MACHINE LEARNING MODEL FOR 3D POINT CLOUD PROCESSING AND CONTACTLESS MULTIPLE ACTIVITY DETECTION

A Final Report in
ELG 5901 - Electrical Engineering Project

Presented to Professor Miodrag Bolic

by
XILAI WANG

Ottawa, Canada
2024

MACHINE LEARNING MODEL FOR 3D POINT CLOUD PROCESSING AND CONTACTLESS MULTIPLE ACTIVITY DETECTION

Xilai Wang, 300346284

University of Ottawa, August 2024

ABSTRACT

This study presents the development and implementation of a machine learning model designed for processing 3D point cloud data generated by a mmWave radar sensor. The primary objective of this project is to create a contactless and non-invasive system capable of real-time multiple activity detection, with a focus on fall detection. The study involves introduction of dataset, preprocessing of the data, and the training of a deep learning model combining PointNet and LSTM architectures. The model's performance is evaluated on various activity classes, demonstrating its potential as an effective solution for monitoring systems. The results indicate a high accuracy rate in detecting critical activities such as falls, with a weighted average accuracy across all classes. This research contributes to the ongoing efforts to develop affordable and reliable activity recognition systems for healthcare applications.

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

I would like to thank my supervisor in this project, Dr. Bolic.

I would like to thank my colleagues in this project: Amr Mourad, Charlie Labaki, Saad Rhanmouni, and Zixiong Han.

Thank you for your kindness, professionalism, creativity, collaboration, and support, which made this summer an extraordinary one.

CHAPTER 1

INTRODUCTION

Falls are a major risk for the elderly, necessitating reliable, real-time detection systems. This project addresses the need for a non-invasive, affordable solution using mmWave radar. The goal is to develop an accurate, easy-to-implement system for real-time activity recognition, emphasizing fall detection and immediate caregiver notification. This report mainly introduces the machine learning model deployed to this project.

1.1 Background and Motivation

Real-time fall detection and activity recognition is important in elderly home care scenario. Falls are the leading cause of injury for adults ages 65 years and older. According to [1], approximately 2500 to 4000 adults age 65 or older die of unintentional falls per month, as depicted in Figure 1.

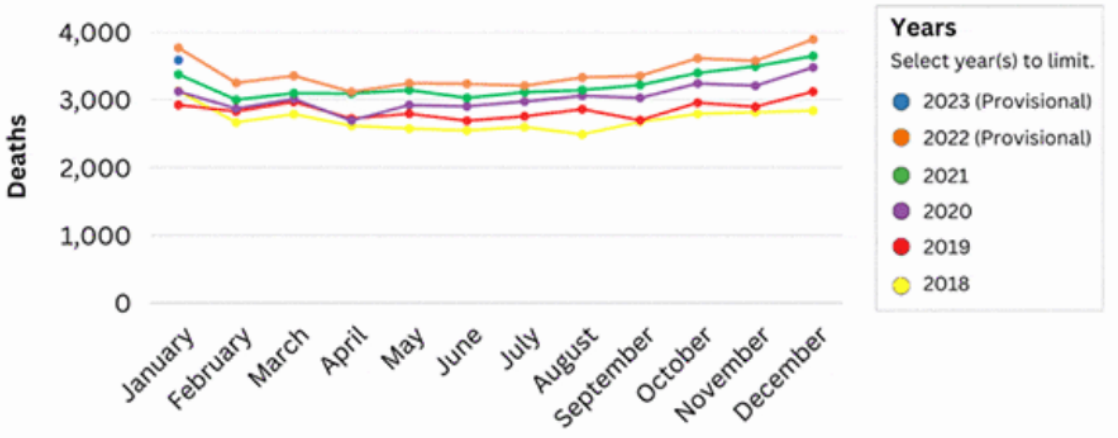

Figure 1: Unintentional older adult fall trends [1]

1.2 Problem Statement

In elderly home care settings, there is an urgent need for a system which can monitor activities of elderly in real time, being affordable and noninvasive simultaneously. Among mainstream options for sensor in the market, LiDAR can capture abundant point cloud yet being expensive, camera can capture clear information about human activity while invading privacy. mmWave radar is an appropriate choice being affordable and noninvasive simultaneously.

1.3 Objectives

Design an easy to implement real-time activity recognition system used in elderly homes such that is accurate, contactless, noninvasive, camera-free, cost-efficient with instant notification to the nurses when detected a falling.

1.4 Contributions

The key contributions of this study are:

- Data analysis on multi-class activity classification dataset.
- Data preprocessing on multi-class activity classification dataset.
- A machine learning model which accurately classifies multiple classes of activities using radar point cloud.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This section reviews HAR approaches utilizing point cloud data. Here is a glimpse of innovative ideas worth considering for our project found in the literature: utilizing joint coordinates (OpenPose skeleton) exclusively in pretraining [5], using feature maps instead of point clouds as input for deep learning models [6], and incorporating binary neural networks and edge intelligence [7].

IEEE Sensors Journal is an important journal on point clouds based HAR. Most of articles in this section are from IEEE Sensors Journal. Most influential articles were found at this journal.

2.2 Sensors

Most point cloud data are generated by radar, LiDAR can also be used for data collection, with the weakness of being expensive. [8] uses IWR1443-Boost radar. [6, 9] uses AWR1243 radar. [10] uses NodeNs ZERO 60 GHz IQ radar. [11] uses 3D solid-state LiDAR with model number HPS-3D160 and a TI 79-81 GHz mmWave radar without declaring the exact model number to generate point cloud. [12] did not declare the radar used for the article. [5, 7] are based on existing point cloud dataset hence no sensor is needed.

2.3 Datasets

Currently, 3 point cloud datasets are available:

- **MMActivity [13]** is the first point cloud dataset for HAR, published on the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNets) 2019 (co-located with MobiCom). Available at: https://github.com/nesl/RadHAR with point cloud and raw data.

- **DGUHA [14]** includes both point cloud and Kinect v4 skeleton data. It is collected from 19 subjects and contains seven human activities including running, jumping, sitting down and standing up, both upper limb extension, falling forward, right limb extension, and left limb extension. The point cloud data in the DGUHA dataset is collected by using a mmWave radar sensor, TI's IWR1443-BOOST radar, which was mounted parallel to the ground at a height of 1.2 m in the experiment. Available at: https://github.com/AIC-DGU/MTGEA.

- **MARS [15]** Available at: https://github.com/SizheAn/MARS, containing data with / without outlier with labels in the form of Kinect v2 skeleton of 25 joints. A pretrained model with recorded accuracy is also provided.

All the three datasets above are collected by TI IWR1443-Boost radar.

2.4 Algorithms

Generally, the HAR task can be divided into three phases: data preprocessing, feature extraction, and classification. Each of these phases encompasses one or more

mainstream solutions.

- **Preprocess:** After Fourier transform, point cloud data is generated. Preprocess of point cloud data includes but not limited to: alignment of frames per second, alignment of amount of points per frame, and denoising. Some algorithms are used for denoising, mainly two: CFAR [6, 8, 9] and DBSCAN [6, 10–12]. In [11], preprocessing also includes background removal and voxelization. In [6], feature mapping of point cloud data is the last step of preprocessing. In [11], BIRCH algorithm is conducted after DBSCAN for a better result.

- **Feature Extraction:** This phase extracts important information from preprocessed point cloud data and input the information to classifier. Mainstream model for feature extraction is CNN [5–9, 12]. In [8], Multilinear principle component analysis is conducted before CNN. [7, 8] used histogram to assist feature extraction. In [5], 3D joint coordinate, output of CNN and input of GNN, is the intermediate form of data.

- **Classification:** LSTM [9, 10] and Dense Layer [6, 7, 12] are the mainstream methods for classification, the last step of HAR. Other models such as SVM [8], Adaptive order hidden markov model (AO-HMM) and crossover path disambiguation algorithm (CPDA) [11], and Graph neural network (GNN) [5] are also used for classification.

2.5 Comparison

All articles discussed in this section are compared in table 1.

Author	Process	Feature	Sensor	Code Avai.	Publication	Dataset
Kim et al. [9]	CA-CFAR →CNN →LSTM	Take only x-z domain point clouds. Aggregate point clouds within 2 seconds.	4 IWR1243's	No	IEEE Sensors Journal 2021	Not published
Yu et al. [10]	DBSCAN →LSTM	Proposed a benchmark dataset which can lead to higher accuracy	NodeNs ZERO 60 GHz IQ radar	No	IEEE Sensors Journal 2022	Not published
Qiao et al. [8]	CA-CFAR →MPCA →CNN →Histogram →SVM	Lightweight network able to directly operate on sparse 3D point cloud data, combining advantages of CNN and Multidimensional PCA.	IWR1443-Boost	No	IEEE Transactions on Geoscience and Remote Sensing 2022	Not published
Luo et al. [7]	Point cloud →histogram →CNN →Dense	Represent point cloud with histogram. Utilize binary neural network and edge device.	None	No	IEEE Transactions on Mobile Computing 2024	MMActivity [13], DGUHA [14]
Alam et al. [11]	Background substracting →Voxel fitting →DBSCAN →BIRCH →AO-HMM →CPDA	Multiple-person tracking, multiple-environment recognition, comprehensive HAR solution.	LiDAR: HPS-3D160, Radar: TI 79-81 GHz mmWave, model not spicified	No	IEEE INFOCOM 2021	Not published
Wu et al. [6]	CFAR →DBSCAN →Association →Point cloud →Feature mapping →CNN →Dense	Utilizing feature maps of 4 channels, multi-person recognition, indoor and aquatic activity recognition	AWR1243	No	IEEE Sensors Journal 2023	Not published
Yu et al. [12]	DBSCAN →CNN →Dense	Utilize dual-view data, data augmentation	Not specified	No	IEEE Sensors Journal 2022	Not published
Lee & Kim [5]	Point cloud →CNN →3D joint coordinate →GNN	Utilizing joint (skeleton) in pretraining ONLY	None	No	Applied Science	MARS [15], MMActivity [13]

Table 1: Point Cloud Methods

CHAPTER 3

DATASET

Dataset used in this study is collected and firstly used by A. Rezaei [4]. The dataset consists of point cloud time series data being labeled using corresponding classes. An mmWave radar was used to collect these data.

3.1 mmWave Radar Sensor

Antenna board IWR6843ISK-ODS by Texas Instruments, Dallas, USA, which has 4 receivers and 3 transmitters was used for data collection. In the study, the Antenna board works in 60 GHz frequency range. The antenna board is as shown in Figure2.

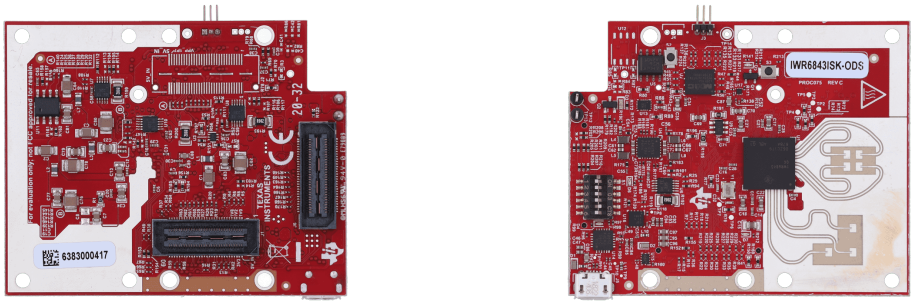

Figure 2: Photo of IWR6843ISK-ODS, top(left), bottom(right) [2]

The antenna board was installed on the carrier board, MMWAVEICBOOST, for power supply and communication with computer. The MMWAVEICBOOST carrier card expands capabilities of select 60 GHz mmWave evaluation modules. This board provides advanced software developement, debug features such as trace and single step via TI's Code Composers compatible debuggers. [3]

The MMWAVEICBOOST sensors carrier card platform is depicted in Figure3.

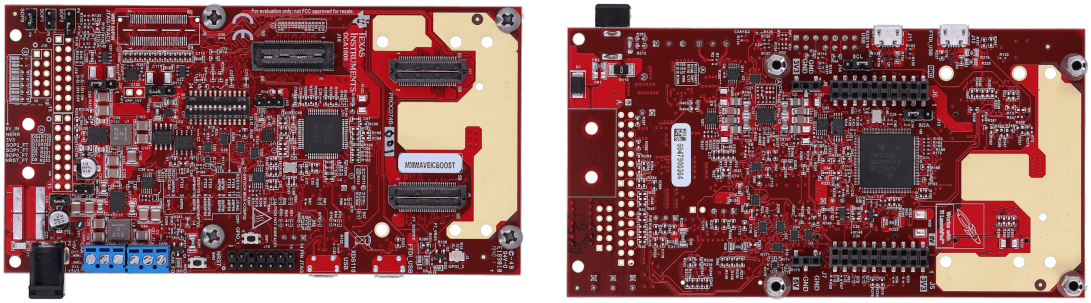

Figure 3: Photo of MMWAVEICBOOSTS, top(left), bottom(right) [3]

3.2 Data Collection Setup

The radar was installed in a room of 5m ×5m ×3m (length, depth, height), with an ambient temperature being controlled between 20°C and 22°C on the side wall and overhead.

For the side mount, the radar was elevated 2m from the floor and tilted approximately 10°down for better coverage.

For the overhead mount, the radar was installed in the center of the room at 3m elevation facing the floor with no tilt. [4]

The room in which dataset was collected is depicted in Figure4.

Figure 4: Experimental room used for data collection [4]

3.3 Radar Configuration

The radar configuration file for data collection is shown in Table 2.

flushCfg
dfeDataOutputMode 1
channelCfg 15 5 0
adcCfg 2 1
adcbufCfg 0 1 1 1
profileCfg 0 60.6 30 10 62 0 0 53 1 128 2500 0 0 30
chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 0 0 0 0 0 4
frameCfg 0 1 128 0 50 1 0
lowPower 0 1
guiMonitor 1 1 0 0
cfarCfg 6 4 4 4 4 16 16 4 4 55 67 0
doaCfg 600 1875 30 1 1 0
SceneryParam -6 6 0.5 6
GatingParam 3 2 2 0
StateParam 10 5 100 100 5
AllocationParam 250 250 0.25 30 1 2
AccelerationParam 1 1 1
trackingCfg 1 2 250 20 52 82 50 90
sensorStart

Table 2: Radar configuration file for data collection [4]

3.4 Data Format and Classes

The original dataset is stored in a CSV file and contains the columns as stated in Table3. The columns in processed dataset is as shown in Table 5.

Column Name	Description
timestamp	Timestamp at which the point was collected, in ISO8601 format, precise to microseconds.
frame_number	The frame number of the radar data, which shows which frame the data point belongs to.
x	The x-coordinate of the point corresponding to this row.
y	The y-coordinate of the point corresponding to this row.
z	The z-coordinate of the point corresponding to this row.
doppler	The Doppler velocity of the target point.
snr	The signal-to-noise ratio, which is the ratio between the strength of the detected radar signal and the background noise.
range	The straight-line distance between the radar and the target point.
azimuth	The azimuth angle, which indicates the horizontal angle of the target point relative to the radar.
elevation	The elevation angle, indicating the vertical angle of the target point relative to the radar.
target_id	Target identification value of the point, for meaning see Table 4.
class_activity	Indicates the class of the current time series data.

Table 3: Columns in original dataset

Column Name	Description
-1	No target is detected.
0-252	Target detected with identification number 0-252.
253	Point not associated, signal to noise ratio is too weak.
254	Point not associated, located outside of the boundary of interest.
255	Point not associated, considered as noise.

Table 4: Meanings of target identification value [4]

Column Name	Description
seq_no	Indicates which time series data is currently among all the time series data
frame_no	Indicates which frame the current frame is within its corresponding time series data
x	Indicates the x-coordinate of the point corresponding to this row.
y	Indicates the y-coordinate of the point corresponding to this row.
z	Indicates the z-coordinate of the point corresponding to this row.
class_activity	Indicates the class of the current time series data

Table 5: Columns in processed dataset

In the original dataset, the class_activity column has several possible values as shown in Table 6.

No.	Class	Description
1	StartSignal	An activity performed by a participant that showed the start of the experiment
2	FinishSignal	An activity performed by a participant that showed the end of the experiment
3	Background	There was no participant in the room.
4	Falling	A participant was performing simulated falls.
5	LayBed	A participant was lying in the bed.
6	LayFloor	A participant was lying on the floor.
7	Sit	A participant was sitting in a chair.
8	SitBed	A participant was sitting in the bed.
9	Stand	A participant was standing in the room.
10	Transition	A participant was transiting from an activity to another activity.
11	Walking	A participant was walking in the room.
12	EndTime	This class indicated the end time of an experiment.

Table 6: Classes in original dataset

In this study, the original class have been modified and merged, resulting in the new values in Table 7. Falling and LayFloor are merged because they have some similar-

ity, and any one of them happening means the person needs help. Sit and SitBed, Stand and Walking are merged respectively because they have similar meaning. StartSignal, FinishSignal, Background, Transition and EndTime are deleted because they are meaningless to this study.

No.	Class	Description
1	Falling+LayFloor	The class after merging Falling and LayFloor.
2	LayBed	Unchanged class LayBed.
3	Sit+SitBed	The class after merging Sit and SitBed.
4	Stand+Walking	The class after merging Stand and Walking.

Table 7: Classes in modified dataset

3.5 Sliding Window Process

The real-time activity recognition model is designed to read data using a sliding window. The window size and hop size of final sliding window are 20 frames and 5 frames respectively. When there are more than 1 label captured in one window, use the rule in Algorithm 3.5 to label the data in the window. The falling class has the shortest time series in terms of frames, but it is also the most critical class. In order to avoid missing falls, a sliding window is given label Falling as long as there are no less than 10 frames with the label Falling in it. Detailed length distribution of Falling is shown in Figure9.

Algorithm 1 Sliding window labelling rule

Input: The window size s of the sliding window, The class_activity label for each frame in the sliding window, $c_0, \ldots, c_{s-1}$.

Output: A single class_activity for the entire sliding window, C.

if $c_0 = \ldots = c_n = \ldots = c_{s-1}$ **then**
    $C \leftarrow c_0$
else if $\sum_{i=0}^{s-1} \mathbf{1}(c_i = 1) \geq 10$ **then**
    $C \leftarrow 1$
else
    $C \leftarrow \text{mode}(c_0, c_1, \ldots, c_{s-1})$
end if
return C

3.6 Dataset Statistics

Some statistic information about dataset used in this study.

3.6.1 Class Distribution

The number of data samples for each class is as shown in Figure5.

3.6.2 Length Distribution (Number of Frames)

The distribution of data sample lengths (number of frames) is shown in Figure6

The 20th and 80th percentile of sequence length in each class is shown in Figure7 and 8.

The sequence length distribution for Falling class is shown in Figure9.

3.6.3 Density of Point Cloud

After eliminating points with unwanted target_id, density of point cloud is as shown in Figure10.

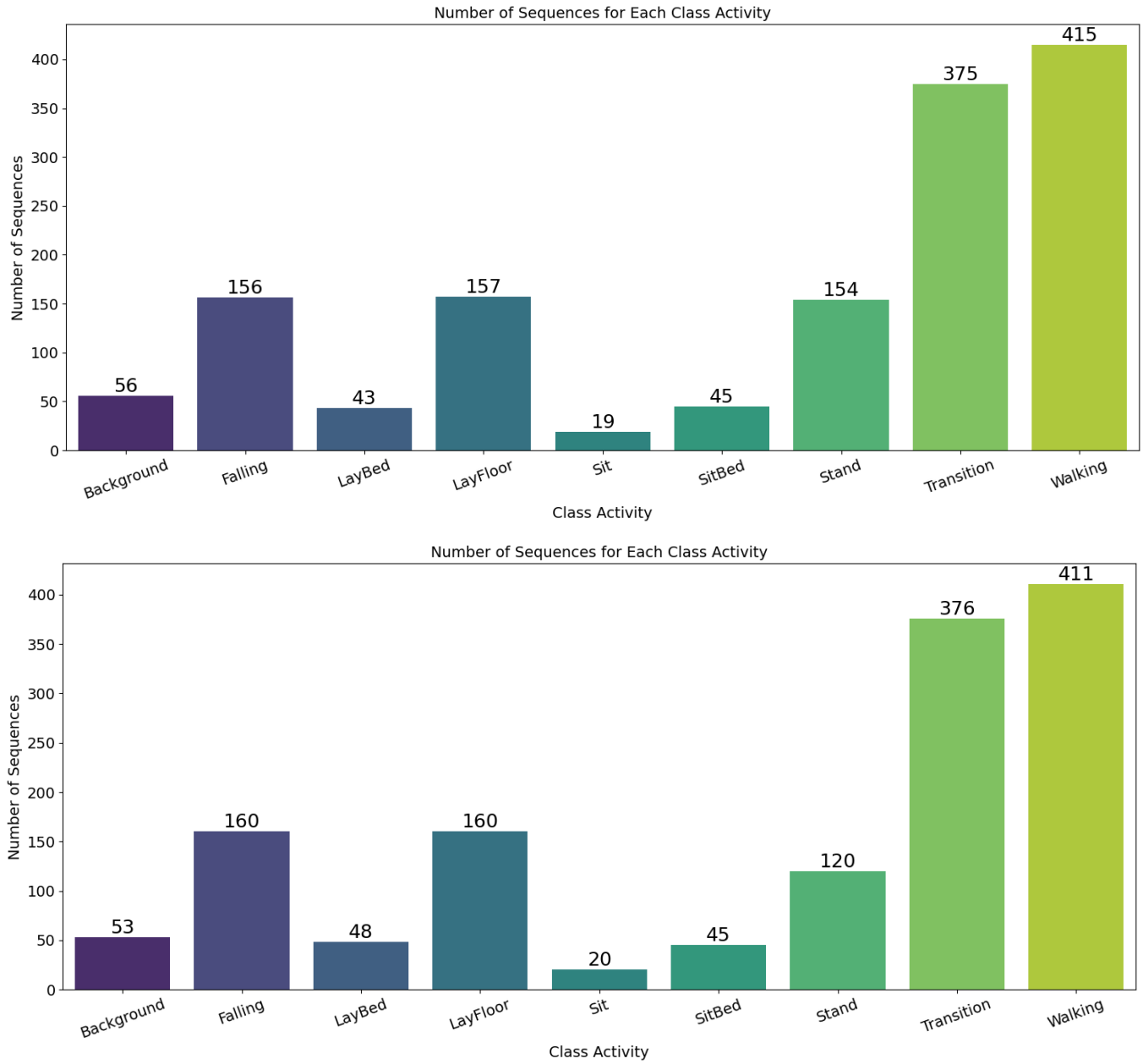

Figure 5: The number of data samples for each class. Top: side mount, bottom: overhead mount.

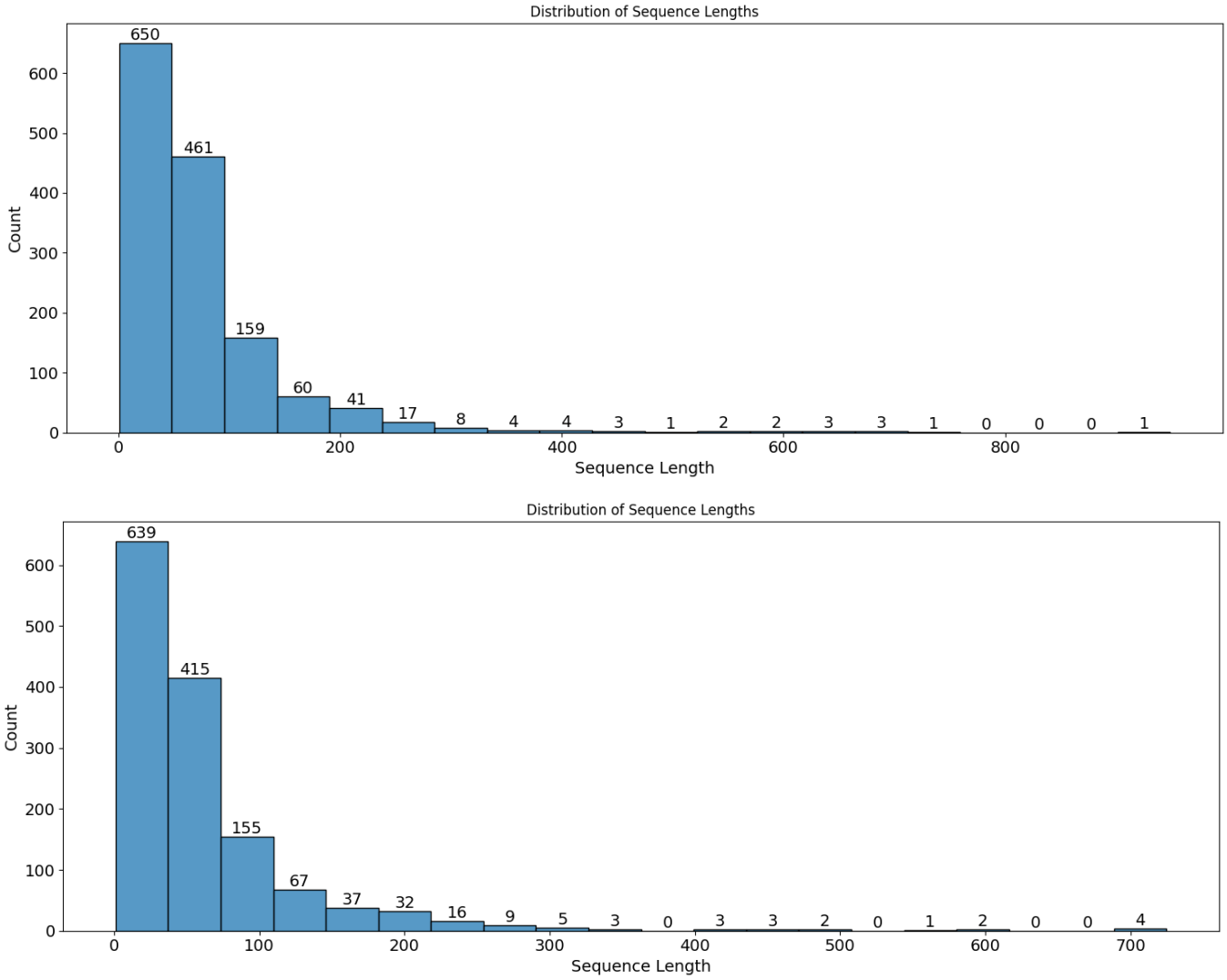

Figure 6: The distribution of data sample lengths (number of frames). Top: side mount, bottom: overhead mount.

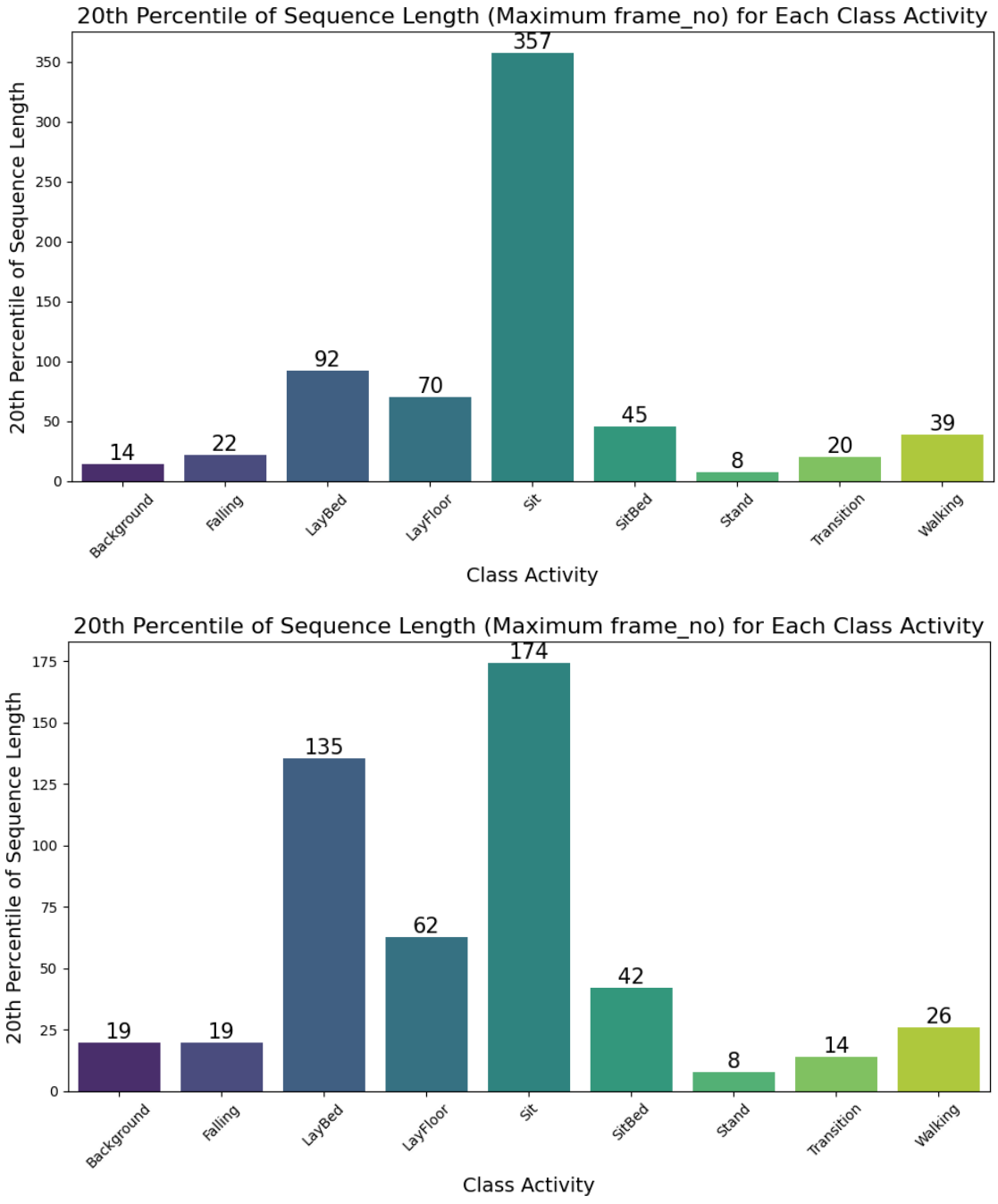

Figure 7: The 20th percentile of data sample lengths (number of frames) in each class. Top: side mount, bottom: overhead mount.

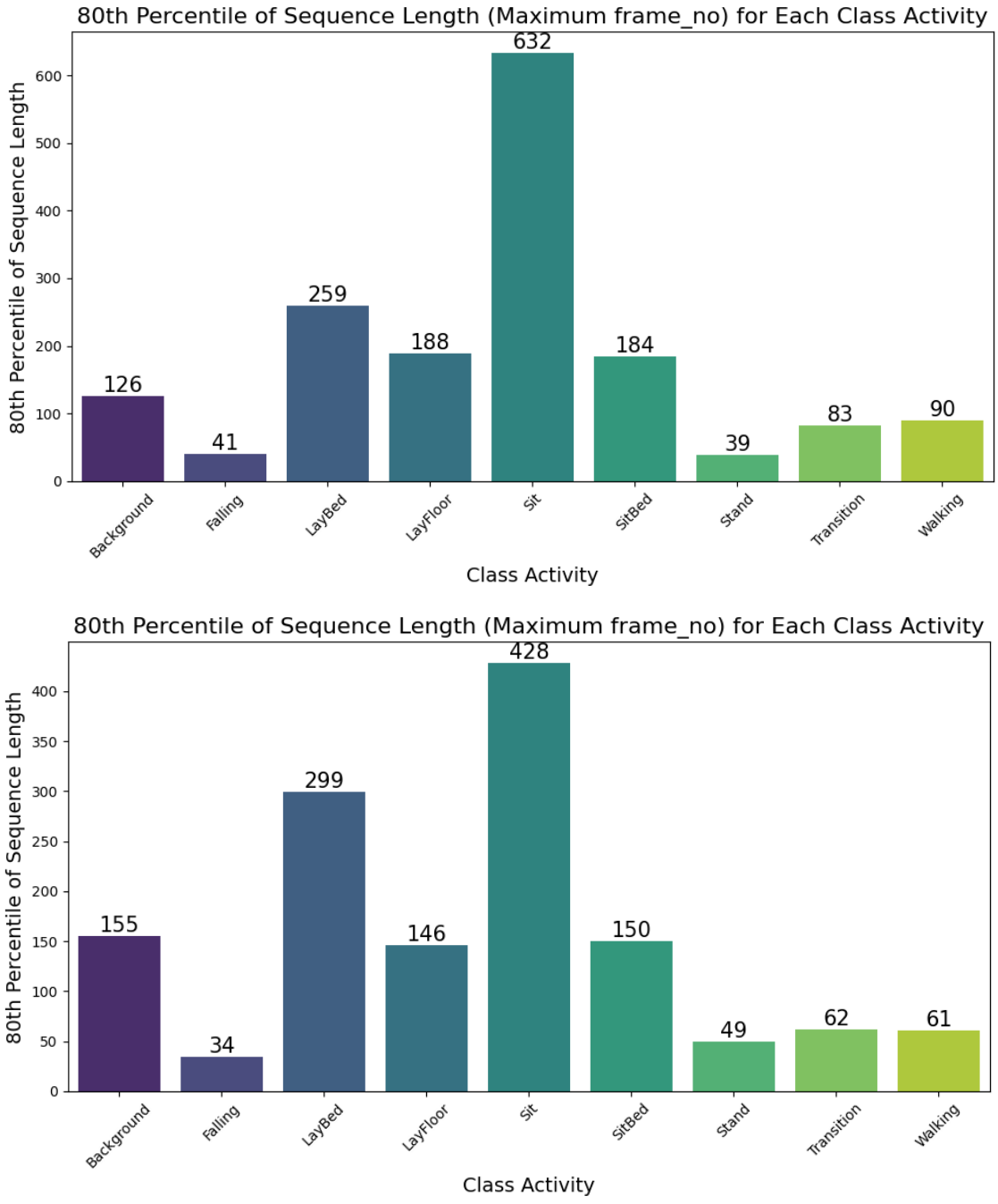

Figure 8: The 80th percentile of data sample lengths (number of frames) in each class. Top: side mount, bottom: overhead mount.

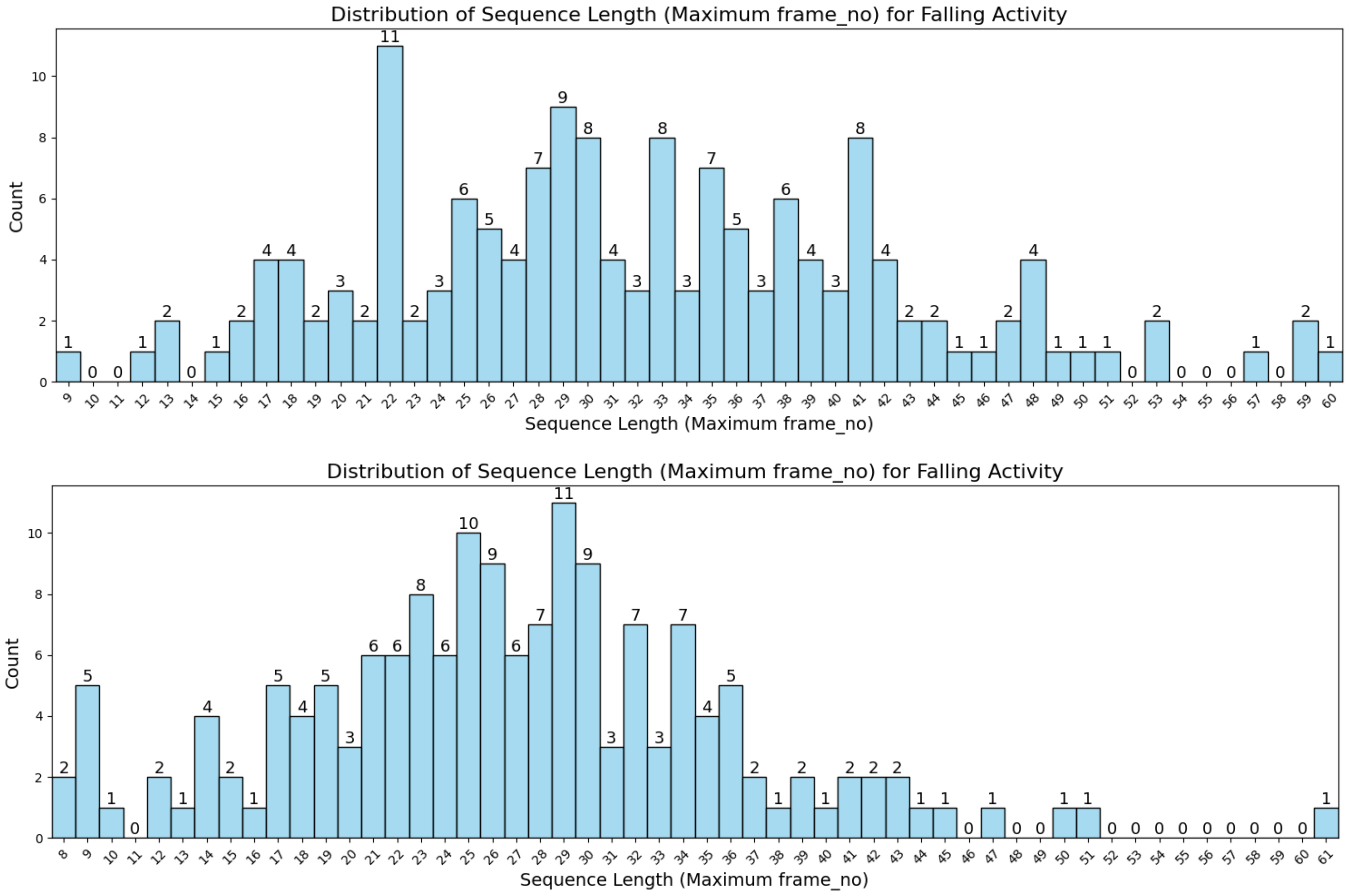

Figure 9: The distribution of sequence length in Falling class. Top: side mount, bottom: overhead mount.

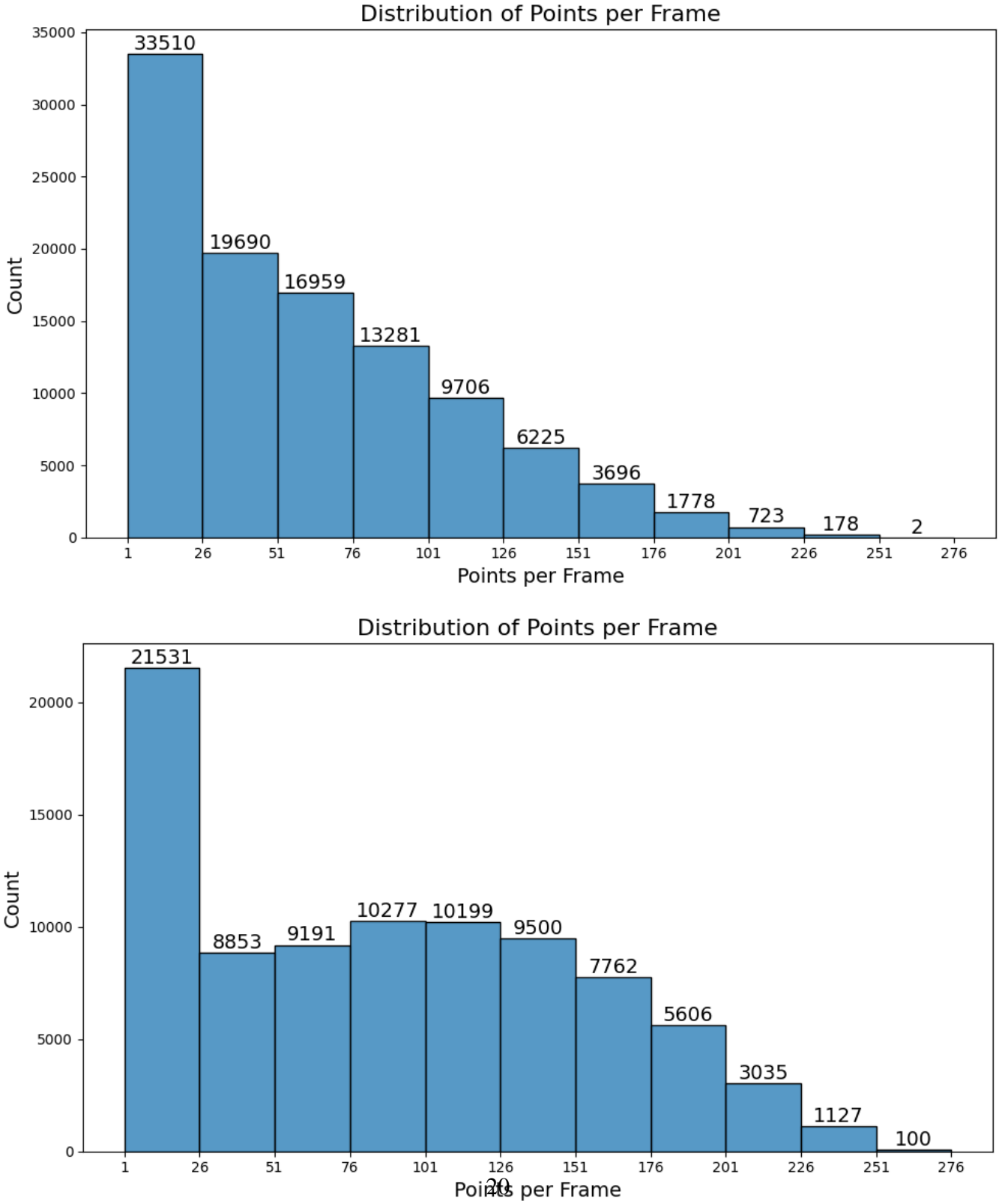

Figure 10: The distribution of Point cloud density (points per frame). Top: side mount, bottom: overhead mount.

CHAPTER 4

METHODOLOGY

The machine learning model used in this study is based on PointNet [16] and LSTM [17].

4.1 Model

The majority of the PointNet model is used as the feature extractor in this study. The adopted portion is enclosed in a red box in Figure 11. The 1×1024 global feature vector is the input to the LSTM.

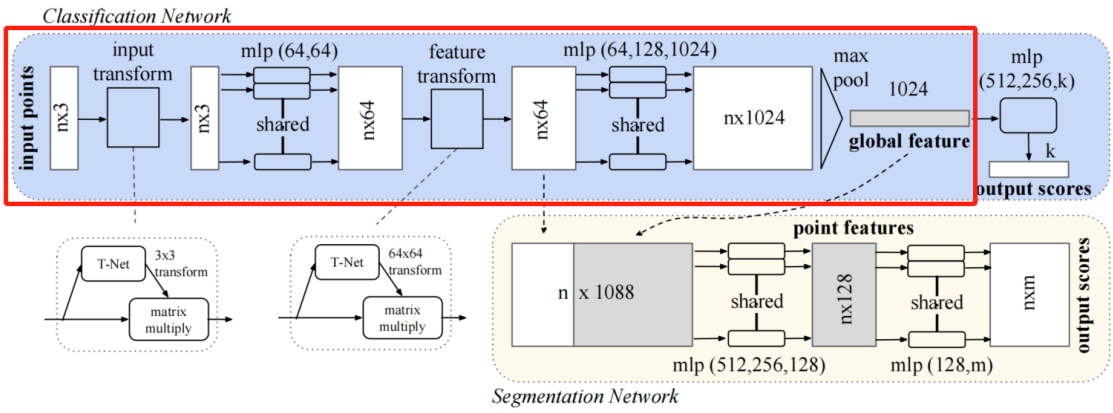

Figure 11: Adopted portion in PointNet architecture

The architecture of the entire model is depicted as Figure 12. Linear layer has a dropout rate of 0.2.

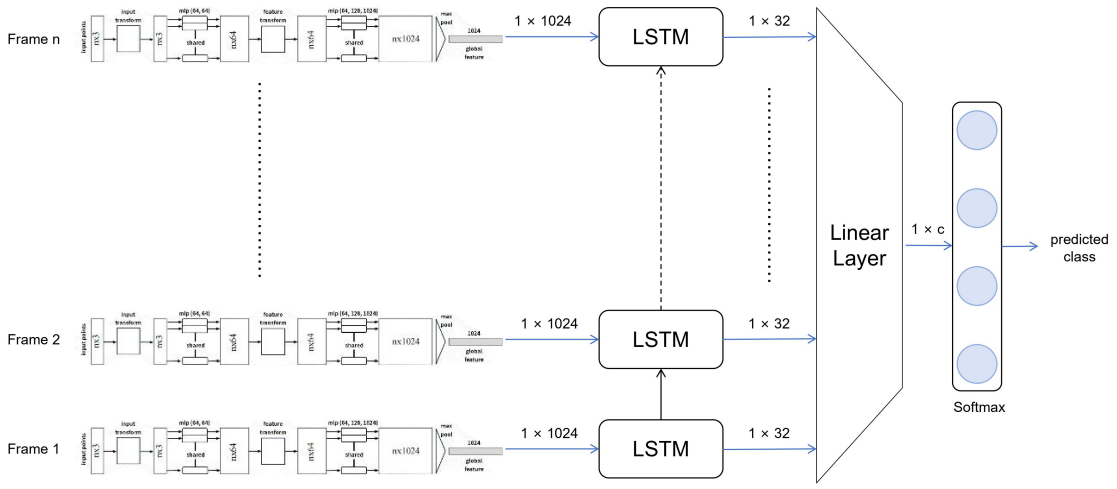

Figure 12: Architecture of the entire model used in this study. n: number of frames of the input. c: number of classes

4.2 Training

The deployed model is trained on the side mount dataset. After classes modification (converting classes from Table 6 to Table 7), training-validation set splitting and sliding window process (processing data using rule in Algorithm 3.5), the class distribution in the training set and validation set is shown in Figure 13.

Training result showS that the model performs poorly on the 3 classes other than Standing+Walking. Therefore undersampling was conducted, resulting in a new training set with class distribution as Figure 14. As shown, Falling+LayFloor has more samples than the other 3 classess, because it is the most important class and is given the highest priority.

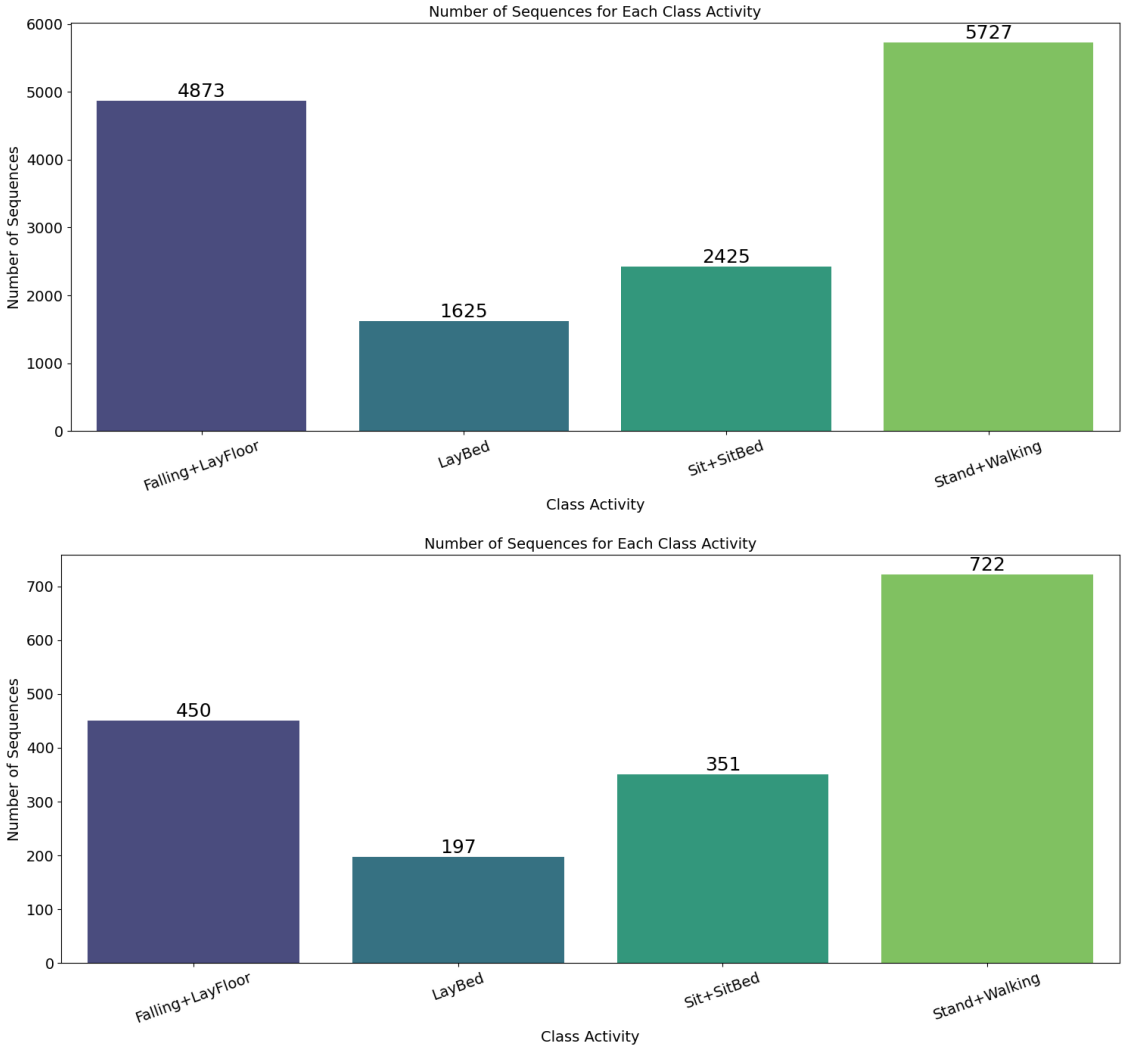

Figure 13: Class distribution of processed side mount dataset, top: training set, bottom: validation set.

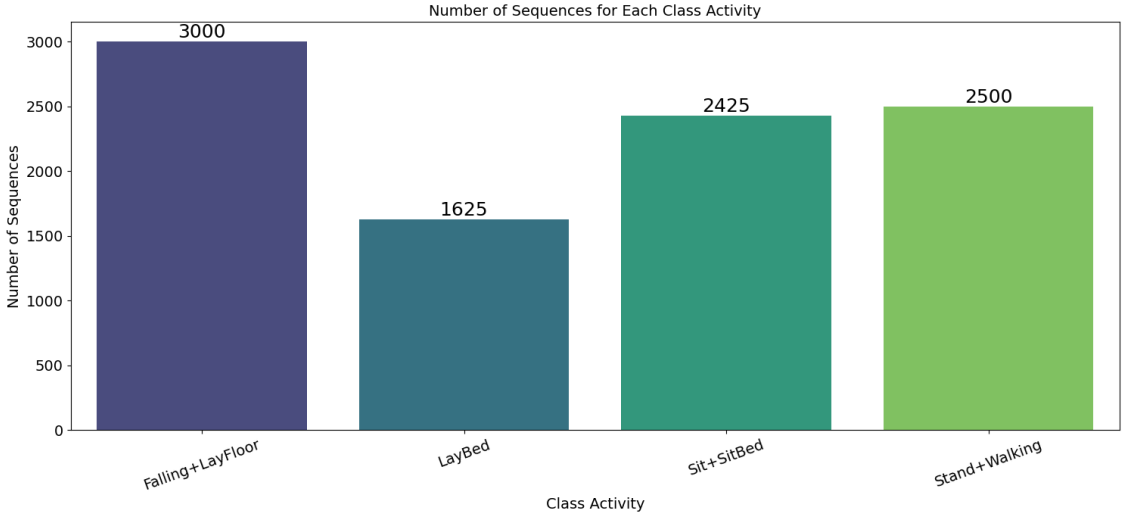

Figure 14: Class distribution of undersampled training set

CHAPTER 5

RESULT

In the experiment, the accuracy of the model on the validation set is calculated separately for each class and then weighted by the proportion of samples in each class within the training set to obtain the overall average.

During the training, the accuracy of model fluctuates after fitting. Among models resulted from each epoch, a model that performs well and is balanced across all classes is selected. The accuracy of the finally deployed model is shown in Table 8.

Class	Accuracy
Falling+LayFloor	0.951
LayBed	0.736
Sit+SitBed	0.823
Stand+Walking	0.861
Overall	0.859

Table 8: Accuracy of deployed model on each class

After the chosen model was deployed, the system was tested in the lab. The test result was shown in Table 9. The radar configuration used for testing is different from data collection. Content of the radar configuration file is shown in Table 10.

Activity	Result
Detected Falls / All Falls	9 / 10
Detected Lay-Floors / All Lay-Floors	3 / 3
Detected Sits / All Sits	15 / 15
Squats Detected as Falling / Actual Squats	0 / 8

Table 9: Test result of the system in the lab

```
sensorStop
flushCfg
dfeDataOutputMode 1
channelCfg 15 7 0
adcCfg 2 1
adcbufCfg -1 0 1 1 1
lowPower 0 0
profileCfg 0 60.75 30 25 59.10 657930 0 54.71 1 96 2950 2 1 36
chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 0 0 0 0 0 2
chirpCfg 2 2 0 0 0 0 0 4
frameCfg 0 2 96 0 55 1 0
dynamicRACfarCfg -1 4 4 2 2 8 12 4 12 5 8 0.40 1 1
staticRACfarCfg -1 6 2 2 2 8 8 6 4 8 15 0.30 0 0
dynamicRangeAngleCfg -1 0.75 0.0010 1 0
dynamic2DAngleCfg -1 3.0 0.0300 1 0 1 0.30 0.85 8
staticRangeAngleCfg -1 0 8 8
antGeometry0 0 0 -1 -1 -2 -2 -3 -3 -2 -2 -3 -3
antGeometry1 0 -1 -1 0 0 -1 -1 0 -2 -3 -3 -2
antPhaseRot 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1
fovCfg -1 70.0 70.0
compRangeBiasAndRxChanPhase 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
staticBoundaryBox -3 3 0.5 7.5 0 3
boundaryBox -4 4 0 8 0 3
sensorPosition 2 0 15
gatingParam 3 2 2 3 4
stateParam 3 3 12 500 5 6000
allocationParam 20 100 0.1 20 0.5 20
maxAcceleration 0.1 0.1 0.1
trackingCfg 1 2 800 30 46 96 55
presenceBoundaryBox -3 3 0.5 7.5 0 3
sensorStart
```

Table 10: Radar configuration file for lab test

CHAPTER 6

CONCLUSIONS

The developed model effectively recognizes activities from mmWave radar data, particularly excelling in fall detection. While the system shows promise for real-time monitoring, future improvements could enhance its accuracy across diverse activities. This work advances affordable, non-invasive monitoring solutions in healthcare.

REFERENCE LIST

[1] U. S. C. for Disease Control and Prevention. (2024) Older adult falls data. [Online]. Available: https://www.cdc.gov/falls/data-research/index.html

[2] T. Instruments. (2024) Iwr6843isk-ods. [Online]. Available: https://www.ti.com/tool/IWR6843ISK-ODS

[3] ——. (2024) Mmwaveicboost. [Online]. Available: https://www.ti.com/tool/MMWAVEICBOOST

[4] A. Rezaei, "Human activity recognition and fall detection using unobtrusive technologies," Ph.D. dissertation, UNSW Sydney, 2023.

[5] G. Lee and J. Kim, "Improving human activity recognition for sparse radar point clouds: A graph neural network model with pre-trained 3d human-joint coordinates," *Applied Sciences*, vol. 12, no. 4, p. 2168, 2022.

[6] Z. Wu, Z. Cao, X. Yu, J. Zhu, C. Song, and Z. Xu, "A novel multi-person activity recognition algorithm based on point clouds measured by millimeter-wave mimo radar," *IEEE Sensors Journal*, 2023.

[7] F. Luo, S. Khan, A. Li, Y. Huang, and K. Wu, "Edgeactnet: Edge intelligence-enabled human activity recognition using radar point cloud," *IEEE Transactions on Mobile Computing*, 2023.

[8] X. Qiao, Y. Feng, S. Liu, T. Shan, and R. Tao, "Radar point clouds processing for human activity classification using convolutional multilinear subspace learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[9] Y. Kim, I. Alnujaim, and D. Oh, "Human activity classification based on point clouds measured by millimeter wave mimo radar with deep recurrent neural networks," *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13 522–13 529, 2021.

[10] Z. Yu, A. Taha, W. Taylor, A. Zahid, K. Rajab, H. Heidari, M. A. Imran, and Q. H. Abbasi, "A radar-based human activity recognition using a novel 3-d point cloud classifier," *IEEE Sensors Journal*, vol. 22, no. 19, pp. 18 218–18 227, 2022.

[11] M. A. U. Alam, M. M. Rahman, and J. Q. Widberg, "Palmar: Towards adaptive multi-inhabitant activity recognition in point-cloud technology," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[12] C. Yu, Z. Xu, K. Yan, Y.-R. Chien, S.-H. Fang, and H.-C. Wu, "Noninvasive human activity recognition using millimeter-wave radar," *IEEE Systems Journal*, vol. 16, no. 2, pp. 3036–3047, 2022.

[13] A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, "Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar," in *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*, 2019, pp. 51–56.

[14] G. Lee and J. Kim, "Mtgea: a multimodal two-stream gnn framework for efficient point cloud and skeleton data alignment," *Sensors*, vol. 23, no. 5, p. 2787, 2023.

[15] S. An and U. Y. Ogras, "Mars: mmwave-based assistive rehabilitation system for smart healthcare," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 20, no. 5s, pp. 1–22, 2021.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.