# Enhanced Vision-Based Human Fall Detection with Mask-RCNN and Autoencoder-LSTM Hybrid Framework

Sritama Chakraborty<sup>1,\*,†</sup>, Mili Ghosh<sup>2,†</sup>

<sup>1</sup>Department of Computer Science & Technology, University of North Bengal <sup>2</sup>Department of Computer Science & Technology, University of North Bengal

#### Abstract

Elderly individuals who reside independently face a heightened risk of experiencing serious harm due to accidental falls, a leading contributor to mortality rates in this demographic. Fall detection is a critical part of health care for older adults. This paper introduces a methodology for identifying falls among elderly individuals utilizing machine learning algorithms. We suggest utilizing Mask R-CNN, Autoencoder- LSTM, and a Hybrid Mask R-CNN-Autoencoder-LSTM framework to detect falls in a video surveillance environment. Primarily, these algorithms are trained to identify deformations in human body shapes and postures, enabling them to detect whether a fall has occurred. At their core, these algorithms are instructed to distinguish shifts in human body structures and movements, empowering them to flag potential fall occurrences. Our findings suggest that the Mask R-CNN and Autoencoder-LSTM models perform well independently, with the Mask R-CNN excelling in spatial feature extraction and the Autoencoder-LSTM effectively modeling the temporal changes in body movements. However, the hybrid Mask R-CNN-Autoencoder-LSTM approach does not fully capitalize on the strengths of both models, leading to a slightly lower performance than the individual models. Despite this, both Mask R-CNN and Autoencoder-LSTM present a reliable solution for fall detection, with the hybrid framework offering potential improvements for future research.

#### Keywords

Human Fall Detection, Mask R-CNN, Autoencoder LSTM, Hybrid Model

### 1. Introduction

In recent times, the prevalence of aging has markedly risen across the population, mirroring demographic changes and extended life expectancies. In the elderly healthcare sector, falls are a critical health issue. Accidental deaths among older adults are most often caused by falls [1]. According to the census, in 2021, around 6.8 percent of people were over 65 years old. According to Elderly in India 2021 from the National Statistical Office (NSO), India's older adult population is expected to reach 194 MM by 2031. This reflects a substantial increase of 41 percent when compared to the figures from the preceding decade, suggesting a notable trend of growth over time. Given the growing population of elderly individuals living independently, it becomes imperative for both governmental and private entities to devise efficient intelligent surveillance systems capable of identifying potential fall risks.

Due to the increasing number of people seeking security and healthcare services, the need for fall detection has been accelerated. Using depth sensors, skeleton joint data, and Hidden Markov Models, [2] research introduces a depth-based life logging system for senior activity detection that shows promise for tracking daily activities and medical treatment. Many research projects have been conducted to create efficient fall detection algorithms. Even though fall incidents cannot entirely be prevented, the ability to precisely identify a fall incidence and issue an emergency notice can save lives. A monitoring system should distinguish fall events from normal activity. Various methods have been proposed to identify falls at right using wearable devices such as accelerators, gyroscopes, and magnetometers [3]. However, these methods could be more effective since it is impractical to wear such devices for a long time [4].

<sup>†</sup>These authors contributed equally.

© 02025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

*The 2024 Sixth Doctoral Symposium on Intelligence Enabled Research (DoSIER 2024), November 28–29, 2024, Jalpaiguri, India* \*Corresponding author.

Sritama.sc@gmail.com (S. Chakraborty); ghosh.mili90@gmail.com (M. Ghosh)

Furthermore, different machine learning techniques can detect various types of falls. The adaptability of machine learning techniques in tackling various contexts and categories of falls has contributed significantly to their widespread adoption and popularity. Some of these include Convolutional Neural Network (CNN), random forest, multi-layer perceptron [5], a statistical model i.e. Hidden Markov Model (HMM), a non-parametric supervised learning parameter i.e. K-Nearest Neighbours (KNN), a supervised machine learning algorithm i.e. Support Vector Machine(SVM), decision trees which employs deep learning method. Using deep convolutional neural networks to organize images and videos has shown outstanding results in computer vision and machine learning [6],[7]. CNNs that learn features from training data are used to produce a workable automatic feature extraction approach for images.

We utilize a set of neural networks known as long-short-term memory networks, allowing for the storage of information over extended durations. Besides its capability to comprehend and forecast patterns in time series, text, speech, and images, LSTM demonstrates remarkable effectiveness in understanding and predicting patterns within sequential data. Using visual attention-guided Bidirectional LSTM fall detection in [8], [9], authors introduced a masked R-CNN to address the challenges posed by complex background environments. This model integrates spatial and temporal information within intricate scenes to effectively tackle the fall detection problem. Our final proposal is to use a hybrid Mask R-CNN-Autoencoder-LSTM approach. The features of a video are extracted using a convolutional neural network, while the video is categorized by an LSTM neural network [10]. In [11], the researchers devised a fusion of a 3D CNN and an LSTM-based visual attention mechanism to capture spatiotemporal features from video sequences related to fall detection, effectively encoding motion information.

Our study aims to implement a video action recognition system that essentially detects human falls. In consequence, we use Mask R-CNN and Autoencoder LSTM to obtain human shape distortion features for fall detection by applying them to each frame picture in a video from a camera. The residuum of this paper is structured as follows. Section 2 describes the UR-Fall Detection data set and proposes fall detection methods using Mask R-CNN, Autoencoder-LSTM, and a Hybrid Mask R-CNN-Autoencoder-LSTM, respectively. Section 3 shows a comparative study of these three experimental methods. Finally, section 4 concludes by presenting the conclusions and future work.

### 2. Background Study

Deep neural networks have garnered increasing popularity with advancements in technology. Deep learning models are trained on vast amounts of data, allowing them to learn complex patterns and relationships within the data. This enables the system to make accurate predictions, classifications, or decisions in various domains such as image recognition, natural language processing, and autonomous systems. Within the spectrum of deep learning algorithms, convolutional neural networks (CNNs) have established themselves as the foremost choice in computer vision technology. In a CNN architecture, different layers serve specific functions, such as convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification or regression tasks. Important body parts are segmented and identified using the Mask R-CNN, which also detects notable posture irregularities that may be signs of a fall. Conversely, a type of RNN known as LSTM(Long Short Term Memory) network functions akin to a feature pooling network but operates at the frame level, enabling it to integrate information over time, much like a CNN. Through the utilization of parameters shared across time, both architectures can maintain constant parameters while capturing the global temporal evolution of a video [12]. The Autoencoder-LSTM analyzes key body components and records movement patterns, identifying significant temporal abnormalities and posture abnormalities that could be signs of a fall. While the Mask R-CNN-Autoencoder LSTM architecture is highly effective theoretically in fall detection due to its capacity to capture both temporal and spatial features.

# 3. Proposed Methodology

This paper presents an extensive analysis of Mask R-CNN, Autoencoder-LSTM, and a combination of both for fall detection. The input data consists of approximately 60 videos of falls and non-falls, which are transformed into individual frames for processing. To enhance the generalization capability of the deep learning models, the frames are scaled using a tool named Keras ImageDataGenerator. The fall detection process involves the utilization of Mask R-CNN, Autoencoder-LSTM, and a hybrid Mask R-CNN-Autoencoder-LSTM model.

In this approach, each model carefully analyzes input frames, identifying key body segments and temporal movement patterns to distinguish between falls and non-falls. Through this comparative analysis, the effectiveness of each model in fall detection is evaluated.

### 3.1. Dataset Description

This research study employs two distinct datasets for fall detection purposes, the UR fall detection data set [4] and the multiple cameras fall data set [13]. The URFall detection data set consists of 70 sequences of both 30 sets of fall sequences and 40 sets of activities of daily living. The multiple cameras fall data set contains 24 falling examples recorded with 8 IP video cameras. In this research study, we utilize videos from multiple datasets, including five original videos created to demonstrate falls and non-falls. Additionally, we randomly selected around 60 videos from various datasets that contain classifications of falls and non-falls. By merging these three datasets, we established a comprehensive and diverse data pool for thorough analysis.

This combined dataset not only increases the sample size but also enhances the diversity of the data, allowing for a more comprehensive examination of fall detection. Moreover, it helps validate our findings across different sources, leading to a more robust analysis. Ultimately, this approach enhances the reliability of the study's conclusions, providing a stronger foundation for the effectiveness of the proposed fall detection methodology.

### 3.2. Feature Selection

Our data set has videos with a frame breadth and height of 640 × 240 pixels. Image analysis is a trendy topic in the subject of computer vision. It involves the extraction of important information from videos. To reduce computational cost and remove noise from images, we rescale to a 64x64 pixel resolution. We chose features from the fall class 15233 and the nonfall class 10540. However, the maximum number of images per class has been set at 10,000. This limitation reflects a conscious decision to control the dataset's size and maintain computational feasibility, all while preserving the equitable representation of both classes. By imposing a limit of 10000 images per class, the study upholds an equitable approach to feature selection while also effectively managing potential resource limitations.

### 3.3. Mask R-CNN

Our experiments use Mask R-CNN, a deep neural network that employs convolutional layers to extract essential features from images and segment body parts relevant to fall detection. The architecture of Mask R-CNN is designed to not only detect objects but also provide pixel-level segmentation. It utilizes a region proposal network (RPN) to generate potential object regions, followed by the application of a mask prediction network to generate segmentation masks for each detected object. The Mask R-CNN model consists of several key components. Initially, it uses a backbone network (such as ResNet or VGG) for feature extraction. This backbone network comprises convolutional layers that adapt their weights and biases through learning to identify important features from input images. Convolutional layers are followed by a Region of Interest (RoI) pooling layer that extracts fixedsize features from each region proposal. Subsequently, the network uses separate branches: one for object classification, one for bounding box regression, and one for predicting segmentation masks. In our experimentation, we utilize a Mask R-CNN model with a backbone network based on ResNet-50. The model is trained to identify

human body parts and detect significant posture changes indicative of falls. The segmentation mask generated by Mask R-CNN helps to precisely locate and delineate body parts, improving the model's ability to recognize fall events. To enhance the model's feature extraction, we incorporate a series of convolutional layers to detect key spatial patterns, followed by the RPN and RoI align layers for precise object localization. Finally, the model outputs a set of classes (such as "fall" and "non-fall") along with segmentation masks that outline the body positions and movements. This design enables Mask R-CNN to not only detect falls but also segment the affected areas of the body, enhancing fall detection accuracy for elderly individuals and supporting real-time safety monitoring in video surveillance scenarios. An overview of our network's configuration can be found in table 1.

Layer Type	Input Shape	Kernel Size	No. of Filters	Strides	Output Size	
Backbone (ResNet/FPN)	(H, W, 3)	Varies	Varies	Varies	(H/32, W/32, N)	
RPN CONV2D	(H/32, W/32, N)	(3, 3)	512	(1, 1)	(H/32, W/32, 18)	
Rol Align	(H/32, W/32, 512)	-	-	-	(14, 14, 256)	
Mask CONV2D	(14, 14, 256)	(3, 3)	256	(1, 1)	(14, 14, 1)	
Dense for Classification	(8192)	-	-	-	(2)	
Dense for Bounding Box	(8192)	-	-	-	(4)	
Mask Output	(64, 64, 1)	-	-	-	(64, 64, 1)	

Table 1

Mask R-Convolutional Neural Network Model Architecture

#### 3.4. Autoencoder LSTM

Autoencoder-LSTM networks are a combination of Autoencoders and Long Short-Term Memory (LSTM) networks, designed to learn both spatial features and temporal dependencies in sequential data. The autoencoder component captures the key features of the input data by learning an efficient representation in a compressed form, while the LSTM part models the temporal dynamics and dependencies between frames in a sequence.

In the Autoencoder-LSTM model, the architecture includes two main components: the encoder-decoder structure of the autoencoder and the LSTM layers that capture temporal relationships between the frames. The autoencoder focuses on encoding the input data into a lower-dimensional latent space and then decoding it back to the original input. This process helps to extract meaningful features while reducing dimensionality, which is particularly beneficial for fall detection in video sequences. The encoder part of the autoencoder compresses the input into a latent representation, typically using convolutional layers or fully connected layers, and the decoder reconstructs the input data from this latent representation. The LSTM layers are then applied to the encoded features to capture the sequential nature of the video frames. Two states exist in the cell. These are New Cell State or new long-term memory  $c_{(T)}$  and New Hidden State or new short-term memory  $h_{(T)}$ . Here is a brief examination of each gate.

1. Forget Gate: The forget gate decides what information should be eliminated from the cell state. It uses the sigmoid activation function which takes the previous hidden state  $h_{(T-1)}$  and current state input xT. The function outputs a value between 0 and 1, with 0 being forgetting and 1 being retaining.

$$f1_{(T)} = \sigma(W_{(fh)} \cdot h_{T-1} + W_{fx} \cdot x_T + b_f)$$
(1)

2. Primary Input Gate: The input gate updates the new long-term memory state. The sigmoid activation function is contained in the input gate, while the tanh activation function is contained in the input node.

$$i1_{(T)} = \sigma(W_{(ih)} \cdot h_{T-1} + W_{ix} \cdot x_T + b_i)$$
<sup>(2)</sup>

$$g1_{(T)} = \sigma(W_{(qh)} \cdot h_{T-1} + W_{gx} \cdot x_T + b_g)$$
(3)

3. Output Gate: The LSTM output gate is responsible for selecting which information to use further.

$$o1_{(T)} = \sigma(W_{(oh)} \cdot h_{T-1} + W_{ox} \cdot x_T + b_o)$$
(4)

New cell State or long-term memory  $c_{(T)}$  are upgraded by the middle level forget gate and primary input gate. The LSTM cell state and hidden state equations are in the following forms.

$$c(T) = f(T) \otimes c(T-1) + i_T \otimes g_T \tag{5}$$

$$y(T) = h(T) = o(T) \otimes \tanh(c_T) \tag{6}$$

To enhance the generalization capability of the Autoencoder-LSTM model and prevent overfitting, dropout layers are incorporated into the architecture. These layers randomly drop certain connections during training, which helps the model to generalize better to unseen data. Finally, the model includes a dense layer that outputs the final classification, which can be used to distinguish between "fall" and "non-fall" instances. Table 2 shows the Autoencoder LSTM model.

Layer (Type)	Output Size	Parameters	
lstm (Encoder)	(None, 192, 128)	98,816	
dropout (Encoder)	(None, 192, 128)	0	
lstm1 (Encoder)	(None, 128)	131,584	
repeatvector (Encoder)	(None, 192, 128)	0	
lstm2 (Decoder)	(None, 192, 128)	131,584	
dropout1 (Decoder)	(None, 192, 128)	0	
lstm3 (Decoder)	(None, 128)	131,584	
dense (Decoder)	(None, 64)	8,256	
dropout2 (Decoder)	(None, 64)	0	
dense1 (Decoder)	(None, 2)	130	

#### Table 2

Autoencoder LSTM Model Architecture

#### 3.5. Hybrid

In this research study, a hybrid Mask R-CNN - Autoencoder-LSTM algorithm was developed to identify falls in elderly individuals. The images are first analyzed by Mask R-CNN, which performs semantic segmentation and extracts key body parts from the input images. These features are then processed by the Autoencoder-LSTM network to capture temporal dependencies in the sequence of frames. The Mask R-CNN - Autoencoder-LSTM network architecture comprises several key components: the Mask R-CNN model for spatial feature extraction and segmentation, followed by the Autoencoder-LSTM model for analyzing the temporal sequence of frames. Mask R-CNN begins with a backbone network (such as ResNet) to extract features, followed by region proposal and segmentation mask prediction, which isolates relevant body parts and postures. The Autoencoder-LSTM component then processes these segmented features, leveraging the encoder-decoder structure of the autoencoder to compress and reconstruct spatial features, while the LSTM captures the temporal relationships in the sequence. The Mask R-CNN performs feature extraction and body part segmentation, while the Autoencoder-LSTM captures sequential patterns in the data, making it suitable for fall detection across video sequences. To enable the Autoencoder- LSTM to effectively process the extracted features, we place Time-Distributed Layers before the LSTM, allowing it to handle convoluted image features over time. While the hybrid approach aims to combine the strengths of Mask R-CNN for spatial feature extraction and Autoencoder-LSTM for temporal sequence modeling, our findings revealed that it did not perform as well as anticipated. The overall accuracy was lower than expected, suggesting that individual Mask R-CNN and Autoencoder-LSTM models might be more effective for this task. This underscores the challenges of combining these two architectures and the need for further optimization to fully leverage their complementary strengths.

Layer (type) Output Shape time-distributed-layer1 (None, 1, 14, 14, 64) time-distributed-layer2 (None, 1, 7, 7, 64) time-distributed-layer3 (None, 1, 4, 4, 128) time-distributed-layer4 (None, 1, 1, 1, 128) time-distributed-layer5 (None, 1, 1, 1, 256) time-distributed-layer6 (None, 1, 1, 1, 256) time-distributed-layer7 (None, 1, 1, 1, 512) time-distributed-layer8 (None, 1, 1, 1, 512) time-distributed-layer9 (None, 1, 512) time-distributed-layer10 (None, 1, 256) time-distributed-layer11 (None, 1, 1024) LSTM(Encoder) (None, 1024) Dense(Decoder) (None, 2)

### 4. Results & Analysis

In this section, we evaluate the effectiveness of the three methods on our dataset, which consists of 70 sequences comprising 30 sets of fall sequences and 40 sets of activities of daily living. We employed training and testing data sets with the information provided by several cameras fall non-fall data sets throughout these tests. The outcome of our work is measured using three metrics: accuracy metric, sensitivity metric, and specificity metric.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
(7)

Sensitivity = 
$$\frac{TP}{TP + FN}$$
 (8)

Specificity = 
$$\frac{TN}{TN + FP}$$
 (9)

True Positive (TP) and True Negative (TN) refer to the appropriate number of falls and non-falls, respectively. In both the fall and non-fall datasets, instances of false positive (FP) and false negative (FN) identifications are documented. When it comes to detecting a fall, sensitivity or SE refers to the ability to detect a fall event. However, a fall can only be detected with specificity or SP. There is an imbalance between the fall and non-fall data sets used for the study. To take advantage of GPUs, every experiment was carried out in Python 3.11.0 machine learning methods utilising sklearn3 and keras4 and tensorflow2 for CNNs.

In the experimental configuration, the datasets designated for training and testing purposes underwent division into distinct proportions. In particular, the dataset was divided, dedicating 90% of the data for training the model and allocating the remaining 10% for model testing purposes. Furthermore, a separate division was implemented, assigning 60% of the dataset for model training and preserving the remaining 40% as an independent testing set. This approach facilitated thorough evaluation of the model's ability to generalize to unseen data. Figure:2 and Figure:3 show loss and accuracy between training and validation data using our CNN based architecture. All three experiments demonstrated that the maximum number of epochs for training was set at 128 and 256, while 5 and 10 epochs were utilized in specific instances for optimal performance. The results, summarized in Table 3, illustrate the overall accuracy achieved by each model: both the Mask R-CNN and Autoencoder LSTM architectures



Figure 1: Training Loss vs Validation Loss



Figure 2: Training Accuracy vs Validation Accuracy

reached nearly 98% accuracy, indicating their strong performance in detecting falls. In contrast, the combined model achieved an accuracy of 93%. This performance disparity highlights the strengths of the individual models; the Mask R-CNN effectively captures spatial features, while the Autoencoder LSTM excels in understanding temporal dynamics. Despite the hybrid model's potential for integrating both spatial and temporal data, it appears to fall short of the accuracy levels reached by the standalone models. This suggests that, for this particular application, leveraging the strengths of Mask R-CNN and Autoencoder LSTM separately may yield better results in fall detection for elderly individuals shown in table 3.

No	Train Features	Test Features	Epochs	Batch	Mask RCNN	LSTM	Hybrid
1	9000	1000	5	128	0.971	0.982	0.938
2	9000	1000	5	256	0.986	0.987	0.943
3	9000	1000	10	128	0.985	0.976	0.957
4	6000	4000	5	256	0.995	0.997	0.948
5	6000	4000	10	256	0.986	0.983	0.936
6	6000	4000	5	128	0.989	0.993	0.954
7	6000	4000	10	128	0.994	0.989	0.941

#### Table 3

Comparison of Mask R-CNN, Autoencoder LSTM, Hybrid Model Architecture

# 5. Conclusion & Future Direction

This paper presents a novel approach for automatic fall detection in video monitoring scenarios, employing Mask R-CNN, Autoencoder-LSTM, and a hybrid fusion of both algorithms. Each frame in the video is directly analyzed using these algorithms to segment and identify body parts and detect temporal anomalies associated with human movement. We conduct a comparative analysis of the accuracy of the three approaches. Our experimental results demonstrate that Mask R-CNN and Autoencoder-LSTM individually show promising potential for real-time fall detection in video streaming, making them valuable for video monitoring systems, especially in contexts like elderly care. In the future, live footage from CCTV cameras could be examined to validate our experimental findings. It's worth noting that the images used in our study predominantly originate from three sources with similar backgrounds, which could influence our approach. Consequently, future work will involve training the two algorithms using images captured from various perspectives to assess their performance under different conditions.

# **Declaration on Generative Al**

The author(s) have not employed any Generative AI tools.

# References

- S. Deandrea, E. Lucenteforte, F. Bravi, R. Foschi, C. L. Vecchia, E. Negri, Risk factors for falls in community-dwelling older people: A systematic review and meta-analysis, Epidemiology 21 (2010) 658–668.
- [2] A. Jalal, S. Kamal, D. Kim, A depth video sensor- based life-logging human activity recognition system for elderly care in smart indoor environments, Sensors 14 (2014) 11735–11759.
- [3] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida, O. Tanaka, A wearable airbag to prevent fall injuries, IEEE Trans. Inf. Technol. Biomed 13 (2009) 910–914.
- B. Kwolek, M. Kepskie, Human fall detection on embedded platform using depth maps and wireless accelerometer, Elsevier 117 (2014) 489–501.
- [5] S. Münzner, P. Schmidt, A. Reiss, M. Hanselmann, R. Stiefelhagen, R. Dürucheb, Cnn based sensor fusion techniques for multimodal human activity recognition, ACM International Symposium on Wearable Computers - ISWC (2017).
- [6] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: Deep networks for video classification, Computer Vision and Pattern Recognition (2015).
- [7] B. Jokanovic, M. Amin, F. Ahmad, Radar fall motion detection using deep learning., Radar Conferenc (2016) 1–6.
- [8] Y. Chen, W. Li, L. Wang, J. Hu, M. Ye, Vision-based fall event detection in complex background using attention guided bi-directional lstm, IEEE Access 8 (2020) 161337 – 161348.
- [9] Y. Chen, W. Li, L. Wanga, J. Hu, M. Ye, Vision-based fall event detection in complex background using attention guided bi-directional lstm, IEEE Access 8 (2020) 161337 161348.
- [10] K. He, X. Zhang, S. Ren, J. Sun., Deep residual learning for image recognition, Computer Vision and Pattern Recognition (2016) 770–778.
- [11] N. Lu, Y. Wu, L. Feng, J. Song, Deep learning for fall detection: 3d-cnn combined with lstm on video kinematic data, IEEE Journal of Biomedical and Health Informatics (2018) 2168–2194.
- [12] C. Ismael, O. Mar´ıa, E. Buemi, J. J. Berlles, Cnn-lstm architecture for action recognition in videos, SAIV, Simposio Argentino de Imágenes y Visión (2020).
- [13] E. Auvinet, C. Rougier, J. Meunier, A. StArnaud, Multiple cameras fall dataset, Technical report 1350 DIRO Université de Montréal (July 2010).