# Vision-based human fall detection - Measuring the effect of differing clip lengths on vision model performance

Michail Bakalianos    Georgios Tsouderos    Tomas Grahn

## I. Introduction

Falls are a significant health risk, particularly for the elderly, often leading to severe injuries and even death. Statistics show that falls are the leading cause of injury-related death for people aged 79 and the second most prevalent cause of injury-related (unintentional) mortality for adults of all ages [1]. The rapid response time following a fall is critical in mitigating the severity of injury. Therefore, there is a need for accurate and timely fall detection systems. This paper aims to focus on the real-world application of these systems, specifically, outlining the trade-offs between choices that must be made. Choices of parameters such as clip length or sampling strategy stems from real-world constraints: longer clips increase computational demands and potentially delay response times, while shorter clips might miss crucial information leading to inaccurate detection. This paper aims to investigate the performance impacts of differing clip lengths on state of the art vision based machine learning models. In a broader sense, this paper aims to direct current research to better demonstrate such trade-offs to highlight choices that can be made to fit specific real-world use cases.

## II. Related Work

Currently, there are broadly two categories of fall detection datasets available for vision-based models, simulated and real world. In the simulated setting more variables relating to the environment are controlled such as location or lighting, whereas in the real case the sequences are taken from sources with high variance. Previous research has focused more heavily on simulated datasets due to their availability, size and lower-complexity. For example the LE2i dataset [2] which has been available since 2013 with 143 fall sequences and the High Quality Fall Dataset (HQFDS) [3] which has been available since 2016 and contains 55 long-running fall sequences from 5 different camera angles. However, the need for accurate fall detection in real-life scenarios has led researchers to construct datasets that better reflect these conditions. Such cases are the Youtube Fall Dataset (YTFD) [4] consisting of 430 falling incidents and 176 normal activities, as well as the real-world fall dataset (RFDS) [5], consisting of 120 fall sequences containing diverse and complex scenes, including outdoor, indoor, sparsely and densely populated situations.

In the vision fall detection research space, there have been many contributions, many of them achieving high accuracy scores (see Table  I) by utilizing a variety of methods (RGB, sensor data, skeleton features, etc.) on the UP-Fall dataset [6], another highly used simulated dataset. However, there is a lack of documentation regarding the suitability of the model in a real-case scenario, where the inference time and the computational resources needed by the model become paramount. We believe that the use of fall-detection systems in situ requires the exploration of these resource and accuracy trade-offs.

TABLE I
Comparison of Different Approaches in Terms of Type, Accuracy

| Study | Type | Accuracy |
|---|---|---|
| Ponce et al. (2020) [7] | Sensor + RGB | 98.72 |
| Waheed et al. (2021) [8] | Sensor | 97.21 |
| Galvão et al. (2021a) [9] | RGB + Sensor | 99.99 |
| Al Nahian et al. (2021a) [10] | Sensor | 96.00 |
| Al Nahian et al. (2021b) [11] | Sensor | 100.00 |
| Ashrapov (2020) [12] | Skeleton | 99.50 |
| Taufeque et al. (2021) [13] | Skeleton | – |
| Galvão et al. (2021b) [9] | Skeleton | 98.62 |
| Ramirez et al. (2021) [14] | Skeleton | 99.34 |
| Ramirez et al. (2022) [15] | Skeleton | 99.81 |
| Ramirez et al. (2023) [16] | Skeleton | 81.14 |
| Martínez-Villasenor et al. (2019) [17] | Sensor | 95.49 |
| Chahyati and Hawari (2020) [18] | Sensor | – |
| Chahyati and Hawari (2020) [18] | RGB + Sensor | – |
| Ramirez et al. (2021) [14] | Skeleton | 99.45 |
| Le et al. (2022) [19] | Sensor | – |
| Mohan Gowda et al. (2022) [20] | RGB + Sensor | 99.20 |
| Islam et al. (2023) [21] | RGB + Sensor | 97.90 |
| Yan et al. (2023) [22] | Skeleton + Sensor | 98.05 |

## III. Methodology

### A. Data

The method uses two of the aforementioned datasets, RFDS and HQFDS. The timestamps of the falls were required for the clip generation and previously existed for HQFDS but not RFDS. Hence, These labels were manually produced for the 120 fall sequences. The analysis of the box-plots in figure 1 in combination with GPU memory restrictions motivated the choice of 15, 25, 35 and 50 frame sequences for classification.

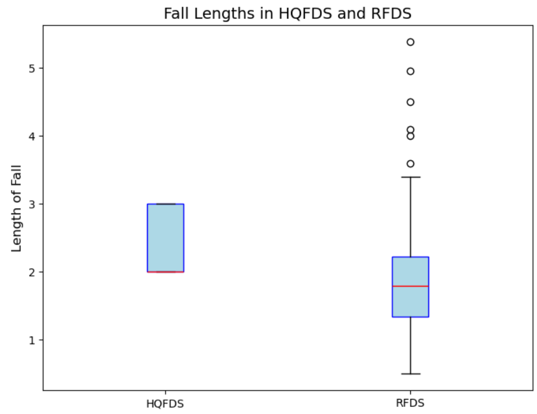

Fig. 1. Boxplots of dataset fall durations

Inspecting figure 1 highlights how identifying the exact moment a fall occurs is subjective, as there is no universally agreed-upon criterion. A fall might be considered to have started when an individual begins losing balance, when rapid, unintentional motion occurs, or even when unusual posture or acceleration is detected. Conversely, some may define a fall only once the individual makes contact with the ground and remains there for a period, potentially indicating an inability to recover. For example, the fall length in the HQFDS are all 2 or 3 seconds, as it was created using actors to specifically simulate falls that occur within this time frame. On the other hand, the RFDS dataset contains falls that typically last between 1.5 and 2.5 seconds, with some outliers that may extend to 5 seconds or more. This variability in fall duration across datasets highlights the need for models to handle a range of fall behaviors and timeframes in real-world applications.

In order to produce the clips classified by the model the following sequence was carried out for each combination of dataset and clip length with a sliding window of 10 frames. First, all fall sequences were sampled from the start of fall minus the sliding window, iterating by the sliding window length, until the produced clips no longer contained any falling frames. These clips were then labeled as containing a fall. Next, an equal number clips of equivalent length were randomly sampled (without replacement) from outside the range of the fall. These clips were then labeled as containing no fall. This process produced the number of clips seen in table II.

TABLE II
Total Clips Counts for Dataset with Clip Length (Frames)

| **Clip Length (Frames)** | ***15*** | ***25*** | ***35*** | ***50*** |
|---|---|---|---|---|
| HQFDS | 4584 | 5128 | 5672 | 6216 |
| RFDS | 1310 | 1398 | 1440 | 1444 |

Before being classified by the models the clips are resized to 224 by 224 pixels and are randomly flipped half of the time for augmentation. Furthermore, during each classification the entire clip is uniformly sampled by the model. Finally, the clips were split into a validation and training dataset containing 20% and 80% respectively.

### B. *Models*

We utilized the MMAction [23] framework, a tool designed for video-based action recognition tasks. MMAction provides a modular and efficient framework to implement and evaluate state-of-the-art vision models for analyzing human actions in video sequences. Its extensive library of pretrained models and compatibility with custom datasets allowed us to streamline the development process and the results. For this task, we employed two advanced models: VideoMAE V2 and Uniformer V2.

1) VideoMAE V2 [24] is a video representation learning model that extends the Video Masked Autoencoder (VideoMAE) framework. It employs a dual masking strategy to improve computational efficiency by applying masking to both the encoder and decoder. This allows for scalable pre-training with billions of parameters, making it ideal for learning robust video features. By leveraging self-supervised learning, VideoMAE V2 achieves state-of-the-art results on benchmarks such as Kinetics-400 and Something-Something V2, demonstrating its effectiveness for video understanding tasks. Furthermore, given the results by Grutschus et al [25] there is precedence for this model achieving high levels of accuracy for fall detection tasks. The model was loaded with pre-trained weights from the Kinetics-400 [26] dataset.
2) Uniformer V2 [27] is a cutting-edge video understanding model that integrates convolutional and self-attention mechanisms into a unified architecture. It builds upon the original UniFormer by optimizing the design of vision transformers (ViTs) for video tasks, combining local and global relation aggregators to achieve a balance between accuracy and computational efficiency. UniFormerV2 delivers state-of-the-art performance on various benchmarks, including being the first to achieve 90% top-1 accuracy on Kinetics-400, making it a robust solution for complex video analysis. We chose UniFormerV2 due to its strong performance in benchmarks and its innovative architecture that combines convolutional and self-attention mechanisms. As vision transformers (ViTs) are currently at the forefront of AI research, we aimed to leverage their state-of-the-art capabilities for video understanding tasks, making UniFormerV2 a fitting choice for our project. The model was loaded with a base architecture of ViT-B/16 and with pre-trained weights based on training with CLIP-400 [28].

Both models were trained utilising the AdamW [29] optimizer for gradient descent, and different learning rates, weight decays, and gradient clipping values, as shown in Table III.

TABLE III
OPTIMIZATION SETTINGS FOR VIDEOMAE V2 AND UNIFORMER V2

| Model | *Optimizer* | *LR* | *Weight Decay* | *Clip Grad* |
|---|---|---|---|---|
| VideoMAE V2 | AdamW | 1e-3 | 0.1 | Max Norm: 5 |
| UniFormer V2 | AdamW | 1e-5 | 0.05 | Max Norm: 20 |

Both models follow a linear warm up schedule for the first 5 epochs with differing parameters followed by cosine annealing for epochs 5 to 35 for the VideoMAEV2 model and epochs 5 to 55 for the UniformerV2 model, both then maintain a constant learning rate.

Early stopping based on validation accuracy was also implemented. Initially, early stopping was set to trigger after 25 epochs without improvement in validation loss. If the model did not improve in it's initial 25 epochs the model was retrained using a 50 epoch trigger.

For the two datasets, we trained the models using different GPUs based on the frame rate of the video clips. For clips with 15 frames and 25 frames, training was done using a single A40 GPU, while for 35 frames and 50 frames, an A100 (80GB memory) GPU was utilized. This difference in GPU's was necessary, given the model's large memory requirements for the 35 and 50 frame cases.

## IV. RESULTS AND DISCUSSION

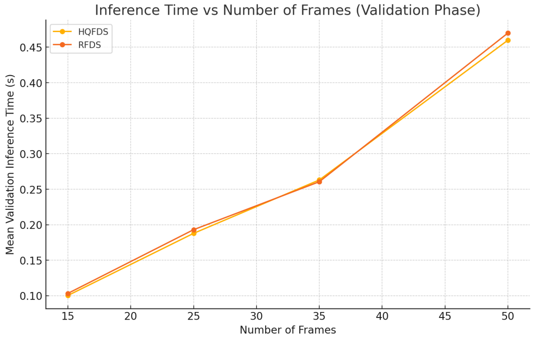

Fig. 2. VideoMaeV2 Inference Time vs Number of Frames

In Figures 2, 3, we report the inference times for the VideoMaeV2 and UniformerV2 models, compared to the input frames. As previously mentioned, we selected 15, 25, 35 and 50 frame sequences, and the data was interpolated to produce the plots. The results are compared to the baseline of 25 frames per second, since the available clips were shot in that framerate. Cases where the number of frames divided by the inference time is lower than 25, create overhead which significantly impacts the suitability of the model for real-time fall detection. In comparing these results, it's important to note again that in the cases of 15 and 25 frames the A40 GPU was used whereas the more powerful A100 (80GB memory) was utilized for the rest.

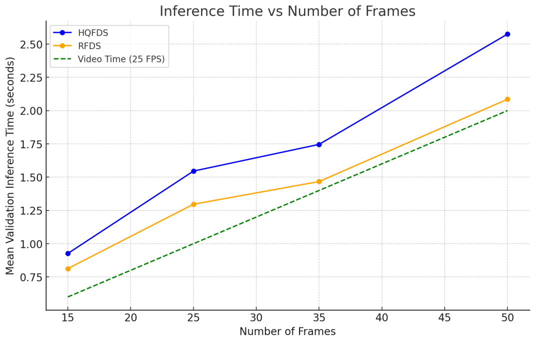

Fig. 3. UniformerV2 Inference Time vs Number of Frames

In figure 2 we observe that the inference time, despite displaying an increase over the amount of frames, stays significantly lower than our desired threshold, indicating the possibility of sampling longer or more clips. However, in figure 3, the inference time is substantially over the threshold, even when the size of the clip is short. In both cases, it is important to highlight the importance of balancing the trade-off between number of frames, the accuracy of the model and inference time, which from experimentation seems to be a parameter-sensitive task that requires further analysis.

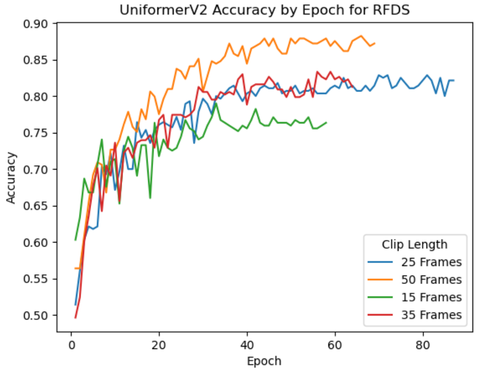

Fig. 4. UniformerV2 RFDS Accuracy

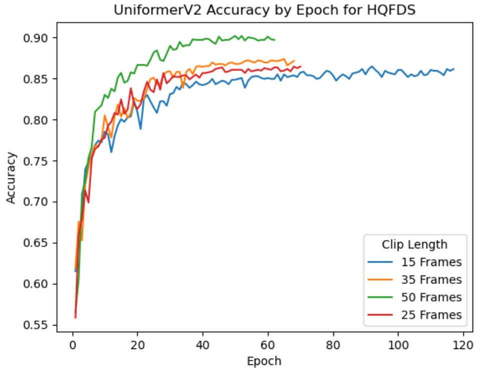

Fig. 5. UniformerV2 HQFDS Accuracy

Observing graphs 4 and 5 show strong support that increasing the number of frames generates increasing accuracy. This is because in both cases there is ascending accuracy performance as the frames are increased. Notably, In both cases there is a slightly larger jump in the increase from 35 to 50 frames. This could be explained by some falls and non-falls only being possible to distinguish with more frames.

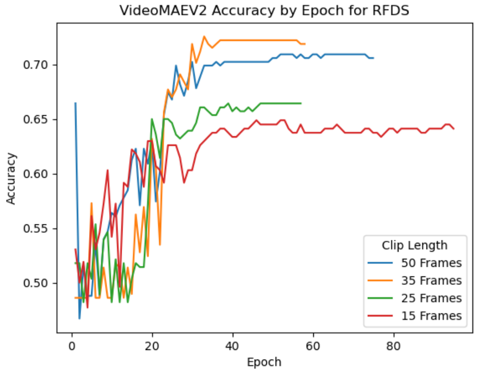

Fig. 6. VideoMaeV2 RFDS Accuracy

This general trend is also supported by the results shown in figure 6, however, the 35 frame clips outperform the 50 frame clips. In this case, referring to table IV reveals that it is only for accuracy and precision and not recall. This result indicates that depending on the model and situation shorter clip lengths can be favored.

The best results in figure 7 are generally in line with the accuracy trends with the 50 frame then 35 frame models giving the highest respectively.

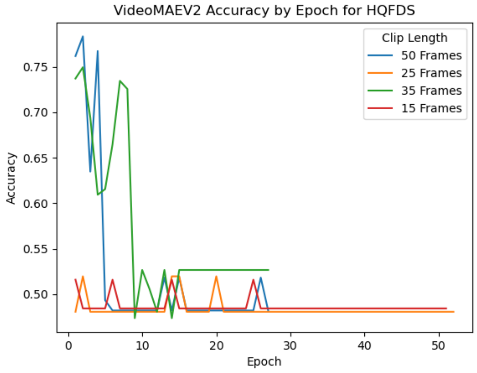

Fig. 7. VideoMaeV2 HQFDS Accuracy

However, these models showed strange training behavior with the 15 and 25 frame models not training in the first 50 epochs and the 35 and 50 frame models rapidly dropping in accuracy after an initial increase. Given the other stable results our belief is that this was caused by incorrect settings in the learning rate or optimizer.

TABLE IV
PERFORMANCE METRICS FOR VIDEOMAEV2 AND UNIFORMERV2 ON HQFDS AND RFDS DATASETS WITH VARYING CLIP LENGTH (FRAMES)

| **VideoMAEV2 - HQFDS** | ***15*** | ***25*** | ***35*** | ***50*** |
|---|---|---|---|---|
| Accuracy | 0.5158 | 0.5195 | 0.7496 | 0.7836 |
| Precision | 0.0000 | 1.0000 | 0.8342 | 0.7577 |
| Recall | 0.0000 | 0.5194 | 0.6967 | 0.8119 |
| **VideoMAEV2 - RFDS** | ***15*** | ***25*** | ***35*** | ***50*** |
| Accuracy | 0.6489 | 0.6643 | 0.7257 | 0.7093 |
| Precision | 0.6097 | 0.5777 | 0.6689 | 0.6013 |
| Recall | 0.6303 | 0.6782 | 0.7674 | 0.7807 |
| **UniformerV2 - HQFDS** | ***15*** | ***25*** | ***35*** | ***50*** |
| Accuracy | 0.8648 | 0.8645 | 0.8739 | 0.9019 |
| Precision | 0.9054 | 0.9043 | 0.9217 | 0.8990 |
| Recall | 0.8305 | 0.8456 | 0.8305 | 0.9103 |
| **UniformerV2 - RFDS** | ***15*** | ***25*** | ***35*** | ***50*** |
| Accuracy | 0.7901 | 0.8286 | 0.8333 | 0.8824 |
| Precision | 0.8373 | 0.7481 | 0.7972 | 0.8378 |
| Recall | 0.7463 | 0.8782 | 0.8676 | 0.9253 |

In table IV accuracy, precision and recall have been reported. Precision and recall have been included as these measures reflect the importance of reducing false positive and false negatives respectively. False positives in this scenario could potentially result in emergency services being called in a ’false alarm’. False negatives, more dramatically, could result in a lack of emergency services being called for a fall. Generally, the models performance improves across all three metrics as the clip lengths increase. One exception is in the performance of UniformerV2 between 35 and 50 frames in which the increase overall accuracy is driven by increasing recall and decreasing precision. This demonstrates a trade-off in these models to be more ’sensitive’ to positive or negative classifications and emphasizes the importance of clearly analyzing which cases you wish to optimize for.

A potential issue with the method used is with data leakage between the validation and training sets. Given that the sliding window length is less than than frame length many of the produced clips have overlapping frames. In the worst case of the 50 frame clips, up to 40 frames could be shared between clips in the validation and training set. Increasing the sliding window length would reduce these overlapping frames however would reduce the number clips. This reduction in clips could prove problematic for training if not enough data is available. One potential solution for this is to use some additional forms of augmentation such as random crops. On a additional level there is also leakage with a fall sequences clips being shared between training and validation. This motivates generating the training and validation splitting on whole sequences to avoid leakage. These issues indicate how the consideration of leakage in further experiments should be done with care to ensure robust results.

## V. FUTURE DIRECTION

In further improving performance, for example, on the 50 frame UniformerV2 a first point of analysis would be to extract the clips that are not successfully classified and attempt to look for patterns in these particular clips. The patterns noticed could then be translated into the model to further boost accuracy.

This analysis could also reveal data problems, for example, perhaps in these clips it appears ambiguous if the subject is truly falling or not in which case newer more concrete labels need to be produced with a clearer definition of a fall.

From figure 1 we see that in the HQFDS dataset the maximum duration of a fall is around 3 seconds whereas in RFDS the mean value is around 2, with some individual cases where they reach about 5. The strategy of splitting up the video into smaller clips aims to reduce the necessary GPU memory the models need, making them more cost efficient. In our experiments we utilised an A40 and an A100 (80GB memory) GPU, but were more resources available, it would open pathways to further analysis and sampling of additional frames.

In addition, a direction worth considering is applying different sampling methods. Instead of processing each video frame within a small clip, which as mentioned can prove to be quite computationally exhaustive, sampling could prove to be a good avenue to both increase the clip size and reduce the memory resources needed by the model. By making the input clip larger, it is possible to retain more information about the event, which from experimentation has proved to yield higher accuracies (Figures  2, 3), while applying sampling could provide a more well-rounded clip that better describes that event. We believe that by balancing these two techniques, better accuracy scores are possible, while maintaining a cost-efficient model, and is something worth exploring further.

Other avenues for fall classification also exists such as pose estimation and utilizing optical flow. Chenyang et al [30] utilized Kinect RGBD cameras, which include depth information that was used in handling illumination changes and identity protection. Their model analyzes the deformation of joints during falls which significantly differs in other actions of daily living. Using the OpenPifPaf model [31], key points of the human skeleton are extracted, tracked, and compared to predefined vectors to detect falls. Núñez-Marcos et al [32] preprocessed the video frames in order to generate optical flow data, then utilized a convolutional neural network for classification. Both these examples point to the benefit of using some additional preprocessed data alongside the input frames to assist classification. Incorporating these, while potentially costly to inference time, could prove beneficial in achieving higher accuracy.

Finally, given the current availability of many datasets such as LE2i, HQFDS, UP-Fall, YTFD and RFDS and high performance methods there is the possibility of a two-step training sequence. Firstly, a general fall detection model could be trained on all available fall datasets to create a more generalized fall detection model. This general method should also explore the aforementioned sampling and clip length strategies to find a range of models with differing hardware requirements and corresponding accuracies. This model could then be fine-tuned to a particular scenario. Considering that the real-application of this model would likely be for static cameras in a particular location this baseline model could prove useful for quickly creating a high-accuracy fine-tuned model.

## References

[1] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. Laighin, V. Rialle, and J.-E. Lundy, "Fall detection-principles and methods," in *2007 29th annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 2007, pp. 1663–1666.

[2] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Optimised spatio-temporal descriptors for real-time fall detection: comparison of svm and adaboost based classification," *Journal of Electronic Imaging (JEI)*, vol. 22, no. 4, p. 17, 2013.

[3] G. Baldewijns, G. Debard, G. Mertes, B. Vanrumste, and T. Croonenborghs, "Bridging the gap between real-life data and simulated data by providing a highly realistic fall dataset for evaluating camera-based fall detection algorithms," *Healthcare technology letters*, vol. 3, no. 1, pp. 6–11, 2016.

[4] Y. Fan, M. D. Levine, G. Wen, and S. Qiu, "A deep neural network for real-time detection of falling humans in naturally occurring scenes," *Neurocomputing*, vol. 260, pp. 43–58, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217304666

[5] L. Wu, C. Huang, S. Zhao, J. Li, J. Zhao, Z. Cui, Z. Yu, Y. Xu, and M. Zhang, "Robust fall detection in video surveillance based on weakly supervised learning," *Neural Networks*, vol. 163, pp. 286–297, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608023001776

[6] 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, p. 1988, 2019.

[7] H. Ponce, L. Martinez-Villasenor, and J. Nunez-Martinez, "Sensor location analysis and minimal deployment for fall detection system," *IEEE Access*, vol. 8, pp. 166 678–166 691, 2020.

[8] M. Waheed, H. Afzal, and K. Mehmood, "Nt-fds—a noise tolerant fall detection system using deep learning on wearable devices," *Sensors*, vol. 21, no. 6, p. 2006, 2021.

[9] Y. M. Galvao, L. Portela, J. Ferreira, P. Barros, O. A. D. A. Fagundes, and B. J. Fernandes, "A framework for anomaly identification applied on fall detection," *IEEE Access*, vol. 9, pp. 77 264–77 274, 2021.

[10] M. J. Al Nahian, T. Ghosh, M. H. Al Banna, M. A. Aseeri, M. N. Uddin, M. R. Ahmed, M. Mahmud, and M. S. Kaiser, "Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features," *IEEE Access*, vol. 9, pp. 39 413–39 431, 2021.

[11] M. J. Al Nahian, T. Ghosh, M. H. Al Banna, M. N. Uddin, M. M. Islam, K. A. Taher, and M. S. Kaiser, "Social group optimized machine-learning based elderly fall detection approach using interdisciplinary time-series features," in *2021 international conference on information and communication technology for sustainable development (icict4sd)*. IEEE, 2021, pp. 321–325.

[12] I. Ashrapov, "Tabular gans for uneven distribution," *arXiv preprint arXiv:2010.00638*, 2020.

[13] M. Taufeeque, S. Koita, N. Spicher, and T. M. Deserno, "Multi-camera, multi-person, and real-time fall detection using long short term memory," in *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications*, vol. 11601. SPIE, 2021, pp. 35–42.

[14] 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. 33 532–33 542, 2021.

[15] H. Ramirez, S. A. Velastin, P. Aguayo, E. Fabregas, and G. Farias, "Human activity recognition by sequences of skeleton features," *Sensors*, vol. 22, no. 11, p. 3991, 2022.

[16] H. Ramirez, S. A. Velastin, S. Cuellar, E. Fabregas, and G. Farias, "Bert for activity recognition using sequences of skeleton features and data augmentation with gan," *Sensors*, vol. 23, no. 3, p. 1400, 2023.

[17] 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, p. 1988, 2019.

[18] D. Chahyati and R. Hawari, "Fall detection on multimodal dataset using convolutional neural netwok and long short term memory," in *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 2020, pp. 371–376.

[19] H.-L. Le, D.-N. Nguyen, T.-H. Nguyen, and H.-N. Nguyen, "A novel feature set extraction based on accelerometer sensor data for improving the fall detection system," *Electronics*, vol. 11, no. 7, p. 1030, 2022.

[20] V. Mohan Gowda, M. P. Arakeri, and V. Raghu Ram Prasad, "Multimodal classification technique for fall detection of alzheimer's patients by integration of a novel piezoelectric crystal accelerometer and aluminum gyroscope with vision data," *Advances in Materials Science and Engineering*, vol. 2022, no. 1, p. 9258620, 2022.

[21] M. M. Islam, S. Nooruddin, F. Karray, and G. Muhammad, "Multi-level feature fusion for multimodal human activity recognition in internet of healthcare things," *Information Fusion*, vol. 94, pp. 17–31, 2023.

[22] J. Yan, X. Wang, J. Shi, and S. Hu, "Skeleton-based fall detection with multiple inertial sensors using spatial-temporal graph convolutional networks," *Sensors*, vol. 23, no. 4, p. 2153, 2023.

[23] "Openmmlab's next generation video understanding toolbox and benchmark," https://github.com/open-mmlab/mmaction2, 2020.

[24] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao, "Videomae v2: Scaling video masked autoencoders with dual masking," 2023. [Online]. Available: https://arxiv.org/abs/2303.16727

[25] T. Grutschus, O. Karrar, E. Esenov, and E. Vats, "Cutup and detect: Human fall detection on cutup untrimmed videos using a large foundational video understanding model," *arXiv preprint arXiv:2401.16280*, 2024.

[26] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017. [Online]. Available: https://arxiv.org/abs/1705.06950

[27] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, L. Wang, and Y. Qiao, "Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer," 2022. [Online]. Available: https://arxiv.org/abs/2211.09552

[28] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "LAION-400M: open dataset of clip-filtered 400 million image-text pairs," *CoRR*, vol. abs/2111.02114, 2021. [Online]. Available: https://arxiv.org/abs/2111.02114

[29] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019. [Online]. Available: https://arxiv.org/abs/1711.05101

[30] C. Zhang, Y. Tian, and E. Capezuti, "Privacy preserving automatic fall detection for elderly using rgbd cameras," in *Computers Helping People with Special Needs*, K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 625–633.

[31] S. Kreiss, L. Bertoni, and A. Alahi, "Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association," 2021. [Online]. Available: https://arxiv.org/abs/2103.02440

[32] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras, "Vision-based fall detection with convolutional neural networks," *Wireless communications and mobile computing*, vol. 2017, no. 1, p. 9474806, 2017.