 커터칼과 유리병의 학습을 위해 기 라벨링 된 이미지 데이터셋을 활용하였다. 구글의 오픈 이미지 데이터셋에서 객체 인식을 위한 박스와 라벨링 정보가 포함된 이미지를 각 200개씩 다운로드하여 학습에 사용하였다. 방대한 이미지 중 원하는 이미지를 선택해서 다운로드할 수 있도록 도와주는 OIDv4 툴킷[15]을 사용하였다.

2) 행위 인식

본 절에서는 행위 인식 기능을 설명한다. 본 논문에서는 자세 추정 모델인 오픈포즈[16,17]를 사용하여 신체 관절 위치 정보를 통해 사람의 행위를 추정한다. 오픈포즈는 히트맵을 사용하여 관절을 먼저 추정하고, 이후 사람별 관절의 좌표 사이를 연결한다[18]. 이후에 전 단계에서의 정보들을 바탕으로 각 프레임에서 각 관절의 위치를 측정한 후, 사람 별 관절 위치를 저장한다. 여기서 관절 위치 값은 키포인트 값이라 호칭한다.

저장된 키포인트 값을 바탕으로 두 가지 행위를 측정한다. 위협 행동으로 때리기와 발차기 행동 두 가지를 선정하였다.

먼저, 때리기는 양팔의 키포인트 값을 활용하여 판단한다. 총 17개의 키포인트 중 팔에 해당하는 키포인트는 손목, 팔꿈치, 어깨이다. 오른손은 Fig. 2의 2, 3, 4에 해당하고 왼손은 Fig. 2의 5, 6, 7에 해당한다. 손목 y 좌표가 팔꿈치 y 좌표보다 위에 있고, 어깨 y 좌표보다 팔꿈치 y 좌표가 내려간 경우에 팔을 올려 때릴 가능성이 있는 행동으로 판단한다. 두 번째로, 발차기 행동은 골반과 무릎 키포인트 값을 사용하여 판단한다. 오른발은 Fig. 3의 8, 9에 해당하고 왼발은 Fig. 3의 11, 12에 해당한다. 골반 y 좌표보다 무릎 y 좌표가 더 위에 있을 때 발차기를 할 가능성이 있는 행동으로 판단한다.

추가적으로, 본 논문에서는 행위 인식 시 인식된 사람이 기존 인식한 사람과 동일인 인지에 대한 여부를 확인한다. 동일인 여부는 오픈포즈로 얻은 사람 별 키포인트 위치 값과 YOLO v4로 추출한 사람 위치와 매핑하는 것을 통해 확인 가능하다. 구체적인 과정은 다음과 같다. 사람별로 정리된 키포인트 값에서 각 좌표의 최대, 최솟값을 구한다. 관절의 키포인트 값은 x와 y 좌표로 구성되는데, 인식 오류로 인한 무의미한 값을 제외시킨다. 계산한 키포인트 좌표의 최대, 최솟값이 YOLO v4를 통해 인식한 사람의 위치와 같거나 포함되면 동일한 사람으로 판단한다.

Fig. 2. Keypoint position in hitting behavior

Fig. 3. Keypoint location of the kick action

Fig. 4. Determination Accessibility by Comparing Image and Central Coordinates of Body (left: approach, right: do not approach)

3) 접근성 판별

본 절에서는 접근성 판별 기능을 소개한다. 본 기능은 사용자에게 위협하는 행동을 취하는 사람이나 위험 객체를 든 사람이 접근하는지 판별하는 기능이다. 이는 촬영된 이미지의 중심 좌표와 오픈포즈를 통해 인식된 사람의 신체 관절 키포인트의 중심 좌표를 비교하여 판별한다[19]. 이미지의 좌표 평면에서 상대가 사용자로부터 가까운지 멀리 있는지는 y 좌표를 통해 확인할 수 있고, x 좌표를 통해서는 어느 방향으로 다가오는지 알 수 있다.

Fig. 4는 라즈베리파이를 목에 건 사용자의 시점에서 위협을 가할 수 있는 사람의 접근성 정도를 판단하는 그림이다. 위협을 가할 수 있는 사람의 신체 관절 중심 x좌표와 이미지 중심 좌표인 빨간 점의 위치를 비교하여 일정 기준 안에 들어오면 사용자의 방향으로 접근한다고 판단한다. 판별된 접근성 데이터는 다음 위험도 판별과 데이터 저장을 위해서 람다 함수를 통해 AWS 다이나모디비에 저장한다.

4) 사람과 위험 객체의 연관성 판별

본 절에서는 사람과 위험 객체의 연관성을 판별하는 방법을 소개한다. 이 방법은 위험 객체가 인식되었을 때 사용자에게 다가오는 상대가 물체를 들고 있는지 판별하는 기능이다. 이는 YOLO v4로 인식한 위험 객체와 인식된 사람의 바운딩 박스 영역을 비교하여 판별한다[20]. 사람이 어떤 물체를 들고 있을 시 인식된 물체의 영역과 사람의 영역은 겹치게 된다. Fig. 5를 보면 사람이 야구 방망이를 들고 있다. 흰색 테두리 안에 사람이 인식되고, 검정 테두리로 야구 방망이가 위

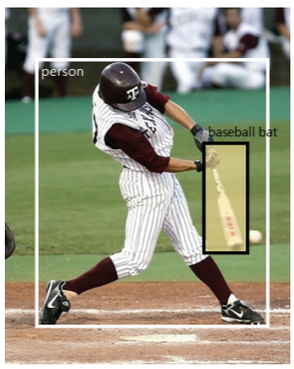

Fig. 5. Relationship Between a Person and a Dangerous Object

Fig. 6. Relationship Between Two People and One Dangerous Object

험 객체로 인식된 경우이다. Fig. 5의 노란색 영역처럼 두 영역이 겹치게 되는데, 이러한 원리를 사용하여 일정 이상 겹쳤을 경우 위험 객체를 특정 사람이 들고 있다고 판단한다.

객체와 사람 간의 여러 가지 경우의 수가 있다. 먼저 Fig. 6처럼 두 사람이 객체 하나에 영역이 겹칠 경우, 객체가 더 가까운 사람에게 속한다고 판단한다. 각 객체가 겹친 사람의 경우를 이차원 리스트에 저장해두고, 객체와 사람의 중심 좌표가 두 점 사이의 거리 공식을 이용해 더 가까운 쪽에 속하도록 한다. 그리고 한 사람이 객체 두 개와 영역이 겹칠 경우, 더 위험한 객체만을 사람과 연결한다. 판별된 위험 객체 데이터는 다음 위험도 판별과 데이터 저장을 위해서 람다 함수를 통해 다이나모디비로 저장된다.

4.2 위험도 평가

본 절에서는 이미지 분석으로 얻은 데이터를 기반으로 위험도를 평가하는 방법을 소개한다. 이를 위하여 4.2.1 절에서는 위험도 계산 알고리즘을 소개하고 4.2.2절에서는 자동화시킨 위험도 평가모델에 대하여 설명한다.

1) 위험도 계산 알고리즘

본 절에서는 위험도 계산 알고리즘을 소개한다. 위험도 계산 알고리즘은 앞 단계에서 얻은 객체, 행위, 접근성, 객체와 사람의 연관성 등 데이터들의 관계성을 바탕으로 위험도를 측정하며, 위험도는 총 4단계로 나뉜다. Table 1은 전체적인 평가 알고리즘을 보여준다. 위험도 평가 기준으로 선정한 속성은 행위(Pose), 객체(Object), 접근성의 유/무(Assessment)이다. 영상 인식에서 얻은 데이터의 경우의 수를 기준으로 6가지로 분류 후, 크게 4단계로 나누어 위험도를 판별한다. 또한, 판별 결과에 따라 그 이후 수행될 결괏값을 결정한다.

첫 번째로, 접근성, 객체, 행위 모두 존재하면, 위험도 3으로 측정하고 “매우 위험하여 대피해야 하는 상황”으로 판별한다. 이 경우 위험 상황을 알리는 오디오를 재생한 뒤, 실시간 상황이 일정 시간 녹화되어 저장된다.

Table 1. Risk assessment criteria

	Assessment	Object	Pose	Risk
1	○	○	○	3
2	○	○		2, 1
3	○		○	
4		○	○	1
5		○		0
6	○			

두 번째로, 접근성이 있는데 객체만 존재하거나 행위만 존재하는 경우, 추가 데이터를 이용하여 위험도를 판별한다. 해당 이미지 속 어두움 정도와 사람의 가까운 정도를 확인하여 두 조건 중 한 가지 경우에라도 위험 기준에 부합할 경우 위험도를 2로 측정한다. 두 조건에 모두 부합하지 않는 경우, 위험도를 1로 측정한다. 위험도가 2로 측정되면, “위험을 추후 확인”해야 하는 상황으로 판별한다. 이 경우 주의하라는 오디오를 재생한 뒤, 이미지를 촬영하는 주기를 감소시킨다. 위험도가 1로 측정되면, 추후 확인 필요 단계로 이미지 촬영 주기만 감소하여 다음 상황을 확인한다.

세 번째로, 접근성은 없고 객체와 행위만 존재하는 경우, “추후 확인 필요”인 위험도 1로 측정한다. 위험도 1로 선정됨에 따른 결괏값은 앞서 설명한 값과 동일하다.

마지막으로, 객체만 존재하고 접근성과 행위는 모두 존재하지 않는 경우, 그리고 접근성만 존재하고 객체와 행위 모두 존재하지 않는 경우엔 위험도를 0으로 측정한다. 위험도 0은 “안전”한 상태로, 특별한 변화가 없다.

2) 위험도 평가 모델

앞서 구성한 알고리즘을 바탕으로 시스템의 학습을 위해 머신러닝으로 자동화한다. 앞의 과정에서 얻은 5가지 데이터(공간 밝기, 위험객체 유무, 접근성, 위협 행동, 가까운 정도)를 파라미터로 설정하며, 임의로 상황별 100번씩 총 500개의 이미지로 시스템을 작동시켜 데이터를 수집하였다.

요소별로 정확하게 분류되어 위험도가 결정되는 특성을 고려하여 의사 결정 방식의 지도 학습 모델을 채택했다. 텐서플로우의 의사 결정 트리 모듈에서 그래디언트 부스티드 회귀 트리[21]과 랜덤 포레스트[22]의 두 모델을 학습시켜 둘 중 더 높은 정확도를 지닌 모델을 채택하였다.

랜덤포레스트에서는 0.9551의 정확도를, 그래디언트 부스티드 회귀트리에서는 0.9872를 보인다. 본 논문에서는 더 높은 정확도를 지닌 그래디언트 부스티드 회귀트리 모델을 채택하였다. 그래디언트 부스티드 회귀트리는 랜덤포레스트와 달리 오차를 보완하며 학습하기 때문에 높은 정확도의 결과를 보였다.

4.3 위험도에 따른 결과 처리

본 절에서는 위험도 평가에서 얻은 결과를 바탕으로 후처리하는 과정에 대하여 설명한다.

1) 문장 처리

본 절에서는 문장 처리 방법을 소개한다. 본 논문의 시스템에서는 이미지에서 분석한 내용을 한 줄의 문장으로 텍스트화한다. 앞 절에서 정의한 위험도 수치를 기준으로, 상황을 문장으로 정리하여 오디오를 재생한다.

문장의 성분인 주어, 목적어, 서술어의 순서를 지정해놓고 추출 데이터를 기반으로 단어를 저장하는 리스트를 각 문장의 성분별로 생성한다. 이렇게 저장된 리스트는 각 이미지의 상황별로 해당하는 단어들을 선택하여 문장을 완성한다.

예를 들면, 어떤 사람이 다가와 위험 객체를 들고 때리려고 한다고 가정해보면 주어는 ‘다가오는 사람이’, 목적어는 ‘야구 방망이를’, 서술어는 ‘들고 때리기 행위를 취하고 있습니다.’를 선택하여 문장을 완성시키며 완성된 문장은 “다가오는 사람이 야구방망이를 들고 있습니다.”, “다가오는 사람이 때리기 행위를 취하고 있습니다.” 등과 같다.

정리된 문장은 AWS 폴리[24]를 통해 텍스트에서 음성 파일로 변환되고, 변환된 음성 파일은 AWS S3에 저장한다.

2) 클라이언트와 서버 간 통신

본 절에서는 클라이언트와 서버 간 통신 방법을 설명한다. 라즈베리파이와 AWS는 엠큐티티를 통해 통신한다. 라즈베리파이에서 등록한 AWS 계정의 접근 키와 엔드 포인트 등을 사용하여 AWS와 연동한다.

본 논문에서는 클라이언트와 서버 간 통신을 위해 라즈베리파이를 AWS S3와 연동하여 데이터를 보내고 불러올 수 있도록 하고, 엠큐티티를 통하여 AWS 람다 함수의 결과를 라즈베리파이에 전송한다. 라즈베리파이는 엠큐티티로부터 받은 값을 기준으로 작동한다.

엠큐티티의 메시지는 위험도 결과에 따라 4가지로 정한다. 위험도 0일 경우, “변화 없음”을 전송한다. 위험도 1일 경우 “촬영 주기 감소”를 전송한다. 이때, 평소 촬영 주기보다 감소하도록 감소할 주기 값도 함께 전송한다. 위험도 2일 경우, “촬영 주기 감소와 음성 출력"을 전송한다. 해당 상황에서는 라즈베리파이가 AWS S3에서 받은 키 값의 이름인 음성 파일을 찾아 저장하고 음성을 출력한다. 그 후, 촬영 주기를 감소한다. 위험도 3일 경우, “음성 출력과 영상 녹화”를 전송한다. 가장 위험한 상황이므로 바로 위험 상황을 알리기 위하여 앞 단계와 같은 방법으로 음성 파일을 저장하여 출력한다. 또한, 사용자에게 벌어지는 상황을 녹화하여 추후 사용자에게 도움이 되도록 한다.

5. 구 현

본 장에서는 본 논문에서 제안하는 Saying CCTV 구현의 전체적인 동작 과정에 대해 정리하여 설명한다.

Fig. 7. Image Taken on Raspberry Pi

Fig. 8. Image in Which Actions and Objects are Recognized

전체 과정의 흐름은 다음과 같다. 먼저 라즈베리파이를 사용하여 이미지를 촬영하여 S3에 저장한다. 이미지가 저장된 후에는 위험도를 계산하는 Lambda 함수가 트리거된다. 구체적인 예시는 Fig. 7-8과 같다. Fig. 7은 사람이 한쪽 팔을 든 채 위협하는 사진이다. Fig. 8에서는 이미지에서 객체와 행위를 인식한다. Fig. 9 ①에서 두 가지 요소 간의 관계성을 분석하여 텍스트화한다. Fig. 9 ②에서는 텍스트를 분석하여 위험도 수치를 계산한다. 위험도 수치 계산 후에는 음성 파일


```
{
  "Date": {
    "S": "20210912234711"
  },
  "Danger_Step": {
    "N": "2"
  },
  "Person_num": {
    "N": "1"
  },
  "Person0": {
    "L": [
      {
        "S": "no danger"
      },
      {
        "S": "access"
      },
      {
        "S": "43656.0"
      },
      {
        "S": "오른손을 든"
      }
    ]
  },
  "Brightness": {
    "S": "49.04"
  },
  "Video_path": {
    "S": "ras_video/20210912234711.h264"
  }
}
```


Fig. 9. Data stored in AWS DynamoDB


```
START RequestId: 1ad44f04-c781-4a7d-b03c-310d20fff022 Version: $LATEST
{'Date': {'S': '20210912234711'}, 'Brightness': {'S': '49.04'}, 'Person_nu
1
Person0
[['no danger', 'access', 43656, '오른손을 든', 2]]
[[-1], [-1], [-1, 0], [-1], [-1]]
위험도2
text: 사람이 오른손을 든 포즈로 접근하고 있습니다.
outputM: mt3
새 DB에 내용 저장
20210912234711.mp3
END RequestId: 1ad44f04-c781-4a7d-b03c-310d20fff022
```


Fig. 10. AWS Lambda log information


```
mqtt : mt3
key: 20210912234711
get mp3 file : 20210912234711
sound start
sound end
```


Fig. 11. Run log screen in Raspberry Pi

(20210912234711.mp3)을 S3에 저장한다. 또한, 위험도 수치가 높지 않을 경우 엠큐티티를 통해 라즈베리파이에게 촬영 주기 감소 요청 신호를 전달하고 음성 (outputM: mt3)을 출력을 요청한다. Fig. 9와 같이 이미지 분석 과정에서 판별된 모든 값은 다이나모디비에 저장된다. Fig. 10은 AWS 람다 로그 정보 값들을 보여준다.

Fig. 11은 라즈베리파이에서 실행된 파이썬 함수의 로그 값을 보여준다. 라즈베리파이는 전달 받은 데이터(Mqtt : mt3)에 따라 촬영 주기를 조절시키며 S3에 음성 파일 (get mp3 file : 20210912234711)을 저장한다. 저장 이후에는 음성 (“사람이 오른손을 든 포즈로 접근하고 있습니다.”)을 출력한다.

6. 사용자 테스트 및 성능평가

본 장에서는 다양한 화질에서의 이미지에서 위험도를 판단하는 사용자 테스트 및 성능평가를 진행한다.

6.1 화질에 따른 이미지 인식

본 절에서는 다양한 이미지 화질에 따른 위험도 판단 성능평가 실험을 설명한다. 성능평가는 동일한 이미지 한 장을 네 가지의 화질로 변경하여 위험 객체, 접근성, 위험 행위를 인식하며 이를 종합적으로 고려하여 위험도를 평가한다.

실험 데이터로는 사람이 다가오며 오른손을 들어 때리는 포즈의 이미지를 사용하였다. 동일한 이미지를 240px(Fig. 12), 360px(Fig. 13), 480px(Fig. 14), 1080px(Fig. 15)의 다양한 화질로 변경하여 실험에 사용하였다. 모델은 위험도 평가 모델 구축 시 가장 우수한 성능을 보인 그래디언트 부스티드 회귀트리 모델을 사용한다.

Table 2. Comparison of Measured Results by Image Quality

Resolution	240	360	480	1080
Expected risk	1	1	1	1
Risk object	×	×	×	×
Accessment	○	○	○	○
Risk action	'오른손을 위로 든'	'오른손을 위로 든'	'오른손을 위로 든'	'오른손을 위로 든'
Risk result	1	1	1	1
Accuray	100%	100%	100%	100%

Fig. 12. 240px Image

Fig. 13. 360px Image

Fig. 14. 480px Image

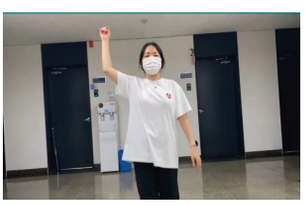

Fig. 15. 1080px Image

Table 2는 각 화질 별 이미지에 대한 위험도 평가 결과를 보여준다. 예상 위험도(Expected risk)는 위험도 평가 그라운드 트루스를 의미한다. 위험 객체(Risk object)는 이미지에 위험 객체가 포함되었는지를 확인하는 항목이다. 접근성(Accessment)는 사람의 접근성 여부를 판단하는 항목이다. 위험 행위(Risk action)은 이미지에 나타난 사람의 행위를 의미한다. 위험도 판단 결과(Risk result)는 모델이 실제로 예측한 위험도를 의미한다.

실험 결과는 다음과 같다. 네 가지 화질의 이미지에서 모두 위험 객체는 인식하지 않았다. 또한, 접근성은 있다고 판단하였다. 위험 행위는 '오른손을 위로 든'이 인식되었다. 상기 위험도 판단 요소들로 위험도를 예측한 결과 네 가지 이미지에서 모두 그라운드 트루스와 동일한 1을 예측하였고 모든 화질에서 동일한 성능을 보여주고 있다.

본 논문에서 제안하는 Saying CCTV는 AWS를 활용한 클라우드 컴퓨팅을 통해 높은 컴퓨팅 성능으로 이미지 분석이 가능하다. 이는 영상을 촬영하는 라즈베리파이 기기에서 저사양의 성능을 가져도 정확한 판단을 가능케 한다. 또한, 화질이 낮을수록 적은 용량의 이미지로 인해 처리시간이 감축되는 효과를 얻는다.

7. 결 론

본 논문에서는 장소와 기기의 성능 제약 없이 실시간으로 위협을 감지할 수 있는 1인칭 영상 인식 시스템인 Saying CCTV을 제안하였다.

본 논문에서 제안하는 Saying CCTV는 AWS 서버와 라즈베리파이 클라이언트로 구성된다. 클라이언트에서는 이미지를 촬영해 AWS 서버에 전송한다. 서버에서는 객체 인식 YOLO v4와 행위 인식을 수행하는 오픈포즈를 통해 이미지를 인식하고 둘의 관계성을 파악하는 알고리즘으로 위험도를 계산한다. 계산한 위험도를 기반으로 사용자에게 전달할 문장을 생성한다. 모든 결괏값을 데이터베이스에 저장하며, 클라이언트에서는 서버로부터 최종 결괏값을 전송받아 사용자에게 음성을 출력한다.

또한, 본 논문에서 기기 사양의 제한 없이 시스템 사용이 가능한지 여부를 확인하는 실험을 진행하였다. 이를 위하여 낮은 해상도의 이미지에서 객체, 행동 인식을 진행하였다. 실험결과, 제안하는 Saying CCTV는 높은 해상도 뿐만 아니라 낮은 해상도에서도 정확한 객체, 행동 인식 성능과 그에 따른 위험도 판단 능력을 보였다. 따라서 본 논문에서 사용하는 라즈베리파이를 사용하여도 Saying CCTV의 내부 알고리즘들이 정상적으로 작동하는 것을 확인 할 수 있었다.

향후 연구는 지역, 시간, 범죄자 등의 실시간 데이터를 추가적으로 수집하고 이를 기반으로 범죄 통계를 자동화하여 실시간 위협을 예측하여 Saying CCTV를 고도화 할 예정이다.

References

[1] Prosecutors, "Crime Analysis 2020," Supreme prosecutors' Office, 2021.

[2] J. S. Yoon, J. S. Park, S. H. Ahn, and M. J. Kim, "Violent offending with unspecified motivation toward strangers," *Korean Criminological Review*, pp.1-179, 2014.

[3] J. B. Park, S. J. Park, J. J. Jeong, and K. W. Kim, "Development of lntelligent video surveillance technology to solve problem of deteriorating arrest rate by improving CCTV constraint," *Journal of The Korean Institute of Communication Sciences*, Vol.37, No.1, pp.17-24, 2019.

[4] J. K. Han, "A study on establishment and management of the crime prevention CCTV," *Journal of Public Society*, Vol.8, No.4, pp.109-137, 2018.

[5] Y. W. Joo and S. J. Lee, "Intelligent CCTV trends and performance improvement measures," *KISA(Korea Internet & Security Agency) Technological Trend Data*, 2014.

[6] J. B. Park, S. J. Park, J. J. Jeong, and K. W. Kim, "Core principles and problem solving methods of intelligent video security technology to support the improvement of CCTV restrictions," *Journal of The Korean Institute of Communication Sciences*, Vol.37, No.1, pp.17-24, 2019.

[7] C. H. Lee and J. W. Kim, "Leveraging cloud-native edge cluster for intelligent edge CCTV service," in *Proceedings of the Korean Institute of Communication Sciences Conference*, Jeju, pp.1507-1508, 2021.

[8] W. C. Choi and J. Y. Na, "Development of CCTV cooperation tracking system for real-time crime monitoring," *Journal of Korea Academia-Industrial cooperation Society*, Vol.20, No.12, pp.546-554, 2019.

[9] S. I. Jang, L. Battulg, and A. Nasridinov, "Detection of dangerous situations using deep learning model with relational inference," *Journal of Multimedia Information System*, Vol.7, No.3, pp.205-214, 2020.

[10] M. Nakib, R. T. Khan, M. S. Hasan, and J. Uddin, "Crime scene prediction by detecting threatening objects using convolutional neural network," in *Proceedings of 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, Rajshahi, pp.1-4, 2018.

[11] AWS, "What is AWS Lambda?" [Internet], https://docs.aws.amazon.com/ko_kr/lambda/latest/dg/welcome.html

[12] A. K. Shetty, I. Saha, R. M. Sanghvi, S. A. Save, and J. Patal, "A review: Object detection models," In *Proceedings of 6th International Conference for Convergence in Technology (12CT 2021)*, Maharashtra, pp.1-8, 2021.

[13] C. K. Park, "A study on the revision of armed with a deadly weapon or other dangerous thing," *Chosun Law Journal*, Vol.17, No.3, pp.283-303, 2010.

[14] COCO, Common Objects in Contest [Internet], https://cocodataset.org/#download

[15] EscVM, OIDv4_ToolKit [Internet], https://github.com/EscVM/OIDv4_ToolKit

[16] CMU-Perceptual-Computing-Lab, OpenPose [Internet], https://github.com/CMU-Perceptual-Computing-Lab/openpose

[17] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, pp. 7291-7299, 2017.

[18] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learing based 2D human pose estimation: A survey," *Tsinghua Science and Technology*, Vol.24, No.6, pp.663-676, 2021.

[19] T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira, and H. Mizoguchi, "Development of a person following robot with vision based target detection," In *Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, pp.5286-5291, 2006.

[20] R. Debnath, A. Singha, B. Saha, and M. K. Bhowmilk, "A comparative study of background segmentation approaches in detection of person with gun under adverse weather conditions," in *Proceedings of 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, pp.1-7, 2020.

[21] M. Pham, M. Tanjil, and M. Ruppert-Stroescu, "Application of gradient boosting through SAS Enterprise Miner 12.3 to classify human activities," in *SAS Global Forum*, Lasvegas, 2016.

[22] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognition Letters*, Vol.27, No.4, pp.294-300, 2006.

[23] J. H. Lee, J. C. Kim, and D. H. Seo, "A study on image caption algorithm based on object detection," *Journal of Advanced Marine Engineering and Technology*, Vol.41, No.7, pp.683-689, 2017.

[24] AWS, "What is Amazon Polly?" [Internet], https://docs.aws.amazon.com/ko_kr/polly/latest/dg/what-is.html

김 예 영

https://orcid.org/0000-0002-1995-3552
e-mail : yeyong0423@naver.com
2018년 ~ 현 재 숙명여자대학교
IT공학전공 학사과정
관심분야 : Cloud Computing, Deep Learning

정 수 현

https://orcid.org/0000-0003-1390-5071
e-mail : jshamy@naver.com
2022년 숙명여자대학교 IT공학전공(학사)
관심분야 : 클라우드

박 소 현

https://orcid.org/0000-0003-1022-1790
e-mail : sohyunpark@dongguk.ac.kr
2016년 숙명여자대학교 멀티미디어과학과 (석사)
2020년 숙명여자대학교 IT공학과(박사)
2020년 ~ 2023년 숙명여자대학교 빅데이터 활용 연구센터 책임연구원
2023년 ~ 현 재 동국대학교 WISE캠퍼스 컴퓨터공학과 조교수
관심분야 : 인공지능, 빅데이터, 영상처리

박 영 호

https://orcid.org/0000-0002-5284-9589
e-mail : yhpark@sm.ac.kr
1992년 동국대학교 컴퓨터공학과(석사)
2005년 한국과학기술원 전산학과(박사)
1993년 ~ 1999년 한국전자통신연구원 교환전송연구단 선임연구원
2005년 ~ 2006년 한국과학기술원 첨단정보기술연구센터 연구원
2005년 ~ 2006년 동국대학교 컴퓨터멀티미디어학과 겸임교수
2006년 ~ 현 재 숙명여자대학교 인공지능공학부 교수
관심분야 : 데이터베이스, XML, IR(정보검색), 멀티미디어 데이터베이스, Bio정보공학, 영상미디어, 예술&공학인터페이스, 데이터베이스 관리시스템, 머신러닝, 빅데이터, 데이터분석, Telecommunication System