

치매 환자 이상 행동 감지 모니터링 및 치매 예방 게임

2020270657 하진영 2021270632 박형준

목차

- 과제의 필요성
- 과제의 목표
- 기대효과
- 방법

과제의 필요성



고령화사회

- 노인 인구 비율 증가에 따른 사회적, 경제적 부담 증가



모니터링

- 효과적인 모니터링 및 간호 부담 감소



예방 게임

- 예방 게임을 통한 정신적 건강 노화 지연

과제의 목표



치매 환자 이상 행동 감지 모니터링 시스템, 예방 게임 개발 목표



실시간 행동 감지로 돌봄 제공자의 부담 감소 및
조기 대응이 가능하도록 하여 환자의 안전 강화



인지 기능을 자극하는 게임을 통해 노인의 정신적 건강 유지 및
치매 예방에 기여



이를 통한 고령화 사회에서 발생하는 문제 해결에 기여 및
보다 지속 가능한 노인 돌봄 환경 조성을 목표

기대효과

치매 환자 돌봄 부담 완화

치매 예방 및 인지 기능 향상

의료, 복지 비용 절감

방법

YOLOv8

- 영상에서 관절점 추출
- 관절점 좌표 => 시계열 데이터 변환

LSTM

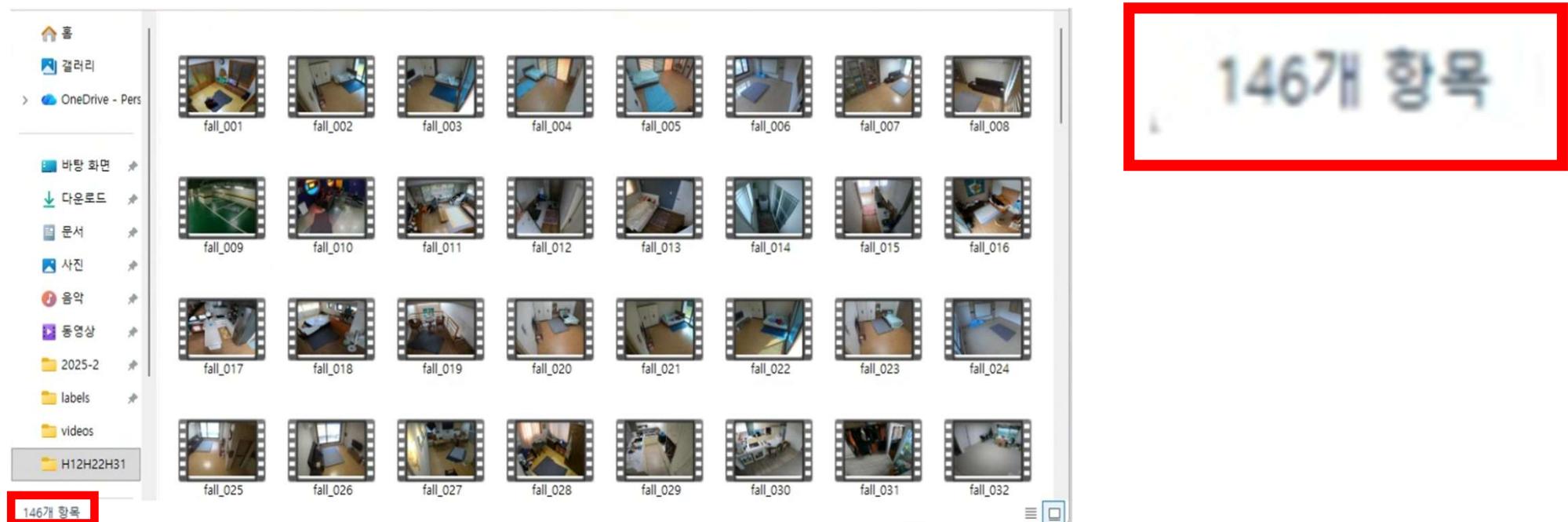
- 시계열 데이터(입력) => LSTM
- 관절점의 변화량 및 기울기 변화 기반 낙상 예측

게임

- Card.cs => 개별 카드 로직
- CardController.cs => 게임 관리 로직

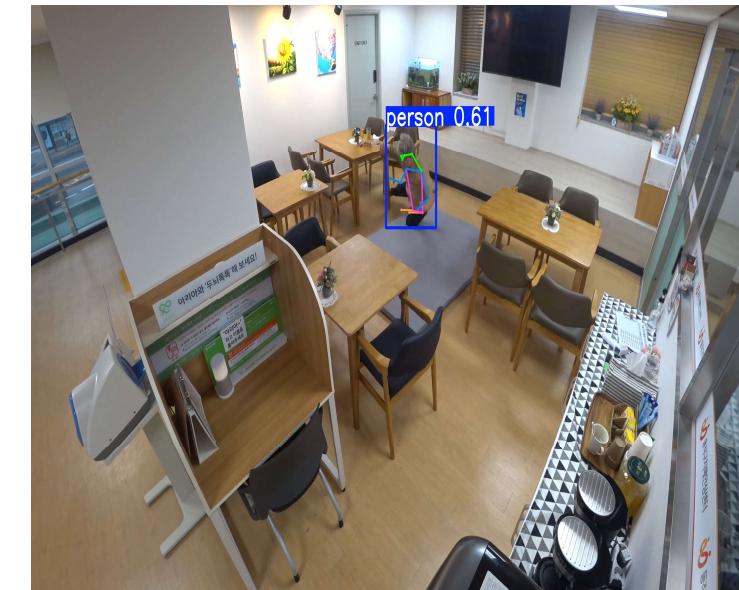
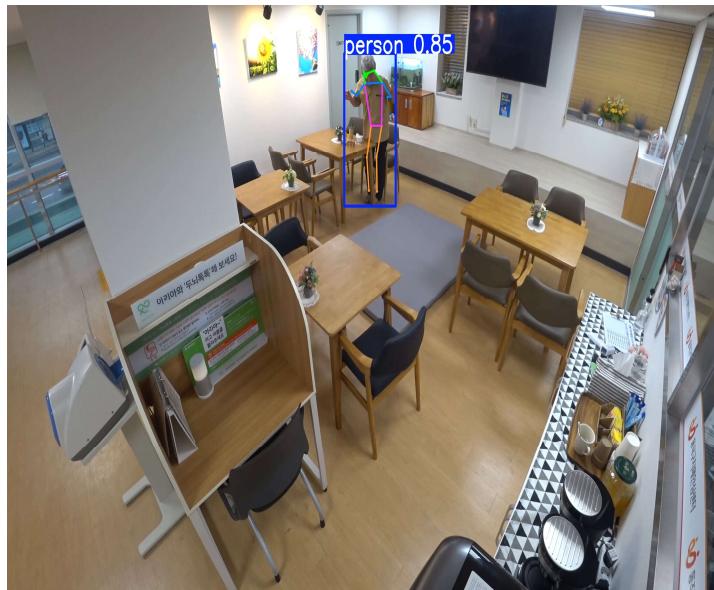
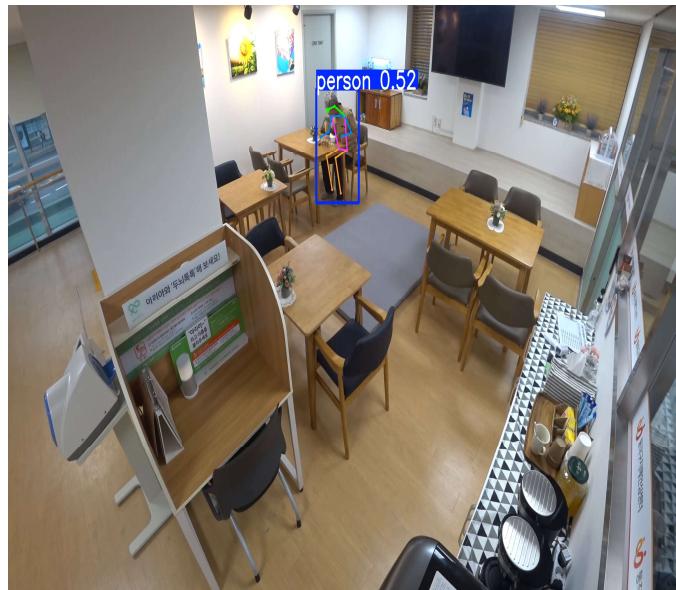
YOLOv8

학습용 데이터는 Aihub의 시니어 이상행동 영상을 활용
총 학습한 영상 데이터 146개



YOLOv8

학습된 YOLOv8 모델을 토대로
영상에서 사람을 감지하고 사람의 관절점을 추출

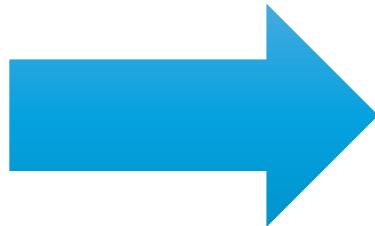


YOLOv8

- 추출한 관절점들을 시계열 데이터로 저장
- LSTM의 입력데이터로 사용하기 위해 시계열데이터 변환 작업 수행

[사람 ID] [x0 y0 conf0] [x1 y1 conf1]
[x2 y2 conf2] ... [x16 y16 conf16]

관절 번호	x 좌표	y 좌표	신뢰도 (confidence)
0	0.519894	0.221104	0.114985
1	0.245840	0.502671	0.142169
2	0.905272	0.504668	0.136164
3	0.889697	0.500761	0.137635



평탄화된 형태: (34,) # 17개의 관절 × 2개 좌표
예: [x0, y0, x1, y1, ..., x16, y16]

frame	Nose_x	Nose_y	Neck_x	Neck_y	...	Ankle_x	Ankle_y
0	122.3	245.8	130.1	240.2	...	310.4	415.6
1	123.1	246.2	131.5	241.0	...	311.2	416.0
2	124.0	248.0	132.7	243.1	...	312.8	418.2

LSTM

```
# 데이터 로드  
  
X = np.load("C:/Users/HaJinYeong/2025-2/lstm_input/X.npy")  
y = np.load("C:/Users/HaJinYeong/2025-2/lstm_input/y.npy")  
  
# 입력 데이터 정규화  
  
X_mean = X.mean()  
X_std = X.std() + 1e-6  
X = (X - X_mean) / X_std  
  
# 데이터 셔플  
  
X, y = shuffle(X, y, random_state=42)  
  
# 클래스 불균형 처리 (개선 공식)  
  
normal_count = np.sum(y == 0)  
fall_count = np.sum(y == 1)  
total = len(y)  
  
class_weight = {  
    0: total / (2 * normal_count),  
    1: total / (2 * fall_count)  
}  
  
print("class_weight:", class_weight)
```

- 저장된 관절점 데이터 로드
- 영상마다의 값 범위가 달라 정규화를 통해 학습의 안정성 향상
- 데이터 셔플을 통해 낙상과 정상을 균등 있게 학습할 수 있도록 패턴 조정
- 낙상의 경우 영상에서 짧은 비중을 차지하기에 클래스 불균형을 처리하여 낙상 샘플의 가중치를 높여 모델이 낙상에 대해 더 민감하게 반응하도록 조정

LSTM

```
# LSTM 모델 (2단 LSTM 구조)

model = Sequential([
    LSTM(64, return_sequences=True, input_shape=(30, 8)),
    Dropout(0.3),

    LSTM(32, return_sequences=False),
    Dropout(0.3),

    Dense(32, activation='relu'),
    Dropout(0.3),

    Dense(1, activation='sigmoid')
])

model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# Early Stopping

es = EarlyStopping(
    patience=10,
    restore_best_weights=True,
    monitor='val_loss'
)
```

- 첫 번째 LSTM은 30프레임 전체를 처리하기 위해 모든 프레임에 대해 출력을 유지하기 위함
- 두 번째 LSTM은 한 프레임의 최종 요약 벡터만 출력
- 요약하면
첫 번째 LSTM은 전체 30 프레임 처리
두 번째 LSTM은 낙상인지 정상인지
판단할 최종 시계열 결과 요약
- 과적합을 막기 위해 val_loss가 10 epoch동안 향상되지 않으면 학습을 중단(Early Stopping)

LSTM

```
# 중앙점 계산  
sc = center(pts[L_SHOULDER], pts[R_SHOULDER])  
hc = center(pts[L_HIP], pts[R_HIP])  
wc = center(sc, hc)  
  
ang1 = angle(sc, wc)  
ang2 = angle(sc, hc)  
  
# ---- 기울기 추가 계산 ----  
shoulder_angle = line_angle(pts[L_SHOULDER], pts[R_SHOULDER])  
hip_angle = line_angle(pts[L_HIP], pts[R_HIP])  
  
# 수평 판정 (각도 절대값이 작을수록 수평)  
is_horizontal = (abs(shoulder_angle) < 0.35 and abs(hip_angle) < 0.35)  
  
feature = np.concatenate([sc, hc, wc, np.array([ang1, ang2])])
```

- 골반 관절점과 어깨 관절점의 중앙점들의 중앙을 허리 관절점으로
- 골반과 허리의 관절점의 **기울기의 변화와 기울기의 변화**에 따라 낙상 여부 판단
- 추가적으로 어깨 관절점과 골반 관절점의 **기울기가 수평에 가까울수록** 낙상에 가깝다고 판단
- 즉, **상체의 기울기가 수평일수록** 낙상으로 판단

치매 예방 게임

```
// ID 매개변수 추가 및 할당
public void InitializeCard(CardController manager, Sprite sp, int id)
{
    // Button 컴포넌트 참조 (없으면 찾고, 없으면 에러)
    if (button == null) button = GetComponent<Button>();

    // 초기화 동안 클릭 방지
    if (button != null) button.interactable = false;

    controller = manager; // Controller 할당
    iconSprite = sp; // 앞면 스프라이트 할당
    spriteID = id; // ID 할당

    // 초기화 완료 후 클릭 허용
    if (button != null) button.interactable = true;
}
```

```
public void Show()
{
    if (iconImage == null)
    {
        Debug.LogError("Card의 iconImage가 인스펙터에 연결되지 않았습니다!");
        return;
    }

    // 클릭 시 앞면 이미지를 표시
    iconImage.sprite = iconSprite;
    isSelected = true;
}
```

public void
InitializeCard(...)

초기화 및 안전성 확보. CardController 인스턴스(manager),
앞면 스프라이트, 그리고 spriteID를 할당받는다. 할당 중에는
Button.interactable을 false로 설정하여 클릭 오류를 방지한다.

public void Show() /
Hide()

시각적 상태 전환. iconImage.sprite를 앞면(iconSprite) 또는
뒷면(hiddenIconSprite)으로 교체하여 카드를 뒤집는 효과를
구현한다.

치매 예방 게임

```
void ShuffleIDs(List<int> idList)
{
    for (int i = idList.Count - 1; i > 0; i--)
    {
        int randomIndex = Random.Range(0, i + 1);

        int temp = idList[i];
        idList[i] = idList[randomIndex];
        idList[randomIndex] = temp;
    }
}

IEnumerator CheckMatching()
{
    if (firstSelected == null || secondSelected == null) yield break;

    // 1초 대기
    yield return new WaitForSeconds(1f);

    // Sprite 객체 비교 대신 ID를 비교
    if (firstSelected.spriteID == secondSelected.spriteID)
    {
        // 짹이 맞으면 파괴 (또는 비활성화/보이게 유지)
        firstSelected.isMatched = true;
        secondSelected.isMatched = true;

        Destroy(firstSelected.gameObject);
        Destroy(secondSelected.gameObject);
    }
    else
    {
        firstSelected.Hide();
        secondSelected.Hide();
    }

    // 초기화 및 클릭 재개
    firstSelected = null;
    secondSelected = null;
    canClick = true;
}
```

ShuffleIDs(List<int>
idList)

랜덤 배치 구현. Fisher-Yates 알고리즘을 사용하여 카드 짹의 ID 리스트를 무작위로 섞어준다. 이 순서대로 카드가 배치되어야 게임이 랜덤성을 가진다.

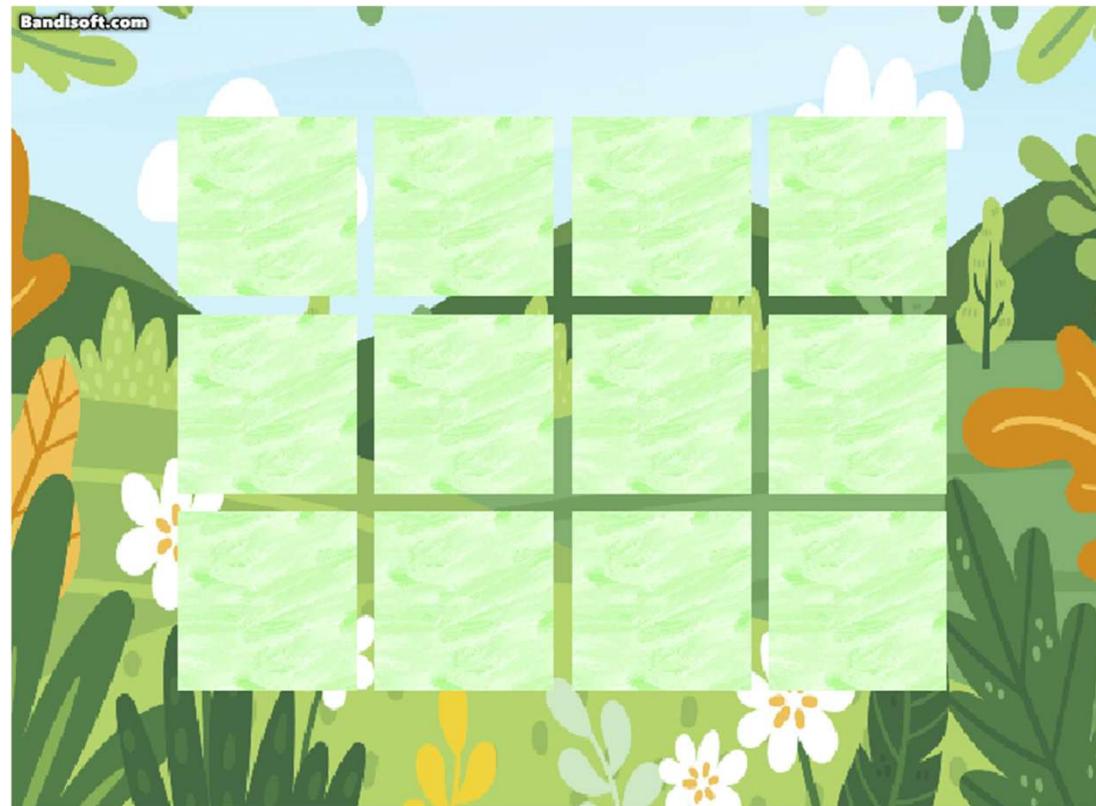
IEnumerator
CheckMatching()

쫙 판별 로직. yield return new WaitForSeconds(1f)로 잠시 대기하여 사용자에게 카드를 보여준다. 핵심은 firstSelected.spriteID == secondSelected.spriteID로 짹을 판별하는 것이다.

주요결과 1



주요결과2



결론 및 한계

- YOLOv8 + LSTM을 결합한 모델을 통해 치매 환자의 이상행동 중 하나인 낙상 여부를 판단할 수 있는 모델을 개발
- 그러나 기존의 YOLOv8 모델을 불러와 사용하여 사람 탐지와 관절점 추출이 안되는 경우가 발생하여 성능이 떨어짐
- 또한, 상체의 기울기만 보기에 허리를 굽히는 등 낙상이 아님에도 낙상이라고 판단하는 한계가 존재
- 노인들에게 맞는 UI/UX를 적용하여 인지 기능 향상을 위한 간단한 과일 카드 짹 맞추기 게임을 개발

향후 계획

YOLOv8

- 사람 탐지 및 관절점 추출이 잘 될 수 있도록 YOLOv8 모델의 추가적인 학습 및 개선 작업
- 다양한 카메라 각도의 영상 데이터 확보

LSTM

- 하체 기울기 변화와 상하체 기울기의 연관성 등을 추가하여 모델의 낙상 여부 판별 개선

치매 예방 게임

- 난이도 맞춤형 패턴 도입
- 폰트 및 UI 요소의 명암비 최적화 및 오디오 피드백 강화