

횡단보도 보행자 인지 향상을 위한 ByteTrack 기반 다중객체추적

노유진, 이유진, 이상민[†]
광운대학교 소프트웨어융합대학

ByteTrack-Based Multi-Object Tracking for Recognition of Pedestrian in Crosswalk

Yujin Roh, Yujin Lee, Sangmin Lee[†]
College of Software and Convergence, Kwangwoon University

According to traffic accident statistics, 42% of traffic accidents between 2017 and 2021 were crossing accidents. To reduce these accidents, the research on the collision avoidance assistance for pedestrian safety is in the spotlight in machine learning field. However, there are few studies to improve the accuracy of recognizing the crossing multiple pedestrians and estimating the residual time to the crossing. We here propose using hybrid approach combining Yolo-X and ByteTrack to obtain the accurate detector of pedestrians. We gathered the actual data set of the pedestrian crossing for the accurate multi-pedestrian tracking model. The proposed approach can detect and track the crossing pedestrians in real time. Furthermore, we can estimate the pedestrian's walking direction, speed, acceleration, and distance to the crosswalk for inferring the motion of pedestrians entering the crosswalk. The experimental results present that the proposed approach outperforms the alternatives in terms of tracking accuracy and computation cost.

Keywords: Pedestrian Recognition, Multi-Object Tracking, Autonomous Vehicle, ByteTrack, CDNet

논문접수일 : 2022.10.08.

심사완료일 : 2022.12.26.

게재확정일 : 2022.12.27.

[†] Corresponding Author: smlee5679@gmail.com

1. 서론

도시 물류에서 자동화 물류 개념을 도입할 경우 물류시스템의 효율성 못지않게 중요한 가치로 보행자의 안전과 편의 증진을 들 수 있다. 이와 더불어 자율주행의 발전은 물류산업 영역에 적용되어 물류시스템의 효율성과 안전성 연구에 기여하고 있다.(김도영 et al., 2018; 김태선 et al., 2021; 박상미와 김정우, 2020; 웨이치 et al., 2018). 교통사고 분석시스템에서 발표한 최근 5개년 통계 보고에 따르면 2017년부터 2021년까지 전체 차대 사람 간의 교통사고 209,616건 중 횡단 중 보행자 충돌사고 건은 87,786건으로 42%를 차지했다(Traffic Accident Analysis System., 2021). 그중 사망자 수는 3,385명, 부상자 수는 90,045명으로 횡단 중 인 보행자에 대해 대응할 수 있는 충돌방지보조(collision avoidance assistance: CAA) 시스템에 관한 연구가 필요하다. 이러한 안전사고를 방지하기 위해 차량에는 다양한 센서들이 장착되고 있으며, CAA 시스템에 관한 연구가 활발하게 진행되고 있다(Akhauri et al., 2020; Althoff et al., 2009; Cho et al., 2019; Chu et al., 2012; Grigorescu et al., 2020; Noh and Han, 2014; Park et al., 2018; Shalev-Shwartz et al., 2016). 최근 각광받고 있는 자율주행 자동차에서는 이러한 CAA 시스템이 필수적이며 카메라 센서, 레이다(radio detection and ranging: RADAR) 센서, 라이다(light detection and ranging: LIDAR) 센서 등을 이용하고 있다. 하지만 RADAR는 작은 크기의 물체 식별이 어려우며, 또한 LIDAR는 원거리 측정이 어렵다는 문제점을 지닌다(Lee et al., 2021; Jin et al., 2018). 많은 CAA 시스템의 발전에도 불구하고 RADAR, LIDAR 센서에 의존적인 경향이 있는 CAA 시스템은 두 센서의 문제점을 가지게 된다. 이 문제점을 해결하기 위해 본 논문에서는 작은 크기의 물체 식별과 종합적인 상황 인식에 적합한 카메라 센서를 이용하여 다중객체 추적(multi object tracking: MOT) 기반으로 두 문제점을 해결하고자 한다.

본 논문은 횡단보도와 횡단보도 인근 보행자를 탐지하여 1) 횡단하려는 다중 보행자를 정확하게 인식하고, 2) 다중 보행자 중 횡단보도까지의 잔여 시간을 추정하는 알고리즘을 개발한다. 자율주행차량의 운행에 있어 이 두 가지 문제를 풀어야 교차로 상황에서 갑자기 뛰어들거나 RADAR 센서 사각지대에 놓여진 보행자에 대한 인식율을 높일 수 있다. 특히, 정확도 성능을 높이기 위해서 제안 기법은 객체 탐지 부문과 객체 추적 부문에 있어서 가장 성능이 높게 나온 두 가지 최신 기법을 활용하였다. 횡단보도 인식에는 도로 교통 객체탐지에 특화된 CDNet(crosswalk detection network) 모델(Zhang Z, et al., 2022)을 활용하고, 인근 보행자의 움직임을 정확히 추적하기 위해 MOT에 특화된 ByteTrack(Zhang Y, et al., 2021)을 학습시켜 활용한다. 제안하는 알고리즘은 2단계로 구성된다. 첫째, 카메라 센서로부터 수집된 데이터인 동영상 데이터를 인공지능 알고리즘이 학습하기 위해 전처리하고 MOT 알고리즘인 ByteTrack을 통해 보행자를 추적한다. 둘째, 보행자의 진행방향, 속도, 가속도, 보행자와 횡단보도 간 거리를 계산한다. 추정된 수치를 이용해서 보행자의 진행방향이 횡단보도를 향하고, 횡단보도까지 남은 거리가 계속 줄어들며 가속도가 감소하지 않는다면 횡단할 가능성이 있는 보행자로 추정한다. 이후 탐지된 보행자 중 가장 먼저 횡단보도에 도착할 보행자를 탐지한 후 해당 보행자의 보행 속도를 반영하여 횡단보도까지의 도착시간을 예측한다. 본 논문에서 제안하는 알고리즘을 통해 보행자의 횡단보도 진입시간을 유추할 수 있으며, 이는 횡단 중 차량과 보행자 간의 충돌사고 빈도를 최소화와 더불어 보행자의 충돌에 대응하는 자율주행 CAA 시스템의 기반기술이 되리라 기대한다.

본 논문의 구성은 다음과 같다. 2장은 본 연구의 관련 연구 및 선행 연구들에 대해 기술하였으며, 3장은 본 연구를 수행하기 위한 알고리즘에 대해 기술한다. 4장은 제안하는 알고리즘의 성능지표와 연구 결과에 대해 기술하고, 5장에서는 연구 결과에 대한 분석과 그 의미에 대해 기술한다. 마지막으로 6장에서 결론 및 향후 연구과제에 대해 기술한다.

2. 선행연구

Figure 1에서 볼 수 있듯 2017년부터 2021년까지 보행자 횡단 중 차량에 의한 충돌사고가 매해 약 20,000 건가량 발생하고 있다. 또한, 매년 약 15,000명 이상이 횡단 중 교통 사상사고를 당하고 있다(Traffic Accident Analysis System., 2021). 이러한 교통사고를 줄이기 위해 횡단 중인 보행자에 대해 대응할 수 있는 충돌방지보조(collision avoidance assistance: CAA) 시스템에 관한 연구가 활발히 이루어지고 있으며, CAA 시스템은 차량에 부착된 여러 센서 데이터들을 수집하고 처리하여 교통사고와 같은 충돌을 미연에 방지하도록 보조한다(Akhauri et al., 2020; Althoff et al., 2009; Cho et al., 2019; Chu et al., 2012; Grigorescu et al., 2020; Noh and Han, 2014; Park et al., 2018; Shalev-Shwartz et al., 2016).

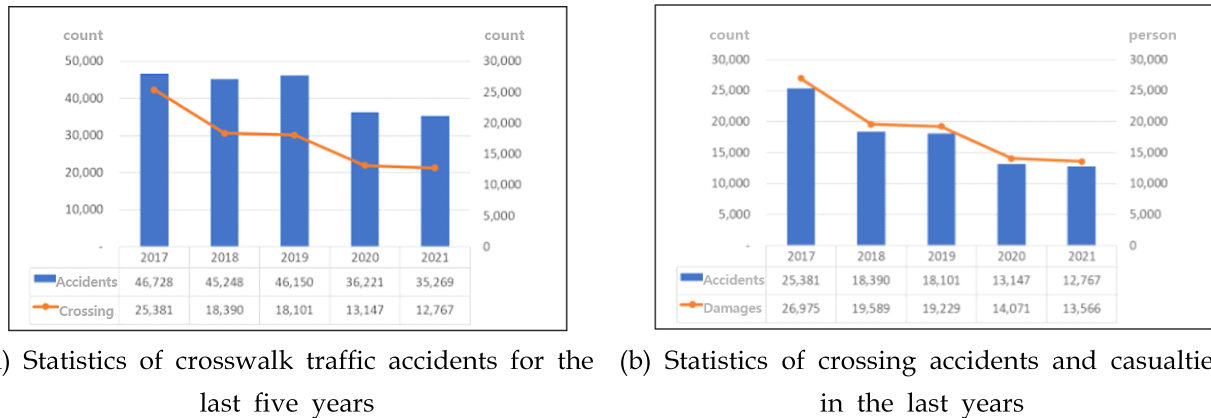


Figure 1. Statistics of traffic accidents in South Korea between 2017 and 2021
(Traffic Accident Analysis System, 2022)

최근에는 정보기술의 발달과 센서들의 정밀도가 크게 향상되었으며, 다수 센서 데이터를 활용하는 기술 개발이 많은 관심을 받고 있다. 또한, 이 데이터들을 효율적으로 분석 및 처리하기 위해 다양한 인공지능 기술들이 연구되고 있다(Almalioglu et al., 2022; Al-Qizwini et al., 2017; Farag et al., 2018; Kwon et al., 2021; Li et al., 2021; Okuyama et al., 2018; Pak A et al., 2022; Prabhakar et al., 2017). 특히, 이미지 데이터에 특화된 합성곱신경망(convolution neural networks: CNNs)은 고사양 컴퓨팅 자원의 등장으로 인해 의료, 산업, 자율주행 등의 분야에서 두드러지는 성과를 보이고 있다(He et al., 2016; Krizhevsky et al., 2017; LeCun et al., 1998; Simonyan et al., 2014; Szegedy et al., 2015). CNNs 기반 이미지 인식 분야에서는 분류(classification), 객체분할(object segmentation), 객체탐지(object detection), 다중객체추적(multi object tracking: MOT) 등으로 구분된다(Girshick, 2014; Redmon et al., 2016). 이 중 CAA 시스템을 포함한 자율주행 자동차 기술에서는 주로 MOT를 활용한 연구가 진행 중이다.

MOT는 동영상에서 객체탐지 및 객체분할과 동시에 인식하고자 하는 객체가 움직이는 경로상 이전 프레임(frame)에서 탐지된 객체와 동일 객체인지 여부를 인식하는 추적 기술을 지칭한다. 최근까지도 MOT를 수행하기 위한 많은 알고리즘들이 개발되고 있는데, 대부분 임계값 보다 높은 탐지박스와 경로조각(tracklet)을 연결하여 동일 객체를 추적하게 된다. 객체추적의 정확도를 높이기 위해서는 조명 및 조도(brightness and contrast) 변화 문제 외 객체 간 중첩(occlusion)되거나 자체 중첩(self-occlusion)되는 경우가려진 객체에 대한 경계박스(bounding box) 추정이 어려워지는 문제 등을 해결할 필요가 있다(Bochinski et al., 2017; Wang et al., 2020; Xu et al., 2022; Zeng et al., 2021; Zhanget al., 2021).

최근 제안된 ByteTrack은 기존 MOT 알고리즘에서 발생하는 가려진 물체의 경계박스를 제거하는 문제를 tracklet의 유사점을 활용하여 높은 경계박스뿐만 아니라 낮은 경계박스도 연결하여 객체를 추적하는 방법으로 해결했다(Zhang Y, et al., 2021). Tracklet는 객체의 ID 값을 유지하는데 구하는 짧은 구간에서의 경

로를 의미한다. ByteTrack은 YOLO(You Only Look Once) 기반의 YOLO-X 아키텍처를 사용하였다. YOLO는 객체탐지를 수행하는 인공지능 알고리즘 중의 하나이다. YOLO 이전의 객체탐지 알고리즘은 분류와 지역화(localization)의 두 가지 일을 순차적으로 학습하기 때문에 학습 수렴(convergence) 속도가 느린 단점이 있었다. YOLO는 기존 알고리즘들과 달리 이 두 가지 일을 동시에 처리하는 one-stage-detection 방법을 채택하여 실시간 객체탐지가 가능하게 만들었다. 따라서 YOLO는 추론 속도 면에서 빠른 실시간 처리가 필요한 자율주행에서 각광 받고 있다. YOLO는 YOLOv1, YOLOv2, YOLOv3, YOLOv4, 그리고 YOLOv5로 발전해왔다. 하지만 해당 논문은 YOLOv4, YOLOv5는 파이프라인을 anchor 위주로 최적화가 진행되어있기 때문에, 일반적인 성능이 오히려 떨어질 수 있다고 판단하여서 YOLOv3를 YOLO-X의 베이스로 삼았다.

YOLO-X는 YOLOv3를 기반으로 사용하지만, head 부분에서는 기존 YOLOv3와 달리 decoupled head를 사용한다(Ge Z, et al., 2021). 분류와 경계박스 회귀(bounding box regression)는 서로 다른 특성을 갖는다. 분류에는 완전연결레이어(fully connected layer)가 효과적이지만, 반면에 지역화에는 convolution head가 효과적이다. 따라서 기존 연구에서 이러한 head 부분을 double-head 방식으로 변경하여 분류에는 완전연결 head를, 지역화에는 convolution head를 적용함으로써 성능을 향상시켰다. 분류에는 이진교차엔트로피(binary cross entropy) 손실함수(loss function)를 사용하고 지역화에는 IoU(intersection over union) 손실함수를 사용하여 학습을 진행한다. 실험 결과, 해당 방식이 기존 coupled head 방식보다 학습 수렴 속도가 빠르고 AP(average precision)가 향상되었다고 한다. 또한, YOLOv3-ultralytics 알고리즘의 성능(AP: 44.3)보다 YOLO-X의 성능(AP: 51.2)이 향상된 성능을 보여준다.

더불어 자율주행을 위해 실시간 횡단보도 검출 알고리즘에 대한 연구 또한 활발히 진행되고 있다(Chen et al., 2019; Uddin and Shioyama., 2005; Zhang Z, et al., 2022). 최근 제안된 CDNet(crosswalk detection network)은 자동차에 카메라 센서와 Jetson nano device를 부착하고, 카메라 센서로부터 수집된 데이터를 Jetson nano device에서 실시간으로 횡단보도를 검출하기 위한 프레임워크를 제안한다(Zhang Z, et al., 2022). CDNet은 실시간 횡단보도 검출을 위해 YOLOv5 모델을 기반으로 실제 기상환경(구름, 맑음, 비, 안개 및 밤)과 같은 실제 복잡한 시나리오에서 적용하기 위해 다양한 데이터 증강 방법이 적용되었다. CDNet은 Jetson nano device에서도 33.1 fps로 처리 속도가 빠르고 F1 score 94.83%의 높은 성능을 보였다.

기존연구에서는 보행자의 인식 및 보행자 횡단 예측이 자율주행에서 중요한 과제로 인식되었으며, 다양한 연구가 진행되고 있다. 하지만 보행자 신호가 있는 횡단보도 또는 단순히 보행자의 횡단 유무를 추정하는 연구에 그쳤다. 이는 무단횡단이나 보행자 신호등이 없는 경우 적용이 어려우며 단순히 횡단 유무를 예측하는 것은 보행자가 횡단보도까지 도달하는 시간을 알기 어렵기 때문에 차량이 보행자보다 먼저 횡단 가능한지 여부를 알기 어렵다. 이는 교통 흐름을 방해하는 요소가 된다. 최근 차량이 우회전시 보행자가 횡단 여부가 있다면 무조건 정지 해야 하는 법규가 시행되었다. 하지만 보행자의 횡단 예측 어려움으로 운전자가 보행자 신호 동안 정지해서 출발지연이 발생하는 교통 흐름 방해 문제가 야기되었다. 따라서 무신호 횡단보도에서 실시간으로 보행자를 인지 및 추적해서 보행자 횡단시간을 예측할 수 있는 연구가 필요하다(Hashimoto et al., 2015).

본 논문에서는 자율주행의 특성에 필수요소인 실시간 처리를 위해 YOLO를 백본모델(backbone)로 사용하는 ByteTrack과 CDNet을 활용하여 횡단보도 인근의 보행자 탐지 및 횡단 예측 연구를 진행하였다. 횡단보도 검출에는 CDNet 모델과 공개되어 있는 가중치(weight)를 활용하였으며, 보행자 검출을 위해 ByteTrack 알고리즘을 학습시켰다. CDNet과 ByteTrack의 출력값(output)들을 이용해 횡단보도 인근의 보행자 탐지 및 횡단 예측 연구 방법을 제안한다.

3. 본론

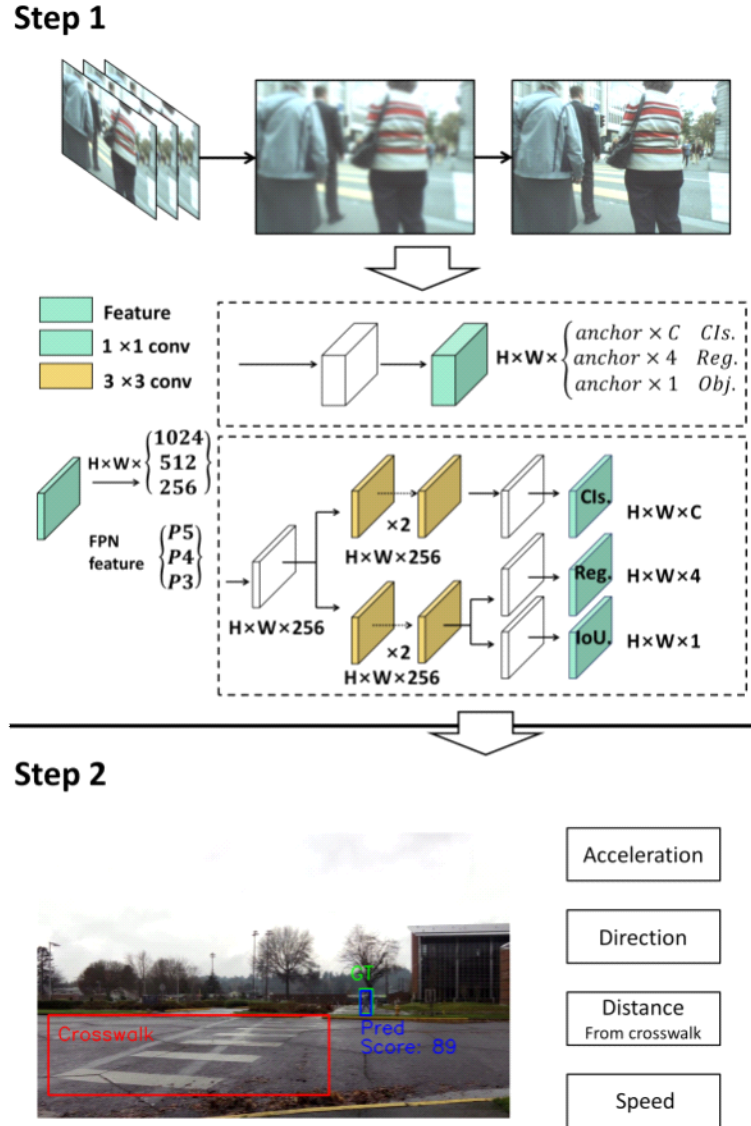


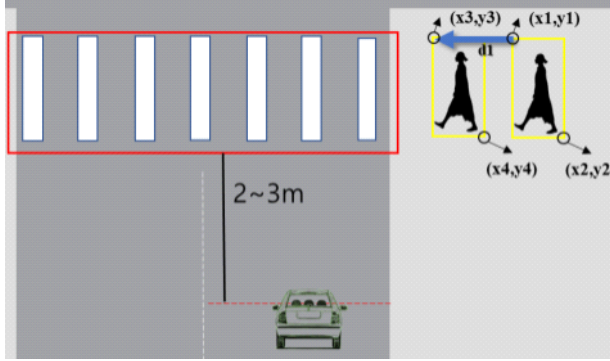
Figure 2. Overall procedure of the proposed approach

3장에서는 횡단보도 인근 보행자 탐지 및 횡단 예측 알고리즘에 대해 기술한다. 제안하는 알고리즘은 Figure 2와 같이 도식화할 수 있으며 총 2단계로 구성된다. 첫째, 카메라 센서로부터 수집된 데이터인 동영상 데이터를 인공지능 알고리즘이 학습할 수 있도록 전처리하는 과정이다. 전처리기(pre-processor)에서는 동영상을 단일 이미지로 만들고 이미지 강조 처리(image enhancement)를 한다. 이후 ByteTrack을 통해 보행자를 인식 및 추적한다. 두 번째 단계에서는 추정된 보행자 객체를 인지하여 진행방향, 속도, 가속도, 보행자와 횡단보도 간 거리를 계산한다.

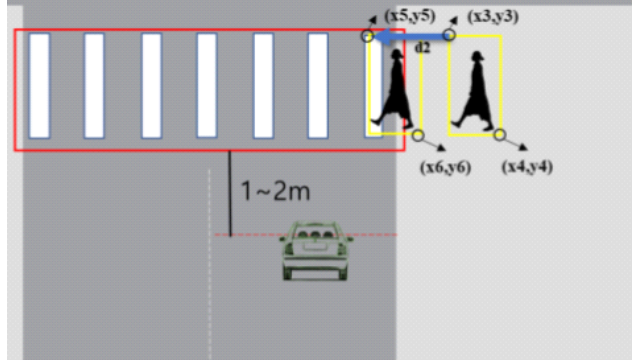
첫 번째 단계에서는 인공지능 알고리즘에 입력하기 위한 데이터를 전처리한다. 데이터 전처리에는 동영상을 프레임(frame) 단위로 추출하여 단일 이미지로 만들고, 이미지의 밝기(brightness)와 대비(contrast), 선명화(sharpening) 등의 처리를 거쳐 특징을 강조한다. ByteTrack을 통해 실시간으로 보행자를 추적한다. ByteTrack 알고리즘에 보행자를 학습(fine-tuning)하였다. 또한, ByteTrack 알고리즘의 강건한 예측성능을 확보하기 위해 데이터증강(data augmentation) 기법을 적용하여 학습데이터를 구성하였다. 이미지에 대한 데이터증강 시 좌우반전(horizontal flip), 회전(rotation), 이동(translation) 등의 실제 데이터와의 이질성

(heterogeneity)이 적은 증강 기법만을 사용하였다.

각 프레임은 초당 30장이 수집된다고 가정할 때, 우선 ByteTrack으로 탐지한 보행자의 bounding box로부터 추적하며 이동 방향 및 속도에 대한 정확한 추정을 위해 초기 프레임 f_0 과 τ 시점 이후의 두번째 프레임 $f_{+\tau}$ 을 샘플링한다.



(a) An example schematically illustrating a method of the proposed system. It detects crosswalks (red box) and navigates pedestrians towards crosswalks (yellow box). Additionally, the proposed system performs object tracking on a frame-by-frame basis to identify movement for the pedestrian.



(b) An example illustrating a method of estimating the remaining time until the pedestrians arrive at a crosswalk. The speed of the pedestrian and the remaining time of arrival at the crosswalk are estimated according to Equations (1) to (9).

Figure 3. Scenarios of the proposed system to detect the crosswalk, track the pedestrians, and estimate the remaining time to the crosswalk.

두 번째 단계에서는 보행자의 속도, 가속도, 진행방향, 횡단보도와 남은 거리를 계산한다. Figure 3은 첫 번째 단계에서 탐지한 결과의 예를 도식화한 것이다. Figure 3 (a)의 가장 우측에 있는 경계박스는 첫 번째 프레임에서 탐지된 보행자이며, 좌측 경계박스는 두 번째 프레임에서 탐지된 보행자를 의미한다. Figure 3 (b)의 경계박스는 두 번째 프레임에서 탐지된 보행자(우측)와 세 번째 프레임에서 탐지된 보행자(좌측)를 표시한 것이다.

보행자의 진행방향, 속도, 가속도, 이동거리, 횡단보도와 보행자 간의 거리를 계산하기 위해 경계박스의 좌표를 이용한다. Figure 3 (a)를 예를 들면, 첫 번째 프레임에서의 보행자 경계박스의 좌표를 (x_1, y_1) , (x_2, y_2) 로 두고, 보행자가 움직인 다음 프레임에서의 좌표를 (x_3, y_3) , (x_4, y_4) 라고 가정한다. 이때, 보행자의 진행방향(\vec{v}_2), 속도(s_2), 그리고 이동거리(d_2)는 각각 식 (1)~(3)과 같다. 이동거리는 유클리디안 거리(Euclidean distance)를 이용하며, 속도는 이동거리를 시간(frame per second: fps)으로 나누어 계산한다.

$$\vec{v}_2 = \begin{cases} LEFT & (x_3 - x_1 < 0) \\ RIGHT & (x_3 - x_1 > 0) \\ FORWARD & (y_3 - y_2 > 0) \text{ and } (y_4 - y_2 < 0) \\ BACKWARD & (y_3 - y_1 < 0) \text{ and } (y_4 - y_2 > 0) \\ STOP & (x_3 - x_1 \leq \epsilon) \text{ and } (y_3 - y_1 \leq \epsilon) \end{cases} \quad (1)$$

$$d_2 = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} \quad (2)$$

$$s_2 = \frac{d_2}{\tau} \quad (3)$$

식 (2)와 (3)의 방식으로 Figure 3 (b)의 보행자 이동거리 및 속도는 (4), (5)과 같이 유도할 수 있다.

$$d_3 = \sqrt{(x_5 - x_3)^2 + (y_5 - y_3)^2} \quad (4)$$

$$s_3 = \frac{d_3 - d_2}{\tau} \quad (5)$$

위 수식들을 일반화하면 아래 식 (6)~(8)과 같다. 이때, f 는 프레임을 의미하며, $x_{(f,lu)}$, $y_{(f,rd)}$ 는 각각 해당 프레임에서의 좌측상단(left-up), 우측하단(right-down)에서의 x 와 y 좌표로 정의한다.

$$d_f = \sqrt{(x_{(f,lu)} - x_{(f-1,lu)})^2 + (y_{(f,lu)} - y_{(f-1,lu)})^2} (f \geq 1) \quad (6)$$

$$s_f = \begin{cases} \frac{d_f}{t_f} & (f = 1) \\ \frac{d_f - d_{f-1}}{t_f - t_{f-1}} & (f \geq 2) \end{cases} \quad (7)$$

$$a_f = \begin{cases} s_f & (f \leq 2) \\ s_f - s_{f-1} & (f \geq 3) \end{cases} \quad (8)$$

또한, 위 수식들의 결과를 바탕으로 보행자가 횡단보도에 도착하기까지의 잔여시간(remaining time) $t_{(f,cp)}$ 은 식 (9)와 같다. $d_{(f,cp)}$ 는 횡단보도와 보행자 간의 거리로 정의한다. 식 (3)에서 측정된 τ 시점 이후 움직인 거리 s_2 은 픽셀로 환산할 수 있는데, 예를 들어 $\tau=30(\text{frame})$ 에서 180 픽셀만큼 이동했다면, 등속도를 가정할 때 프레임 당 6 픽셀만큼 이동했음을 추정할 수 있다. 그러므로, 보행자의 위치로부터 횡단보도까지의 거리인 d_2 에 도달하는데 잔여시간 $t_{(f,cp)}$ 은 식 (9)와 같다. 예를 들어, τ 시간 내 180픽셀만큼 이동했는데 횡단보도까지 360 픽셀이 남은 경우, 잔여시간은 $\frac{360}{180}\tau$ 이다.

$$t_{(f,cp)} = \frac{d_{(f,cp)}}{d_f} \times fps \quad (f \geq 1) \quad (9)$$

4. 실험 및 결과

4.1 연구배경 및 목적

인공지능 알고리즘을 학습시키기 위해서는 고성능의 그래픽 처리 장치(graphics processing unit: GPU)가 필요하다. 본 연구에서는 3장에서 제안한 인공지능 알고리즘 학습을 위해 GPU는 TITAN Xp를 사용하였고, 소프트웨어는 python 3.8.13, openCV 4.6.0.66, torch 1.12.1를 사용하였다.

ByteTrack 알고리즘을 학습시키기 위한 데이터셋으로 MOT17, MOT20, ETHZ를 이용하였다(Dendorfer., 2020; Ess et al., 2018; Milan., 2016). 해당 데이터들은 보행자와 관련한 빅데이터로 여러 카테고리들을 분할(segmentation)하거나 탐지(detection), 추적(tracking)하는 것을 목표로 한다. 데이터셋은 총 50,426장의 이미지이며, 해상도는 640x480~1920x1080, 그리고 fps는 14~30로 이루어져 있다. 전체 데이터셋 중 43,311장을 학습(train) 데이터셋으로, 7,115장을 검증(validation) 데이터셋으로 사용하였다.

또한, 본 논문에서 제안한 횡단보도 인근에서의 보행자 검출과 보행자 상태를 계산하기 위해 추가적으로 pedestrian dataset을 사용하였다(Meschke., 2017). pedestrian dataset은 횡단보도(crosswalk), 사거리(fourway), 밤(night)에 촬영한 3개의 동영상으로 구성된다. 해상도는 1920x1080이며, 30 fps로 이루어져 있다. 3장의 전처리기를 이용하여 동영상 데이터를 횡단보도 378장, 사거리 1,659장, 그리고 밤 565장의 단일 이미지로 변환하여 사용하였다.

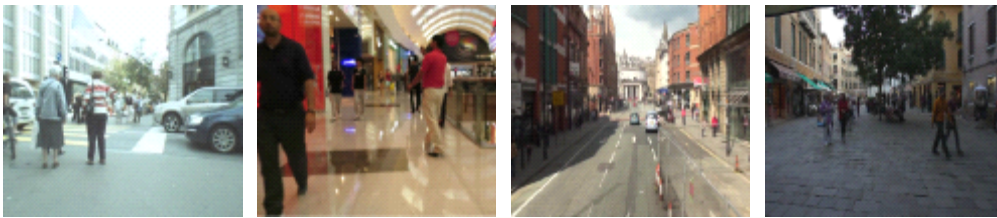


Figure 4. Sample frames of tracking people in MOT17 dataset

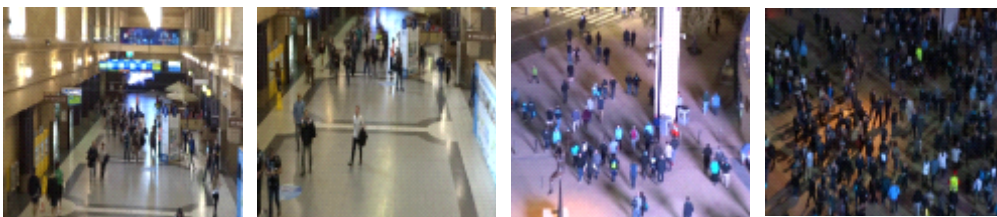


Figure 5. Sample frames of tracking people in MOT20 dataset

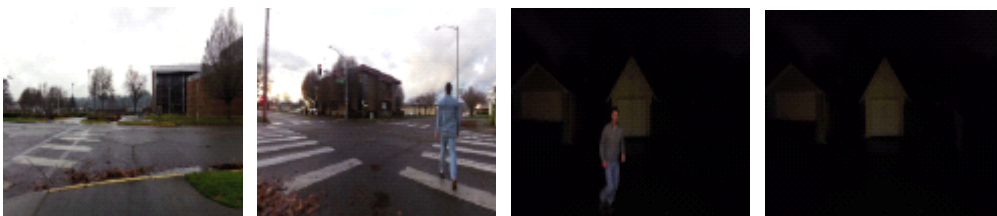


Figure 6. Sample frames in Pedestrian dataset

4.2 성능평가

1) 성능지표

본 논문에서는 성능 평가를 위한 주요 성능지표로 MOTA(multi-object tracking accuracy)와 IDF1(identification F1 score), 그리고 IDs(ID switches)를 사용하였다. MOTA는 Bernardin K et al(2006)이 MOT 알고리즘에서의 정확도를 측정하기 위해 제안한 성능지표이다. MOTA의 식은 (10)과 같다. 식 (10)에서 t 는 시간을 의미하며, m 은 인식하지 못한 객체의 개수를 의미한다. fp 와 mme 는 찾은 객체들 중 잘못된 찾은 것을 의미하며 각각 위양성(false positive: FP)과 위음성(false negative: FN)에 해당하는 개수를 의미한다. g 는 존재하는 모든 물체의 개수를 의미한다.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (10)$$

IDF1은 MOT 알고리즘이 추적하고 있는 ID가 정답군(ground truth)과 얼마나 일치하는지를 평가하기 위한 지표로 ID에 대한 F1 점수이다. IDF1은 식 (13)과 같으며, IDR와 IDP는 각각 ID에 대한 재현율(identification recall)과 정밀도(identification precision)로 식 (11), (12)와 같다.

$$IDR = \frac{IDTP}{IDTP + IDFN} \quad (11)$$

$$IDP = \frac{IDTP}{IDTP + IDFP} \quad (12)$$

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \quad (13)$$

IDs는 탐지한 결과와 실제 객체에 할당되어 있는 ID가 다른 것을 의미한다. 그 외 추가적인 성능지표로 MT(number of mostly tracked trajectories), ML(number of mostly lost trajectories), FP, 그리고 FN을 사용하여 성능을 평가하였다.

2) 결과

<Table 1> Performance evaluation of ByteTrack regarding MOT17 and MOT20 datasets. The experiments were conducted on ByteTrack without additional training phase.

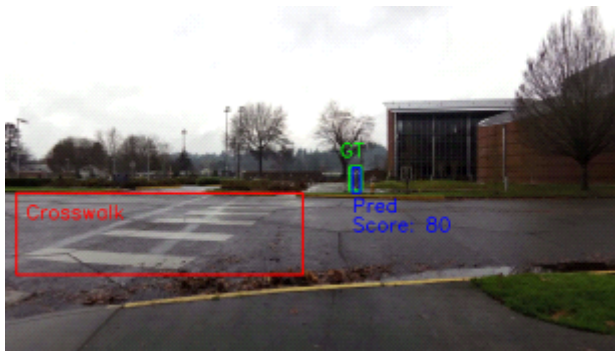
Dataset (N)	MOTA ↑	IDF1 ↑	IDs ↓	MT ↑	ML ↓	FP ↓	FN ↓
MOT17 (2,652)	75.3%	77.1%	134	186	55	2,101	11,093
MOT20 (4,463)	63.5%	66.2%	1,320	642	201	65,462	157,580

ByteTrack 알고리즘의 성능을 평가하기 위해 학습되지 않은 MOT17와 MOT20의 데이터를 이용하였다. 성능 평가 결과는 아래 <Table 1>과 같다. MOT17에서는 MOTA가 75.3%로 꽤 높은 성능을 보였으나 MOT20 데이터셋에서는 다소 떨어진 63.5%의 성능을 나타냈다. 이는 MOT17 데이터셋의 객체(objects) 개수는 53,890개로 MOT20의 615,137개에 비해 현저히 적기 때문으로 판단된다. 특히, MOT20 데이터셋은 군집되어 있는 객체가 다수 포함되어 있어 MOT17과 비교하여 상대적으로 낮은 성능을 보였다.

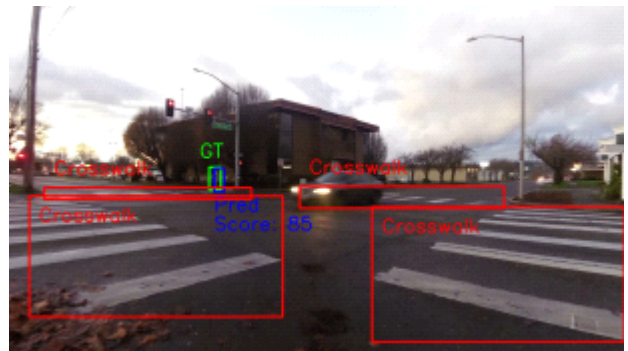
<Table 2> Performance evaluation of Pedestrian dataset
in terms of detection and tracking accuracy.

Dataset (N)	MOTA ↑	IDF1 ↑	IDs ↓	MT ↑	ML ↓	FP ↓	FN ↓
Crosswalk (378)	100.0%	100.0%	0	1	0	0	0
Fourway (1,659)	90.9%	28.8%	5	1	0	52	59
Night (565)	49.9%	55.5%	2	0	0	13	268
Overall (2,224)	82.1%	47.3%	7	2	0	65	327

<Table 2>는 Pedestrian dataset에서의 성능 평가 결과이다. 전체적인(overall) 성능은 MOTA가 82.1%로 각각의 데이터셋에서 100%, 90.9%, 49.9%의 성능을 보였다. 이 중 밤에 해당하는 데이터셋은 객체가 잘 보이지 않는 이미지의 포함으로 인해 FN이 증가하여 상대적으로 낮은 성능을 보였다. Figure 7는 MOT 알고리즘이 탐지한 경계박스와 정답군을 그림으로 나타낸 것이다. 녹색 경계박스는 정답군이며, 파랑은 탐지한 경계박스, 그리고 빨강은 횡단보도의 경계박스이다. (단, 여기서 빨강 object는 횡단보도에 대한 CDNet과 추론 결과를 토대로 별도 표기하였다. Figure 7 (a), (b)는 탐지점수(pred score)가 높고(80 이상), 경계박스도 정답군과 어느 정도 일치함을 확인할 수 있다. 그러나 Figure 7 (c)와 같이 탐지점수가 낮거나 Figure 7 (d)처럼 이미지가 너무 어두운 경우 탐지하지 못하는 경우가 발생하였다.



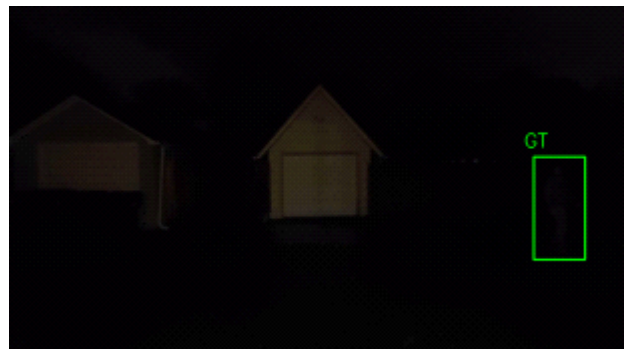
(a) Example of detecting crosswalk and a pedestrian



(b) Example of detecting multiple crossroads and a pedestrian



(c) Example of detecting a pedestrian in dark situations



(d) Example of in-screen detection of insufficient light, not even pedestrians

Figure 7. Examples of pedestrian detection and tracking results

Figure 8과 Figure 9은 Pedestrian dataset의 횡단보도 데이터셋을 이용해 본 논문에서 제안하는 알고리즘의 결과를 계산하여 그림에 표기한 것이다. Figure 8은 1 프레임 ~ 8 프레임까지의 결과이며, Figure 9은 보행자가 횡단보도에 도착하여 건널 때까지(62 프레임 ~ 69 프레임)의 결과이다. 경계박스가 보행자를 잘

탐지하고 있음을 확인할 수 있다. 제안하는 알고리즘을 이용한 보행자의 진행방향, 속도, 가속도, 횡단보도에 도달하기까지의 시간 또한 시간의 흐름에 따라 보행자의 상태를 잘 반영하고 있음을 확인할 수 있다. 다만, 프레임마다 예측하다 보니 Figure 8의 8번째 프레임에서 횡단보도 도착 시간 결과($t_{(f, cp)}$)처럼 어느 정도 튀는 값이 존재한다. 이는 추후 보행자의 보행 속도를 프레임 단위로 표준화한 후 표준화시킨 프레임마다 슬라이딩 윈도우 방식을 적용하여 튀는 값을 조정할 수 있을 것이다. 또한, 횡단보도의 경계박스와 보행자의 경계박스가 교차하는 지점부터는 보행자의 상태를 '건너고 있음(crossing)'으로 정의하였다. 이는 Figure 9의 63 프레임 ~ 69 프레임에서 확인할 수 있다.

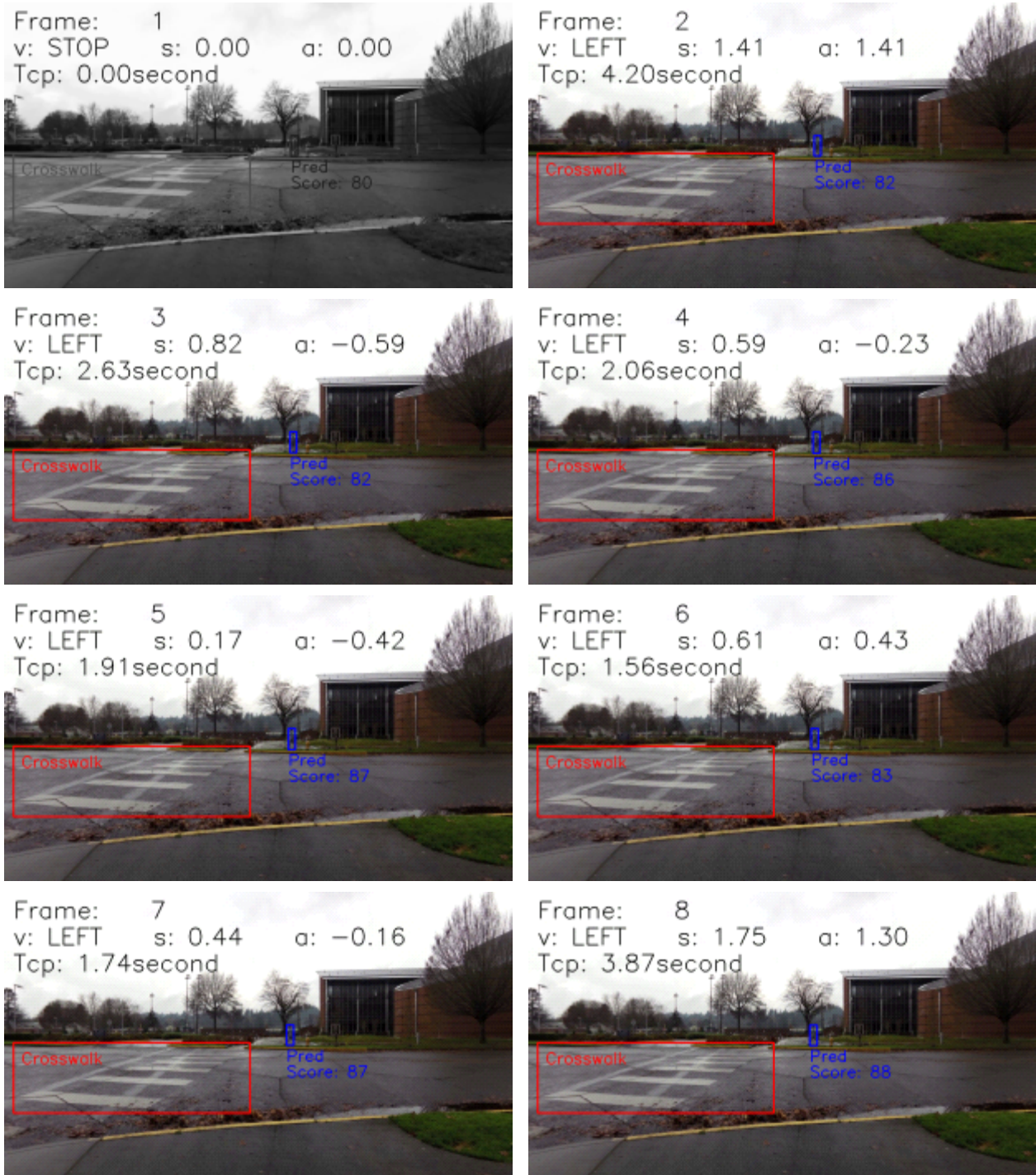


Figure 8. Examples of tracking a pedestrian at crosswalks and estimating the remaining time through eight sequential frames(1-8) (direction: v, speed: s, acceleration: a, remaining time until arrival at crosswalk: Tcp, status: crossing)



Figure 9. Examples of tracking a pedestrian at crosswalks and estimating the remaining time through eight sequential frames(62-69) (direction: v, speed: s, acceleration: a, remaining time until arrival at crosswalk: Tcp, status: crossing)

4.3 적용사례

본 절에서는 직접 촬영한 영상을 제안한 알고리즘으로 처리한 결과에 대해 기술한다. 영상 촬영 장소는 광운대학교 앞 인근 횡단보도이며, 촬영기기는 아이폰 14 프로를 이용하여 HD(1280x720), 30 fps로 촬영하였다. 실제 상황과 유사한 환경에서 촬영하기 위해 보도의 연석 부분에서 촬영을 진행하였으며, 안전을 위해 신호등은 보행가능 신호일 때만 촬영을 진행하였다. 다양한 상황에서 알고리즘을 평가하기 위해 다수의 보행자가 있는 환경에서 촬영하였으며 보행자가 횡단보도까지 걸어서 지나가는 경우, 횡단보도까지 걷다가 멈추는 경우, 그리고 횡단보도까지 걷다가 뛰는 경우를 추가 실험하였다.

1) 여러 보행자 인지



Figure 10. Example of detecting multiple pedestrians around crosswalks

Figure 10은 다수의 보행자들이 있을 때 알고리즘이 인지한 보행자를 표시한 것이다. 결과를 보면 다수의 보행자들에 대해서도 잘 인지하고 있음을 확인할 수 있었으며, Figure 10의 상단 그림과 같이 보행자가 겹쳐있거나, Figure 10의 하단 그림처럼 보행자 영역 일부가 잘린 경우에도 보행자로 인지하고 있음을 알 수 있다.

2) 횡단보도까지 걸어서 이동하는 경우

Figure 11은 특정 보행자(ID: 1)가 걸어서 횡단보도까지 이동하여 건너는 경우를 제안 알고리즘으로 처리한 결과이다. 보행자의 경계박스와 횡단보도의 경계박스가 겹치는 경우 상태를 '건너고 있음(crossing)'으로 처리하였다. Figure 11을 보면 보행자가 횡단보도까지 걸어가는 동안 남은 시간이 줄어들고 있음을 알 수 있다.



Figure 11. An example of detecting the ID#1 pedestrian approaching the crosswalk (direction: v, speed: s, acceleration: a, time remaining until arrival at crosswalk: Tcp, status: crossing)

3) 횡단보도까지 걸다가 멈추는 경우와 뛰는 경우

Figure 12는 특정 보행자(ID: 1)가 걸어서 횡단보도까지 이동하여 건너는 경우를 제안 알고리즘으로 처리한 결과이다. Figure 12를 보면 보행자가 횡단보도까지 걸어가는 동안 남은 시간이 줄어들다가 보행자가 멈추는 경우 상태는 '멈춤(stop)'으로 바뀌고 남은 시간이 0으로 변화하였음을 알 수 있다.



Figure 12. An example of detecting multiple pedestrians approaching a crosswalk (direction: v, speed: s, acceleration: a, time remaining until arrival at crosswalk: Tcp, status: crossing)

Figure 13은 특정 보행자(ID: 11)가 걸어서 횡단보도까지 뛰어서 이동하여 건너는 경우를 제안 알고리즘으로 처리한 결과이다. Figure 13을 보면 보행자가 횡단보도까지 뛰어가는 동안 남은 시간이 빠르게 줄어듦을 알 수 있다. 또한, 보행자와 횡단보도의 경계박스가 겹치는 경우 상태가 '건너고 있음(crossing)'으로 변화한 것을 확인할 수 있다.

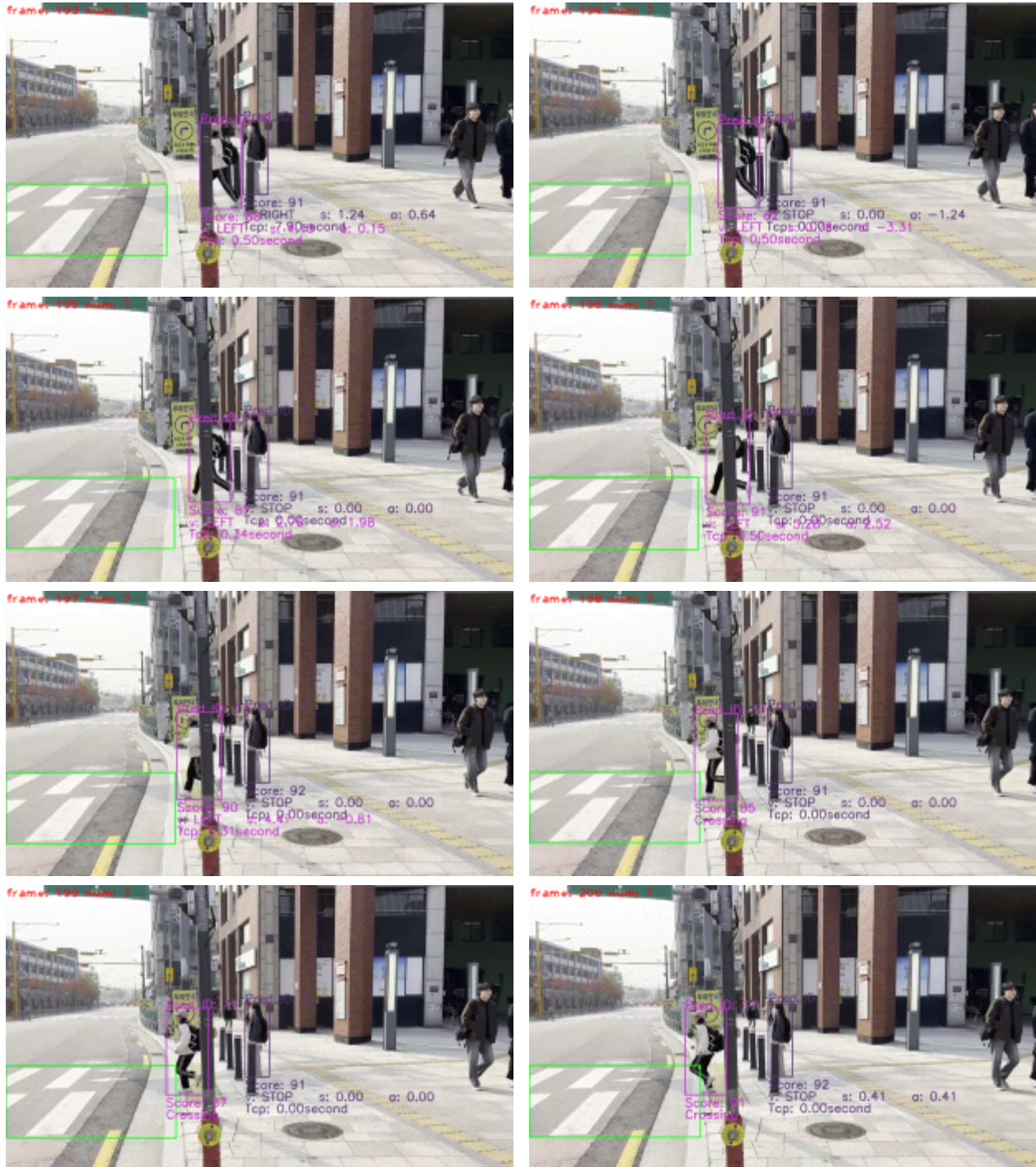
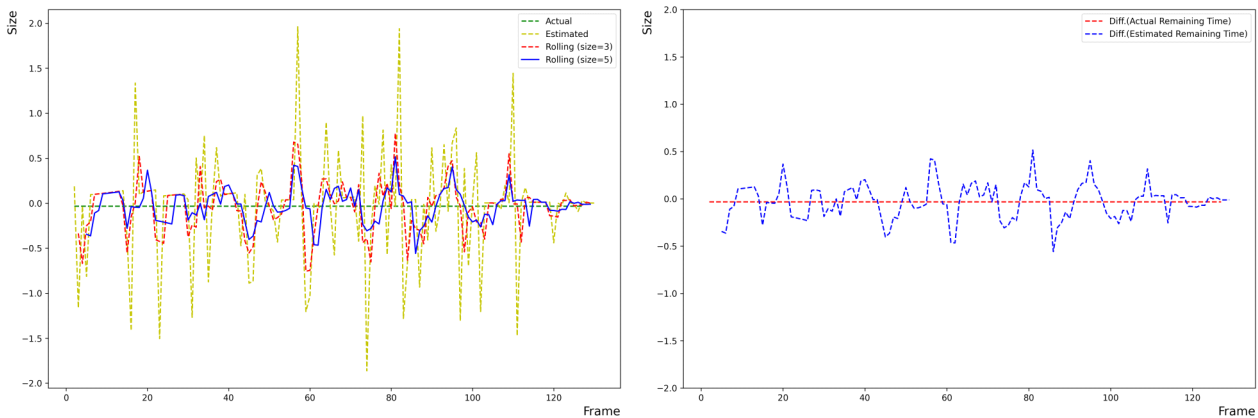


Figure 13. An example of detecting the ID#11 pedestrian running into a crosswalk (direction: v, speed: s, acceleration: a, time remaining until arrival at crosswalk: Tcp, status: crossing)

5. 결과분석 및 토론

5장에서는 4장에서의 결과에 대해 분석하고, 그 의미에 대해 토론한다. 먼저, 보행자 인지 모델의 성능은 MOT17 데이터셋에서 75.3%, MOT20에서 63.5%의 MOTA를 보였으며, 테스트 데이터셋에서 82.1%의 성능을 보여주었다. 장당 처리 속도는 GPU TITAN Xp 환경에서 약 31.2 fps를 보여 실시간 처리에도 적용이 가능함을 보여주었다.

또한, 인공지능 모델들에서의 결과를 기반으로 제안 알고리즘을 실제 환경에 적용하여 보행자의 방향, 속도, 그리고 횡단보도까지 남은 시간 등을 계산하였다. Figure 14는 보행자가 횡단보도에 도달하기까지의 실제 남은 시간과 제안 알고리즘의 예측결과를 그래프로 비교한 것이다. Figure 14에서 녹색은 실제시간, 노랑은 예측시간, 그리고 빨강, 파랑은 후처리된 예측시간 의미한다. 프레임 간 추정 오차를 보정하기 위해서 이동평균(moving average; MA) 필터를 적용하였으며, 윈도우 크기(time window size)는 각 3(빨강)과 5(파랑)으로 표시하였다. Figure 14 (b)에서 확인할 수 있듯 일정한 패턴을 그리며 점차 줄어들고 있음을 알 수 있다. Figure 14를 통해 제안하는 알고리즘이 보행자가 횡단보도에 도달하기까지의 시간을 반영하고 있음을 알 수 있었다. 이를 활용하여 차량 운전자의 보행자가 횡단보도에 진입하기까지의 시간을 알림 서비스 형태로 제공하여 안전사고를 예방하고, 나아가 자율주행자동차의 차량제어 시스템에 활용될 수 있음을 확인하였다.



(a) Experimental results showing the difference between the actual and estimated remaining time of pedestrians crosswalk arrival. Green dotted line: actual remaining time, yellow dotted line: estimated remaining time, red dotted line: MA(3), blue solid line: MA(5)

(b) Experimental results showing the difference between the actual remaining time and the estimated remaining time with MA(5). Overall, the predicted values of the remaining time is accurately estimated to be within 0.5 seconds of those of the actual value.

Figure 14. Experimental results of estimating the time the pedestrian arrives at the crosswalk

6. 결론

본 논문에서는 도시 물류에서 있어 자동차에 의한 보행 안전사고의 위험을 줄이기 위해 횡단보도 인근 보행자 탐지 및 횡단 예측 시스템을 제안하였다. 횡단 예측 시스템에서는 횡단보도 인지와 보행자 인지 및 추적을 위한 동작을 분리함으로써 부하 분산 및 추론시간의 단축을 꾀하였다. 또한, 횡단보도 인지를 위해서는 CDNet을 활용하였고 다수 보행자를 정확하게 추적하여 횡단까지 잔여시간을 추정하기 위해 ByteTrack을 사용하였다. CDNet와 ByteTrack에서 추출한 값을 가지고 보행자의 횡단 여부를 예측하기 위한 값인 보행자의 진행 방향, 속도, 가속도, 횡단보도와의 거리를 계산해서 횡단보도에 도착까지 잔여시간을 추정하였다.

본 연구에서는 ByteTrack이 실제 활용 목적에 적합한지 여부를 판단하기 위해 MOT17와 MOT20의 데이터를 활용하여 검증하였다. 실험 결과, MOT17에서는 MOTA가 75.3%, MOT20에서 63.5%의 성능을 보였다. MOT20은 보행자의 밀집도가 크다 보니 MOT17 데이터셋에 비해 상대적으로 낮은 성능을 보였다. 또한, Pedestrian 데이터셋에서 성능 평가 결과는 MOTA 82.1% 성능을 나타냈다. 추가로 다양한 상황에서 알고리즘을 탐지 및 추적 성능을 평가하기 위해 실제 환경에서 직접 촬영을 통해 시나리오 검증을 수행하였다. 특히, 보행자가 횡단보도까지 걸어서 지나가는 경우와 횡단보도까지 건다가 멈추는 경우, 그리고 횡단보도까지 건다가 뛰는 경우를 추가로 실험하여 제안 방식의 정확도 성능을 보였다. 제안 알고리즘은 다양한 상황에 적용 가능성을 보여주며 전반적으로 높은 MOTA 성능을 보여주었다.

제안 알고리즘은 교차로 교통사고를 줄이기 위해 다양한 방식으로 실제 서비스 상용화에 기반기술로 활용될 수 있을 것으로 기대된다. 첫째, 보행자가 횡단보도에 진입하기까지의 시간을 차량 운전자에 알려주는 알림 서비스로 활용될 수 있다. 둘째, 대한민국을 포함하여 다수 나라의 교차로 법규에 따라 범용적으로 자율주행차량에 응용될 수 있다. 셋째, 비전 기반의 보행자 충돌에 대응하는 충돌방지도조시스템의 핵심 기술의 기반이 될 것으로 기대한다.

다만, 본 논문에서 제안한 알고리즘에서의 횡단 여부 예측 정확도의 경우 프레임마다 예측하기 때문에 이상치(outlier)가 존재하여 현시점에서 자율주행자동차에 접목하기에는 강건함이 떨어진다. 따라서 추후 논문에서는 본 논문에서의 제안 알고리즘에 칼만 필터(Kalman filter) 등을 이용해 보정(calibration)하고, 보행자의 속도를 프레임 단위로 표준화하여 정확도를 높일 계획이다. 또한, 슬라이딩 윈도우 방식을 적용하여 잡음(noise)을 최소화할 계획이다. 더불어 시뮬레이터를 사용해서 복잡한 시나리오에서 학습을 진행하여 보행자 횡단 여부 예측 정확도를 강화하고, 보행자 신호등과 같은 도로 인프라 데이터셋을 수집하여 더욱 강건한 시스템을 연구할 생각이다.

참고문헌

- Akhauri, S., Zheng, L. Y., and Lin, M. C. (2020). Enhanced transfer learning for autonomous driving with systematic accident simulation. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (pp. 5986-5993). IEEE.
- Almalioglu, Y., Turan, M., Trigoni, N., and Markham, A. (2022). Deep learning-based robust positioning for all-weather autonomous driving. *Nature Machine Intelligence*, 1-12.
- Al-Qizwini, M., Barjasteh, I., Al-Qassab, H., and Radha, H. (2017). Deep learning algorithm for autonomous driving using googlenet. In 2017 IEEE Intelligent Vehicles Symposium (IV), (pp. 89-96). IEEE.
- Althoff, M., Stursberg, O., and Buss, M. (2009). Model-based probabilistic collision detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 10(2), 299-310.
- Bernardin, K., Elbs, A., and Stiefelwagen, R. (2006). Multiple object tracking performance metrics and evaluation in a smart room environment. In Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV, (Vol. 90, No. 91). Citeseer.
- Bochinski, E., Eiselein, V., and Sikora, T. (2017). High-speed tracking-by-detection without using image information. In 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS), (pp. 1-6). IEEE.
- Chen, N., Hong, F., & Bai, B. (2019). Zebra crossing recognition method based on edge feature and Hough transform. *Journal of Zhejiang University of science and technology*, 31(6), 476-483.
- Cho, M. G. (2019). A study on the obstacle recognition for autonomous driving RC car using lidar and thermal infrared camera. In 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), (pp. 544-546). IEEE.
- Chu, K., Lee, M., and Sunwoo, M. (2012). Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE transactions on intelligent transportation systems*, 13(4), 1599-1616.
- Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., and Leal-Taixé, L. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*.
- Ess, A., Leibe, B., Schindler, K., and Van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, (pp. 1-8). IEEE.
- Farag, W., and Saleh, Z. (2018, April). Traffic signs identification by deep learning for autonomous driving. In Smart Cities Symposium 2018, (pp. 1-6). IET.
- Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 580-587).

- Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3), 362-386.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770-778).
- Jin, Y., Jeon, H., Shin, Y. N., and Hyun, E. (2018). Data Analysis for Sensor Fusion of AVM Camera and Automotive Radar. *Korean Society of Automotive Engineers Spring Conference*, 581-583.
- Kim, D., Kim, Y., Go, S., Kim, Y., Lee, Y., Kim, C., (2018). A Study on the Application of Autonomous Vehicles to the Logistics Industry : Focusing on the Change of the Logistics Hub. *Journal of the Korean Society of Supply Chain Management* 18(2), (pp. 81-90).
- Kim, T., Kim, S., Kim, K., Oh, Y., Lee, J., Cho, W., Kim, K., (2021). Logistics Sorting System using Autonomous Driving Robot. *Journal of The Korea Socceity of Computer and Information*, 29(2), (pp. 491-492).
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.
- Kwon, T. W., Lee, S. P., Kim, D., Jang J., Lee, M., Kang, S. U., Kim, H., Oh, K., On, J., Kim, Y. J., Yun, S. J., Jin, K. W., Kim, E. Y., Kim, and K. G. (2021). Diagnostic performance of artificial intelligence model for pneumonia from chest radiography. *Plos one*. 16.4: e0249399.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lee, S., Baek, S., and Lee, S. (2021). A Study on the Collision Avoidance Algorithm Using FMCW Radar. *Proceedings of the Korean Telecommunications Society Conference*, 668-669.
- Li, G., Yang, Y., Qu, X., Cao, D., and Li, K. (2021). A deep learning based image enhancement approach for autonomous driving at night. *Knowledge-Based Systems*, 213, 106617.
- Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., and Leibe, B. (2021). Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2), 548-578.
- Meschke, S. (2017). Pedestrian Dataset. Retrieved 2022 from <https://www.kaggle.com/datasets/smeschke/pedestrian-dataset>.
- Milan, A., Leal-Taixé, L., Reid, I., Roth, S., and Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.
- Noh, S., and Han, W. Y. (2014). Collision avoidance in on-road environment for autonomous driving. In *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, (pp. 884-889). IEEE.
- Okuyama, T., Gonsalves, T., and Upadhyay, J. (2018). Autonomous driving system based on deep q learnig. In *2018 International conference on intelligent autonomous systems (ICoIAS)*, (pp. 201-205). IEEE.
- Pak, A., Manjunatha, H., Filev, D., and Tsiotras, P. (2022). CARNet: A Dynamic Autoencoder for Learning Latent Dynamics in Autonomous Driving Tasks. *arXiv preprint arXiv:2205.08712*.
- Park, S., Kim, J., (2020). Development and Validation of a Picker Following Automated Guided Vehicle Design Process for Pickers in a Logistics Environment. *Korean Journal of Logistics*, 28(5), (pp. 11-26).

- Park, Y., Yang, J. H., and Lim, S. (2018). Development of complexity index and predictions of accident risks for mixed autonomous driving levels. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), (pp. 1181-1188). IEEE.
- Prabhakar, G., Kailath, B., Natarajan, S., and Kumar, R. (2017). Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. In 2017 IEEE region 10 symposium (TENSYP), (pp. 1-6). IEEE.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 779-788).
- Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In European conference on computer vision, (pp. 17-35). Springer, Cham.
- Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295.
- Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 1-9).
- Traffic Accident Analysis System (2021). Traffic Accidents by Accident Type. Retrieved 2022 from http://taas.koroad.or.kr/sta/acs/gus/selectAcCntTyTfCacd.do?menuId=WEB_KMP_OVT_MVT_TAG_ATT.
- Uddin, M. S., & Shioyama, T. (2005). Detection of pedestrian crossing using bipolarity feature-an image-based technique. IEEE Transactions on Intelligent transportation systems, 6(4), 439-445.
- Wang, Z., Zheng, L., Liu, Y., Li, Y., and Wang, S. (2020). Towards real-time multi-object tracking. In European Conference on Computer Vision, (pp. 107-122). Springer, Cham.
- Wei, Q., Won S., (2018). Conceptual Design and Importance Analysis of Industry 4.0 Based Products and Services Applied in Container Terminals, Korean Journal of Logistics, 26(3), (pp. 91-110).
- Xu, S., Wang, X., Lv, W., Chang, Q., Cui, C., Deng, K., Wang, G., Dang, Q., Wei, S., Du, Y., and Lai, B. (2022). PP-YOLOE: An evolved version of YOLO. arXiv preprint arXiv:2203.16250.
- Y. Hashimoto, Y. Gu, L. -T. Hsu and S. Kamijo, "Probability estimation for pedestrian crossing intention at signalized crosswalks," 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2015, pp. 114-119, doi: 10.1109/ICVES.2015.7396904.
- Zeng, F., Dong, B., Wang, T., Zhang, X., and Wei, Y. (2021). Motr: End-to-end multiple-object tracking with transformer. arXiv preprint arXiv:2105.03247.
- Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2021). Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864.
- Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision, 129(11), 3069-3087.

Zhang, Z. D., Tan, M. L., Lan, Z. C., Liu, H. C., Pei, L., & Yu, W. X. (2022). CDNet: a real-time and robust crosswalk detection network on Jetson nano based on YOLOv5. *Neural Computing and Applications*, 1-12.