

A Study on the Fall Detection
Using Yolov8-pose and Accelerometer Data for Elderly Care

요약

본 연구에서는 Yolov8-pose 모델을 활용하여 고령자의 낙상 감지 시스템을 개발하고 이를 실시간으로 평가하였다. 본 시스템은 낙상과 일상적인 활동을 효과적으로 구분하며, mAP50-95가 0.671로 높은 정확도를 나타내었다. 기존의 YOLO 기반 시스템에 가속도 데이터를 결합하여 감지 정확성을 높였다. 가속도 데이터는 급격한 움직임이나 균형을 잃은 상황에서 발생하는 변화를 실시간으로 감지할 수 있게 해주며, 이를 통해 시스템의 신뢰성을 더욱 강화하였다.

I. 서론

1.1. 서론

고령자의 낙상 사고는 심각한 인명 피해로 이어질 가능성이 높기 때문에 초기 단계에서 감지하는 것이 중요하다. 따라서 고령자의 낙상과 다양한 행동을 신속하고 정확하게 감지하며, 위험 상황을 조기에 예측하여 이에 대비할 수 있어야 한다.

본 논문에서는 Yolov8-pose 모델과 가속도 데이터를 결합한 새로운 낙상 감지 시스템을 제안한다. 가속도 데이터는 고령자의 움직임 변화를 정밀하게 감지할 수 있어 낙상을 더욱 정확하게 판단할 수 있다. 시스템은 Yolov8-pose 모델을 통해 사람의 위치와 자세를 감지하고, 가속도계 데이터를 바탕으로 낙상 여부를 판단한다.

본 연구에서는 가속도 데이터를 CSV 파일로 저장하여 분석에 활용했으나, 추후 실시간 시스템으로 구현할 계획이다. 이를 통해 기존의 영상 기반 감지 시스템보다 더 신뢰할 수 있는 고령자 안전 모니터링 시스템을 구축할 수 있다.

1.2. 관련 연구

최근 연구들은 고령자 낙상 감지 시스템 개발에 있어 비디오 기반 접근 방식의 장점을 강조한다. Raza et al.(2022)의 연구에서는 YOLO 모델과 그 변형(YOLOv1~v4 및 Tiny-YOLOv4)을 활용하여 낙상을 실시간으로 감지하고, 비전 기반 시스템의 우수성을 입증하였다. 특히 Tiny-YOLOv4 모델은 mAP 96%와 6.71ms의 낮은 추론 시간을 기록하며, 기존의 센서 기반 시스템보다 높은 효율성과 정확도를 보였다. 이러한 모델은 Raspberry Pi와 OAK-D와 같은 엣지 디바이스에서 구현되어 고비용 하드웨어의 필요성을 감소시키고, 다양한 환경에서 실시간 감지를 가능하게 했다.

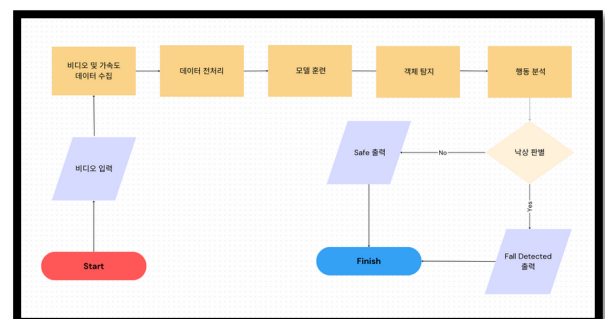
이 연구 또한 UR Fall Detection Dataset 을 활용하여 RGB 비디오 및 가속도 데이터를 기반으로 낙상 감지 모델을 학습하였다. 기존의 SVM 이나 KNN 기반 접근법과 비교했을 때, YOLO 기반 모델은 더 높은 정확도와 정밀도를 기록하며 낙상 감지의 신뢰성을 강화하였다.

II. 본론

2.1. 가속도를 고려한 Yolov8-pose

본 연구의 주요 개선점 중 하나는 기존의 Yolo 기반 낙상 감지 시스템에 가속도 데이터를 추가하여 감지의 정확성을 높이고, 실제 낙상 여부를 동시에 고려하여 더욱 정밀하게 낙상을 감지하고 위험 상황을 파악할 수 있는 것이다.

본 연구를 통해 가속도 데이터를 결합하여 움직임의 강도와 속도 변화까지 고려한 낙상 감지를 구현했다. 이 연구에서는 특정 임계치를 초과하는 가속도 변화와 함께 관절 위치의 급격한 이동이 감지되면 이를 낙상으로 판단하여 표시한다.



<그림 1> Flowchart

2.2. 연구 과정

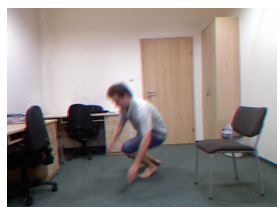
<그림 1>을 보면, 시스템은 우선 실시간으로 입력된 비디오와 가속도계를 통해 가속도 데이터를 수집한다. 수집된 비디오와 가속도 데

이터는 모델에 입력되기 전에 전처리 과정을 거친다. 이 과정에서, 비디오의 각 프레임에서 사람의 주요 신체 부위(예: 팔, 다리, 목, 등)의 키포인트를 추출하고, 해당 프레임에 맞는 가속도 데이터 또한 추출한다.

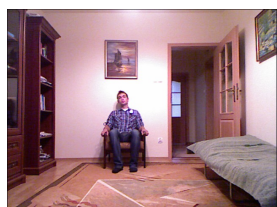
다음으로, 이 데이터가 모델에 입력되면, 낙상 여부를 예측한다. 모델은 두 가지 주요 입력을 결합하여, 낙상과 비낙상을 구분한다. 또한, 객체(사람)를 탐지하고, 키포인트가 어떻게 움직이는지 분석한다. 키포인트의 변화를 추적하고, 가속도 데이터로부터 특정 임계값을 초과하는 급격한 변화를 감지하면 낙상으로 판단한다. 낙상을 감지하면, 'Fall Detected'로 상태 결과가 출력된다. 반면, 변화가 일정 범위 내에서 안정적이면 'Safe'를 화면에 출력하여 상태를 알 수 있다.

2.3. 연구 방법

모델 학습은 UR Fall Detection Dataset(<그림 2>, <그림 3> 참고)을 기반으로, 낙상(Fall)과 일상 활동(Activities of Daily Living) 비디오 데이터를 프레임별로 분할하여 6887개(2995개의 Fall 프레임과 3892개의 ADL 프레임)를 학습에 사용하였다. ([표 1] 참고) ADL 비디오는 일상 생활 행동(걷기, 앉기, 눕기 등)을 포함한다. [표 2]를 보면 알 수 있듯이, 데이터의 80%인 5509개(낙상 2396개는 train data로, 20%인 1378개(낙상 599개 + 일상 활동 779개)는 test data로 나누어 학습을 진행하였다.



<그림 2> UR Fall Detection Dataset - FALL



<그림 3> UR Fall Detection Dataset - ADL

모델 학습 과정에 적용된 하이퍼 파라미터들은 [표 3]에 명시되어 있으며, 이는 성능과 정확도를 동시에 최적화하기 위하여 선택되었다.

Dataset	총 6887개
Fall sequences	2995개
ADL sequences	3892개

[표 1] 학습에 사용된 데이터셋

Train data	5509개
Test data	1378개

[표 2] Train-test split

Epochs	50
Batch Size	16
Optimizer	Adam
Learning rate	0.001
Loss Function	Binary Cross Entropy

[표 3] 하이퍼 파라미터

가속도 임계값인 fall_threshold는 가속도가 일정 값 이상이면 낙상 가능성이 있다고 판단하게 된다. 또한, 키포인트 변화 임계값은 키포인트 변화가 일정 값을 초과하면 낙상으로 간주되도록 설정된다. check_falling 함수는 y_values와 가속도를 비교하여 낙상 여부를 판단한다. first_point_diff와 second_point_diff는 키포인트 변화의 차이를 계산하여, 사람이 갑작스럽게 기울어진 경우를 감지하게 된다. 가속도가 fall_threshold 값을 초과하고, 키포인트 변화가 각각 first_point_threshold와 second_point_threshold를 초과하면 낙상으로 판단하게 된다. <그림 4>는 앞서 설명한 부분의 핵심 코드이다.

```

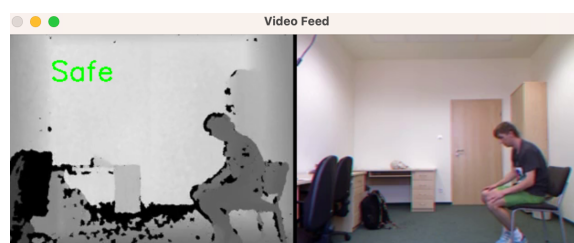
fall_threshold = 1.5
first_point_threshold = 50
second_point_threshold = 50

def check_falling(y_values, acceleration):
    global previous_y_values, fallen_state
    if previous_y_values is not None and len(y_values) >= 6 and len(previous_y_values) >= 6:
        first_point_diff = abs(previous_y_values[0] - y_values[5])
        second_point_diff = abs(previous_y_values[5] - y_values[10])
        if acceleration > fall_threshold and first_point_diff > first_point_threshold and second_point_diff > second_point_threshold:
            fallen_state = True
            print("Fall Detected")
        else:
            if not fallen_state:
                print("Safe")
    previous_y_values = y_values
    
```

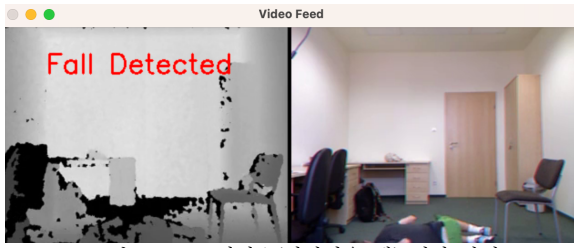
<그림 4> 중간 코드 사진

2.4 결과

<그림 5>와 <그림 6>에서 보듯이, 사용자에게 실시간으로 "Safe" 또는 "Fall Detected"로 고령자의 상태를 출력하였다.



<그림 5> ADL 비디오(앉았을 때) 결과 사진



<그림 6> Fall 비디오(넘어졌을 때) 결과 사진

모델의 성능 평가에는 [표 4]에 나열된 지표들을 사용했다.

	Fall	ADL
Accuracy	0.90	0.925
Precision	0.987	0.974
Recall	0.90	0.925
F1 score	0.947	0.961
mAP50-95	0.697	0.645

[표 4] 성능 지표

본 연구에서는 모델의 성능을 평가하기 위해 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1 Score, mAP50-95 지표를 사용하였다.

정확도는 모델이 예측한 값 중 실제로 맞은 예측의 비율을 의미한다. [표 4]에 따르면, Fall 클래스의 정확도는 0.90, ADL 클래스는 0.925로 나타났다. 이는 두 클래스 모두에서 모델이 매우 높은 정확도로 예측을 수행했음을 보여준다.

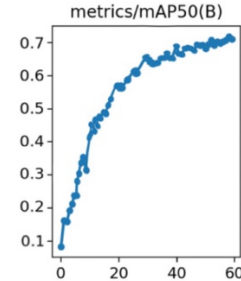
정밀도는 모델이 양성(예: 특정 행동이 감지됨)이라고 예측한 경우 중 실제로 양성인 비율을 나타낸다. 즉, 모델이 얼마나 정확하게 낙상을 감지했는지를 보여준다. [표 4]에서 Fall 클래스의 정밀도는 0.987, ADL 클래스는 0.974로, 두 클래스 모두 높은 정밀도를 기록하여 모델이 매우 정확한 예측을 하고 있음을 알 수 있다.

재현율은 실제 양성 케이스 중 모델이 얼마나 많은 경우를 올바르게 감지했는지를 나타낸다. Fall과 ADL 클래스의 재현율은 각각 0.90과 0.925로 나타났으며, 이는 모델이 실제로 존재하는 객체를 잘 감지하고 있다는 것을 의미한다.

F1 Score는 정밀도와 재현율의 조화 평균으로, 두 지표의 균형을 평가한다. Fall 클래스의 F1 Score는 0.947, ADL 클래스는 0.961로 나타났다. 이는 모델이 두 클래스에서 모두 높은 정밀도와 재현율을

유지하며 균형 잡힌 성능을 보였음을 의미한다.

Fall 클래스의 mAP50-95는 0.697, ADL 클래스는 0.645로 나타났다. 이 값은 모델이 다양한 IoU 임계값에서 객체를 효과적으로 감지하고 있다는 것을 나타낸다.



<그림 6> YOLOv8-pose의 mAP50-95

<그림 6>은 YOLOv8-pose 모델의 학습 과정 중 mAP50-95 값의 변화를 나타낸다. x축은 학습 횟수(epochs), y축은 모델의 mAP50-95 값을 나타내며, 학습이 진행될수록 성능이 향상되는 추세를 보인다. YOLOv8-pose 모델이 다양한 IoU 임계값에서 객체를 효과적으로 인식하며 뛰어난 성능을 발휘하고 있다는 것을 시각적으로 보여준다.

따라서 본 연구를 통해 고령자 안전 모니터링 시스템에서 보다 정확하고 신뢰성 있는 성능을 제공하며, 낙상 감지 뿐만 아니라 일상적 활동도 효과적으로 처리할 수 있는 시스템을 제시하였다.

III. 결론

본 연구에서는 YOLOv8-pose 모델을 활용하여 고령자의 낙상 감지 시스템을 개발하고, 이를 실시간으로 평가하였다. 모델은 낙상과 일상적인 활동을 효과적으로 구분할 수 있었으며, 높은 정확도와 정밀도를 기록하였다. 특히, 낙상 비디오와 일반적인 활동 비디오에서 모두 우수한 성능을 보였고, 정밀도와 재현율 간의 균형을 잘 맞추어 F1 점수에서도 긍정적인 결과를 얻었다. 시각적 정보만으로 발생할 수 있는 False Positive를 줄이고, 낙상 감지의 정확성과 신뢰성을 크게 향상시켰다.

그러나 데이터셋의 크기와 환경적 다양성에서 한계가 있었으며, 향후 더 다양한 환경에서 수집된 데이터와 함께 모델을 개선할 필요가 있다. 또한, 가속도 센서와 비디오 데이터의 동기화 및 정확도 문제를 해결하면 시스템의 성능을 더욱 향상시킬 수 있을 것이다.

참고문헌

- [1] Raza, A., Yousaf, M. H., & Velastin, S. A., “Human fall detection using YOLO: A real-time and AI-on-the-edge perspective.”, 2022
- [2] Rasyid, M., Zakaria, R., & Mahayuddin, Z. A., “Smart elderly fall detection system using deep learning approach.”, 2021
- [3] He, J., Pan, X., & Li, Y., “Fall detection using machine learning approaches and wearable sensors.”, 2022
- [4] Mauldin, T. R., Canby, M. E., Metsis, V., & Ngu, A. H. H., “SmartFall: A smartphone-based fall detection system using machine learning.”, 2020
- [5] Chen, L., Chen, X., Zhang, J., & Wang, S., “Real-time elderly fall detection using deep learning on embedded devices.”, 2022