

YOLOv8 · MediaPipe · GRU 기반 건설현장 폭력 자동 탐지 모델

목차



서론

본론

결론

참고문헌

주제 선정 배경

문제 정의 및 연구 목표

설계 순서도

데이터 수집 및 전처리

모델 학습

분석 결과 및 해석

다른 모델과의 성능 비교

예상 결과

활용 방안

기대효과

건설 산업의 문제, "건설현장 폭력"

경찰청 건설현장 폭력행위 단속 보고서

보도시점 2023. 8. 23.(수) 조간 누리망방송 2023. 8. 22.(화) 12:00

「건설현장 폭력행위 특별단속」 성과

- 불법행위 근절할 때까지 강력한 단속 지속

- 총 4,829명 송치, 이 중 148명 구속/금품 갈취가 3,416명(70.7%)으로 최다(8.14. 기준)
- 조직적으로 갈취행위를 한 5개 단체에 대해 범죄단체(집단)조직죄 적용 / 장애인 없는 장애인노조·사이비 기자·유령 환경단체 검거 등 고질적 폭력행위 척결
- 「건설현장 폭력행위 특별단속」에 준하는 강력한 상시 단속체제 구축, 국토부 등 관련 부처와 협력을 강화하여 불법행위가 재발하지 않도록 엄정 수사 / 피해자 보호 철저 예정, 보복범죄 엄단

뉴스1) 건설현장 집단 폭행

더 차이나

“임금 안 줘서”...공사장 작업반장 집단폭행 중국인들 검거

중앙일보 | 입력 2025.02.20 01:42 업데이트 2025.02.20 08:57

뉴스2) 건설현장 작업자 간 다툼

기획 사설/기고 사람들 독자공간

경기신문

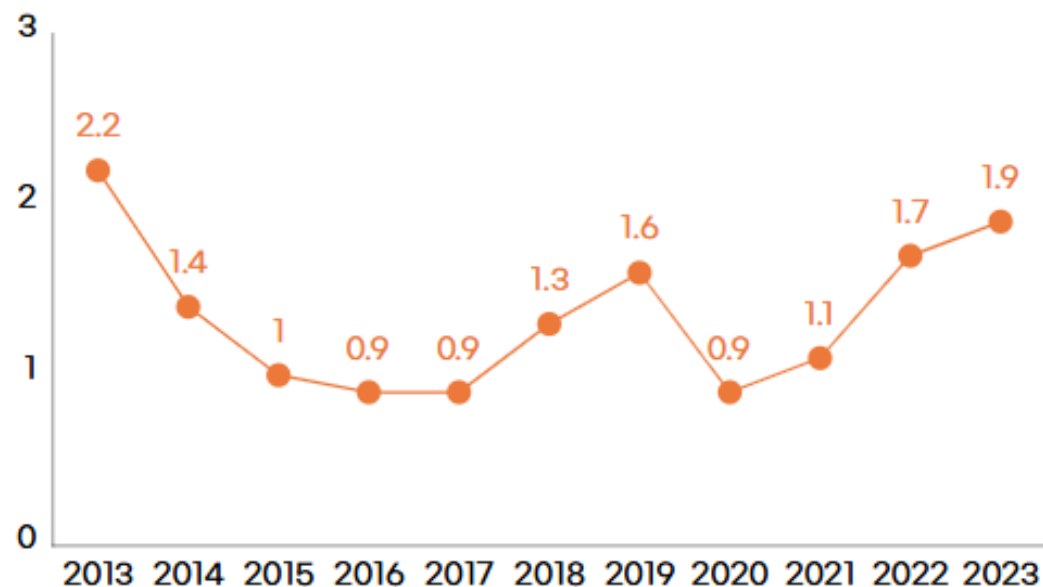
파주 신축 건설현장서 작업자 간 폭행 사태 발생

장진 기자 gija@kgnews.co.kr | 등록 2025.03.13 17:30:12

📄 📧 📞 📱 + -

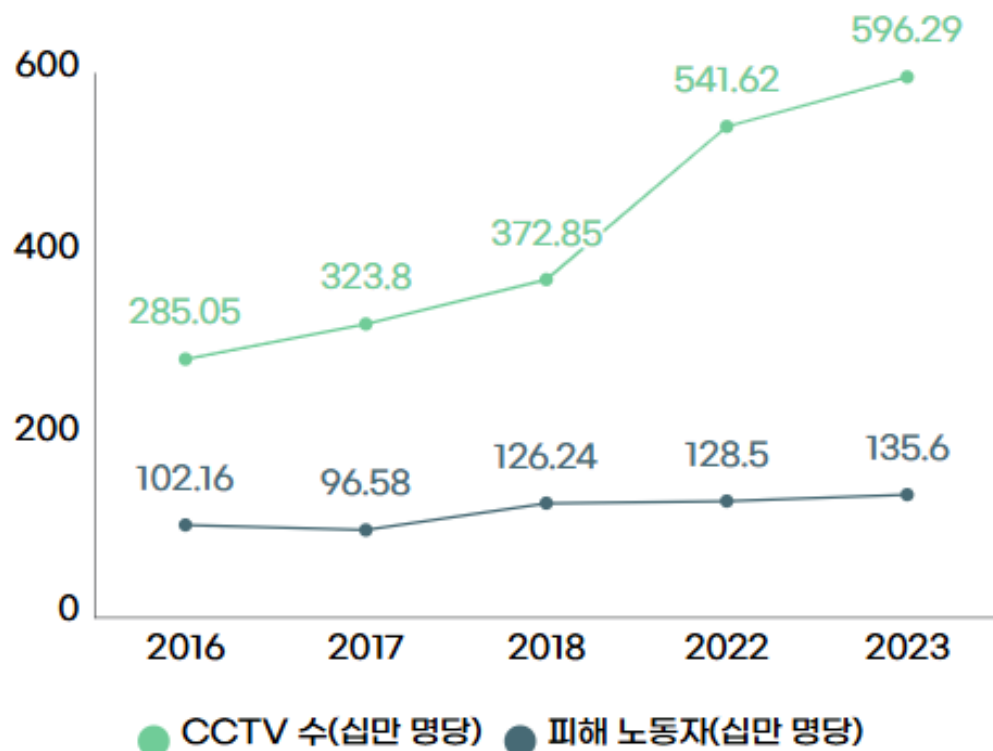
다시 증가하는 "건설현장 폭력"

다시 증가 추세를 보이는 건설현장 폭력 피해율



<건설현장 폭력 피해율, 2013-2023>

증가하는 현장 CCTV 수 & 감소하지 않는 현장 폭력 피해 수



● CCTV 수(십만 명당) ● 피해 노동자(십만 명당)

문제 정의

건설 현장 내 폭력 발생 가능성 존재 및 안전관리의 사각지대

건설 현장은 잦은 인력 변동, 고강도 작업, 높은 스트레스로 인해 작업자 간의 언어/신체적 갈등 및 폭력 발생 위험이 높으나, 현재의 안전 관리 시스템은 사고 예방(추락, 장비 등)에만 초점되어 폭력 예방은 사각지대에 놓여 있음.

CCTV 기반 수동 모니터링의 한계와 실시간 대응 부재

현장 CCTV를 통해 폭력 상황이 녹화될 수는 있지만, 관리 인력이 이를 실시간으로 전부 모니터링하는 것은 불가능하며, 사건 발생 후 사후 확인만 가능하여 즉각적인 개입과 추가 피해 방지가 어려움

기존 산업 AI 모델의 행동(폭력) 분석 복잡성 부재

기존 산업 AI 안전 모델은 안전 장비 착용 여부 등 '정적 상태' 탐지에 집중하며, '폭력'과 같이 시간의 흐름에 따른 복잡한 '행동'을 분석하는 모델은 부재하여 기술적 진보가 필요

연구 목표

“

공사 현장 환경에서 **작업자의 복합적인 행동 변화를 인식**하여
폭력 행위를 실시간으로 자동 탐지하는 고성능 산업 안전 AI 시스템 개발

”

순서도

▶ 전체 Vibe Coding 진행

▶ 활용 Tool: Visual Studio Code(환경) + Cursor(LLM)

데이터 수집

이상행동 데이터
실내(편의점, 매장) 사람 이상행동
데이터

EDA

XML 데이터 분석

데이터 가공

프레임 조정 및 영상 추출
데이터 라벨링

데이터 전처리

객체 탐지

YOLOv8



관절 추론

MediaPipe

모델 학습

시계열 분석

GRU



분석 결과

Accuracy
Precision, Recall, F1 score
ROC curve

성능 비교

다른 모델과의 비교

LSTM
Bidirectional LSTM

실시간 테스트

웹캠을 이용한
실시간 폭력 탐지 테스트

활용 데이터

8가지의 이상행동(전도, 파손, 방화, 흡연, 유기, 절도, 폭행, 이동 약자) 중 폭행 데이터 400건 사용



영상 데이터



라벨링 데이터(XML)

```

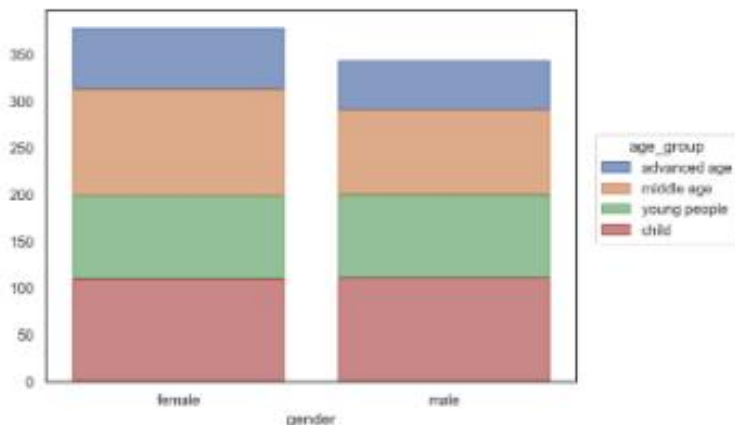
<track id="1245" label="Left knee" source="wma">
  <source frame="109" duration="3" offset="C" <frame"> <source"> <source">125,447" <source">3"
    <attribute name="ID" <value">125,447" />
  </source>
  <source frame="110" duration="1" offset="C" <frame"> <source"> <source">125,447" <source">3"
    <attribute name="ID" <value">125,447" />
  </source>
</track>
<track id="1246" label="Left knee" source="wma">
  <source frame="109" duration="3" offset="C" <frame"> <source"> <source">125,455" <source">3"
    <attribute name="ID" <value">125,455" />
  </source>
  <source frame="110" duration="1" offset="C" <frame"> <source"> <source">125,456" <source">3"
    <attribute name="ID" <value">125,456" />
  </source>
</track>
<track id="1247" label="Left knee" source="wma">
  <source frame="109" duration="3" offset="C" <frame"> <source"> <source">125,547" <source">3"
    <attribute name="ID" <value">125,547" />
  </source>
  <source frame="110" duration="1" offset="C" <frame"> <source"> <source">125,547" <source">3"
    <attribute name="ID" <value">125,547" />
  </source>
</track>

```

데이터 전처리

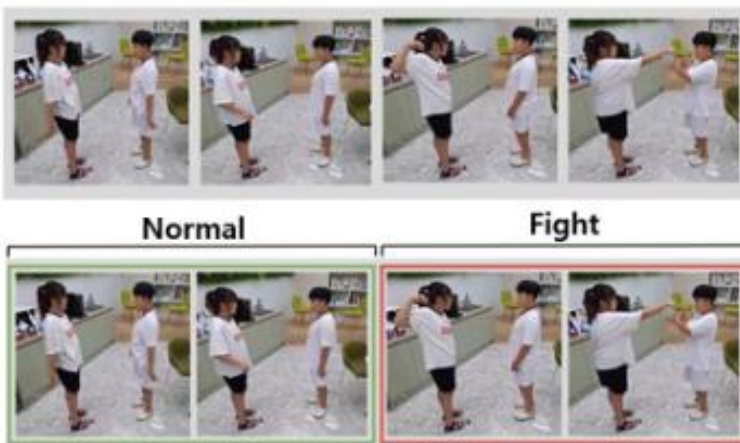
EDA

- 데이터 구조 파악
- XML 내에 있는 폭력 시작, 끝 프레임을 사용하여 폭력 발생 구간 설정



영상 데이터 추출

- moviepy 라이브러리를 이용해 영상 데이터에서 폭력이 발생한 구간, 폭력이 발생하지 않은 구간 각각 10초씩 추출



프레임 조정 및 라벨링

- 폭력 영상은 1, 비폭력 영상은 0으로 라벨링
- 각 영상은 30 fps으로 구성 되어 있으며, 프레임 별로 분할 후 640x640 크기로 조정

label	coordinate
0	[[0.5294928550720215, 0.12906429171562195, 0.5...
1	[[0.8016193509101868, 0.09111222624778748, 0.7...
0	[[0.5931543707847595, 0.10530948638916016, 0.6...
0	[[0.649429440498352, 0.023449301719665527, 0.6...
0	[[0.6384149789810181, 0.11879301071166992, 0.6...

데이터 전처리

객체 탐지

- YOLOv8: 실시간 객체 탐지에 사용되는 대표적인 신경망
- 높은 정확도와 빠른 속도로 다양한 객체를 실시간으로 탐지하는 데 적합
- YOLOv8을 통해 바운딩 박스를 영역으로 설정하여 객체 탐지



관절 좌표 추출

- Mediapipe: 비디오 프레임에서 33개의 관절을 추론하는 자세 추적을 위한 모델
- 실시간성, 유연성이 뛰어남
- 바운딩 박스 내 사람 관절의 좌표를 추출



데이터 증강

- Flipping을 통해 데이터를 2배로 늘려 데이터셋의 다양성을 확보하고 과적합 방지

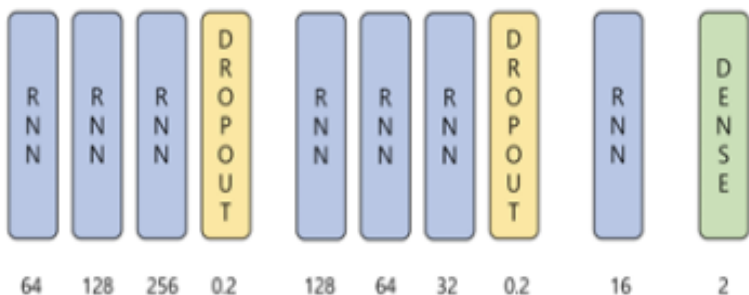


- Flipping을 통해 증강 처리한 데이터를 80%, 10%, 10%의 비율로
Train set(640건),
Validation set(80건),
Test set(80건)
으로 무작위 분할 진행

모델 학습

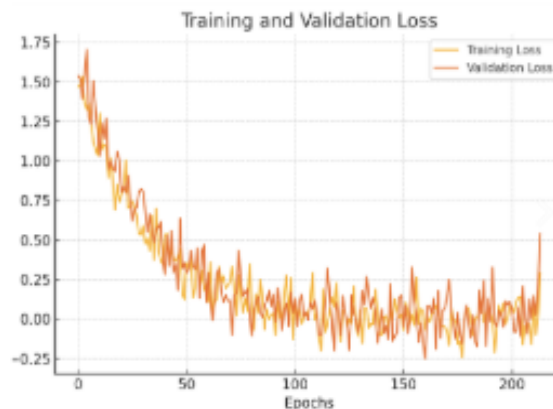
GRU

- 시계열 데이터를 처리하기 위한 순환 신경망(RNN)의 일종으로 LSTM을 단순화한 구조
- 두 개의 게이트(업데이트 게이트와 리셋 게이트)를 사용하여 정보를 효율적으로 처리
- 학습 속도가 빠르고, 비교적 적은 데이터로도 효과적인 성능을 발휘



학습 결과

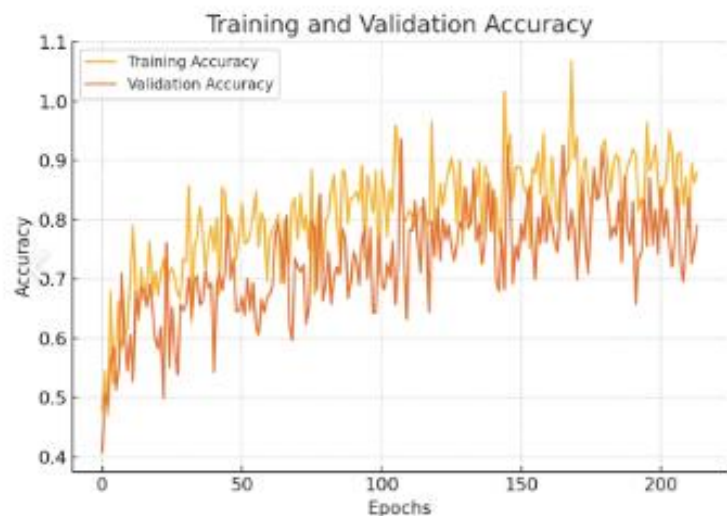
- 전처리한 데이터를 Epoch 300, Batch size 8으로 설정하여 학습
 - Patience 50으로 Early stopping하여 과적합 방지
- 손실 함수: Cross-entropy로, Optimizer: Adam으로 설정
- Train loss: 0.29, Validation loss: 0.54



분석 결과 해석

Accuracy

- Train accuracy: 88%
- Validation accuracy: 79%
- Test accuracy 84%

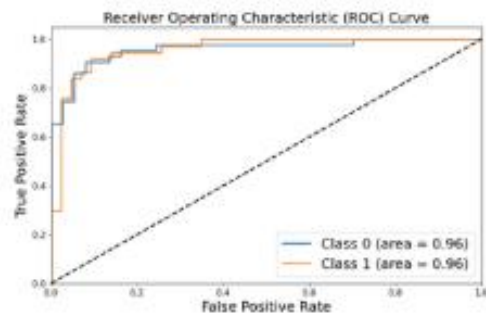


모델 성능 지표

- Precision, Recall, F1 score 측정

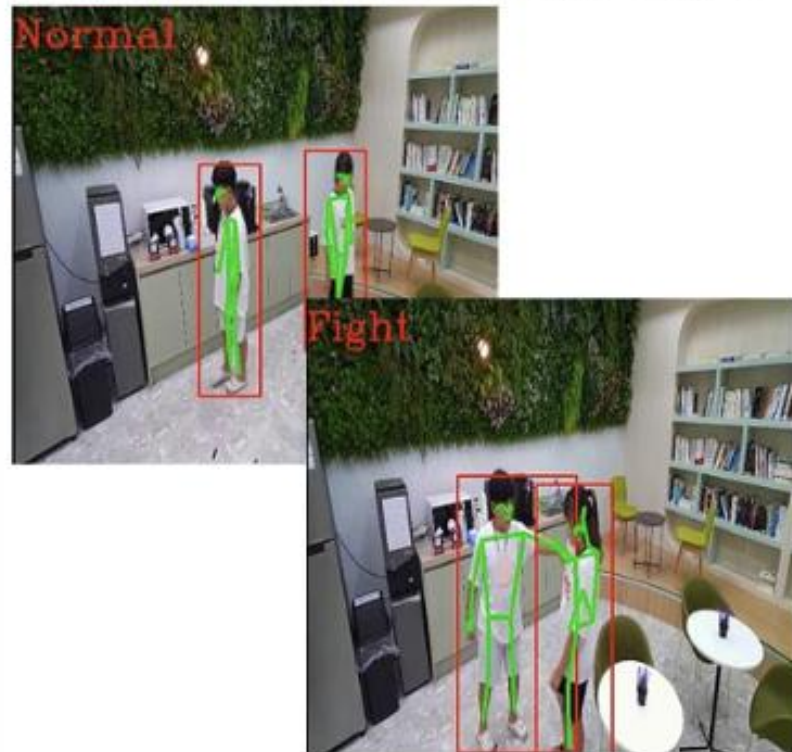
	Precision	Recall	F1 score
비폭력	0.88	0.76	0.82
폭력	0.81	0.9	0.85

- 분류 모델의 전반적인 성능을 평가하는 지표로서 ROC curve를 확인
- AUC(ROC의 면적)이 0.96으로 좋은 성능을 보임



테스트

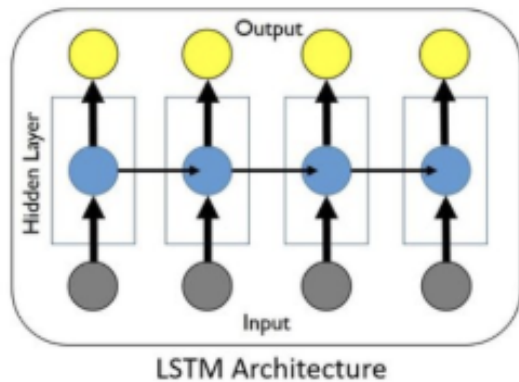
- AI-Hub CCTV 영상을 활용하여 모델 테스트



다른 모델과의 성능 비교

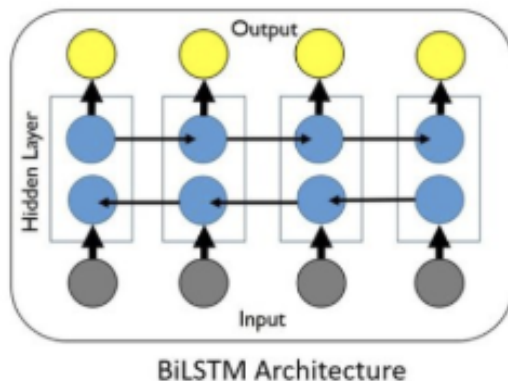
LSTM

- 시계열 데이터를 처리하기 위해 설계된 순환 신경망(RNN)의 일종
- 과거 정보를 모두 반영하는 대신 forget gate 와 input gate를 이용하여 정보를 선별
- 필요한 정보는 많이 채택하고 불필요한 정보는 버리는 방식을 취함으로써 장기 의존성 문제를 해결



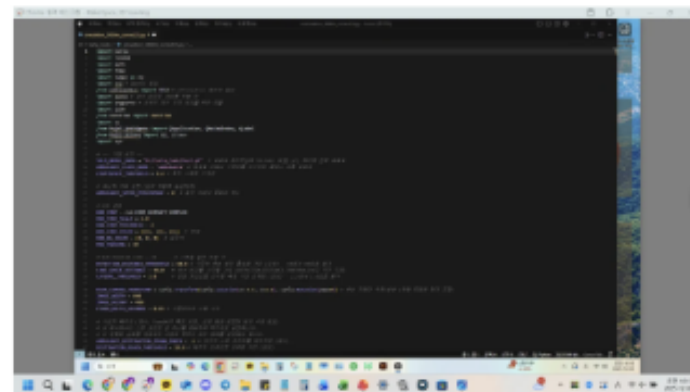
Bidirectional LSTM

- 두 개의 LSTM을 사용하여 입력 시퀀스를 순방향과 역방향으로 처리
- 특정 상황에서는 LSTM 보다 더 뛰어난 성능



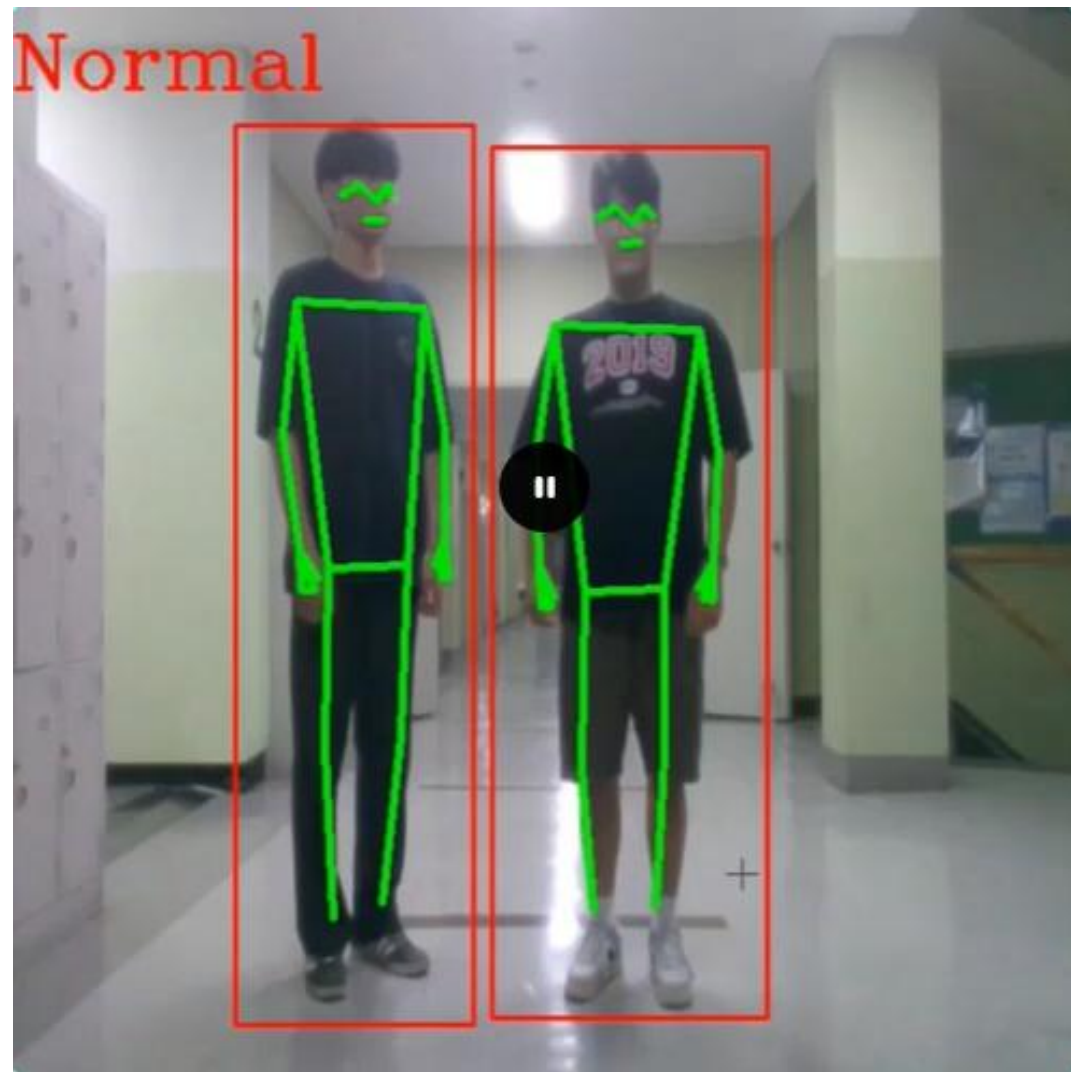
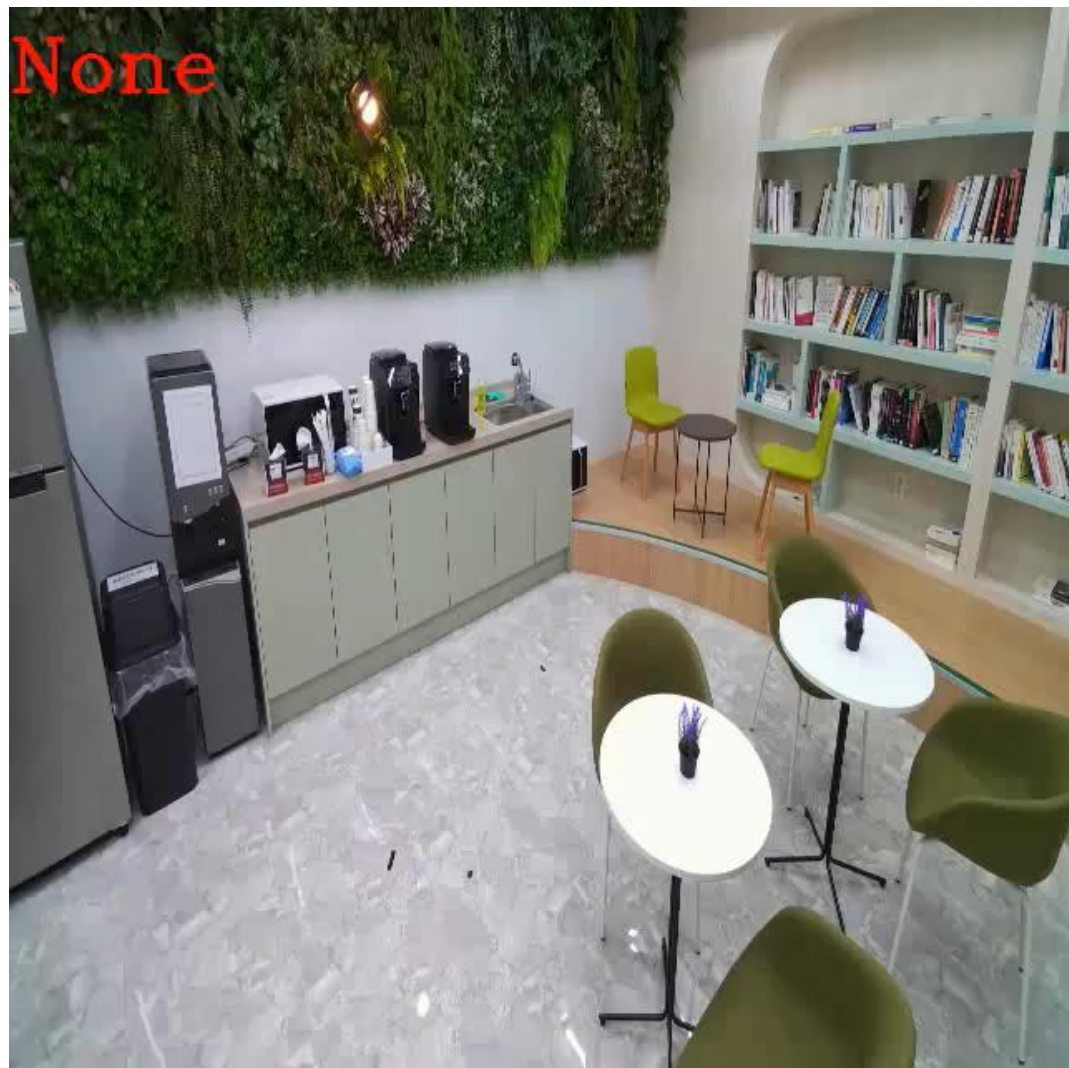
모델 비교

- 본 프로젝트에서 제안한 GRU 모델과 동일한 hyperparameter를 적용시켜 학습한 LSTM, Bidirectional LSTM과 성능을 비교



	Accuracy	Precision	Recall	F1 score
LSTM	0.799	0.824	0.772	0.795
BiLSTM	0.773	0.821	0.733	0.757
GRU	0.805	0.819	0.819	0.806

실시간 테스트



예상 결과



모델 적용 후



활용방안: 실시간 건설현장 폭력 알림 서비스

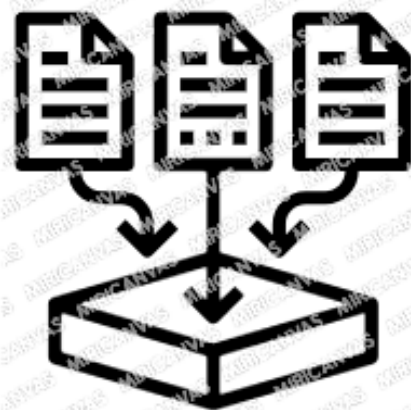
건설현장 폭력 모바일 알림 서비스



실시간 건설현장 폭력 탐지 시스템 기반 어플리케이션

- ▶ 현장 폭력 발생 시, 즉각적 경고 알림 전송
- ▶ 신속한 대응을 통한 건설현장 폭력 피해 최소화
- ▶ 건설 노동자들의 심리적 안정

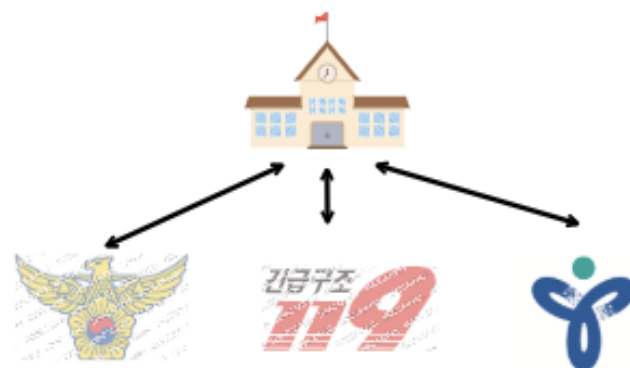
데이터 통합 관리 시스템



데이터 통합 관리 시스템 구축

- ▶ 현장 폭력 실시간 모니터링을 통한 데이터 수집
- ▶ 데이터 분석을 통한 건설현장 폭력 발생 패턴 파악
- ▶ 맞춤형 방안 마련 등 대책수립 가능

협력 네트워크



실시간 협력 지원

- ▶ 경찰서, 소방서, 건설기업과 실시간 연계 및 긴급지원 협력 네트워크 구축

기대 효과

01

실시간 현장 폭력 탐지를 통한 즉각 대응

- ▶ 사후 증거 수집에 불과하던 기존 CCTV의 한계점 극복
 - ➡ 건설현장 폭력 상황에 발빠른 대처

02

데이터 통합 관리 시스템 구축

- ▶ 폭력 종류, 발생 시간, 장소 등 데이터 수집 및 분석 진행
 - ➡ 폭력 발생 위험 지역 · 시간대 도출을 통한 경비 강화 등 예방 조치

03

안전관리 · 관제 인력 부족 문제 해결

- ▶ 24/7 모니터링 및 폭력 탐지시 자동 알림
 - ➡ 업무 부담 경감, 인력 효율성 극대화

04

다른 사회문제 확대적용

- ▶ 폭력 外 다른 범죄(흡연, 욕설, 현장 파손, 절도 등) 영상 데이터를 추가 학습
 - ➡ 건설 현장 내 다른 사회적 문제를 추가적으로 탐지 및 해결하는 모델로 확장

- 통계청, '한국의사회동향', 2023,
- 경찰청, '건설현장 불법행위 단속 결과(반부패공공범죄)', 2024-11-11
- 김평석, '용인 반도체 클러스터 현장서 폭력사건...경찰 조사 중', 뉴스1, 2025-05-07
- 장진, "파주 신축 건설현장서 작업자 간 폭행 사태 발생", 경기신문, 2025-03-13
- 고은지, "CCTV 관제센터 1인당 평균 958대 담당...적정수준 19배", 연합뉴스, 2022-11-09
- 류경근, 최용철, & 이덕규. (2020). GRU를 활용한 악성코드 탐지의 관한 연구. 한국정보처리학회 학술대회논문집, 27(1), 254-257.
- 이재진, 홍현지, 송재민, & 염은섭. (2021). Gated recurrent unit (GRU) 신경망을 이용한 적혈구 침강속도 예측. 한국가시화정보학회지, 19(1), 57-61
- e-나라지표, '건설현장 폭력 행위 및 산업재해현황'
- 심태웅, 김도윤, 최종인, & 박광영. YOLOv5 기반의 폐기물 자동인식 시스템 설계
- Kwon, Y., & Kim, D. (2022). Real-Time workout posture correction using OpenCV and MediaPipe. 한국정보기술학회논문지, 20(1), 199-208
- Chen, K. Y., Shin, J., Hasan, M. A. M., Liaw, J. J., Yuichi, O., & Tomioka, Y. (2022). Fitness 10 movement types and completeness detection using a transfer-learning-based deep neural network. Sensors, 22(15), 5700.
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016).
- LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.
- 정성우, 김은철, & 유준혁. (2021). CAPS: CCTV 영상을 이용한 자율형 딥러닝 기반 아동학대 감지 시스템. 제어로봇시스템학회 논문지, 27(12), 1029-1037
- 박성우, 정승민, 문재욱, & 황인준. (2022). BiLSTM 기반의 설명 가능한 태양광 발전량 예측 기법. 정보처리학회논문지. 소프트웨어 및 데이터 공학, 11(8), 339-346
- 주일택, & 최승호. (2018). 양방향 LSTM 순환신경망 기반 주가예측모델. 한국정보전자통신기술학회 논문지, 11(2), 204-208

활용 코드 첨부 (Visual Studio Code + Cursor)

```
import cv2
import numpy as np
import mediapipe as mp
from ultralytics import YOLO
import tensorflow as tf
from collections import deque

# --- 설정 ---
SEQUENCE_LENGTH = 30 # GRU에 들어갈 프레임 수 (편도우 크기)
CONFIDENCE_THRESHOLD = 0.5 # 폭력 판정 임계값

# 1. 모델 로드 (학습된 모델이 없으면 다시 모델로 작동)
try:
    model = tf.keras.models.load_model('violence_detection_gru.h5')
    print("학습된 GRU 모델을 로드했습니다.")
    USE_MODEL = True
except:
    print("경고: 'violence_detection_gru.h5' 파일을 찾을 수 없습니다.")
    print("테스트를 위해 랜덤 예측 모드로 실행합니다.")
    USE_MODEL = False

# 2. YOLO 및 MediaPipe 초기화 (PDF Page 6, 9)
yolo_model = YOLO('yolov8n.pt') # 가벼운 모델 사용
mp_pose = mp.solutions.pose
pose = mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5)
mp_drawing = mp.solutions.drawing_utils

# 3. 데이터 전처리 함수 (PDF Page 8, 9)
def extract_keypoints(results):
    if results.pose_landmarks:
        # 33개의 랜드마크에서 x, y, z, visibility 추출 후 평면화 (flatten)
        landmarks = np.array([[res.x, res.y, res.z, res.visibility] for res in results.pose_landmarks.landmark]).flatten()
    else:
        # 판지되지 않으면 0으로 채움
        landmarks = np.zeros(33 * 4)
    return landmarks

# 4. 웹캠 실행 (PDF Page 13)
cap = cv2.VideoCapture(0)
sequence = deque(maxlen=SEQUENCE_LENGTH) # 최근 30프레임 저장용 큐
label = "Normal"
color = (0, 255, 0) # 초록색

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # 영상 크기 조정 (속도 최적화)
    frame = cv2.resize(frame, (640, 480))

    # YOLOv8로 객체 탐지 (사람만 필터링: class 0 is person)
    # PDF Page 9: 바운딩 박스를 임의로 설정
    yolo_results = yolo_model(frame, classes=0, verbose=False)

    # YOLO 박스 그리기 및 ROI 설정
    for result in yolo_results:
        boxes = result.boxes
        for box in boxes:
            # 바운딩 박스 좌표
            x1, y1, x2, y2 = map(int, box.xyxy[0])

            # MediaPipe 적용을 위해 ROI 추출 (옵션)
            # 현재 화면에 대해 Pose를 추정하는 것이 굳이 필요할 수 있으나,
            # PDF 의도를 살려 사람 영역을 중심으로 처리한다고 가정하거나,
            # 단순히 시각화를 위해 YOLO 박스를 사용합니다.

            # --- MediaPipe Pose 추정 ---
            # 이미지를 RGB로 변환
            img_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            pose_results = pose.process(img_rgb)

            # 관절 좌표 그리기 (PDF Page 13 처럼)
            if pose_results.pose_landmarks:
                mp_drawing.draw_landmarks(frame, pose_results.pose_landmarks, mp_pose.POSE_CONNECTIONS)

            # 키포인트 추출 및 시퀀스에 추가
            keypoints = extract_keypoints(pose_results)
            sequence.append(keypoints)

            # 데이터가 충분히 모였을 때(30프레임) 예측 수행
            if len(sequence) == SEQUENCE_LENGTH:
                if USE_MODEL:
```

```
                # 4. 웹캠 실행 (PDF Page 13)
                cap = cv2.VideoCapture(0)
                sequence = deque(maxlen=SEQUENCE_LENGTH) # 최근 30프레임 저장용 큐
                label = "Normal"
                color = (0, 255, 0) # 초록색

                while cap.isOpened():
                    ret, frame = cap.read()
                    if not ret:
                        break

                    # 영상 크기 조정 (속도 최적화)
                    frame = cv2.resize(frame, (640, 480))

                    # YOLOv8로 객체 탐지 (사람만 필터링: class 0 is person)
                    # PDF Page 9: 바운딩 박스를 임의로 설정
                    yolo_results = yolo_model(frame, classes=0, verbose=False)

                    # YOLO 박스 그리기 및 ROI 설정
                    for result in yolo_results:
                        boxes = result.boxes
                        for box in boxes:
                            # 바운딩 박스 좌표
                            x1, y1, x2, y2 = map(int, box.xyxy[0])

                            # MediaPipe 적용을 위해 ROI 추출 (옵션)
                            # 현재 화면에 대해 Pose를 추정하는 것이 굳이 필요할 수 있으나,
                            # PDF 의도를 살려 사람 영역을 중심으로 처리한다고 가정하거나,
                            # 단순히 시각화를 위해 YOLO 박스를 사용합니다.

                            # --- MediaPipe Pose 추정 ---
                            # 이미지를 RGB로 변환
                            img_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                            pose_results = pose.process(img_rgb)

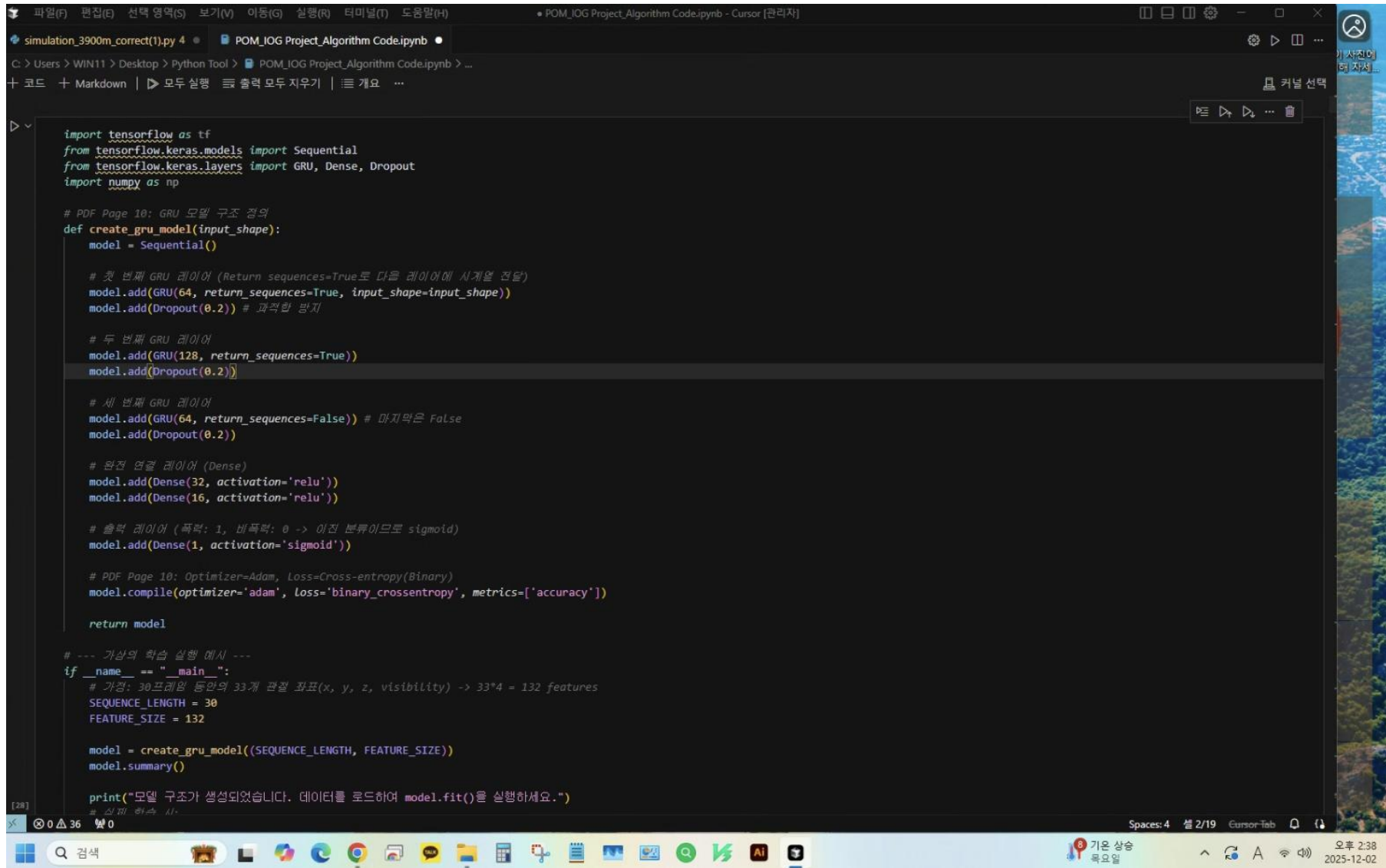
                            # 관절 좌표 그리기 (PDF Page 13 처럼)
                            if pose_results.pose_landmarks:
                                mp_drawing.draw_landmarks(frame, pose_results.pose_landmarks, mp_pose.POSE_CONNECTIONS)

                            # 키포인트 추출 및 시퀀스에 추가
                            keypoints = extract_keypoints(pose_results)
                            sequence.append(keypoints)

                            # 데이터가 충분히 모였을 때(30프레임) 예측 수행
                            if len(sequence) == SEQUENCE_LENGTH:
                                if USE_MODEL:
```

▶ 실시간 폭력 탐지 코드(YOLOv8 + MediaPipe)

활용 코드 첨부 (Visual Studio Code + Cursor)



```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Dropout
import numpy as np

# PDF Page 10: GRU 모델 구조 정의
def create_gru_model(input_shape):
    model = Sequential()

    # 첫 번째 GRU 레이어 (Return sequences=True로 다음 레이어에 시계열 전달)
    model.add(GRU(64, return_sequences=True, input_shape=input_shape))
    model.add(Dropout(0.2)) # 과적합 방지

    # 두 번째 GRU 레이어
    model.add(GRU(128, return_sequences=True))
    model.add(Dropout(0.2))

    # 세 번째 GRU 레이어
    model.add(GRU(64, return_sequences=False)) # 마지막은 False
    model.add(Dropout(0.2))

    # 완전 연결 레이어 (Dense)
    model.add(Dense(32, activation='relu'))
    model.add(Dense(16, activation='relu'))

    # 출력 레이어 (출력: 1, 비확률: 0 -> 이진 분류이므로 sigmoid)
    model.add(Dense(1, activation='sigmoid'))

    # PDF Page 10: Optimizer=Adam, Loss=Cross-entropy(Binary)
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

    return model

# --- 가상의 학습 실행 예시 ---
if __name__ == "__main__":
    # 가정: 30프레임 동안의 33개 관측 좌표(x, y, z, visibility) -> 33*4 = 132 features
    SEQUENCE_LENGTH = 30
    FEATURE_SIZE = 132

    model = create_gru_model((SEQUENCE_LENGTH, FEATURE_SIZE))
    model.summary()

    print("모델 구조가 생성되었습니다. 데이터를 로드하여 model.fit()을 실행하세요.")
    # 실제 학습 코드
```

▶ GRU 모델 구조 및 학습 코드

Thank you 😊