LIGHTWEIGHT FALL DETECTION ON EDGE DEVICES VIA KNOWLEDGE DISTILLATION

Quang Nhut Pham, Duy Khanh Ninh*

The University of Danang – University of Science and Technology, Vietnam

*Corresponding author: nkduy@dut.udn.vn

(Received: April 16, 2025; Revised: June 08, 2025; Accepted: June 20, 2025)

DOI: 10.31130/ud-jst.2025.23(9D).568E

Abstract - Falls pose a significant risk to the elderly, often leading to severe injuries and medical emergencies. Timely and accurate fall detection is crucial for effective intervention. While vision-based approaches offer high accuracy and non-intrusive monitoring, they typically require large-scale deep learning models, making deployment on resource-constrained edge devices impractical due to high computational demands. To address this challenge, we propose an efficient fall detection framework using knowledge distillation to transfer knowledge from a high-performance teacher model to a compact student model. This approach significantly reduces model complexity without sacrificing accuracy. We apply cross-background and cross-person validation for robust evaluation. Experimental results show our model improves F1-score by up to 7% while requiring only 1/200 of the teacher's parameters, making it suitable for real-time edge deployment.

Key words - Keypoint fall detection; Spatial-Temporal Graph Convolutional Networks; knowledge distillation; edge device.

1. Introduction

Falls are a major public health concern, ranking as the second leading cause of unintentional injury-related deaths worldwide. Each year, approximately 684,000 fatalities occur, with older adults being the most vulnerable group [1]. In Vietnam, fall incidence increases with age, peaking at 32.7% among those over 80 [2]. If untreated, fall-related complications can persist for months [3].

Fall detection systems are categorized into wearable and non-wearable approaches [4]. Wearable solutions provide high accuracy but suffer from user discomfort and limited battery life [5]. Non-wearable methods, particularly vision-based systems, enable unobtrusive monitoring but require efficient real-time processing [6].

Recent studies have explored model optimization techniques to enable fall detection on edge devices [7, 8, 9]. However, there remains a significant gap in applying knowledge distillation (KD) to compress computer vision-based models, especially skeleton-based architectures, for fall detection tasks with real-time edge deployment. To the best of our knowledge, no existing work has combined KD with optimized skeleton-based vision models for this purpose.

This study proposes a lightweight skeleton-based fall detection system optimized for real-time deployment. We leverage KD [10] to compress the Spatial-Temporal Graph Convolutional Networks (ST-GCN) [11] model into a 3-layer architecture with only 17,023 parameters while maintaining high accuracy. The model achieves a 7% F1-score improvement and is deployable on edge devices. Our main contribution is a lightweight ST-GCN model optimized with KD for real-time fall detection on edge devices, striking a practical balance between efficiency and accuracy.

2. Related works

Human activity recognition, particularly fall detection can be broadly categorized into sensor-based and vision-based approaches. While sensor-based systems rely on wearable devices to capture motion data, vision-based methods use cameras to track human movement without requiring any physical contact. The latter offers a more convenient and non-intrusive alternative, with recent advances enabling the extraction of skeletal features for accurate activity recognition [12].

2.1. Vision-based fall detection

Vision-based fall detection typically uses RGB, depth, and infrared (IR) data, with RGB favored for its accessibility and low cost [12]. Recent deep learning models exploit spatio-temporal features for higher accuracy [13].

Skeleton-based methods are popular in HAR and fall detection due to their efficiency and interpretability [12, 14]. Using pose estimation like YOLO [15], models such as LSTMs and GCNs can capture spatial-temporal dependencies [16, 17]. While optimization techniques like lightweight models [7], quantization [8], and parameter pruning [9] improve performance, advanced training strategies for balancing compression and accuracy remain limited. To address this, we propose using KD to optimize fall detection models.

2.2. Knowledge distillation

KD enables knowledge transfer from a high-capacity teacher model to a smaller student model, improving efficiency while maintaining performance. Hinton et al. introduced the foundational KD framework using soft targets to guide the student model [10]. Over time, various KD strategies have emerged, including relational and structural KD [18], adversarial learning [19], and self-distillation [20].

Although KD has been extensively applied in action recognition [21], it remains unexplored for fall detection. This study bridges that gap by applying KD to optimize fall detection models on benchmark datasets [22-25].

3. Fall detection framework using KD

This work focuses on improving fall detection by leveraging KD for skeleton-based action. The main hypothesis is that the student model can achieve comparable performance to the teacher model while significantly reducing computational complexity, making real-time applications more feasible. The KD method in Section 4 is applied to transfer knowledge from a high-capacity teacher to a compact student, enhancing the student's performance on fall detection tasks.

3.1. Comparative results

Given a skeleton sequence represented as a graph $G = (V, E)$, where V denotes the body joints and E represents the skeletal connections, the Spatio-Temporal Graph Convolutional Network (ST-GCN) is employed to extract spatio-temporal features from the sequence. The core operation of ST-GCN is defined as:

$$H^{(l+1)} = \sigma\left(\sum_k^{\Lambda_k^{-1/2}} A_k \Lambda_k^{-1/2} H^{(l)} W_k\right), \quad (1)$$

where A_k is the normalized adjacency matrix encoding joint connectivity, $H^{(l)}$ is the feature matrix at layer l, W_k is the learnable weight matrix, and σ is a non-linear activation function.

To transfer knowledge from the teacher ST-GCN model to the student ST-GCN model, KD is employed. This process combines classification loss and distillation loss. The distillation loss is computed using the Kullback–Leibler (KL) divergence between the softened output distributions of the teacher and student models:

$$L_{KL} = T^2 \sum_{i=1}^{N} p_T(y_i) log\left(\frac{p_T(y_i)}{p_S(y_i)}\right), \quad (2)$$

where $p_T(y_i)$ and $p_S(y_i)$ represent the output probabilities of the teacher and student models, respectively, softened by a temperature parameter T:

$$p(y_i) = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}, \quad (3)$$

where z_i is the logit output for class i. The standard cross-entropy loss for the ground-truth label is

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left|\left|f_T(x_i) - f_S(x_i)\right|\right|_2^2, \quad (4)$$

where N is the batch size, and the squared L2 norm corresponds to the Mean Squared Error (MSE) between the teacher and student features.

The three losses cross-entropy, distillation, and feature are combined into a single objective

$$L = \alpha L_{KL} + \beta L_{MSE} + (1 - \alpha - \beta) L_{CE}, \quad (5)$$

where α balances between the supervised cross-entropy loss and the KD loss, and β controls the strength of the feature-level mimicry.

By minimizing this objective, the student ST-GCN not only learns to reproduce the softened output distribution of the teacher but also aligns its internal representations, leading to more effective and robust distillation while maintaining a compact model, as illustrated in Figure 1.

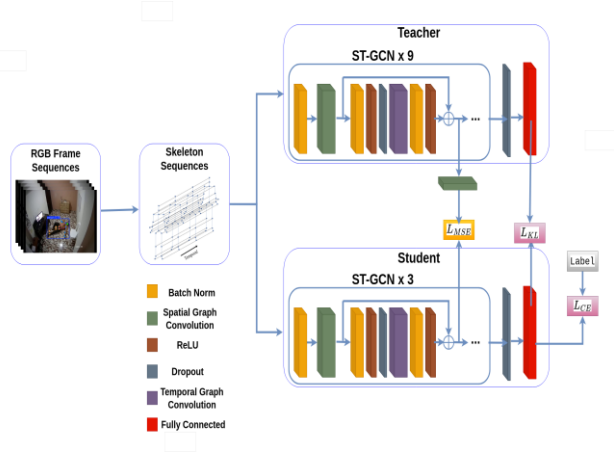

Figure 1. *Workflow of KD framework for skeleton-based fall detection*

3.2. Feature extraction and evaluation

To enable efficient fall detection on edge devices, we employ YOLOv11 [15] Pose to extract 2D skeletal keypoints from video frames, where each keypoint includes pixel coordinates and a confidence score (see Equation 6). To ensure robustness across varying resolutions and camera perspectives, we apply two normalization techniques: Min-Max Normalization (Equation 7), which scales keypoints to a [0,1] range, and Body-Length Normalization (Equations 8 and 9), which normalizes keypoints relative to the Euclidean distance between the neck and the midpoint of the hips. These steps yield resolution-independent features suitable for skeleton-based detection on resource-constrained platforms.

$$K = (x_i, y_i, c_i), i \in (1, \ldots, N), \quad (6)$$

$$\hat{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \hat{y}_i = \frac{y_i - y_{min}}{y_{max} - y_{min}}, \quad (7)$$

$$L = \left|\left|L_{neck} - L_{hip}\right|\right|, \quad (8)$$

$$\tilde{x}_i = \frac{x_i - x_{hip}}{L}, \tilde{y}_i = \frac{y_i - y_{hip}}{L}. \quad (9)$$

For performance evaluation, we use Leave-One-Subject-Out (LOSO) cross-validation, where in each iteration, data from one subject is held out for testing while the rest is used for training (Equation 10). This ensures the model generalizes to unseen individuals. We report the F1-score as the primary evaluation metric, defined in Equation 11 as the harmonic mean of precision and recall, which are computed using Equations 12 and 13. The F1-score provides a balanced measure of performance, especially in scenarios with class imbalance, making it a reliable indicator of fall detection effectiveness.

$$S_{train} = S \setminus S_i, S_{test} = S_i, \quad (10)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \quad (11)$$

$$Precision = \frac{TP}{TP + FP}, \quad (12)$$

$$Recall = \frac{TP}{TP + FN}. \quad (13)$$

4. Experiments

4.1. Dataset

We evaluate our approach using multiple publicly available datasets, including CaucaFall, GMDCSA-24, FallVision, URFall, and UPFall, which vary in sample size, environment, and subject diversity. To ensure consistency and fair comparison, we standardize all datasets by merging various fall-related actions into a single Fall class and all other activities into a Non-Fall class. This binary classification allows for a unified performance evaluation across different datasets despite variations in annotation schemes (see Table 1).

Table 1. *Datasets used in experiments*

Dataset	Samples	Subjects	Cross-Validation
CaucaFall [22]	100	10	Cross-Person
GMDCSA-24 [23]	162	4	Cross-Person
FallVision [24]	6,864	4	Cross-Person
URFall [25]	100	30	Cross-Background
UPFall [7]	1,007	17	Cross-Person

4.2. Implementation details

As shown in Table 2, our experiments utilize one student model and two teacher models with varying capacities. The student model is a lightweight 3-layer ST-GCN composed of three STGC blocks with 8, 16, and 32 channels, respectively. In contrast, the teacher models are deeper and more expressive: a 4-layer ST-GCN with 64, 128, 256, and 512 channels, and a 9-layer ST-GCN configured with blocks of 64×4, 128×3, and 256×3 channels. These configurations provide a range of teacher-student complexity for evaluating the effectiveness of the KD approach. All models are trained for 60 epochs using the Adam optimizer (learning rate = 0.001, weight decay = 1e−5), with a multi-step learning rate scheduler (milestones at 30, 40, and 50 epochs, decay factor = 0.1). Training is performed with weighted cross-entropy loss, and model performance is evaluated using the F1-score. For KD, we employ a temperature T = 4 and balance the loss with soft target loss weight = 0.7 and cross-entropy loss weight = 0.3. The student model learns from both teacher models through KD. Model training and evaluation are conducted on a GPU-accelerated setup.

Table 2. *List of models used in the experiments*

Model	Layers	Parameters	Runtime (ms)
Student (3-layer)	(8, 16, 32)	17,203	4.003
Teacher (4-layer)	(64, 128, 256, 512)	3,662,286	9.655
Teacher (9-layer)	(64×4, 128×3, 256×3)	3,031,584	7.885

4.3. Results

Table 3 presents the performance comparison between the teacher and student models across different datasets. The teacher model, which is significantly larger in terms of parameters, achieves higher F1 scores and lower loss values. However, when applying KD, the student model improves its performance, closing the gap with the teacher model. Notably, the F1 score of the student model with KD surpasses the baseline student model across all datasets, demonstrating the effectiveness of knowledge transfer.

In addition to accuracy metrics, we also compare the inference efficiency of our student model with other baseline models. Table 4 shows the average inference time per sample and the corresponding frames per second (FPS) achieved on the same hardware. As shown, our lightweight student model achieves significantly faster inference, making it more suitable for real-time applications, particularly in embedded or resource-constrained environments.

Table 3. *Performance comparison between student and teacher models across different datasets*

Dataset	Student model	Teacher model	Teacher		Student		Student with KD	
			Loss	F1	Loss	F1	Loss	F1
CaucaFall [22]	3-layer (8, 16, 32)	4-layer (64, 128, 256, 512)	0.1267	0.9687	0.3553	0.8398	0.3674	0.9054
GMDCSA-24 [23]	3-layer (8, 16, 32)	9-layer (64×4, 128×3, 256×3)	0.2254	0.9302	0.3894	0.8600	0.3067	0.9069
FallVision [24]	3-layer (8, 16, 32)	4-layer (64, 128, 256, 512)	0.3545	0.8802	0.5604	0.8008	0.4074	0.8339
URFall [25]	3-layer (8, 16, 32)	4-layer (64, 128, 256, 512)	0.1323	1.0000	0.3018	0.7868	0.3115	0.8571
UPFall [7]	3-layer (8, 16, 32)	9-layer (64×4, 128×3, 256×3)	0.2828	0.9288	0.2735	0.9155	0.2167	0.9265

Table 4. *Comparison of frame rate across different model optimization approaches*

Model	FPS
LSTM [21]	4.15
RNN [21]	3.90
MLP [21]	4.45
3D-CNN [22]	2.15
Lightweight OpenPose + RNN-LSTM [23]	10.00
Distilled ST-GCN (Ours)	4.20

4.4. Edge device deployment

To enable real-time human fall detection on edge platforms, we deploy our system on the NVIDIA Jetson Nano, a low-power, ARM-based device with a 128-core Maxwell GPU and 4GB RAM (see Figure 2). Given its resource constraints, both the YOLOv11 Pose and ST-GCN models are converted to ONNX format for compatibility with optimized inference engines such as TensorRT. The ST-GCN model is further quantized to 8-bit precision, reducing memory usage and accelerating computation while preserving accuracy. The implementation is built using Python, with OpenCV for video handling and ONNX Runtime for model execution. Keypoints are collected in a sliding window

of 32 frames and passed to the ST-GCN model once the buffer is filled.

Despite these optimizations, real-time performance remains limited. YOLO inference takes approximately 0.2 seconds per frame, even with ONNX acceleration. As a result, the system is most suitable for low-frame-rate or event-triggered applications, such as fall detection or gesture classification, where frame skipping is acceptable. This setup showcases how deep learning pipelines for pose-based fall detection can be adapted to run efficiently on edge devices, enabling privacy-aware, always-on systems without relying on cloud-based processing.

Figure 2. *The edge device, NVIDIA Jetson Nano Developer Kit B01, used in the experiments*

In our deployment, the pipeline runs successfully on the Jetson Nano, achieving an average inference speed of 4.2 FPS, demonstrating the feasibility of edge-based human fall detection with constrained resources (see Figure 3 for detailed system implementation).

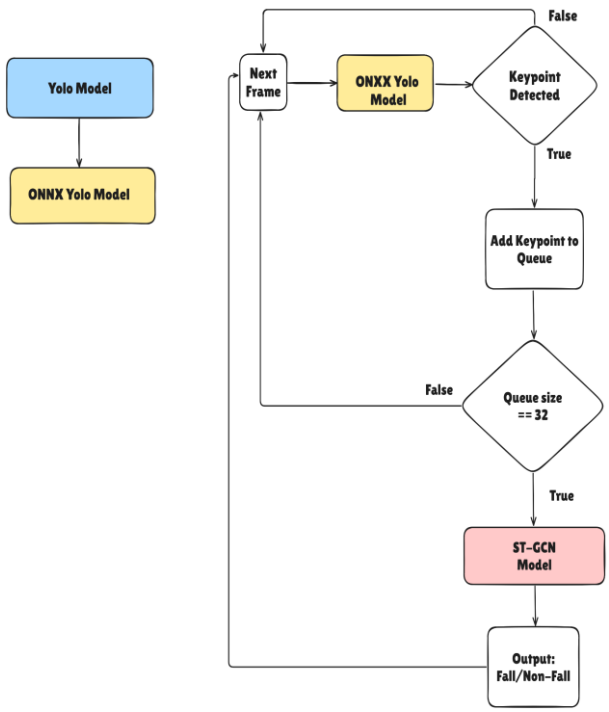

Figure 3. *Details on edge device system implementation*

5. Conclusion and future work

In this paper, we proposed an efficient fall detection system using KD to transfer knowledge from a high-performance teacher model to a compact student model. Our approach effectively reduces model complexity while maintaining strong predictive performance, making it highly suitable for real-time deployment on resource-constrained edge devices. We demonstrated the feasibility of this method through experiments on multiple public benchmark datasets, including CaucaFall, GMDCSA-24, UP-Fall, UR-Fall, and FallVision. The results of our experiments show that the KD-based student model achieved a significant improvement in F1-score by up to 7%, despite having only 1/200 of the teacher model's parameters. This reduction in model size, combined with the ability to maintain performance, highlights the potential of our approach in practical fall detection applications. The use of KD enables the deployment of fall detection systems on edge devices with minimal computational overhead. In summary, our use of KD offers a novel and efficient approach to the challenge of deploying accurate fall detection models in resource-constrained settingsOur framework not only provides an efficient fall detection system but also opens the door for further research into applying KD to other fall detection tasks in real-world, resource-constrained environments.

REFERENCES

[1] World Health Organization, "Fall: Fact sheets", *WHO, Geneva, Switzerland*, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls. [Accessed April 8, 2025].

[2] D. K. Hoang, N. M. Le, U. P. Vo-Thi, H. G. Nguyen, L. T. Ho-Pham, and T. V. Nguyen, "Mechanography assessment of fall risk in older adults: the Vietnam Osteoporosis Study", *J. Cachexia Sarcopenia Muscle,* vol. 12, no. 5, pp. 1161–1167, 2021. https://doi.org/10.1002/jcsm.12751

[3] H. M. Vu, L. H. Nguyen, H. L. T. Nguyen, G. T. Vu, C. T. Nguyen, T. N. Hoang, et al., "Individual and environmental factors associated with recurrent falls in elderly patients hospitalized after falls", *Int. J. Environ. Res. Public Health*, vol. 17, no. 7, 2441, 2020. https://doi.org/10.3390/ijerph17072441

[4] A. Muro-De-La-Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, "Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications", *Sensors,* vol. 14, no. 2, pp. 3362–3394, 2014. https://doi.org/10.3390/s140203362

[5] M. A. Khan, A. Saboor, H. C. Kim, and H. Park, "A systematic review of location aware schemes in the Internet of Things", *Sensors,* vol. 21, no. 9, 3228, 2021. https://doi.org/10.3390/s21093228

[6] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, and G. Farias, "Fall detection and activity recognition using human skeleton features", *IEEE Access,* vol. 9, pp. 33532–33542, 2021. doi: 10.1109/ACCESS.2021.3061626

[7] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "UP-Fall Detection Dataset: A multimodal approach", *Sensors*, vol. 19, no. 9, 1988, 2019. https://doi.org/10.3390/s19091988

[8] N. Noor and I. K. Park, "Factorized 3D-CNN for real-time fall detection and action recognition on embedded system", *IEEE Access,* vol. 12, pp. 112852–112863, 2024. doi: 10.1109/ACCESS.2024.3443618

[9] W. J. Chang, C. H. Hsu, and L. B. Chen, "A pose estimation-based fall detection methodology using artificial intelligence edge

computing", *IEEE Access,* vol. 9, pp. 129965–129976, 2021. doi: 10.1109/ACCESS.2021.3113824

[10] G. Hinton, O. Vinyals, and J. Dean, "*Distilling the knowledge in a neural network*", arXiv preprint arXiv:1503.02531, 2015.

[11] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition", *in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, Louisiana*, USA, 2018, pp. 7444 – 7452.

[12] F. X. Gaya-Morey, C. Manresa-Yee, and J. M. Buades-Rubio, "Deep learning for computer vision-based activity recognition and fall detection of the elderly: A systematic review", *Appl. Intell*., vol. 54, no. 19, pp. 8982–9007, 2024.

[13] J. Nogas, S. S. Khan, and A. Mihailidis, "Deepfall: Noninvasive fall detection with deep spatio-temporal convolutional autoencoders", *J. Healthc. Inform. Res.,* vol. 4, no. 1, pp. 50–70, 2020.

[14] H. Le Viet, H. Le Hoang Ngoc, K. Tran Dinh Minh, and S. Than Van Hong, "A deep learning framework for gym-gesture recognition using the combination of transformer and 3D pose estimation", *Cybern. Phys.,* vol. 13, no. 2, pp. 161–167, 2024. doi: 10.35470/2226-4116-2024-13-2-161-167

[15] R. Khanam and M. Hussain, *"Yolov11: An overview of the key architectural enhancements",* arXiv preprint arXiv:2410.17725, 2024.

[16] Z. Guan, S. Li, Y. Cheng, C. Man, W. Mao, N. Wong, and H. Yu, "A video-based fall detection network by spatio-temporal joint-point model on edge devices", *in Proc. Design, Autom. & Test Eur. Conf. Exhib. (DATE), Grenoble, France*, 2021, pp. 422–427.

[17] O. Keskes and R. Noumeir, "Vision-based fall detection using ST-GCN", *IEEE Access*, vol. 9, pp. 28224–28236, 2021. doi: 10.1109/ACCESS.2021.3058219

[18] F. Tung and G. Mori, "Similarity-preserving knowledge distillation", *in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Seoul, Korea (South),* 2019, pp. 1365–1374.

[19] Y. Tian, D. Krishnan, and P. Isola, "*Contrastive representation distillation*", arXiv preprint arXiv:1910.10699, 2019.

[20] D. Q. Vu, N. Le, and J. C. Wang, "Teaching yourself: A self-knowledge distillation approach to action recognition", *IEEE Access,* vol. 9, pp. 105711–105723, 2021. doi: 10.1109/ACCESS.2021.3099856

[21] F. M. Thoker and J. Gall, "Cross-modal knowledge distillation for action recognition", *in Proc. IEEE Int. Conf. Image Process. (ICIP),* Taipei, Taiwan, 2019, pp. 6–10.

[22] J. C. E. Guerrero, E. M. España, M. M. Añasco, and J. E. P. Lopera, "Dataset for human fall recognition in an uncontrolled environment", *Data in Brief,* vol. 45, 108610, 2022. https://doi.org/10.1016/j.dib.2022.108610

[23] E. Alam, A. Sufian, P. Dutta, M. Leo, and I. A. Hameed, "GMDCSA-24: A dataset for human fall detection in videos", *Data in Brief,* vol. 57, 110892, 2024. https://doi.org/10.1016/j.dib.2024.110892

[24] N. N. Rahman *et al.*, "FallVision: A benchmark video dataset for fall detection", *Data in Brief*, vol. 59, 111440, 2025. https://doi.org/10.1016/j.dib.2025.111440

[25] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer", *Comput. Methods Programs Biomed.,* vol. 117, no. 3, pp. 489–501, 2014. https://doi.org/10.1016/j.cmpb.2014.09.005