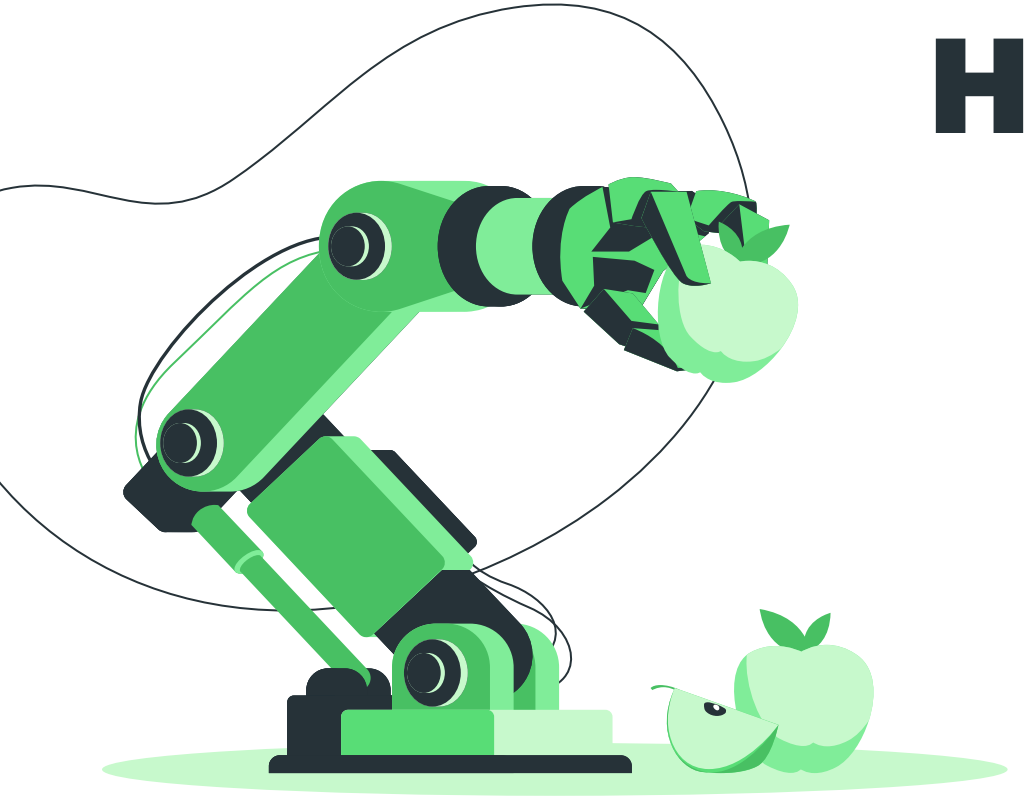


Human Fall Detection System

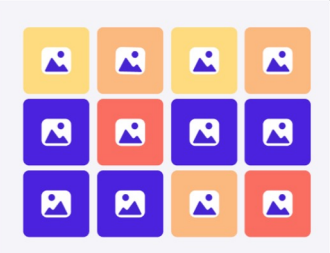
Yinzhou Lu, Hang Yu

Overview

- The project was inspired by recent incidents where some elderly people were not taken care of in time because of falls.
- My goal is to create a system that can detect human movements in real time and determine what state they are in.

- Real-Time Processing
- Adaptability and Accessibility

System Include



Database

The database contains a large number of images as well as tags and people locations corresponding to the images.

Deep Learning (Yolo v5)

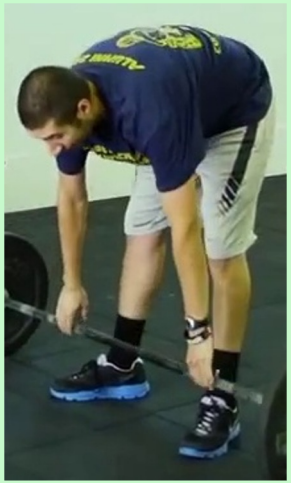
We use a deep learning framework: yolo v5 for real-time object detection. We need to train the data with this framework to get the model that we need.

Camera

In this system, we call the camera of the mobile device to replace the camera in the real world.

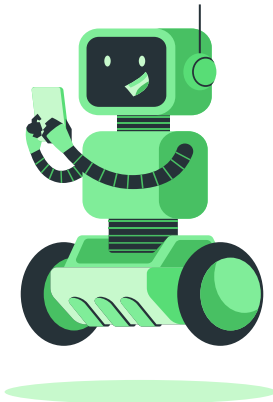
Database: 3 types of images

Up

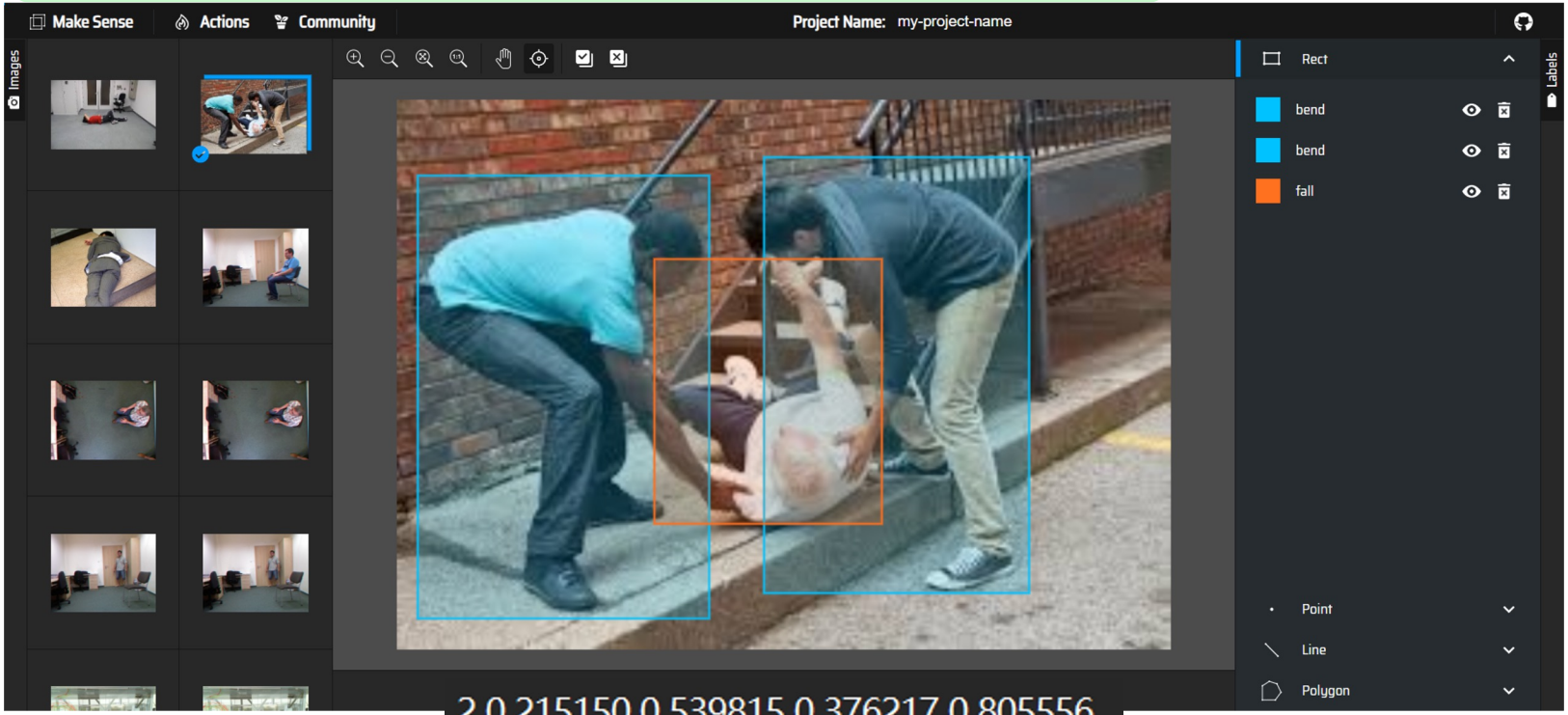


Bend

Fall

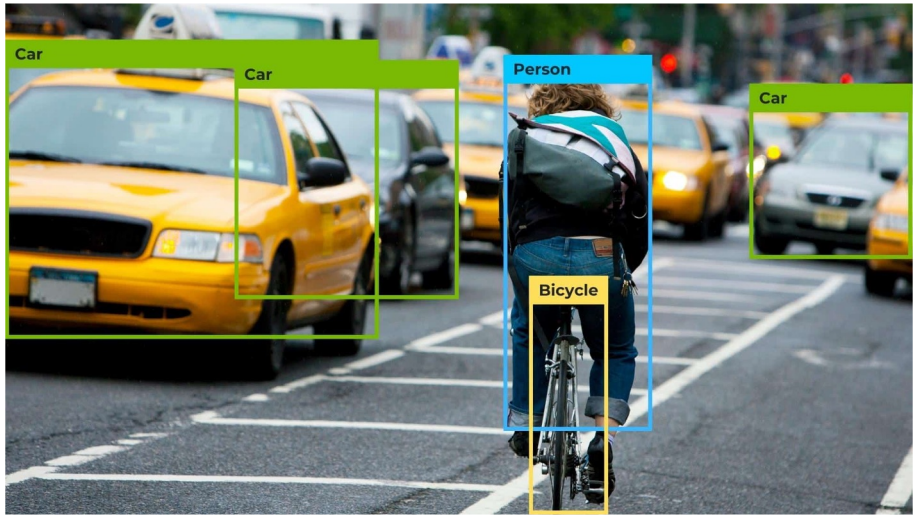


makesense.ai do label

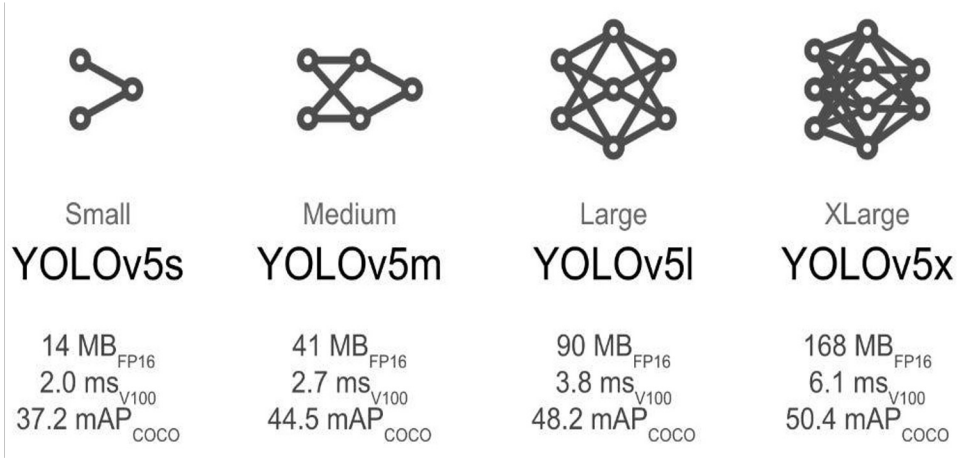


Deep Learning Model (Yolo v5)

- YOLO v5 (You Only Look Once version 5) is a popular deep learning framework for real-time object detection.
- The core idea of the YOLO algorithm is to predict the bounding boxes and categories of objects simultaneously through a single neural network model.



Yolo v5 Pre-trained model



This shows the performance comparison of pre-trained models of different sizes in the YOLOv5 series.

- Model size
- Training speed
- Accuracy

However, due to my computer and time constraints, I used the smallest yolov5s model.

Parameter we changed

Batch size

Batch size refers to the number of samples used for each parameter update during training. We changed it from 16 to 32

Learning rate

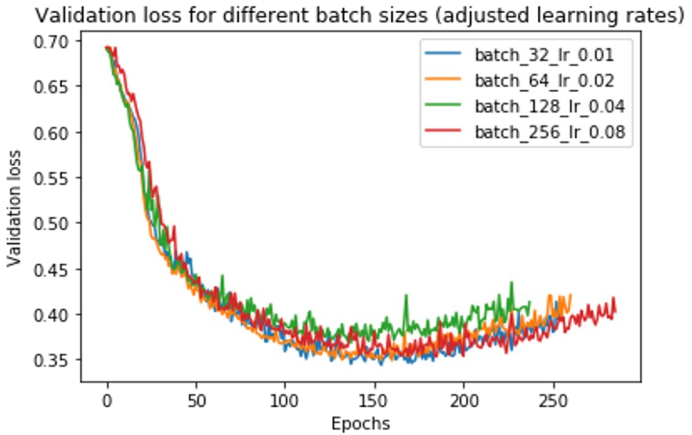
The learning rate determines the magnitude of updating model weights during gradient descent. We reduced the value from 0.01 to 0.005.

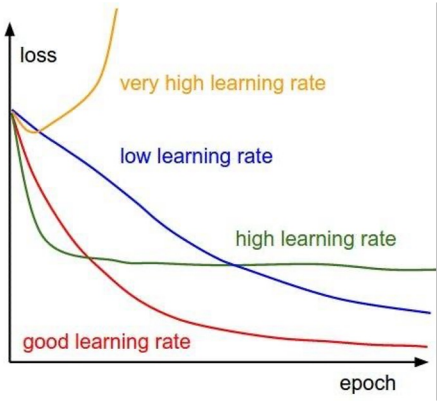
Epoch

More epochs ensure that the model fully learns and adapts to the training data. We increased the epoch from 20 to 40.

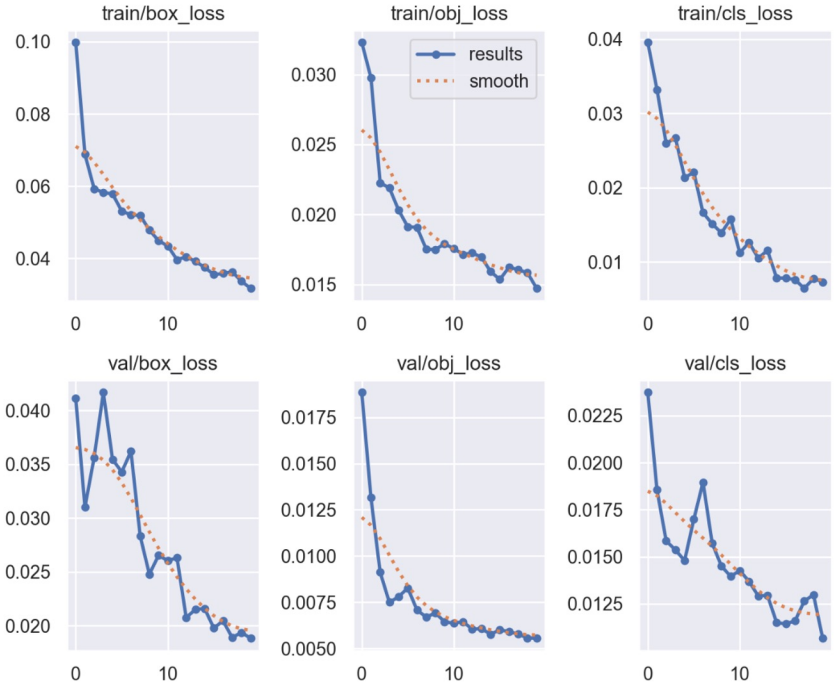
Optimizer

Optimizer is an algorithm used to update and adjust network parameters. The default optimizer for yolov5 is SGD, which we changed to Adam.

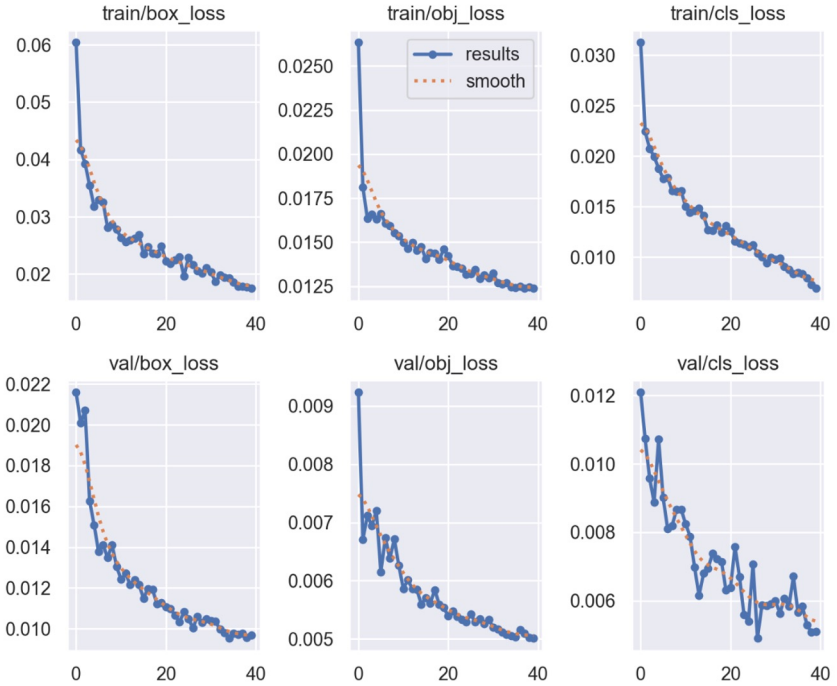




Various indicators in the training process

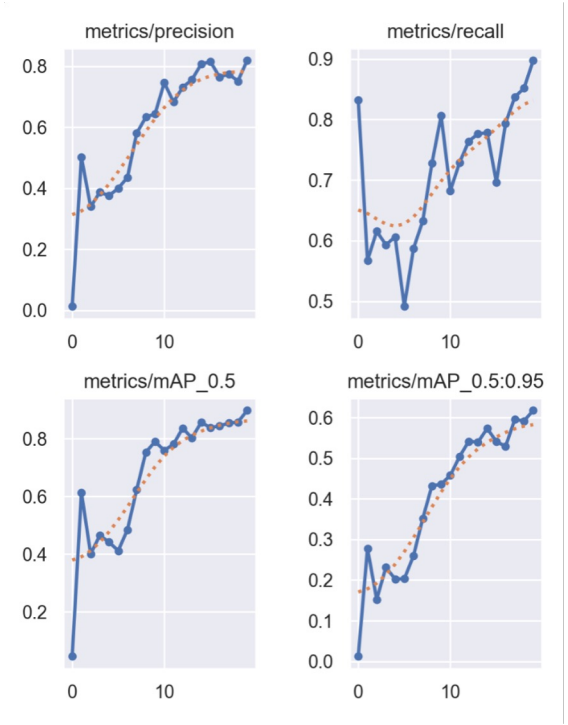


Original

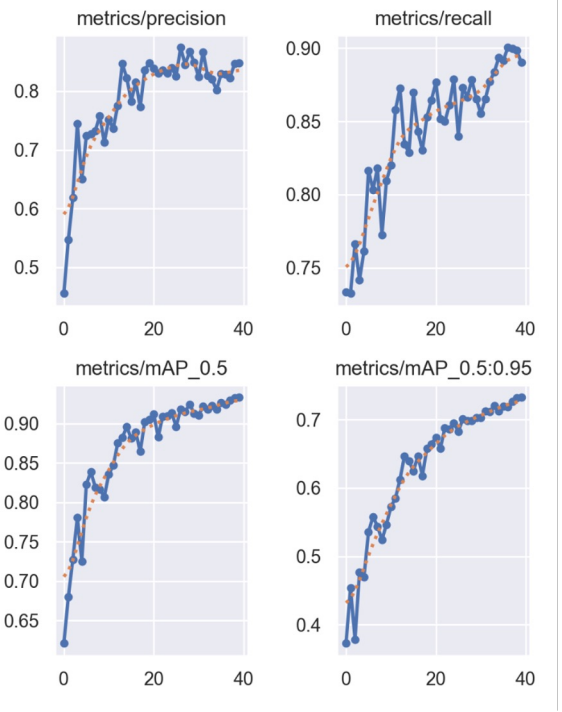


Improved

Various indicators in the training process

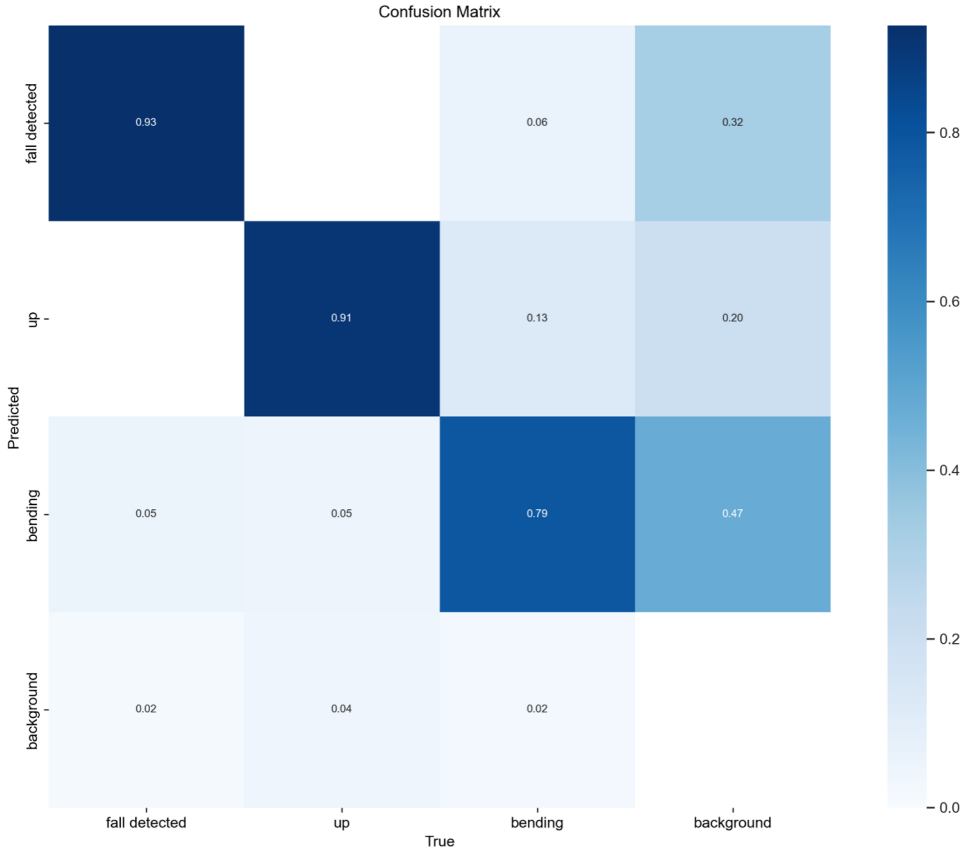


Original



Improved

Confusion Matrix



Camera part

Real Time Monitor

We can call the mobile equipment camera to display it on the computer and simulate the real-time monitoring picture.

Camera code

```python
import cv2
import torch

from ultralytics.utils.plotting import Annotator, colors
model = torch.hub.load('ultralytics/yolov5', 'custom', path='runs/train/40_epochs/weights/best.pt',
                       force_reload=True).to('cuda' if torch.cuda.is_available() else 'cpu')
# Capture a video stream using OpenCV's VideoCapture function
url = 'http://admin:admin@192.0.0.2:8081/video'
cap = cv2.VideoCapture(url)
while cap.isOpened():
    # Reads frames in a video stream
    ret, frame = cap.read()
    # If the frame is read correctly, it is displayed
    if not ret:
        print("Failed to grab frame")
        break
    img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model([img], size=640)

    annotator = Annotator(frame, line_width=2, example=str(model.names))
    for *xyxy, conf, cls in results.xyxy[0]:
        c = int(cls)
        label = f'{model.names[c]} {conf:.2f}'
        annotator.box_label(xyxy, label, color=colors(c, True))
    cv2.imshow('YOLOv5 Detection', annotator.result())
    # Press 'q' to exit
    if cv2.waitKey(1) == ord('q'):
        break
# Release the catcher and close all OpenCV Windows
cap.release()
cv2.destroyAllWindows()
```

IP Camera

The Python code utilizes the OpenCV library to capture and display a video stream from an IP camera.


bending 0.79

Reference

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. 2020.

[2] Jos´e Camilo Eraso Guerrero, Elena Mu˜noz Espa˜na, Mariela Mu˜noz A˜nasco, and Jes´us Emilio Pinto Lopera. Dataset for human fall recognition in an uncontrolled 9 environment. Data in Brief, 45:108610, 2022.ISSN 2352-3409.doi: https://doi.org/10.1016/j.dib.2022.108610. URL https://www.sciencedirect.com/science/article/pii/S2352340922008162.

[3] Jun Peng, Yuanmin He, Shangzhu Jin, Haojun Dai, Fei Peng, and Yuhao Zhang.Improved yolov5 method for fall detection.pages 504–509, 2022.doi:10.1109/ICIEA54703.2022.10006129.

[4] Guto Leoni Santos, Patricia Takako Endo, Kayo Henrique de Carvalho Monteiro,Elisson da Silva Rocha, Ivanovitch Silva, and Theo Lynn. Accelerometer-based hu-man fall detection using convolutional neural networks.Sensors, 19(7), 2019. ISSN1424-8220. doi: 10.3390/s19071644. URLhttps://www.mdpi.com/1424-8220/19/7/1644.

[5] Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don’t decaythe learning rate, increase the batch size, 2018.

[6] Thanh-Hai Tran, Thi-Lan Le, Dinh-Tan Pham, Van-Nam Hoang, Van-Minh Khong,Quoc-Toan Tran, Thai-Son Nguyen, and Cuong Pham. A multi-modal multi-viewdataset for human fall analysis and preliminary investigation on modality.pages1947–1952, 2018. doi: 10.1109/ICPR.2018.8546308.

[7] Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012.10


Thank you
for
listening!