

골격 좌표 벡터 및 LSTM 모델을 이용한 넘어짐 검출

시종욱*¹, 손현철*², 김다슬*³, 김문년*⁴, 정지연*⁵, 김규리*⁶, 김영형**^{*}, 김성영***^{*}

Fall Detection using Skeletal Coordinate Vector and LSTM Model

Jongwook Si*¹, Hyeoncheol Son*², Daseul Kim*³, Moonnyeon Kim*⁴, Jiyeon Jeong*⁵,
Gyuree Kim*⁶, Younghyung Kim**^{*}, and Sungyoung Kim***^{*}

이 연구는 금오공과대학교 학술연구비로 지원되었음(2018-104-131)

요약

최근 들어 세계 각국에서는 고령화, 1인 가구 및 독거노인 증가 문제를 직면한 국가들이 많이 나타나고 있으며 이에 따라 고령자와 1인 가구의 삶의 질 향상을 위한 연구의 필요성이 대두되고 있다. 본 논문에서는 1인 가구의 삶의 질 향상 방법 중 안전이라는 측면에 초점을 두어 넘어짐의 여부를 판단하는 방법을 제안하였다. 제안한 시스템의 방법으로는 골격 추출 기술을 사용하여 비디오 영상에서 골격 좌표를 추출하고 이 좌표를 벡터화하여 상관관계를 나타낸 후 이를 LSTM에 입력으로 넣어 넘어짐을 판단한다. 추출한 골격 좌표 중에서는 일부만을 사용하여 넘어짐의 판단 성능을 향상할 수 있도록 하였다. 제안한 시스템의 성능을 5가지의 기준 데이터 세트를 사용하여 평가하였고, 대부분의 데이터 세트에서 우수한 성능을 보였다.

Abstract

In recent years, many countries around the world have faced the problems of aging, single-person households, and the increasing number of elderly people who live alone. These problems have led to the need for research to improve the quality of life for the elder and single-person households. In this paper, the method of detecting falls was proposed with a focus on the aspect of safety among the methods of improving the quality of life of single-person households. First, we extract key points of human skeletons based on the existing method, vectorize them to represent correlation between them, and input them into LSTM to determine whether falls have occurred. We try to enhance the performance by using only representative feature points, not all feature points. Five reference datasets were used to evaluate the performance of proposed system and performed well in most datasets.

Keywords

fall down detection, anomaly detection, deep learning, LSTM, skeletal coordinate vector

* 금오공과대학교 컴퓨터공학과

- ORCID¹: <http://orcid.org/0000-0003-2092-2769>

- ORCID²: <http://orcid.org/0000-0003-2036-9190>

- ORCID³: <http://orcid.org/0000-0002-1280-4300>

- ORCID⁴: <http://orcid.org/0000-0001-6147-7476>

- ORCID⁵: <http://orcid.org/0000-0002-6222-3504>

- ORCID⁶: <http://orcid.org/0000-0002-8209-9037>

** 금오공과대학교 IT융합학과 교수

- ORCID: <http://orcid.org/0000-0001-5708-854X>

*** 금오공과대학교 컴퓨터공학과 교수(교신저자)

- ORCID: <http://orcid.org/0000-0002-7722-6759>

• Received: Aug. 07, 2020, Revised: Nov. 23, 2020, Accepted: Nov. 26, 2020

• Corresponding Author: Sungyoung Kim

Dept. of Computer Engineering, Kumoh National Institute of Technology, 61 Daehak-ro (yangho-dong), Gumi, Gyeongbuk, [39177] Korea

Tel.: +82-54-478-7530, Email: sykim@kumoh.ac.kr

1. 서 론

오늘날 세계 각국에는 고령화 문제를 직면한 국가들이 많이 나타나고 있다. 또한, 1인 가구와 독거 노인 비율도 증가하고 있다. 통계청의 2019 고령자 통계 자료[1]에 따르면 2019년을 기준으로 우리나라의 인구피라미드는 30~50대가 많은 항아리형이며 65세 이상의 고령자는 전체 인구의 14.9%를 차지하고 있다. 이와 달리 2060년에는 60대 이상이 많은 역삼각형의 항아리형으로 65세 이상의 고령자가 전체 인구의 43.9%를 차지할 것으로 예상된다. 또한, 65세 이상 고령자 중 1인 가구 비율이 34.2%로 독거노인 문제도 심각한 것을 볼 수 있다. 통계청의 인구 총 조사에 나타난 1인 가구의 현황 및 특성 자료[2]에 따르면 2017년 우리나라의 1인 가구는 전체 가구의 28.6%를 차지하고 있으며 이는 2000년대비 152.6% 증가한 수치이다. 반면에 일반 가구는 2000년대비 37.5% 증가하였다. 이는 1인 가구의 비율이 매우 빠르게 증가하고 있음을 보여준다. 이러한 현상들이 지속됨에 따라 고령자와 1인 가구의 삶의 질 향상을 위한 연구의 필요성이 대두되고 있다.

1인 가구와 고령자처럼 혼자 있는 경우가 많은 이들은 갑자기 머리가 아프거나 속이 안 좋아서 쓰러지는 상황 등의 위급한 상황이 발생한다면 제대로 된 대처를 하지 못할 수도 있다. 늦은 대처는 고독사로 이어질 가능성이 크다. 고독사는 따로 통계를 내지 않기 때문에 연도별 무연고 사망자 자료[3]를 통해 살펴보면 무연고 사망자는 2014년부터 2018년까지 지속해서 증가하고 있다. 하지만 무연고 사망자는 유가족이 없거나 유가족이 시신 인수를 거부해 지방자치단체에서 사후 처리를 맡기는 사망자들을 말하는 것으로 고독사는 그 수가 훨씬 많을 것으로 예상된다. 이러한 상황들이 발생하는 것을 방지하기 위해 본 논문에서는 갑자기 쓰러진 상황을 검출하는 연구를 진행하였다.

넘어짐 판단에 대한 연구는 이전부터 지속적으로 이루어지고 있다. 기존의 넘어짐 연구에서는 넘어짐을 검출하고자 하는 대상 객체(사람)가 센서를 직접 착용 또는 부착하거나 객체 주변에 센서를 설치하는 등으로 관련 데이터를 얻은 후 이를 분석하여

넘어짐을 검출하는 연구[4]-[6] 등이 있다. 센서를 사용하는 연구는 움직임 실시간으로 정교하게 파악할 수 있다는 장점이 있지만, 실생활에서 항상 장비를 착용해야 한다는 문제와 배터리 문제 등이 발생할 수 있기 때문에 한정된 영역에서만 사용 가능하다. 이 밖의 다른 넘어짐 연구는 CCTV 영상과 같은 비디오 데이터에서 객체를 검출하여 행동을 분석하는 연구[7]-[12] 등이 있다. 이러한 방법은 사람이 미처 인지하지 못하는 유용한 특징들을 데이터를 통해 직접 찾아낼 수 있다는 장점이 있다. 또한, 자세 추정이 점점 정확해지면서 그 응용 분야도 자세 교정, 행동 인식, 이상 행동 감지, 안전 예방 시스템, 증강현실 등으로 확대되고 있다. 최근에는 2차원 자세 분석뿐만 아니라, 3차원 좌표계를 이용한 연구가 활발히 진행 중이다. 하지만 3차원 좌표계를 이용할 경우 마이크로소프트사에서 개발한 키넥트 센서와 같이 비싼 센서를 필요로 한다. 따라서 본 연구에서는 키넥트와 같은 비싼 센서를 사용하지 않고 비디오 영상만을 이용하여 자세를 분석하고 넘어짐을 검출할 수 있는 방법을 제안한다.

II. 관련 연구

2.1 넘어짐 판단

넘어짐 판단은 노인 또는 환자가 보호자 없이 홀로 있다가 갑자기 쓰러지는 등의 위험 상황을 검출하여 신속하게 대응할 수 있게 한다.

2.1.1 센서 기반 넘어짐 판단

기존의 많은 넘어짐 판단 연구는 센서 기반과 영상 기반 등으로 나눌 수 있다. 센서 기반의 넘어짐 판단은 센서로부터 얻어온 센서 데이터를 사용하여 상황을 판단한다. 센서는 웨어러블 기기나 신체 부착형 센서 등과 같이 넘어짐을 판단하고자 하는 객체(사람)가 직접 착용 또는 부착하는 경우와 바닥 압력 센서나 진동 센서와 같이 객체 주변에 설치하는 설치형 센서 등이 있다. Fañez 연구팀은 넘어짐 판단에서 일반적으로 사용되는 3DACC 센서가 내장되어 있는 웨어러블 스마트 시계를 사용하여 사

용자의 움직임 패턴을 측정하고, 이 패턴을 분석하여 넘어짐 여부를 판단한다[4]. 해당 연구는 사용자 중심의 온라인 학습과 전이 학습을 융합하여 검출 성능을 높였다. 하지만 전이 학습 적용 후에도 잘못 검출한 경우와 아예 검출하지 못한 경우가 많이 나타났다. Dovgan 연구팀은 엉덩이, 무릎 등에 태그를 부착한 후 태그의 위치를 감지하는 시스템을 사용하여 값을 수집한다. 이후 이 값을 사용하여 넘어짐을 포함한 비정상적인 행동을 탐지하였고 높은 성능을 보였다[5]. Fañez의 연구팀과 Dovgan 연구팀이 같이 착용 및 부착형의 센서를 사용할 경우 항상 착용의 번거로움 및 배터리 등이 문제가 발생할 수 있다. 이와 달리 Zigel 연구팀은 바닥 진동 센서와 소리 센서를 조합하여 넘어짐을 판단하였다[6]. 해당 연구의 경우 넘어짐 판단의 대상자가 센서를 직접 착용 및 부착하지 않아도 되고 저비용으로 넘어짐 판단이 가능하다는 장점이 있다. 하지만 배경 소음과 센서의 위치가 결과에 영향을 줄 수 있고, 진동 센서 등의 설치형 센서의 짧은 수명 등의 문제가 발생할 수 있다.

2.1.2 영상 기반 넘어짐 판단

영상 기반의 넘어짐 판단의 경우 카메라를 이용하여 촬영된 영상 데이터를 처리하여 상황을 판단한다. Chen은 가우시안 혼합 모델을 사용하여 배경 및 노이즈를 제거하고 움직이는 객체를 파악한 후 객체의 높이와 종횡비, 무게중심을 구한다. 구해진 값들을 1초 전의 값들과 비교하여 변화가 크다면 넘어짐으로 판단한다[7]. 해당 연구는 여러 사람의 넘어짐을 검출할 수 있으며 사람이 많아질수록 잘못 검출한 횟수가 많아지지만, 이전의 연구보다 높은 성능을 보였다. Núñez-Marcos 연구팀은 RGB 이미지를 Optical Flow[13] 이미지로 변환하여 CNN에 입력으로 넣고 특징을 추출한 후 FC-NN을 사용하여 넘어짐을 판단한다[8]. 해당 연구는 전이 학습을 사용하여 검출 성능을 높였지만, Optical Flow[13]을 사용하여 계산이 많고 조명 변화에 민감하다. Miao Yu 연구팀은 비디오 데이터에서 배경을 제거하여 인체의 실루엣을 추출한 후 CNN을 적용하여 서 있는 상태, 누운 상태(넘어짐), 앉은 상태 등으로 분류

한다[9]. 해당 연구는 대부분의 움직임 타입을 잘 검출하였지만, 굽힌 상태를 넘어짐으로 잘못 검출한 경우도 있었다. Fan 연구팀은 넘어짐을 서 있는 상태, 넘어지는 도중, 넘어짐, 움직임 없음의 4단계로 나누어 CNN을 적용했다. 또한, 유튜브에서 수집된 넘어짐 영상을 이용하여 새로운 데이터 셋을 구축하였다[10]. 해당 연구는 기존의 연구와 비교하여 실험을 진행하였고 대부분 좋은 성능을 보였다. Yong Chen은 Mask R-CNN[14]을 사용하여 배경으로부터 객체를 검출한 후 Attention[15] 기법과 LSTM[16]을 사용하여 넘어짐을 검출하였다[11]. 해당 연구는 이전의 연구와 달리 Mask R-CNN[14]을 사용하여 배경으로부터 객체를 검출하였고, 높은 성능을 보였다. 하지만 단일 사람에게만 적용 가능하다는 한계가 존재한다. Chen 연구팀은 OpenPose [17]를 사용하여 인체의 골격 좌표를 추출하고, 이 데이터를 활용하여 몸의 중심선 각도, 외부 직사각형의 폭과 높이 비율 계산하여 넘어짐을 판단한다 [12]. 해당 연구는 OpenPose[17]를 사용하여 저렴한 비용으로 높은 성능을 달성하였다. 하지만 골격 추출 시 모든 관절점이 추출되지 않으면 불완전한 데이터가 생겨 최종 인식에 영향을 줄 수 있다. K.G. Gunale 연구팀은 배경을 제거하여 움직이는 객체를 탐지하고 직사각형 및 타원형 경계 상자도 표시하고 가로, 세로, 비율 등을 고려해 형상을 추출하여 Adaboost 분류기를 사용해 넘어짐을 판단한다[18]. Arisa Poonsri 연구팀은 평균 필터 모델과 결합된 가우스 모델을 사용하여 배경을 제거 한후 사람을 추출하여 인간의 신체적인 움직임의 여섯 가지 자세로 구분한다[19]. 해당 연구는 프레임 간의 정보, 형상의 가로, 세로, 비율 등을 파악하여 넘어짐을 감지한다. 위의 연구를 이용하여 MoG 및 배경 제거를 이용하여 추출한 객체를 PCA를 사용해 특징을 추출한다[20]. 이후에 Human Centroid Tracking을 사용해 자세를 분류하고 자세를 최종적으로 결정한다.

2.1.3 하이브리드 방식 넘어짐 판단

센서와 영상을 모두 활용해서 넘어짐을 판단하는 방법은 정확하게 넘어짐을 검출할 수 있다는 장점이 있다. 하지만, 센서의 신호가 혼재되는 경우 오

작동할 수 있는 단점이 존재한다. 최연성 연구팀은 움직임 감지 센서, WIFI, Bluetooth, 지그비 모듈 등을 통해 센서값을 얻고 활용을 했다[21]. 카메라를 추가하고 영상과 영상에서 추출되는 움직임 벡터를 활용하여 단점을 해결하였다.

2.2 골격 추출

최근 많은 분야에서 마이크로소프트사에서 개발한 키넥트 센서를 활용한 연구가 진행되고 있다. 키넥트 센서는 RGB 카메라의 IR Depth Sensor, IR Emitter, Color Sensor, Microphone Array 및 센서를 상하로 움직일 수 있는 Tilt Motor 등으로 구성된다. 이 센서들을 이용하여, 일반적인 RGB 카메라를 이용해 촬영하는 Color View, 깊이 정보를 보여주는 Depth View 및 사용자의 검출된 골격을 보여주는 Skeleton View로 나타낸다. Skeleton View는 20개의 골격에 대한 위치를 알려주며, 이를 이용하여 해당하는 신체 부위의 골격 정보를 쉽게 알 수 있다.

하지만 키넥트 센서는 RGB 카메라와 IR 카메라가 물리적으로 떨어져 있으므로 각 영상을 일치시키기 위한 작업이 필요하다는 단점이 있다. 라이브러리들을 사용하여 어느 정도를 일치시키는 것은 가능하지만 한계가 있다. 또한 카메라 앞에서 5미터 이내 정도로 인식 거리가 제한된다. 스스로 적외선을 송출하고 그것을 사용해 촬영하므로 일반 카메라를 사용하는 것 보다는 외부조명의 영향을 한결 적게 받지만 조명의 영향으로부터 완전히 자유롭지는 못하다.

키넥트 센서를 활용해 골격 좌표를 추출하는 방법 이외에도 영상에서 골격 좌표를 추출하기 위한 여러 가지 방법들이 있다. OpenPose[17]는 딥러닝의 CNN을 기반으로 하며, 사진에서 실시간으로 여러 사람의 몸, 손, 그리고 얼굴의 특징점을 추출할 수 있으며 키넥트 센서와 같은 장비 없이 일반 카메라에서도 적용이 가능하다. 하지만 학습량이 부족하거나 학습되지 않았던 희귀한 자세, 몸의 일부가 보이지 않는 경우, 여러 사람이 접치는 경우 등은 아직 해결하지 못하였고 실제 사람이 아니지만, 사람과 비슷한 형태를 지니면 자세를 추출하는 단점도 존재한다. OpenPose[17]는 2D 좌표만을 추정하였지만

페이스북 인공지능 연구팀이 공개한 Densepose[22]는 RGB 영상의 모든 인간의 픽셀을 3D 표면에 매핑한다. 또한, 지역 기반 모델로 이루어져 있으며 Cascading을 통해서 정확도를 향상 시킬 수 있고 깊이 정보를 얻을 수 있다. HRNet[23]는 고해상도의 인간 포즈 추정이 가능하다. 기존의 방법들은 고해상도에서 저해상도 네트워크에 의해 생성된 저해상도 표현으로부터 고해상도 표현을 복원하지만 HRNet[23]는 전체 프로세스를 통해 고해상도 표현을 유지할 수 있다. COCO 데이터 세트 기반 벤치마크 결과 중 현재 성능이 가장 우수하다.

2.3 이상 탐지

이상 탐지(Anomaly Detection)는 정상(Normal) 데이터와 비정상(Abnormal) 데이터를 구별해내는 것을 말하며 비디오 감시, 제조업, 시스템 운영, 보안 등의 많은 분야에서 활용되고 있다. 특히 비디오 감시 분야에서는 CCTV 등으로부터 얻은 영상에서 싸움, 교통사고, 넘어짐 등의 비정상적인 상황을 인지하여 신속하게 사고를 예방하거나 대응할 수 있다.

Seymanur Akti 연구팀은 공공장소 등에 설치된 감시카메라로부터 얻은 영상을 이용하여 공개적으로 이용 가능한 싸움 장면 데이터 세트를 구축하였다. 이 데이터 세트에 Attention[15]과 LSTM[16] 기법을 적용하여 싸움 장면을 분류한다[24]. Gracia 연구팀은 이미지 분류에 주로 사용되는 Bag-of-Words framework를 응용해 싸움과 싸움이 아닌 상황을 구별한다[25]. Zu Hui는 영상 데이터에서 배경과 차량을 추출하고 차량의 위치, 가속도, 방향 등의 정보를 이용해 사고를 판단한다[26]. Yu Yao는 차량의 블랙박스 영상으로부터 획득한 교통사고 영상 데이터 세트를 구축하였다. 또한, 이 데이터 세트에 GRU[27]로 구성된 RNN-ED 모델을 적용하여 교통사고를 검출한다[28].

2.4 객체 동작 인식

여러 기술을 이용하여 영상 내에서 객체의 동작을 인식하는 연구가 진행되고 있다. 장재영 연구팀은 스마트폰 환경을 기반으로 오픈포즈와 딥러닝을

이용하여 인간의 행동을 인식하는 방법을 제안한다 [29]. 오병훈은 색상 통신을 이용하여 스마트폰 환경에서 3차원상의 거리와 각도를 이용하여 객체의 동작을 인식하는 방법을 제안한다[30]. 최석규 연구팀은 사람을 재인식하기 위한 반복적인 EM알고리즘 및 3단 계층 네트워크를 포함한 확장된 RNN구조에 특징과 지표 거리를 이용한 방법을 제안한다 [31].

III. 제안 방법

본 논문에서 제안하는 방법의 처리 절차는 다음과 같다. 각 프레임별로 Openpose[17] 기술을 활용하여 해당 영상에서 사람에 해당하는 영역의 골격 좌표를 모두 추출한다. 넘어짐을 판별하기 위해서, 골격 좌표 사이의 상관관계를 크게 보여줄 수 있는 좌표의 정보를 바탕으로 이웃하는 좌표와의 벡터를 구한다. 이를 LSTM[16] 모델의 입력으로 하여 최종적으로 넘어짐을 검출한다. 넘어지는 것은 사람이 서 있는 상태에서 넘어지는 형태로 급격한 변화에 의해 일어나는 경우가 대부분이다. 따라서, 사람의 골격에 따른 벡터의 시간적인 변화가 중요하므로 시간의 흐름을 학습할 수 있는 LSTM[16]을 사용해야 할 필요가 있다.

3.1 골격 좌표 추출

기존의 OpenPose의 골격 추출 시스템은 골격 좌표의 개수에 따라 나눌 수 있다. 넘어짐의 상황을 판단하기 위해서는 OpenPose에서 검출한 모든 골격 좌표를 이용하지 않고 특정한 일부 좌표만을 사용

한다. 이렇게 일부 좌표만을 사용하는 이유는 Openpose가 항상 모든 골격을 정확하게 추출하는 것은 아니기 때문이다. 사람의 형태가 올바르게 않거나 화면 내에 사람의 전신이 위치하지 않으면 OpenPose는 보이는 부분에 대해서만 골격을 검출하게 된다. 따라서 본 논문에서는 필요한 골격이 검출되지 않았을 경우 이전 프레임의 좌표를 사용한다. 그런데 여러 프레임에 걸쳐 골격 추출이 정확하게 검출되지 않을 때에는 이전 프레임의 정보를 사용한 것에도 한계가 있으므로 정확한 넘어짐 검출이 어려울 수 있다. 또한, 넘어지는 형태는 굉장히 방대하므로 모든 골격에 대하여 판단하기에는 큰 무리가 있다. 따라서 본 논문에서는 OpenPose에서 제공하는 25개의 골격 좌표중에서 넘어짐 검출에 가장 적합한 11개의 골격 좌표를 추출하여 사용한다.

3.2 골격 좌표 벡터화

골격 좌표의 상관관계를 통해 넘어짐을 검출하기 위해서는 이웃한 좌표와의 관계를 수치적 표현해야만 한다. 추출된 좌표만을 가지고 판단하기에는 입력 영상의 크기가 다르고, 동일한 골격의 형태라도 위치가 다를 경우 수치상으로 차이가 발생한다. 따라서 본 논문에서는 추출된 모든 골격 좌표 데이터에 대해서 영상 크기에 따라 정규화하여 그림 1과 같이 CSV 파일로 저장하고, 방향을 고려할 수 있도록 벡터화한다. 저장된 CSV는 11개의 골격 좌표이므로 프레임당 22개의 값이 저장된다. 짝수 인덱스는 x 좌표, 홀수 인덱스는 y 좌표를 나타내며 최종적으로 10개의 입력 벡터로 변환한다(그림 2). 그림 3은 벡터화한 골격 정보를 시각화한 내용이다.

0.234635	0.529167	0.246875	0.493519	0.253125	0.556481	0.257812	0.556481	0.314323	0.520833	0.313021	0.507407	0.260677	0.54537	0.26224	0.526389	0.288542	0.550926	0.289844	0.534259	0.288281	0.490741	0.230119	0.488187
0.234635	0.529167	0.247135	0.493519	0.253125	0.556481	0.257812	0.556481	0.314323	0.523611	0.313021	0.507407	0.260677	0.54537	0.26224	0.526389	0.288542	0.550926	0.288542	0.534722	0.286979	0.490741	0.231653	0.488206
0.236198	0.529167	0.248437	0.493519	0.253125	0.556481	0.257812	0.556481	0.314323	0.523611	0.313021	0.507407	0.260677	0.54537	0.26224	0.526389	0.288542	0.550926	0.288542	0.534722	0.286979	0.490741	0.231653	0.488217
0.23776	0.531482	0.248437	0.496296	0.251563	0.556481	0.257812	0.556481	0.314583	0.523611	0.313021	0.510185	0.260938	0.548148	0.26224	0.529167	0.288542	0.550926	0.288542	0.534722	0.286979	0.490741	0.234732	0.48824
0.236198	0.531944	0.248437	0.496296	0.253125	0.556481	0.257812	0.556481	0.314583	0.523611	0.313021	0.510185	0.260677	0.539815	0.26224	0.529167	0.288542	0.548148	0.288542	0.534722	0.286979	0.490741	0.23178	0.488197
0.236198	0.531482	0.246875	0.496296	0.253125	0.556481	0.257812	0.556481	0.314583	0.523611	0.313021	0.5125	0.260677	0.54537	0.260938	0.531482	0.288542	0.550926	0.288281	0.5375	0.286979	0.490741	0.233163	0.488131
0.236198	0.531482	0.246875	0.496296	0.251563	0.556481	0.257812	0.556481	0.314583	0.526389	0.313021	0.5125	0.260677	0.54537	0.26224	0.529167	0.288542	0.550926	0.288542	0.534722	0.286979	0.488426	0.233215	0.488182
0.234635	0.531482	0.24401	0.496296	0.253125	0.556481	0.257812	0.556481	0.314583	0.523611	0.313021	0.512963	0.260677	0.542593	0.26224	0.529167	0.288542	0.550926	0.288281	0.539815	0.286979	0.490741	0.230125	0.488162
0.234635	0.515278	0.245573	0.493519	0.253125	0.556481	0.257812	0.556481	0.314583	0.523611	0.313021	0.512963	0.259375	0.537037	0.26224	0.529167	0.288542	0.550926	0.288281	0.5375	0.286979	0.490741	0.230169	0.488156
0.236198	0.515278	0.24401	0.493519	0.254687	0.556481	0.257812	0.556481	0.314583	0.523611	0.313021	0.5125	0.259375	0.537037	0.260677	0.526389	0.288542	0.550926	0.286979	0.5375	0.286979	0.488426	0.231675	0.48818
0.23776	0.526389	0.245573	0.493519	0.253125	0.556481	0.25625	0.556481	0.314583	0.526389	0.313021	0.515278	0.259375	0.537037	0.26224	0.526389	0.288542	0.550926	0.288281	0.539815	0.286979	0.490741	0.236234	0.488149
0.239323	0.529167	0.245573	0.493519	0.251563	0.556481	0.254687	0.556481	0.314583	0.526389	0.313021	0.512963	0.260677	0.539815	0.26224	0.526389	0.288542	0.550926	0.286979	0.537037	0.286979	0.487963	0.236276	0.488103
0.239323	0.526389	0.240885	0.493519	0.251563	0.556481	0.253125	0.553704	0.314583	0.526389	0.313021	0.515278	0.260677	0.54537	0.260677	0.529167	0.288281	0.550926	0.288281	0.5375	0.286979	0.488426	0.233201	0.488132
0.236198	0.515278	0.24401	0.493519	0.251563	0.558796	0.254687	0.556019	0.314583	0.529167	0.313021	0.515278	0.260677	0.542593	0.260938	0.529167	0.288281	0.550926	0.288281	0.534722	0.286979	0.487963	0.233194	0.485423
0.23776	0.523611	0.24401	0.493519	0.251563	0.556481	0.25625	0.556019	0.314583	0.526389	0.313021	0.515278	0.260677	0.539815	0.260938	0.529167	0.288281	0.548148	0.288281	0.537037	0.286979	0.488426	0.234737	0.488142
0.239323	0.529167	0.245573	0.493519	0.253125	0.556481	0.25625	0.556019	0.313021	0.526389	0.313021	0.515278	0.260677	0.534722	0.26224	0.523611	0.288281	0.550926	0.288281	0.5375	0.286979	0.488426	0.23627	0.488172
0.239323	0.523611	0.25	0.493519	0.253125	0.556481	0.25625	0.556019	0.313021	0.529167	0.313021	0.512963	0.260677	0.539815	0.26224	0.523611	0.288281	0.550926	0.286979	0.5375	0.286979	0.488426	0.237173	0.48818
0.23776	0.520833	0.248437	0.491204	0.253125	0.556481	0.25625	0.556481	0.314583	0.526389	0.313021	0.515278	0.260677	0.539815	0.26224	0.529167	0.288281	0.550926	0.286979	0.534722	0.288281	0.487963	0.236259	0.488081
0.239323	0.526389	0.248437	0.493519	0.253125	0.556481	0.257812	0.556481	0.314583	0.526389	0.313021	0.515278	0.260938	0.542593	0.26224	0.529167	0.288281	0.550926	0.286979	0.534722	0.286979	0.488426	0.237735	0.488143
0.23776	0.529167	0.245573	0.499074	0.251563	0.556481	0.25625	0.556481	0.314323	0.526389	0.313021	0.518056	0.260677	0.539815	0.260938	0.531944	0.288542	0.550926	0.286979	0.537037	0.286979	0.488426	0.236254	0.485499
0.23776	0.526389	0.245312	0.501852	0.253125	0.556481	0.25625	0.556019	0.314323	0.526389	0.313021	0.518056	0.259115	0.5375	0.260677	0.529167	0.288542	0.550926	0.286979	0.5375	0.286719	0.488426	0.236234	0.485452
0.23776	0.531944	0.239323	0.509722	0.254687	0.553704	0.254687	0.553704	0.314583	0.526389	0.314323	0.520833	0.251563	0.50463	0.257812	0.515278	0.288542	0.550926	0.286979	0.542593	0.286719	0.487963	0.23014	0.482772

그림 1. 골격 좌표를 저장하는 CSV 파일
Fig. 1. A CSV file that stores skeletal coordinates

```

Nose_Neck = [array_skeleton[i][2] - array_skeleton[i][0], array_skeleton[i][3] - array_skeleton[i][1]]
Neck_LElbow = [array_skeleton[i][12] - array_skeleton[i][2], array_skeleton[i][13] - array_skeleton[i][3]]
Neck_RElbow = [array_skeleton[i][14] - array_skeleton[i][2], array_skeleton[i][15] - array_skeleton[i][3]]
LElbow_LWrist = [array_skeleton[i][12] - array_skeleton[i][4], array_skeleton[i][13] - array_skeleton[i][5]]
RElbow_RWrist = [array_skeleton[i][14] - array_skeleton[i][6], array_skeleton[i][15] - array_skeleton[i][7]]
Neck_Hip = [array_skeleton[i][20] - array_skeleton[i][2], array_skeleton[i][21] - array_skeleton[i][3]]
Hip_LKnee = [array_skeleton[i][20] - array_skeleton[i][16], array_skeleton[i][21] - array_skeleton[i][17]]
Hip_RKnee = [array_skeleton[i][20] - array_skeleton[i][18], array_skeleton[i][21] - array_skeleton[i][19]]
LKnee_LAnkle = [array_skeleton[i][16] - array_skeleton[i][8], array_skeleton[i][17] - array_skeleton[i][9]]
RKnee_RAnkle = [array_skeleton[i][18] - array_skeleton[i][10], array_skeleton[i][19] - array_skeleton[i][11]]
    
```

그림 2. 골격 좌표 벡터화

Fig. 2. Vectorization on skeleton coordinate

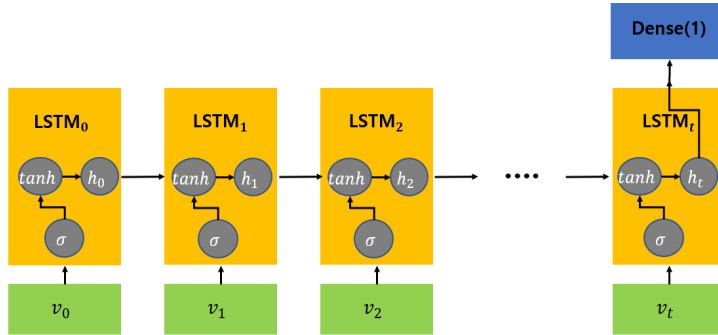


그림 4. 네트워크 구조

Fig. 4. Network structure

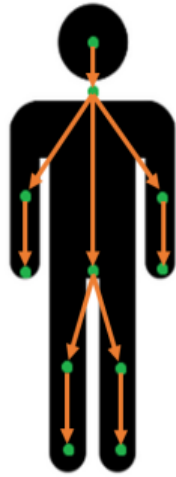


그림 3. 골격 좌표 벡터의 시각화

Fig. 3. Visualization of skeleton coordinate vectors

표 1. 골격 좌표 벡터의 세부정보

Table 1. Details of skeletal coordinate vectors

Vector	Relation	Vector	Relation
v_0	Nose - Neck	v_5	Neck - Hip
v_1	Neck - Leiblow	v_6	Hip - Lknee
v_2	Neck - Relbow	v_7	Hip - Rknee
v_3	Leiblow - Lwrist	v_8	Lknee - Lankle
v_4	Relbow = Rwrist	v_9	Rknee - Rankle

IV. 실험

4.1 데이터 세트

제안하는 방법에서는 객관적인 검출 성능 평가를 확보하기 위하여 넘어짐 검출 분야에서 사용되는 여러 가지 데이터 세트를 이용하여 데이터 세트별 성능 평가를 제시한다. AI Hub 데이터 세트[32]는 화질이 높고 상체의 전면부가 보이게 넘어지는 데이터가 주를 이룬다. UR Fall Detection 데이터 세트 [33]는 카메라의 시점이 전면 및 조감 뷰로 이루어져 있는데 전면부 시점만 성능 평가에 사용한다. 이

3.3 넘어짐 검출 모델 구조

LSTM[16] 모델과 노드가 1개인 Dense 레이어로 단순하게 구성된 네트워크 구조를 사용한다(그림 4). LSTM은 시계열 데이터 처리에 주로 쓰이지만, 한 프레임에 대하여 하나의 골격 좌표 벡터(v_i)가 차례대로 나열되므로 이런 구조가 가능하다. 표 1은 v_i 가 어떤 골격과의 상관관계를 나타내는지 보인다.

데이터 세트는 근거리에서 사람이 넘어지는데 사람의 전신은 드러나지 않는 특징이 있다.

Multiple Cameras Fall 데이터 세트[34]는 하나의 이상 행동에 대해 여러 각도로 촬영한 영상으로 이루어져 있다. Fall Detection 데이터 세트[35]는 주로 2가지 형태(수평과 수직)로 넘어짐이 표현되어 있는데 성능 평가에는 가로의 형태로 넘어진 것만 사용한다. Science Direct 데이터 세트[36]는 근거리에서 측면으로 넘어진 형태로 구성된다.

4.2 훈련

10개의 벡터화된 골격 정보를 입력으로 사용하여 모델을 훈련한다. AI Hub[32]의 크로마키 영상 100개 중에서 55개의 넘어짐 영상과 45개의 일반 영상을 훈련에 사용하였다. 넘어짐은 1, 넘어짐이 아닌 경우 -1로 레이블링 한다. Ubuntu 18.04 LTS 운영체제 환경에서 GTX Geforce 1080 TI 2대의 GPU를 사용하여 학습을 진행하였다.

4.3 성능

데이터 세트마다 정확한 평가를 위해 20개 내외의 영상을 사용하고 데이터 세트에서 골격 좌표가 존재하는 프레임 단위로 평가한다. 넘어짐 판단은 (1)과 같이 진행한다.

$$\begin{cases} \text{if } value > thresh & \text{swoon} \\ \text{else} & \text{normal} \end{cases} \quad (1)$$

넘어짐이 발생한 상황에서, 그림 5는 정확하게 골격을 검출한 경우이고 그림 6은 일부 또는 전체 골격을 검출하지 못한 경우이다.

골격을 제대로 검출하지 못하는 경우는 후면으로 넘어지는 경우(그림 6(a)), 넘어져서 신체 일부가 화면을 벗어나는 경우(그림 6(b))나 넘어지는 방향이 카메라 방향을 향하는 경우(그림 6(c))이다. 그림 6(d)를 보면 사람이 누워있는 것으로 판단할 수 있다. 그런데 골격은 정확하게 추출되었지만, 골격만을 본다면 사람이 서 있는 것과 누워있는 것을 구분하기 어려운 경우이다. 이전 절에서도 언급한 것과 같이 UR Fall Detection 데이터 세트는 그림 6(b)와 같이 신체 일부가 화면을 벗어나는 경우가 빈번하여 낮은 검출 성능을 나타내었다(표 2 참조).

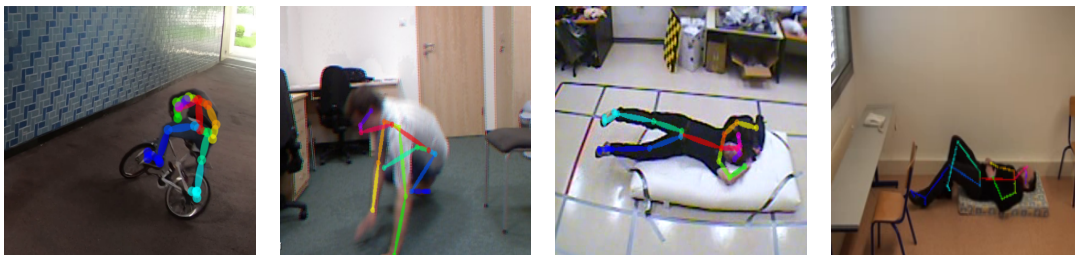


그림 5. 골격을 정확하게 검출한 사례
Fig. 5. Examples of accurate detection on skeletons

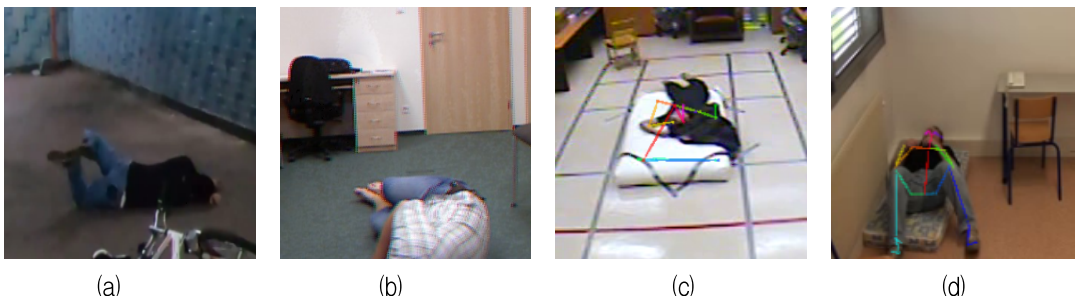


그림 6. 골격을 부정확하게 검출한 사례
Fig. 6. Examples of inaccurate detection on skeletons

표 2. 데이터 세트에 대한 평가 (%)

Table 2. Evaluation on reference datasets (%)

		Accuracy	Precision	Recall	Specificity
AI Hub[32]	10	85.5	95.4	83.0	70.8
	24	62.8	55.0	69.1	71.7
UR Fall Detection[33]	10	73.5	58.9	55.1	79.7
	24	61.0	76.6	53.5	49.0
Multiple Camera Fall[34]	10	83.6	72.4	72.4	88.3
	24	75.5	66.3	73.3	82.3
Fall Detection[35]	10	89.5	82.7	95.3	96.1
	24	76.1	57.9	98.0	98.5
Science Direct[36]	10	76.8	66.3	71.3	83.3
	24	80.5	81.8	69.2	79.8

표 3. 기존 연구 성과와의 비교 (%)

Table 3. Comparison performance of proposed method with existing researches (%)

		Accuracy	Precision	Recall	Specificity
[35]	[18]	79.3	79.7	83.5	73.1
	[19]	86.2	89.1	93.2	64.3
	[20]	91.4	88.7	98.0	60.0
	Proposal	89.5	82.7	95.3	96.1

그림 7은 사용한 모든 데이터 세트에 대한 넘어짐 검출 결과의 성능을 TPR, FPR의 관계에 따른 ROC 그래프를 그린 것이며, AUC는 87%의 성능을 보인다.

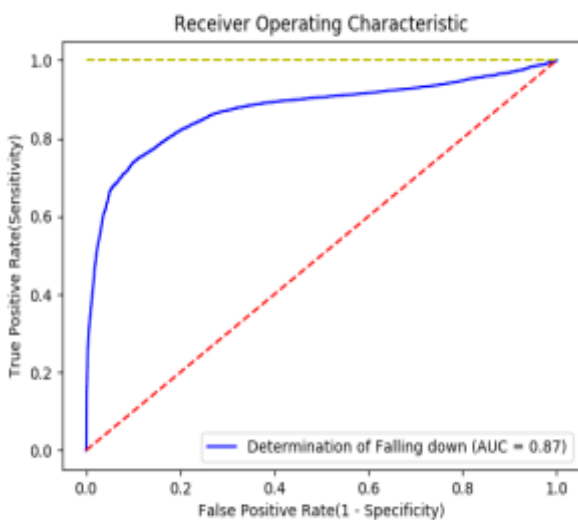


그림 7. ROC 그래프
Fig. 7. ROC graph

V. 결론 및 향후 과제

1인 가구 및 독거노인이 증가하고 있는 상황에서 신속한 처리가 필요한 위험 상황이 갑자기 발생하면 대응이 쉽지 않은 상황이다. 따라서 본 논문에서는 입력 영상에서 검출한 골격 좌표를 사용하여 넘어짐을 판단하기 위해 이들 간의 상관관계를 벡터화하고 딥러닝 모델을 이용하여 최종 넘어짐을 검출하는 방법을 제안하였다. 넘어짐 검출 분야에서 사용되는 여러 가지 데이터 세트를 사용하여 검출 성능을 분석하여 제안하는 방법의 유용성을 확인하였다. 골격 검출 성능은 제안하는 넘어짐 검출 성능에 큰 영향을 미치는데 본 논문에서 제안된 방법의 성능을 높이기 위해서는 가장 먼저 골격 좌표검출 성능을 높여야 하고 이를 위해서 넘어짐에 대한 데이터를 충분히 수집하여 학습해야 한다. 또한, 이전 프레임의 좌표를 고려하는 시계열 데이터 형태로 연구해 본다면 보다 높은 성능을 기대할 수 있을 것이다. 제안하는 방법은 1인 가구 및 독거노인의 삶의 질 향상에 이바지할 수 있을 것이다.

References

- [1] 통계청, 「2019 고령자 통계」, [Accessed: Aug. 31. 2020], http://kostat.go.kr/portal/korea/kor_nw/1/1/index.board?bmode=read&aSeq=377701.
- [2] 통계청, 「인구총조사에 나타난 1인 가구의 현황 및 특성」, [accessed: Aug. 31. 2020], https://kostat.go.kr/portal/korea/kor_nw/1/2/2/index.board?bmode=read&aSeq=370806&pageNo=1&rowNum=10&amSeq=&sTarget=title&sTxt.
- [3] 데이터 슝, 연도별 무연고 사망자 수, [accessed: Aug. 31. 2020], <http://www.datasom.co.kr/news/articleView.html?idxno=101370>.
- [4] M. Fañez, J. R. Villar, E. de la Cal, J. Sedano, and V. M. González, "Transfer learning and information retrieval applied to fall detection", *Expert Systems*, Jan. 2020.
- [5] E. Dovgan, M. Luštrek, B. Pogorelc, A. Gradišek, H. Burger, and M. Gams, "Intelligent elderly-care

- prototype for fall and disease detection", *Zdravniški Vestnik*, Vol. 80, No. 11, pp. 824-831, Nov. 2011.
- [6] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and soundProof of concept on human mimicking doll falls", *IEEE Transactions on Biomedical Engineering*, Vol. 56, No. 12, pp. 2858-2867, Dec. 2009.
- [7] M. C. Chen, "A video surveillance system designed to detect multiple falls", *Advances in Mechanical Engineering*, Vol. 8, No. 4, Apr. 2016. <https://doi.org/10.1177/1687814016642914>.
- [8] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras, "Vision-based fall detection with convolutional neural networks", *Wireless Communications and Mobile Computing*, Vol. 2017, Dec. 2017. <https://doi.org/10.1155/2017/9474806>.
- [9] M. Yu, L. Gong, and S. Kollias, "Computer vision based fall detection by a convolutional neural network", *ACM International Conference on Multimodal Interaction*, Glasgow UK, pp. 416-420, Nov. 2017. <https://doi.org/10.1145/3136755.3136802>.
- [10] Y. Fan, M. D. Levine, G. Wen, and S. Qiu, "A deep neural network for real-time detection of falling humans in naturally occurring scenes", *Neurocomputing*, Vol. 260, pp. 43-58, Oct. 2017.
- [11] Y. Chen, L. Wang, J. Hu, and M. Ye, "Vision-Based Fall Event Detection in Complex Background Using Attention Guided Bi-directional LSTM", *IEEE*, Vol. 8, No. 19980483, pp. 161337-161348, Sep. 2020.
- [12] W. Chen, Z. Jiang, H. Guo, and X. Ni, "Fall Detection based on key points of human-skeleton using openpose", *Symmetry*, Vol. 12, No. 5, p. 744, May 2020.
- [13] J. J. Gibson, "The perception of visual surfaces", *The American Journal of Psychology*, Vol. 63, No. 3, pp. 367-384, Jul. 1950.
- [14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Mar. 2017.
- [15] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *International Conference on Learning Representations*, Sep. 2014.
- [16] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, Nov. 1997.
- [17] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Nov. 2016.
- [18] M. Chamle, K. G. Gunale and K. K. Warhade, "Automated unusual event detection in video surveillance", *International Conference on Information and Computer Technologies (ICICT)*, Coimbatore, India, pp. 1-4, Aug. 2016.
- [19] A. Poonsri and W. Chiracharit, "Fall detection using Gaussian mixture model and principle component analysis", *International Conference on Information Technology and Electrical Engineering (ICITEE)*, Phuket, Thailand, pp. 1-4, Oct. 2017.
- [20] A. Poonsri and W. Chiracharit, "Improvement of fall detection using consecutive-frame voting", *International Workshop on Advanced Image Technology (IWAIT)*, Chiang Mai, Thailand, pp. 1-4, Jan. 2018.
- [21] S. S. Kim, S. W. Kim, Y. Choi, "An Abnormal Activity Monitoring System Using Sensors and Video", *Journal of KIISE*, Vol. 41, No. 12, pp. 1152-1159, Dec. 2014.
- [22] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense Human Pose Estimation in the Wild", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7297-7306, Jun. 2018.
- [23] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep

high-resolution representation learning for human pose estimation", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 5693-5703, Feb. 2019.

[24] Ş. Akti, G. A. Tataroğlu, and H. K. Ekenel, "Vision-Based Fight Detection from Surveillance Cameras", 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, Turkey, pp. 1-6, Nov. 2019.

[25] I. S. Gracia, O. D. Suarez, G. B. Garcia, and T. K. Kim, "Fast fight detection", Journal of Pone, pp. 1-19, Apr. 2015. <https://doi.org/10.1371/journal.pone.0120448>

[26] Z. Hui, Y. Xie, M. Lu, and J. Fu, "Vision-based real-time traffic accident detection", Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, pp. 1035-1038, Jul. 2014.

[27] K. Cho, B. van Merriënboer, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation", 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Jun. 2014.

[28] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised Traffic Accident Detection in First-Person Videos", IEEE International Conference on Intelligent Robots and Systems, Macau, China, pp. 273-280, Nov. 2019.

[29] J.Jang, S. Hong, D. Son, H. Yoo, H. An, "Development of Real-time Video Surveillance System Using the Intelligent Behavior Recognition Technique", Journal of IIBC, Vol. 19, No. 2, pp. 169-174, Apr. 2019.

[30] B. Oh, "Development of Motion Recognition Platform Using Smart-Phone Tracking and Color Communication", Journal of IIBC, Vol. 17, No. 5, pp. 143-150 Oct. 2017

[31] S. Choi, W. Xu, "A Study on person Re-identification System using Enhanced RNN", Journal of IIBC, Vol. 17, No. 2, pp. 15-23, Apr. 2017

[32] AI Hub, [accessed: Aug. 2. 2020], <http://www.aihub.or.kr/aidata/139/download>

[33] UR Fall Detection Dataset, [accessed: Aug. 4. 2020], <http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html>

[34] Multiple Cameras Fall Dataset, [accessed: Aug. 4, 2020], <http://www.iro.umontreal.ca/~labimage/Dataset/>

[35] Fall Detection Dataset, [accessed: Aug. 4. 2020], <http://le2i.cnrs.fr/Fall-detection-Dataset?lang=fr>

[36] Science Direct Dataset, [accessed: Aug. 4. 2020], <https://www.sciencedirect.com/science/article/pii/S2352340919301908>

저자소개

시 종 욱 (Jongwook Si)



2020년 8월 : 금오공과대학교
컴퓨터공학과(공학사)
2020년 9월 ~ 현재 :
금오공과대학교 컴퓨터공학과
대학원(석사과정)
관심분야 : 영상분석, 인공지능

손 현 철 (Hyeoncheol Son)



2020년 8월 : 금오공과대학교
컴퓨터공학과(공학사)
2020년 9월 ~ 현재 :
금오공과대학교 컴퓨터공학과
대학원(석사과정)
관심분야 : 영상분석, 딥러닝

김 다 슬 (Daseul Kim)



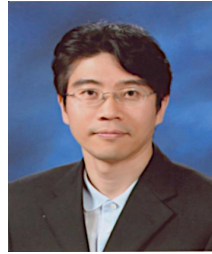
2020년 2월 : 금오공과대학교
컴퓨터공학과(공학사)
2020년 3월 ~ 현재 :
금오공과대학교 컴퓨터공학과
대학원(석사과정)
관심분야 : 영상분석, 이상감지

김 문 년 (Moonnyeon Kim)



2015년 3월 ~ 현재 : 금오공과
대학교 컴퓨터공학과(학사과정)
관심분야 : 컴퓨터비전, 영상처리

김 성 영 (Sungyoung Kim)



1994년 2월 : 부산대학교
컴퓨터공학과(공학사)
1996년 2월 : 부산대학교
컴퓨터공학과(공학석사)
2003년 8월 : 부산대학교
컴퓨터공학과(공학박사)
2004년 ~ 현재 : 금오공과대학교

컴퓨터공학과 교수

관심분야 : 영상처리, 컴퓨터비전, 기계학습, 딥러닝

정 지 연 (Jiyeon Jeong)



2018년 3월 ~ 현재 : 금오공과
대학교 컴퓨터공학과(학사과정)
관심분야 : 영상처리, 멀티미디어

김 규 리 (Gyuree Kim)



2019년 3월 ~ 현재 : 금오공과
대학교 컴퓨터공학과(학사과정)
관심분야 : 멀티미디어, 웹, IT정책

김 영 형 (Younghyung Kim)



1992년 2월 : 금오공과대학교
전자공학과(공학사)
1996년 2월 : 금오공과대학교
대학원 산업경영학과(경영학석사)
2010년 2월 : 금오공과대학교
대학원 산업경영학과(경영학박사)
2014년 ~ 현재 : 금오공과대학교

IT융합학과 교수

2016년 ~ 2017년 : 3D 프린팅 산업협회장

관심분야 : 산업조직, HRD, 리더십, 3D 프린팅, 딥러닝