JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 23, No. 3, pp. 371-379, Mar. 2022

Check for updates

CNN 기반 CCTV 동영상 내 보행자 응급 상황 자동 감지 기술 연구

송 인 서[1] · 박 태 정[2*]
[1]덕성여자대학교 IT미디어공학과 연구원
[2*]덕성여자대학교 IT미디어공학과/사이버보안 전공 교수

Automatic Emergency Detection Based on CNN for Pedestrians in CCTV Video

Inseo Song[1] · Taejung Park[2*]

[1]Researcher, Department of IT Media Engineering, Duksung Women's University, Seoul 01369, Korea
[2*]Professor, Department of IT Media Engineering/Cybersecurity, Duksung Women's University, Seoul 01369, Korea

[요 약]

심정지, 뇌졸중 등과 같이 즉각적인 의료적 처치가 필요한 응급 상황은 일반적으로 매우 급작스럽게 발생하기 때문에 신속한 상황 파악이 중요하다. 본 논문에서는 이차원 이미지 내 물체 감지만을 수행하는 Convolutional Neural Network (CNN) 구현 중 하나인 YOLO(You Only Look Once)를 확장해서 CCTV로 촬영한 동영상 속에서 보행자가 갑자기 쓰러지는 응급 상황을 자동으로 인지하고 웹페이지 지도 상에 위치를 표시함으로써 신속한 의료 처리를 가능하게 하는 시스템을 제안한다. 일반적으로 동영상 내 사물 인식은 CNN과 같은 단일 이미지 인식 기술만으로는 부족하다고 생각되어 동영상에 특화된 많은 연구들이 제시되어 있으나 본 논문에서는 2차원 이미지에 특화된 CNN 기술의 하나인 YOLO를 이용해서 비교적 간단하지만 실용적인 응급 상황 감지 시스템을 구현할 수 있음을 제시한다. 본 논문에서 제안하는 방법의 성능은 실험을 통해 네 가지 성능 지표를 통해 제시한다.

[Abstract]

Since most medical emergencies including cardiac arrest and stroke happen unexpectedly, it is critical to recognize and respond to the situations immediately. In this paper, we propose an AI-based system which recognizes automatically medical emergencies where pedestrians fall unexpectedly due to their health problems captured in real-time video clips from CCTVs and locates the geometric position on a map on a web page to provide with prompt first aids. To this end, we extend the YOLO (You Only Look Once) network, a variant of the Convolutional Neural Network (CNN) which is suitable for 2D still images, not video. Though many researchers have studied on the methods dedicated to recognize objects in video, with a belief that CNN is not enough to recognize motions, we show that it is possible to build a robust but simple medical emergency detection system by extending the YOLO network - a variant of CNN - that only handles 2D images. Also, we report the performance of the proposed system in four performance measures in this paper.

색인어 : 딥러닝, YOLO, 응급상황, 영상인식, CCTV

Keyword : Deep Learning, YOLO, Emergency, Image Recognition, CCTV

http://dx.doi.org/10.9728/dcs.2022.23.3.371

Received 11 January 2022; **Revised** 07 February 2022
Accepted 21 February 2022

***Corresponding Author; Taejung Park**

Tel: +82-2-901-8339
E-mail: tjpark@duksung.ac.kr

Ⅰ. 서 론

1-1 연구 배경 및 필요성

시민의 안전과 치안을 위해 공공 CCTV 설치는 점차 증가하고 있는 추세이다. 그러나 CCTV 영상만으로는 실시간으로 발생 중인 상황에 대한 인식과 대처를 자동화하기가 쉽지 않다. 다시 말해 CCTV는 널리 사용되는 영상 기록 장치로서 비교적 풍부한 자료를 생성하는 수단임에도 불구하고, 사고 발생 후에 사고 당시 상황을 파악하기 위한 보조적인 장치로 활용되는 것이 대부분이다.

보건복지부 조사에 따르면 중증응급환자 적정시간 내 최종 치료기관에 도착하는 비율은 2017년 기준으로 52.4%에 그치고 있다 [1]. 즉, 응급상황 발생 시 2차 피해를 막기 위해서는 신속한 상황 인식 및 대처가 중요하나, 인적이 드문 시간이나 장소에서 발생하는 응급상황은 상황 인식 자체가 어려우므로 적절한 시기에 대처가 쉽지 않다고 볼 수 있다. 따라서 최우선적으로 사건 발생 시점을 인지하는 과정을 ICT 기술을 통해 무인화 및 자동화할 수 있다면 응급 상황에 대한 대처를 개선할 수 있을 것이다.

본 논문에서는 빠르게 물체를 탐색하는 YOLO(You Only Look Once)[2]에 기초하여 CCTV를 통해 촬영된 거리 영상에서 갑자기 쓰러지는 사람을 인지하고 자동으로 응급 구조 기관에 연락을 취하는 실용적인 시스템을 제안하고 그 효용성을 실험을 통해 증명한다.

1-2 관련 연구

1) 2차원 비디오 내 일반적인 사물 인식 연구

2차원 비디오 내에서 일반적인 사물을 인식하고 명칭을 부여하는 다양한 연구가 수행되었으나 현재 대중적으로 가장 널리 사용되는 네트워크들 중 하나는 YOLO(You Only Look Once)[2]이다. YOLO는 single-shot-detection 방식을 사용하는 물체 인식 네트워크로 2016년 발표된 이후 현재는 정확도가 개선된 버전인 YOLO v3가 발표되었다[3].

YOLO는 two-shot-detection 방식을 사용하는 네트워크보다 물체 인식 정확성은 다소 떨어지지만 입력된 이미지를 한 번만 연산하므로 처리 속도가 매우 빨라 높은 실시간성을 보인다는 장점이 있다. YOLO 네트워크는 입력된 이미지를 정방형 셀로 분할하여 각 셀마다 class confidence를 산출해 최종적으로 이미지에서 탐지된 물체의 바운딩 박스와 class, 확률 정보를 출력한다(그림 1).

이러한 2차원 비디오 내 사물 인식과 관련된 최근 연구들 중 가장 주목할만한 연구로 Feichtenhofer와 동료 연구자들이 발표한 SlowFast 네트워크[4]를 들 수 있는데, 이 네트워크는 낮은 프레임 레이트(slow)와 높은 프레임 레이트(fast)를 동시에 파악할 수 있도록 설계되었다.

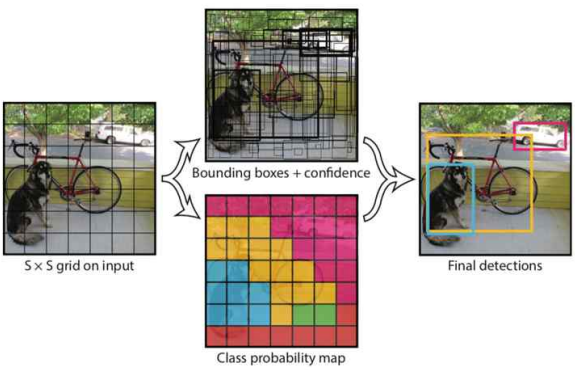

그림 1. YOLO 물체 인식 과정
Fig. 1. Object Detection Process in YOLO

이 연구에서는 SlowFast의 slow 부분은 공간 상의 의미를 파악하며 fast 부분은 시간의 흐름에 따른 유용한 정보를 파악함으로써 물체 자체를 인식할 뿐만 아니라 동작의 의미까지 인식한다.

2) 2차원 이미지 및 비디오 내 인체 정보 추출 연구

최홍석 외 저자[5]는 2차원 동영상 내 인체 정보를 추출하여 3차원 인체 메시(mesh) 모델로 추출하는 기존 방식이 현재 동영상 프레임의 특성에만 지나치게 의존하는 기존 연구를 개선하고 시간의 경과 속에서 일관성을 유지하는 시스템(Temporally Consistent Mesh Recovery System; TCMR)을 제안하였다. 이 연구의 경우 GRU (Gated Recurrent Units) [6]를 이용하여 2차원 동영상 내에서 인체의 움직임에 대한 의미를 파악하기 보다는 3차원으로 추출하는 과정에서 동작의 자연스러운 연결 구현에 중점을 두었다.

Bazarevsky 외 저자들이 제안한 BlazePose[7]는 CNN (Convolutional Neural Network)[8]으로 모바일 장치에서 촬영한 비디오 내에서 인체를 감지하고 인체의 각 부분을 node와 edge로 구성되는 뼈대(skeleton) 구조로 제시한다.

Martinez와 동료 저자들은 GRU 및 LSTM [9]을 기반으로 인체 뼈대 구조에 기초하여 인간의 다음 번 동작을 예측하는 연구를 수행하였다[10].

Pavllo와 동료 연구자들은 인체를 뼈대로 구성하고 동작을 예측할 경우 관절 사이의 각도를 Euler 각도나 exponential map parameterization[11]으로 표현할 경우 오차가 누적되는 문제를 해결하기 위하여 관절 사이의 각도를 quaternion으로 표현함으로써 이러한 오차 누적 문제를 해소하였다.

Dabral과 동료 연구자들이 제안한 HG-RCNN (HourGlass Regions Based CNN) 방식[12]에서는 ResNet[13]과 HourGlass Network[14]를 이용하여 단일 이미지 1장에서 여러 사람의 뼈대 구조를 3차원으로 구성하는 연구를 수행하였다.

Pavlakos와 동료 저자들은 대규모 모션 캡처 데이터베이스를 학습시킴으로써 단일 이미지 상에서 인체의 자세와 얼

굴 표정, 손 동작, 성별까지 파악하여 3차원 메시 모델로 구성하는 연구를 수행하였다[15].

본 연구 수행 중 2차원 비디오에서 이러한 인체의 뼈대 구조를 감지하고 보행 중 의학적 응급 사태로 인해 쓰러지는 사람의 인체의 팔다리 부분 등의 각도들을 DNN (Deep Neural Network)으로 감지하고자 했으나 현재 최신 기술에서도 일부 CCTV 영상에 대해서는 CCTV의 위치와 장애물로 인한 지속적이고 일관적인 인체 뼈대 구조 파악이 어려운 문제가 발견되어 본 논문에서 해결하고자 하는 방향과 부합되지 않는 한계가 있었다. 따라서 본 논문에서 제안하는 방식에서는 2차원 비디오 내 일반적인 사물 인식에 초점을 맞추었으며 응급성이 중요한 애플리케이션의 특성으로 인하여 대규모 CCTV 데이터 스트림에서 빠른 속도로 사물을 인식할 수 있고 보편성이 뛰어난 YOLO를 이용하였다.

Ⅱ. 본 론

2-1 접근 방식

본 연구에서는 2차원 동영상 스트림 중에서 사물 검출을 사용하여 실신 발생 여부를 감지하는 방법을 적용한다. 연구 초기에는 영상 내에서 인물의 3D 자세를 추정한 후 자세를 분류하여 실신 여부를 판단하는 방법을 사용하려 했으나, 3D 자세 추정을 사용하여 실신 여부를 판별하는 경우 몇 가지 문제점이 존재하였다.

먼저 3D 자세 추정을 하기 위해서는 먼저 영상에서 감지된 인물을 바탕으로 2D 자세 관절 연결 정보를 인식한 후, 이를 기반으로 다시 3D 연결 구조를 추정하는 과정이 필요하다. 이 과정에서 실시간 3D 자세 추정을 해 주는 모델인 VNect를 사용하여 테스트 하였으나, 일반적인 보행 동작과는 달리 실신 동작은 프레임 상에서 인물의 관절 부분이 겹쳐져 있어 2D 및 3D 자세에 대한 정상적이고 안정적인 추정이 불가능하였다.

또한 3D 자세 추정을 한 후 인물의 자세를 분류하기 위해서는 영상 내의 공간 정보가 필요하며, 이에 따라 영상의 좌표계를 변환하기 위해 영상의 camera extrinsic parameter 정보가 필요한데, 본 연구에서 사용한 데이터셋에는 해당 정보가 포함되어 있지 않아 영상 분석을 통해 카메라 공간을 파악하는 별도의 연산이 필요하다는 또 다른 문제도 있었다.

종합적으로, 실시간 탐지가 필요한 상황에서 매 프레임마다 거쳐야 하는 연산이 지나치게 복잡하다고 판단되어 보다 일반적이고 단순한 방식으로 문제에 접근하고자 하였다.

따라서 최종적인 구현 방식은 실신 장면이 촬영된 프레임만을 추출하여, 객체 검출 네트워크를 통해 2D 영상에서 실신에 해당하는 객체를 검출하는 방식으로 실신 탐지를 구현하는 방향으로 설정하였다. 그림 2에서는 제안하는 방식에 대한 흐름도를 제시한다.

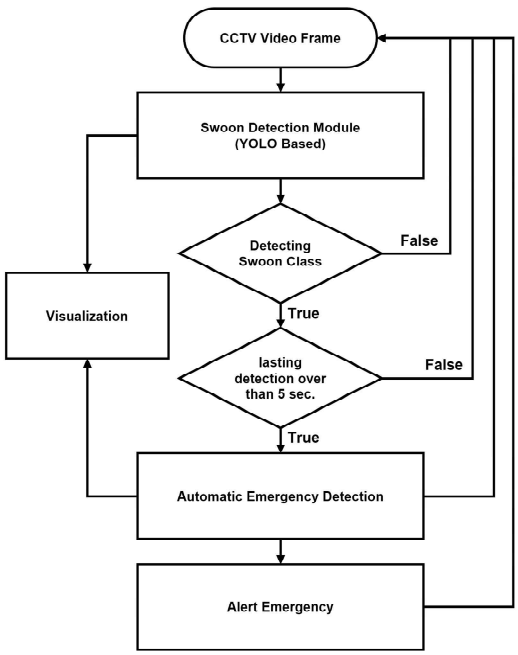

그림 2. 흐름도
Fig. 2. Flowchart

2-2 구현

1) 데이터셋

모델 학습을 위해 한국지능정보사회진흥원(NIA;National Information Society Agency)이 구축하고 AI Hub(aihub.or.kr)에서 제공하는 'CCTV 이상행동 영상 데이터셋(Abnormal Event CCTV Video AI Traning Dataset)을 활용했다. 해당 데이터셋은 12가지 이상행동(폭행, 싸움, 절도, 기물파손, 실신, 배회, 침입, 투기, 강도, 데이트폭력 및 추행, 납치, 주취행동)으로 구성되어 있으며, 각 영상은 라벨링 정보(XML 파일)와 함께 MP4 영상으로 제공된다. 라벨링 정보에는 영상에서 발생하는 이상행동에 대한 카테고리 정보, xy좌표 정보, 이상행동 발생 시간 및 지속시간, 촬영 장소, 촬영 시간대에 대한 정보 등이 포함되어 있다. 개인정보 보호 문제로, 모든 영상은 실제 CCTV 영상 녹화본이 아니라 노상 혹은 크로마키 세트에서 사전에 섭외된 연기자가 설정된 상황을 연출하는 방식으로 촬영되었다.

본 연구에서는 이 중 '실신(Swoon)' 상황에 해당하는 912개 영상(총 84시간 26분 16초 분량)을 사용했다.영상 데이터와 제공되는 XML 형식의 주석 파일을 파싱해 이벤트 시작(/event/strarttime) 및 지속시간 정보 (/event/duration)를 바탕으로, MP4 파일에서 실신 상황에 해당하는 프레임을 추출했다. 추출은 영상 인코딩/디코딩 라이브러리인 FFmpeg을 사용해 일괄처리했으며, 캡처한 프레임은 YOLO 모델 학습에 적합하도록 [16] 960*640px로 크기를 조정해서 JPEG 형식으로 저장했다.

표 1. 라벨링 데이터 구조
Table 1. Data structure for labeling

Level 1	Level 2	Level 3	Level 4
folder			
filename			
source	database		
	annotation		
size	width		
	height		
	depth		
header	duration		
	fps		
	frames		
	location		
	weather		
	time		
event	eventname		
	starttime		
	duration		
object	name		
	position	keyframe	
		keypoint	x
			y
	action	actiontime	
		frame	start
			end

그림 3. 크로마키 영상 처리 예시
Fig. 3. Sample for Processed Chroma Key Images

그림 4. 데이터 라벨링 예시(Yolo_mark 실행 화면 캡처)
Fig. 4. Sample for Data Labelling (from a Screenshot in Yolo_mark)

실신 장면의 특성상 움직임 변화가 크지 않고 같은 프레임이 지속적으로 나타나므로 프레임 캡처 비율은 제공된 영상의 원본 FPS를 사용하지 않고 0.5FPS로 낮춰 추출하였다. 프레임 캡처 중 발생한 중복 프레임에 대해서는 라벨링 과정에서 추가로 제거하는 과정을 거쳐 제외하였다. 표 1에서는 사용한 라벨링 데이터 구조[17]를 제시한다.

2) 시각 정보 보완을 위한 크로마키 영상 처리

앞서 설명한 대로 이 연구에서 이용한 영상 데이터에는 연기자의 실신 연기 장면만 포함되어 있고 주변 환경 정보가 녹색 배경(그림 3의 오른쪽 아래 사각형)으로만 제공되어 실제 상황 영상과는 차이가 있다. 이러한 문제를 해결하기 위해 제공된 영상 중 크로마키 세트에서 촬영된 영상에서 인물을 분리한 후, 보다 사실적인 상황에서의 영상 정보로 증강하기 위해 네이버 지도 서비스[18]에서 캡처한 임의의 거리뷰 영상 30장과 합성하였다. 추출한 크로마키 영상 프레임에서 1차적으로 배경을 제거한 후, Adobe Photoshop batch 스크립트 코딩을 통한 처리로 인물 레이어와 배경 레이어를 합성했다. 합성 과정에서 인물은 임의로 위치와 좌우대칭, 크기, 각도를 변경해 배치했으며, Adobe Photoshop의 Match Color 기능을 사용해 인물 레이어의 색상, 채도, 명도를 각 배경의 색조와 일치하도록 재조정했다.

추출된 프레임 중에서 중복된 프레임을 제거하는 과정을 거쳐 결과적으로 인물 이미지 11장을 사용해 총 330장을 학습 데이터로 생성하였다.

3) 라벨링

YOLO 네트워크는 학습에 오브젝트 정보와 좌표 정보가 라벨링된 이미지를 사용한다. 총 5개 컬럼으로 작성되어 있으며, 오브젝트 클래스 번호, 오브젝트 바운딩 박스 정보(width, height, x 좌표, y 좌표)로 구성되어 있다. 이미지는 복수의 오브젝트를 포함할 수 있으며 이 경우 여러 행으로 작성된다. 본 연구에서 사용하는 학습용 이미지에는 프레임당 하나의 오브젝트만 포함되어 있으므로 각 라벨링 데이터는 1개 행, 5개 열로 구성되어 있다.

라벨링에는 YOLO 네트워크 학습용 데이터 라벨링 도구로 공개 및 배포되고 있는 GUI 도구인 Yolo_mark[19]를 사용했다. 클래스 정의 파일에 1개의 클래스('Swoon')를 작성한 후 추출한 프레임을 Yolo_mark에 로드해 실신 인물에 해당하는 영역에 바운딩 박스를 생성했다(그림 4). 라벨링 완료 후 라벨링 데이터와 프레임은 같은 경로에 저장한 후 shell 스크립트를 사용해 경로 목록을 생성해 학습 데이터 정의 파일에 삽입하였다.

4) 학습 데이터셋 구성

추출한 이미지를 바탕으로 표 2와 같이 학습 데이터셋을 구성하였다. 크로마키 영상을 통해 합성한 이미지는 트레이닝셋에만 포함시켰으며 테스트셋에는 포함하지 않았다. 학습 과정에서 데이터 증강(data augmentation)을 진행할 예정이므로 데이터셋 구성 과정에서 별도의 데이터 증강 과정은 거치지 않고 원본 프레임을 사용했다.

트레이닝셋과 테스트셋의 비율은 과적합을 피하기 위해 보편적으로 전체 데이터에서 70~80% 비율을 트레이닝셋으로 구성한다. 본 연구에서는 크로마키 이미지를 통해 생성한 330장을 제외한 데이터를 먼저 Python 스크립트를 사용해 7:3 비율로 샘플링한 후(트레이닝 셋 651장, 테스트 셋 285장), 크로마키 이미지를 통해 별도로 생성한 데이터를 트레이닝셋에 추가하는 방식으로 데이터셋을 구성했다. 최종 학습 데이터셋은 표 2의 구성과 같다. 결과적으로 확보한 프레임 중 77%를 트레이닝 셋으로 사용하였다.

2-3 학습 결과

1) 학습 환경

Ubuntu Linux에서 docker 환경을 구성하고 이 환경 내에서 시스템을 구현하였다. 실시간으로 이미지 입출력을 하기 위해 OpenCV를 사용해 코드를 구현하였으며 YOLO 네트워크용 프레임워크인 Darknet을 멀티 GPU 환경에서 구동하였다.

2) 모델 학습

YOLO 공식 홈페이지에서 배포하는 YOLOv4 weight를 기반으로 2-1절에서 서술한 방식으로 구성한 거리 풍경과 실신 모습을 합성한 데이터를 이용해서 총 500,200회 단일 클래스 커스텀 학습을 수행했으며 학습 수행 20,000회마다 weight를 백업하였다. 학습 수행 20,000회마다 weight를 백업하도록 설정했으며, 500,200회까지 수행한 후 최종 weight를 저장하고 학습을 종료한다. 또한 다양한 시간대와 계절에 따른 밝기와 색조에 대응하기 위해 saturation, exposure, hue 옵션을 적용해 데이터 강화(data augmentation)를 적용하였다. 학습 과정에서 사용한 주요 매개변수는 표 3에서 정리한다.

표 2. 데이터셋 구성
Table 2. Configuration for Data Set

Size of Training Set		Size of Test Set	Total
981		285	1,266
original	651		
chromakey	330		

표 3. 주요 학습 매개변수
Table 3. Important Training Parameters

Parameters	Values
batch	64
subdivisions	32
width	960
height	640
channels	3
momentum	0.9
decay	0.0005
angle	0
saturation	1.5
exposure	1.5
hue	0.1

Classification (prediction)

Ground truth (actual classes)			
	TP	FN	P=TP+FN
	FP	TN	N=FP+TN
	P'=TP+FP	N'=FN+TN	Total=P+N=P'+N'

그림 5. 혼동행렬
Fig. 5. Confusion Matrix

Ⅲ. 실험

학습 종료 후, 학습 과정에서 백업한 각 weight에 대해 precision, recall, IoU(Intersection of Union), mAP(mean Average Precision) 4가지 지표를 통해 성능을 비교하였다.

모델의 성능 평가는 실제 값(ground truth)과 모델을 통해 예측된 값(prediction)의 비교를 통해 이루어진다. 이를 나타낸 것이 혼동행렬(confusion matrix, 그림 5)이며, TP(True positive)와 TN(True negative)의 비율이 높을수록 모델의 정확도가 높다. FP(false positive)와 FN(false negative)은 각각 type 1 error, type 2 error로, 모델이 예측한 결과가 실제 값과 다름을 의미한다.

컴퓨터 비전에서는 모델 성능 평가를 위해 Precision(정밀

도, 수식 1), Recall(재현율, 수식 2)을 바탕으로 AP(Average Precision, 평균정밀도) 값을 산출한다. 예측 결과는 IoU(Intersection of Union, 수식 3) 값을 임계값으로 하며, 보편적으로 0.5 또는 0.75를 사용한다.

AP는 0.0, 0.1, … 1.0인 Recall 값에 대해 각각 얻을 수 있는 최대 Precision 값의 평균으로, 0에서 1 사이의 값을 가진다. mAP(mean Average Precision)는 앞에서 얻은 AP 값들의 평균이다. mAP 산출은 다소 복잡하지만, 모델의 성능을 단일 값으로 비교할 수 있도록 한다 [20].

$$\text{Presicion} = \frac{\text{TP}}{\text{TP}+\text{FP}} = \frac{\text{TP}}{\text{predictions}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} = \frac{\text{TP}}{\text{ground truths}} \quad (2)$$

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (3)$$

$$AP = \sum_{k=0}^{k=n-1} [Recall(k) - Recall(k+1)] * Precision(k) \quad (4)$$
$$Recall(n) = 0, Precision(n) = 1$$
$$n = Nmber of thresholds$$

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (5)$$

$$AP_k = the\ AP of class\ k$$
$$n = the\ number of classes$$

성능 지표 연산은 darknet에 포함되어 있는 mAP 연산 명령어인 `$ darknet detector map`을 사용해 각 체크포인트에서 백업된 weight에 성능 지표 연산을 수행했다. 본 연구에서 mAP 산출에 사용한 IoU 임계값은 0.5이다.

표 4와 그림 6에서는 산출된 성능 지표를 제시한다. 이 그림에서 가로축은 학습 수행 횟수이며, 눈금 한 칸의 크기는 20,000 epoch를 나타낸다.

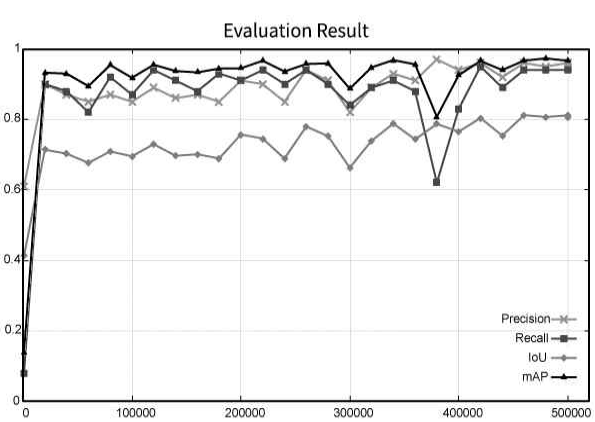

그림 6. 모델 성능 평가 결과
Fig. 6. Evaluation Result

그림 7. 제안하는 방식으로 비디오 스트림에서 검출된 긴급 상황
Fig. 7. Detected Emergency in Video Stream with Our Method

성능 평가 지표를 바탕으로, 최종 학습 결과를 모델에 적용했다. 표 4를 통해 각 iteration에 해당하는 구체적인 성능 지표 값을 확인할 수 있다.

학습에는 총 65시간 30분이 소요되었다.

그림 7은 학습한 모델이 영상에서 객체를 탐지한 결과와 그에 대한 바운딩 박스를 시각화한 것이다. 그림 7 상단과 같이 실신 인물의 일부가 가려진 경우에도 정상적으로 검출해 내고 있는 것을 확인할 수 있다.

Ⅳ. 적용

실신 감지 모듈은 Python으로 작성했으며, 앞에서 학습한 weight를 로드한 후 영상을 입력받아 프레임 단위로 오브젝트 탐지를 수행한다.

입력은 OpenCV를 통해 영상 스트림으로 주어진다. 입력된 프레임에서 오브젝트 탐지를 수행한 후, 탐지 결과에서 실신자가 탐지되는 경우 duration을 카운트하며, 연속된 프레임에서 5초 이상 실신자가 탐지되는 경우에는 응급상황으로 감지한다.

표 4. 모델 성능 평가 결과
Table 4. Evaluation

iteration	precision	recall	Iou	mAP
800	0.61	0.08	0.4127	0.139382
20000	0.9	0.9	0.7139	0.932216
40000	0.87	0.88	0.7025	0.92986
60000	0.85	0.82	0.6768	0.894376
80000	0.87	0.92	0.709	0.953961
100000	0.85	0.87	0.6951	0.918128
120000	0.89	0.94	0.7293	0.95485
140000	0.86	0.91	0.6967	0.936966
160000	0.87	0.88	0.7011	0.933121
180000	0.85	0.93	0.6891	0.944179
200000	0.91	0.91	0.757	0.944822
220000	0.9	0.94	0.7456	0.966287
240000	0.85	0.9	0.6887	0.935802
260000	0.94	0.94	0.7799	0.957075
280000	0.91	0.9	0.7523	0.95913
300000	0.82	0.84	0.6629	0.886985
320000	0.89	0.89	0.7392	0.947542
340000	0.93	0.91	0.7879	0.968043
360000	0.91	0.88	0.7433	0.95602
380000	0.97	0.62	0.7868	0.806952
400000	0.94	0.83	0.7651	0.925723
420000	0.96	0.95	0.8039	0.965539
440000	0.92	0.89	0.7537	0.94
460000	0.96	0.94	0.812	0.966747
480000	0.95	0.94	0.8067	0.972924
500000	0.96	0.94	0.8114	0.966317
final	0.95	0.94	0.8069	0.967415

단일 프레임에서 실신자가 탐지되는지 여부가 아닌 지속시간을 고려한 이유는 오브젝트 탐지 수행 결과 false true가 발생하는 경우 의도한 응급상황 탐지 및 알림을 수행하지 못했기 때문이다. 응급상황의 경우 실신자가 스스로 조치를 취할 수 없거나 타인의 도움을 받지 못하는 상황에 해당하므로 실신 상태가 유지되는지 여부로 응급상황을 판단했다. 지속시간 5초는 임의로 지정한 시간이며, 관련 연구나 통계에 의해 적정한 지속시간이 제시되어야 할 것으로 보인다.

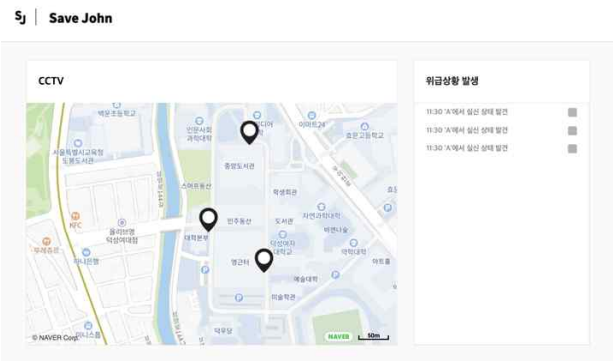

그림 8. 구현된 시각화 웹페이지
Fig. 8. Implemented Visualization Web Page

프레임에서 오브젝트 탐지를 수행하여 실신자가 탐지되는 경우 모델은 class 정보와 탐지에 대한 probability, bounding box 정보가 반환한다. (시각화 모듈을 사용하는 경우) 입력된 프레임에 대해 반환된 추정 정보에 따라, 실시간으로 검출된 오브젝트에 대한 bounding box를 시각화하며 응급상황 발생 시각과 영상 정보를 DB에 저장하도록 했다.

시각화 페이지는 실신 탐지 현황을 실시간으로 업데이트하며, 웹브라우저를 통해 이용할 수 있도록 했다(그림 8). Node.js로 서버를 구성했으며, DB는 MongoDB를 사용했다. 시각화 페이지는 위치 정보를 나타내기 위한 지도 영역과 텍스트 로그 영역으로 구성했으며, 실신 탐지 정보를 저장하는 DB와 연동되어 동작한다.

Ⅴ. 결 론

본 연구에서는 웹페이지를 통한 시각화만을 제공하고 있지만, 실제 알림이나 메시지 전송 등을 통해 직접적인 신고까지 무인화할 수 있도록 개선하고 시스템을 마련한다면 행인 등과 같은 목격자가 없는 상황에서도 적절한 시기에 구조 및 개입이 이루어질 수 있으며 결과적으로 추가적인 피해를 막을 수 있을 것으로 생각된다. 또한, 본 연구에서는 노상에 설치된 CCTV를 가정하였으나 카메라 설치 장소에 따라 병원이나 요양원 등 실신 가능성이 큰 인원이 다수 밀집한 시설에서 응급상황을 보다 빠르게 인지하고 대응하는 용도로 활용할 수 있을 것이다.

학습 데이터 변화 혹은 추가를 통해, 본 연구에서 살펴본 실신 상황 외에도 다양한 상황에 적용할 수 있다. 따라서 다양한 분야와 상황에 대해 서비스 확장이 유연하다. 예를 들어 절도, 도난 상황 등을 학습해 무인 판매 시설 등에서 활용하는 등의 활용이 가능할 것이다.

실시간으로 촬영되는 영상을 기반으로 컴퓨터 비전을 통해 실신 상황 탐지를 기반으로 응급상황을 인식하고, 신고자의 역할을 대체함으로써 CCTV가 좀 더 능동적으로 사회의 안전을 위해 활용되기를 기대한다.

감사의 글

본 연구는 한국연구재단을 통해 과학기술정보통신부의 기초연구사업으로부터 지원받아 수행되었습니다
(과제번호- 2021R1A4A502890711).

참고문헌

[1] Korea Ministry of Health and Welfare (MOHW), “Fundamental Plans for Medical Emergency Response from 2018 to 2022”, 2018

[2] YOLO: Real-Time Object Detection[Internet]. Available: https://pjreddie.com/darknet/yolo/

[3] Redmon, Joseph and Farhadi, Ali, “YOLOv3: An Incremental Improvement”, arXiv:1804.02767, 2018

[4] Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). “SlowFast Networks for Video Recognition,” in IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6201–6210, 2019, https://doi.org/10.1109/iccv.2019.00630

[5] Choi, H., Moon, G., Chang, J. Y., & Lee, K. M. (2021). “Beyond Static Features for Temporally Consistent 3D Human Pose and Shape From a Video,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1964-1973, 2021, https://doi.org/10.1109/cvpr46437.2021.00200

[6] Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., & Bengio, Y. “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 2014, https://doi.org/10.3115/v1/d14-1179

[7] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. “BlazePose: On-device Real-time Body Pose tracking”. in CVPR Workshop on Computer Vision for Augmented and Virtual Reality, Seattle, WA, USA, 2020.

[8] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D., “Backpropagation Applied to Handwritten Zip Code Recognition,” Neural Computation, Vol. 1, No. 4, pp. 541–551, 1989, https://doi.org/10.1162/neco.1989.1.4.541

[9] Hochreiter, Sepp & Schmidhuber, Jürgen. “Long Short-term Memory,” Neural computation. Vol 9. No. 8, pp. 1735-1780. 1997., https://doi.org/10.1162/neco.1997.9.8.1735

[10] Martinez, J., Black, M. J., & Romero, J., “On Human Motion Prediction Using Recurrent Neural Networks,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4674–4683, 2017, https://doi.org/10.1109/cvpr.2017.497

[11] Pavllo, D., Grangier, D., & Auli, M., “QuaterNet: A Quaternion-based Recurrent Model for Human Motion,” ArXiv:1805.06485 [Cs], 2018

[12] Dabral, R., Gundavarapu, N. B., Mitra, R., Sharma, A., Ramakrishnan, G., & Jain, A., “Multi-Person 3D Human Pose Estimation from Monocular Images,” in International Conference on 3D Vision (3DV), pp. 405–414. 2019, https://doi.org/10.1109/3dv.2019.00052

[13] He, K., Zhang, X., Ren, S., & Sun, J., “Deep Residual Learning for Image Recognition,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016, https://doi.org/10.1109/cvpr.2016.90

[14] Newell, A., Yang, K., & Deng, J., “Stacked Hourglass Networks for Human Pose Estimation,” In Computer Vision – ECCV 2016, pp. 483–499, 2016., https://doi.org/10.1007/978-3-319-46484-8_29

[15] Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., & Black, M. J., “Expressive Body Capture: 3D Hands, Face, and Body from a Single Image,“ in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10967-10977, 2019, https://doi.org/10.1109/cvpr.2019.01123

[16] Yolo v4, v3 and v2 for Windows and Linux-How to improve object detection[Internet]. Available:https://github.com/AlexeyAB/darknet#how-to-improve-object-detection

[17] Data structure for abnormal behaviors in CCTV videdo (in xml format)[Internet]. Available:https://aihub.or.kr/aidata/139

[18] Naver map service[internet]. Available: https://map.naver.com

[19] Yolo mark[internet]. Available: github.com/Alexey-AB/Yolo_mark

[20] mAP (mean Average Precision) for Object Detection[Internet]. Available:https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173 https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

송인서(Inseo Song)

2021년 : 덕성여자대학교 IT미디어공학과 (공학사)
2021년 2월~2021년 6월 : 덕성여자대학교 IT미디어공학과 연구원
2021년 7월~12월 : Naver CLOVA 인턴

※관심분야 : HCI, 영상분석, 인공지능

박태정(Taejung Park)

1997년 : 서울대 전기공학부 (공학사)
1999년 : 서울대 전기공학부 대학원 (공학 석사, 반도체 물리 전공)
2006년 : 서울대 전기컴퓨터공학부 대학원 (공학박사, 컴퓨터 그래픽스 전공)

2018년~현 재 : 덕성여자대학교 공과대학 사이버보안/IT미디어공학과 부교수
2013년~2017년 : 덕성여자대학교 정보미디어대학 디지털미디어학과 조교수
2006년~2013년 : 고려대학교 연구교수
※관심분야 : 컴퓨터그래픽스, 병렬처리, 인공지능, 수치해석, 3차원 모델링