

알고리즘 성능

1. 기존 알고리즘 → KFall

전방낙상 포함여부	True	False
민감도	100% (2346/2346)	100% (1411/1411)
특이도	98.31	98.31
Soso	39	39
Early	181	154

2-1. 최적화 (Early 감소) → KFall

전방낙상 포함여부	True	False
민감도	100% (2346/2346)	100% (1411/1411)
특이도	98.68	98.68
Soso	56	47
Early	86	82

2-2. 최적화 (Early 감소) → 연대

전방낙상 포함여부	True	False
민감도	100% (408/408)	100% (240/240)
특이도	99.24 % 524/528)	99.24 % 524/528)
Soso	31	21

3-1. 최적화 (전방낙상 제외 고려) → KFall

전방낙상 포함여부	True	False
민감도	99.62% (2337/2346)	99.86% (1409/1411)
특이도	98.61% (2691/2729)	98.61% (2691/2729)
Soso	128	87
Late	55	55

3-2. 최적화 (전방낙상 제외 고려) → 연대

전방낙상 포함여부	True	False
민감도	99.75% (407/408)	100% (240/240)
특이도	99.81% (527/528)	99.81% (527/528)
Soso	29	7

4. 최종모델 (주요변경사항)

- 낙상판단을 위한 연속판단 윈도우 수: 2 → 3
- 수직속도 감쇄계수 변경: 0.9 → 0.95
- 방향에 대한 규제 추가
 - GyroX가 일정 이상이면서 나머지 조건들 만족 시 낙상 (전방낙상 배제)
 - GyroX가 일정 미만일 때, GyroZ가 일정 이상이면, 나머지 조건들 만족 시 낙상 (측방낙상 및 전측방낙상 고려)
 - down0이 None인 조건 신규 생성 (뚜렷한 낙상 패턴이 보이지만 작은 down값 보일 때)
→ 현재 딥러닝 모델의 한계

```

conditions = [
    {'down': None, 'raw_acc': 80, 'raw_gyro': 100, 'snr_acc_svm': 22,
    'vv': 1.0, 'gyroX': 75, 'gyroZ': 65,},
    {'down': 90, 'raw_acc': 90, 'raw_gyro': 140, 'snr_acc_svm': 23, 'vv':
    None, 'gyroX': None, 'gyroZ': None,},
    {'down': 90, 'raw_acc': 90, 'raw_gyro': 45, 'snr_acc_svm': 22, 'vv':
    0.1, 'gyroX': None, 'gyroZ': None,},
    {'down': 85, 'raw_acc': 100, 'raw_gyro': 60, 'snr_acc_svm': 30, 'vv':
    1.5, 'gyroX': None, 'gyroZ': None,},
    {'down': 80, 'raw_acc': 95, 'raw_gyro': 150, 'snr_acc_svm': 21.5,
    'vv': 0.1, 'gyroX': None, 'gyroZ': None,},
    {'down': 80, 'raw_acc': 110, 'raw_gyro': 15, 'snr_acc_svm': 26.5,
    'vv': 7.0, 'gyroX': 50, 'gyroZ': 50,},
    {'down': 80, 'raw_acc': 90, 'raw_gyro': 90, 'snr_acc_svm': 25, 'vv':
    0.1, 'gyroX': None, 'gyroZ': None,},
    {'down': 80, 'raw_acc': 80, 'raw_gyro': 40, 'snr_acc_svm': 24, 'vv':
    8.0, 'gyroX': None, 'gyroZ': None,},
    {'down': 75, 'raw_acc': 90, 'raw_gyro': 10, 'snr_acc_svm': 34, 'vv':
    1.0, 'gyroX': 50, 'gyroZ': 50,},
    {'down': 75, 'raw_acc': 120, 'raw_gyro': 100, 'snr_acc_svm': 34, 'vv':
    0.3, 'gyroX': None, 'gyroZ': None,},
    {'down': 75, 'raw_acc': 130, 'raw_gyro': 30, 'snr_acc_svm': 17, 'vv':
    0.3, 'gyroX': 20, 'gyroZ': 20,},
    {'down': 70, 'raw_acc': 80, 'raw_gyro': 40, 'snr_acc_svm': 26, 'vv':
    3.5, 'gyroX': 50, 'gyroZ': 50,},
    {'down': 70, 'raw_acc': 80, 'raw_gyro': 40, 'snr_acc_svm': 19, 'vv':
    4.0, 'gyroX': None, 'gyroZ': None,},
    {'down': 70, 'raw_acc': 75, 'raw_gyro': 70, 'snr_acc_svm': 22.5, 'vv':
    2.5, 'gyroX': None, 'gyroZ': None,},
    {'down': 65, 'raw_acc': 70, 'raw_gyro': 35, 'snr_acc_svm': 20, 'vv':
    4.0, 'gyroX': None, 'gyroZ': None,},
    {'down': 60, 'raw_acc': 85, 'raw_gyro': 70, 'snr_acc_svm': 19.0, 'vv':
    5.0, 'gyroX': None, 'gyroZ': None,},
    {'down': 55, 'raw_acc': 60, 'raw_gyro': 80, 'snr_acc_svm': 18, 'vv':
    5.5, 'gyroX': None, 'gyroZ': None,},
    {'down': 25, 'raw_acc': 40, 'raw_gyro': 80, 'snr_acc_svm': 13, 'vv':
    7.5, 'gyroX': 75, 'gyroZ': 75,},
]

```

```

def check_conditions2(down, raw_acc, raw_gyro, snr_acc_svm, vv, gyroX, gyroZ):
    detected_modes = []
    for idx, cond in enumerate(conditions):
        if (
            (
                (cond['gyroX'] is None or gyroX > cond['gyroX']) and
                (cond['down'] is None or down >= cond['down']) and
                (cond['raw_acc'] is None or raw_acc < cond['raw_acc']) and

```

```

        (cond['raw_gyro'] is None or raw_gyro > cond['raw_gyro']) and
        snr_acc_svm < cond['snr_acc_svm'] and
        (cond['vv'] is None or vv > cond['vv'])) or

    (
        (cond['gyroX'] is None or gyroX < cond['gyroX']) and
        (cond['gyroZ'] is None or gyroZ > cond['gyroZ']) and
        (cond['down'] is None or down >= cond['down']) and
        (cond['raw_acc'] is None or raw_acc < cond['raw_acc']) and
        (cond['raw_gyro'] is None or raw_gyro > cond['raw_gyro']) and
        snr_acc_svm < cond['snr_acc_svm'] and
        (cond['vv'] is None or vv > cond['vv']))
    ):
        detected_modes.append(idx + 1) # 모드 인덱스는 1부터 시작
return detected_modes

```

연대 데이터에서의 모드별 성능

```

Mode Detection Rates (percentage of total true positives):
Mode 1: 18.06% (130/720)
Mode 2: 1.53% (11/720)
Mode 3: 8.06% (58/720)
Mode 4: 21.25% (153/720)
Mode 5: 3.47% (25/720)
Mode 6: 0.56% (4/720)
Mode 7: 3.89% (28/720)
Mode 8: 0.56% (4/720)
Mode 9: 25.97% (187/720)
Mode 10: 10.28% (74/720)
Mode 11: 1.94% (14/720)
Mode 12: 0.42% (3/720)
Mode 13: 1.11% (8/720)
Mode 14: 0.42% (3/720)
Mode 15: 1.53% (11/720)
Mode 16: 0.28% (2/720)
Mode 17: 0.14% (1/720)
Mode 18: 0.56% (4/720)

Mode False Detection Rates (percentage of total false positives):
Mode 1: 0.00% (0/3)
Mode 2: 0.00% (0/3)
Mode 3: 0.00% (0/3)
Mode 4: 0.00% (0/3)
Mode 5: 0.00% (0/3)
Mode 6: 0.00% (0/3)
Mode 7: 0.00% (0/3)
Mode 8: 0.00% (0/3)
Mode 9: 0.00% (0/3)
Mode 10: 0.00% (0/3)
Mode 11: 100.00% (3/3)
Mode 12: 0.00% (0/3)
Mode 13: 0.00% (0/3)
Mode 14: 0.00% (0/3)
Mode 15: 0.00% (0/3)
Mode 16: 0.00% (0/3)
Mode 17: 0.00% (0/3)
Mode 18: 0.00% (0/3)

```