Research on real-time human fall detection method based on YOLOv5-Lite

Baolei Chen1, Jianming Zhuo2, Kai Tang3, Jianhang Zou3, Shikai Zuo1, and Yijun Cai1,\*

1Smart Sensing Integrated Circuit Engineering Research Center of Universities in Fujian Province, Xiamen University of Technology, Xiamen 361024, China

2Xiamen Blue Ocean Technology Co., Ltd, Xiamen 361022, China

3Xiamen Wish Information Technology Co., Ltd, Xiamen 361006, China

\*yijuncai@foxmail.com

*Abstract—Fall detection is an important application with various applications in fields such as healthcare and safety monitoring. This paper presents a fall detection method based on the lightweight object detection model YOLOv5-Lite. The proposed method uses a custom dataset for training and optimizing the model to achieve precise detection of human fall states in real-time scenarios. Accuracy, recall rate, mAP, and FPS are used to evaluate the model's performance. Experimental results show that when the IoU is 0.5, the accuracy of the model is 92.1%, with a detection accuracy of 94.0% and a recall rate of 85.6% for fall states. Regarding the running speed, the model runs at 3.12ms FPS when using a camera with a 640×480 size as input on an RTX3060 (6G) graphics card. This study has practical and promotional value and provides technical support and reference for human fall detection in practical scenarios such as elderly care facilities and public areas.*

*Keywords—Fall detection, YOLOv5-Lite, Custom dataset, Real-time scenarios*

# INTRODUCTION

During the period of 2015 to 2018, the National Institute of Senior Services (NISS) collected a total of 205,670 cases of falls among elderly individuals. Research indicates that the highest proportion of falls among the elderly population occurs in their own homes. Falls resulted in moderate to severe injuries in 37.21% of cases, and 22.49% of elderly individuals required hospitalization as a result [1]. Falling poses a significant threat to the health of elderly individuals[2] [3], therefore, proposing a method for detecting falls in the elderly population is crucial.

Existing fall detection technologies can be divided into three categories: wearable, vision-based, and environment-based deployments. Wearable devices are a novel technology, typically comprising an accelerometer and an algorithm, as well as more complex sensors such as barometers [4]. Although this solution is cost-effective and relatively sensitive, wearable devices can be cumbersome to use, and elderly individuals may forget to wear them.Gelmini et al. proposed a method for detecting high falls and triggering the deployment of protective airbags using the gravity algorithm [5]. The authors used a series of signal processing stages to extract the slow dynamic related to the gravitational effect. In experimental data testing, the method was found to detect fall events and protect the elderly effectively. However, considering the inconvenience of using this device for daily travel and the possibility of accidental injury due to misfiring, caution should be exercised.

In the field of computer vision, Nogas et al. proposed a new framework called DeepFall to solve the fall detection problem as an anomaly detection problem. The DeepFall framework uses deep spatio-temporal convolutional autoencoders to learn spatial and temporal features of normal activities and employs a non-invasive sensing mode [6]. Shu et al. proposed a whole-house fall detection system that can detect different types of falls, including falls, slips, fainting, and other types. The system installs eight cameras indoors and analyzes fall patterns using machine learning and rule-making, monitored by conventional available surveillance systems [7]. This approach can be combined with the popular YOLO series object detection algorithms to accurately calculate the location and information of the fallen person. Jun Peng et al. also improved the YOLOv5 algorithm. The method added an ECA module to the main network of YOLOv5s and changed PANet to BiFPN for weighted fusion of different dimension features extracted from the neck network [8]. The experimental results on the Le2i public fall dataset showed that the accuracy of the improved method increased by 2.7 percentage points, which is an improvement over other mainstream algorithms.

The YOLO series algorithms have fast detection speeds and high accuracy. YOLO5-Lite has a model accuracy comparable to YOLOv5 and a parameter size smaller than YOLOv8, which can meet the needs of fall detection. Therefore, this article selects YOLOv5-lite [9], uses genetic algorithm to adjust parameters, trains using cosine annealing method and mosaic data set augmentation to obtain a lightweight and efficient fall detection model.

# YOLOV5-LITE DETECTION ALGORITHM

1. YOLOv5-Lite Network Model

The YOLO algorithm is a one-stage object detection algorithm, which stands for You Only Look Once, meaning that the neural network only needs to scan the input image once to output the detection results.YOLOv1 laid the foundation for the entire series, and subsequent improved versions have continued to innovate. Its core idea is to transform the object detection problem into a regression problem, using the convolutional neural network structure to directly predict bounding boxes and class probabilities. YOLOv5 introduces new improvement ideas based on YOLOv4, significantly improving detection speed and accuracy. The network structure of YOLOv5 can be divided into three parts: Backbone (main part), Neck, and Head. The main part mainly uses CSPDarknet53 as the backbone network, the Neck uses the SPP structure, and the Head uses the YOLOv5 Head structure.Compared with YOLOv4, the entire network structure is more lightweight.

YOLOv5-Lite is a lightweight version of YOLOv5, mainly aimed at industrial applications and optimized for Arm architecture processors. Its network structure is relatively simple compared to YOLOv5, with the main part mainly using shuffle blocks containing shuffle channels, and the detection head using a truncated version of the YOLOv5 Head. The network architecture of YOLOv5-Lite is shown in Figure 1.

1. Input Method

The input method of YOLOv5-Lite uses the same Mosaic data enhancement method as YOLOv5 and YOLOv4. The Mosaic data enhancement combines four images together to enrich the image background. The image size processing can adaptively add the minimum amount of black border to facilitate resizing all images to the same size. Adaptive anchor box calculation is to adaptively calculate a set of anchor boxes based on the size and aspect ratio of the targets in the training set, in order to better train the model.

1. Backbone

The backbone of YOLOv5-Lite mainly uses shuffle blocks containing shuffle channels. The ShuffleNet Block is an image model block that uses channel shuffle operations and deep convolution for efficient architecture design. The ShuffleNet Block is part of the ShuffleNet architecture. The entire process of the Shuffle Block can be divided into three stages: Split, Transform, and Fuse. In the Split stage, the channels of the feature map are divided into two parts. In the Transform stage, each part is convolved and reorganized. In the Fuse stage, the two parts are connected. This design makes the Shuffle Block have features such as partial computation and efficient short paths.

1. Neck

The Neck of YOLOv5-Lite is used to generate the feature pyramid and adopts the PANet structure. PANet is a multiscale feature fusion method that can fuse features at different scales to improve detection accuracy.

1. Prediction

The Prediction part of YOLOv5-Lite includes three feature maps for prediction, which are 40×40×255, 20×20×255, and 10×10×255. During the training process, YOLOv5-Lite uses the BCEWithLogits Loss to calculate the average loss of objectness and classification, and uses CIoU to calculate the loss value of bounding box regression. Specifically, the formula for calculating CIoU is as Equation(1):

|  |  |
| --- | --- |
|  | (1) |

Where  represents the Euclidean distance between the center points of the predicted box

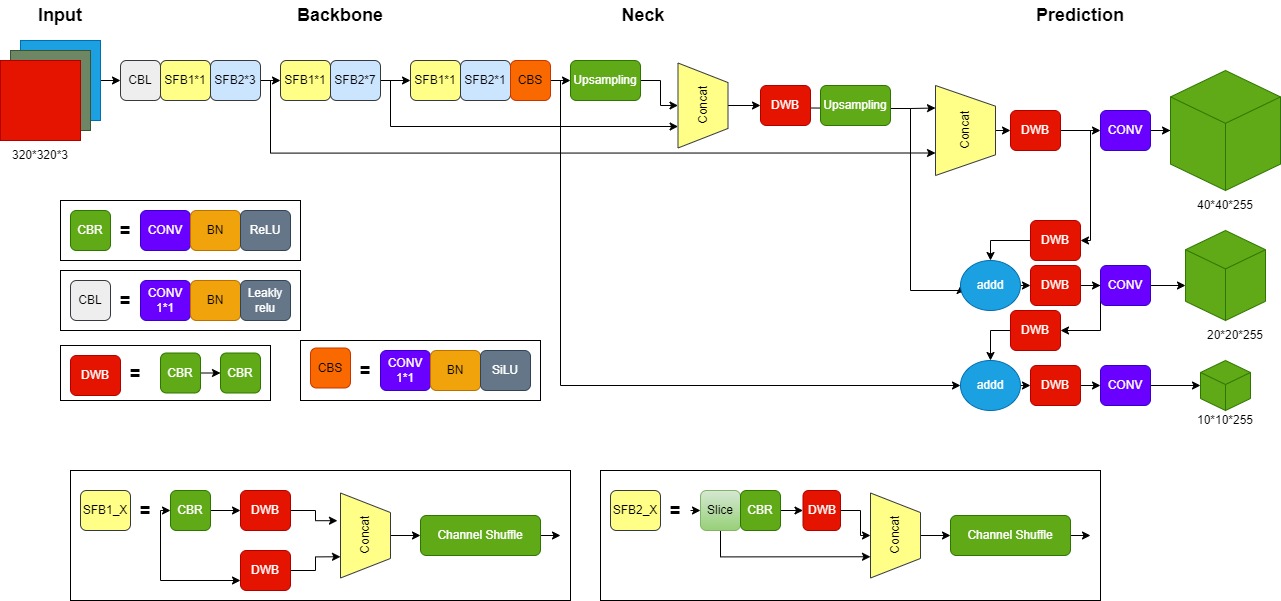


Fig. 1.YOLOv5-lite Network [11]

and the ground truth box. c represents the diagonal distance of the minimum closed region that can contain both the predicted box and the ground truth box. The formulas for a and  are shown in Equations (2) and (3), respectively:

|  |  |
| --- | --- |
|  | (2) |

|  |  |
| --- | --- |
|  | (3) |

# ESTABLISHMENT OF THE FALLING DETECTION DATASET

Currently, available datasets for the task of fall detection are relatively scarce. In this study, we collected images from multiple sources and created a custom dataset for fall detection. The original dataset used for training was annotated by the original author using the makesense.ai website [10], with human poses categorized into three states: ‘Fall’, ‘Walking’, and ‘Sitting’. The training set consisted of 374 images, and the validation set consisted of 111 images. However, we found that directly using this dataset to train models did not yield satisfactory results. Therefore, we further collected 1634 images from the internet, including human poses of ‘Fall’, ‘Walking’, and ‘Sitting’, and annotated them with bounding boxes using labelimg, and converted them into the YOLO format. These images were mainly obtained from news photos. Finally, we obtained a dataset containing 2119 images.

TABLE 1 HYPERPARAMETERS

|  |  |
| --- | --- |
| Parameter | value |
| img-size | (640,640) |
| initial learning rate | 0.00116 |
| final OneCycleLR learning rate | 0.236 |
| momentum | 0.965 |
| weight decay | 0.00043 |
| warmup epochs | 2.96 |
| warmup momentum | 0.799 |
| warmup bias lr | 0.1 |
| Box loss gain | 0.0467 |
| Cls loss gain | 0.0519 |
| Obj loss gain | 1.26 |
| cls BCELoss positive weight | 1.08 |

# EXPERIMENT

1. Computer Configuration

The computer configuration used for training was as follows: CPU was Intel Core i5-10400F, 4000 MHz (40 x 100), with 8GB of RAM; GPU was RTX3060 (12G/ASUS); the operating system was Windows10; the software used for the training environment was Pycharm, python version 3.7, CUDA version 11.6, and PyTorch framework 1.3.

1. Model Training

The pre-trained weights used were v5lite-s, with the file tags and paths saved in fall.yaml. The initial hyperparameters were obtained using hyp.scratch.yaml through hyperparameter evolution on the COCO dataset, where the best hyperparameters for this dataset were obtained using a genetic algorithm. After 5 rounds of hyperparameter evolution for 400 epochs, the optimal hyperparameters were obtained. The batch size was set to 48, and the number of epochs was set to 80. For other hyperparameters, please refer to Table 1.

# EXPERIMENTAL RESULTS ANALYSIS

1. Evaluation Metrics

For the performance evaluation of the YOLOv5-Lite algorithm, we mainly consider the following metrics: precision, recall, mean Average Precision (map@0.5), and the parameter FPS (Frames Per Second) related to image processing speed. The specific calculation formulas are as Equation(4) ,(5) and(6):

|  |  |
| --- | --- |
|  | (4) |

In which TP represents the number of true positive samples, FP represents the number of false positive samples, and FN represents the number of false negative samples.

|  |  |
| --- | --- |
|  | (5) |
|  | (6) |

1. Analysis of Training Results

When Iou is set to 0.5, the detection accuracy of all categories for YOLOv5-Lite is 92.1%, with fall category having a detection accuracy of 94.0% and a recall rate of 85.6%. The training recall and precision results are shown in Figures 2 and 3 Using an RTX3060 (6G) graphics card and a 640×480 input size, the detection speed is 3.12ms per frame. As the epochs approach 30, the precision and recall rates tend to converge.

1. Actual Detection Results

Figure 4 illustrates the actual detection results of the model,where (a) shows the real-time monitoring results obtained by directly accessing the laptop camera, with a real-time frame rate of between 25-33 frames per second. Figures (b), (c), and (d) show the detection results from images obtained from the internet. It is worth noting that even when only the upper body is visible, the model can accurately recognize the posture of the person, which fully demonstrates the effectiveness of the model for practical use. However, there may be false detections when the image is blurry or only a small part of the person’s body is visible.

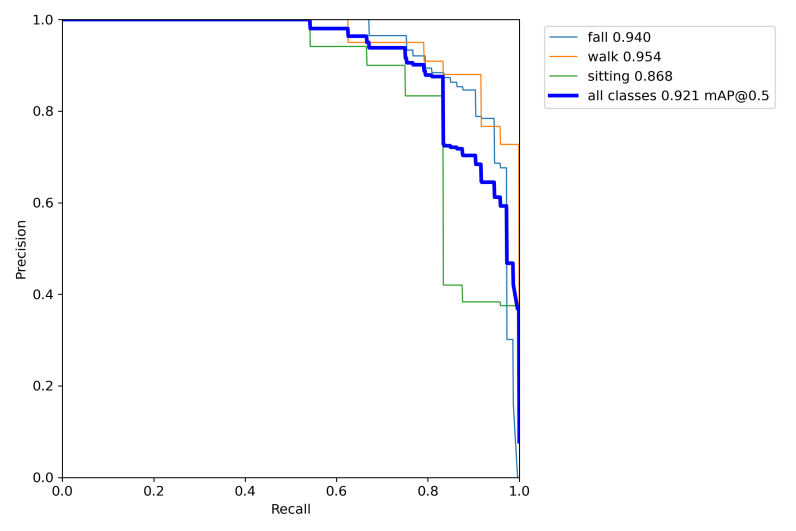


Fig. 2. PR\_curve

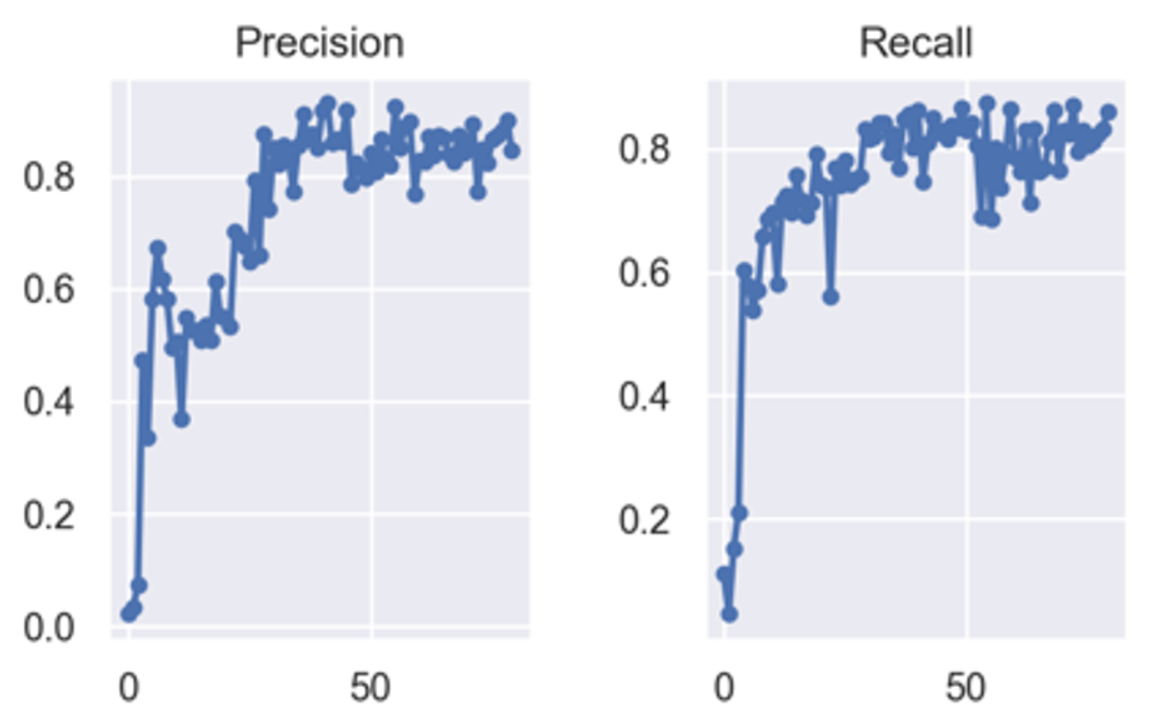


Fig. 3 Recall and accuary results



Fig. 4 Actual Detection Results

# Ⅵ. CONCLUSION

In conclusion, this paper proposed a human fall detection method based on the YOLOv5-Lite model. We kept the original network architecture and collected over 2100 images to construct a comprehensive dataset, and ultimately obtained a network model with an accuracy of 92.1% and a processing speed of 3.12ms. Through experimental validation, the model demonstrated real-time detection speed and accuracy, meeting the requirements of human fall detection. Future research will consider modifying the detection head, improving model scale and fitting ability, especially for recognizing human posture in blurry images.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Fujian Province (2020J01294, 2020J01295).

REFERENCES

1. Lu Zhiming, Wang Yuan, Ye Pengpeng, Er Yuliang, Duan Leilei. Analysis on epidemiologic characteristics of fall in old people: results from Chinese National Injury Surveillance System, 2015-2018. Chinese Journal of Epidemiology, 2021, 42(1): 137-141.
2. Wang Z, Wang X, Li X, et al. Expert consensus on fall risk assessment for elderly people in China (draft) [J]. Chinese Journal of Geriatric Health Care, 2019, 17(04):47-48+50. (in Chinese)
3. Sun L, Wu X, Yang J. Application of structured nursing program based on geriatric comprehensive assessment in prevention of falls among elderly people outside hospital [J]. Nursing Research, 2019, 33(02):355-358. (in Chinese)
4. Warrington, D.J., Shortis, E.J. & Whittaker, P.J. Are wearable devices effective for preventing and detecting falls: an umbrella review (a review of systematic reviews). BMC Public Health 21, 2091 (2021).https://doi.org/10.1186/s12889-021-12169-7
5. S. Gelmini, S. Strada, M. Tanelli, S. Savaresi, A. Guzzon, Automatic detection of human’s falls from heights for airbag deployment via inertial measurements, Automation in Construction, Volume 120, 2020, 103358, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2020.103358. (https:// www.sciencedirect.com/science/article/pii/S0926580520309389)
6. Shu, F., Shu, J. An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box. Sci Rep 11, 2471 (2021). https://doi.org/10.1038/s41598-021-81115-9
7. Nogas, J., Khan, S.S. & Mihailidis, A. DeepFall: Non-Invasive Fall Detection with Deep Spatio-Temporal Convolutional Autoencoders. JHealthc Inform Res 4, 50–70 (2020). https://doi.org/10.1007/s41666-019-00061-4
8. J. Peng, Y. He, S. Jin, H. Dai, F. Peng and Y. Zhang, ”Improved YOLOv5 Method for Fall Detection,” 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 2022, pp. 504- 509, doi: 10.1109/ICIEA54703.2022.10006129.
9. Chen, X., & Gong, Z. (2021). YOLOv5-Lite: Lighter, faster and easier to deploy. Zenodo. https://doi.org/10.5281/zenodo.5241425
10. Uttej Kumar Kandagatla. Fall Detection Dataset [DB/OL]. (2021-09-30)[2022-04-01]. https://www.kaggle.com/datasets/uttejkumarkandagatla/fall-detection- dataset.
11. Chen, L. (2021, October 15). YOLOv5-Lite tutorial: Explaining all the principles and ideas, training your own dataset, and deploying to mobile devices. Zhihu. https://zhuanlan.zhihu.com/p/478630138