

Multi-location Smartphone-based Activities of Daily Life Recognition using Deep Learning Techniques



By:

Natasha Saeed

(Registration No: MSSE-00000401614)

Supervisor:

Dr. Ali Hassan

Co-Supervisor:

Dr. Ahsan Shahzad

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING
COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING,
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

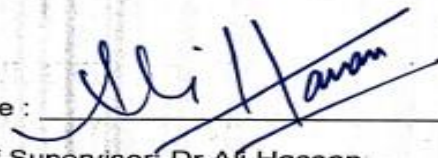
ISLAMABAD,

September 2, 2024

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by NS **Natasha Saeed** Registration No. **0000401614**, of College of E&ME has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the thesis.

Signature :



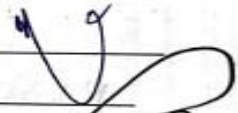
Name of Supervisor: Dr Ali Hassan

Date: 02-09-2024

Signature of HOD:

(Dr Usman Qamar)

Date: 02-09-2024



Signature of Dean:

(Brig Dr Nasir Rashid)

Date: 02-09-2024



Multi-location Smartphone-based Activities of Daily Life Recognition using Deep Learning Techniques

By:

Natasha Saeed

(Registration No: MSSE-00000401614)

A thesis submitted to the National University of Sciences and Technology,
Islamabad
in partial fulfillment of the requirements for the degree of

Master of Sciences in Software Engineering

Supervisor:

Dr. Ali Hassan

Co-Supervisor:

Dr. Ahsan Shahzad

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING
COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING,
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD,

September 2, 2024

*Dedicated to my parents whose
tremendous continuous support and
endless prayers led me to this
accomplishment.*

Acknowledgement

I begin by expressing my deepest gratitude to Allah Almighty, the most merciful and kind, for His countless blessings and for giving me the strength to complete this work. I am profoundly grateful to my supervisor, Dr. Ali Hassan, for his steadfast intellectual support and invaluable insights throughout my research journey. His inspiring mentorship, technical guidance, and moral encouragement have been instrumental in sharpening my skills and critical thinking, enabling me to achieve my research goals. This dissertation would not have been possible without his ongoing support.

I am deeply thankful for the motivation by my co-supervisor, Dr. Ahsan Shahzad, for his thoughtful suggestions and guidance. I also extend my gratitude to my parents for their constant support.

Abstract

The rapid growth of the elderly population has underscored the importance of accurately recognizing Activities of Daily Living (ADLs) for effective health monitoring and timely interventions, particularly in regions such as Korea, where the aging demographic presents distinct challenges. Traditional methods often struggle with processing complex sensor data and optimizing model performance, especially when dealing with varied activities captured from multiple body locations. To address these limitations, we propose an advanced deep learning framework that utilises Long Short-Term Memory (LSTM) networks to analyze time-series sensor data, enabling the model to capture temporal dependencies and patterns within the signals. Additionally, we transform these time-series signals into Short-Time Fourier Transform (STFT) spectrogram images, which are subsequently processed using Convolutional Neural Network (CNN), EfficientNet_B0, and Vision Transformer (ViT) models through transfer learning techniques. To mitigate the class imbalance inherent in the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) is employed to generate synthetic samples for underrepresented classes. Furthermore, a Butterworth low-pass filter is applied to remove background noise, ensuring higher quality data for model training. The proposed models exhibited robust performance across multiple sensor placements, with the Efficient-Net_B0 model demonstrating superior accuracy, achieving 99%, 98%, 99%, and 97% for sensor placements on the bag, belly, hand, and thigh, respectively. These results highlight the potential of our approach for enhancing ADL recognition in health monitoring systems.

Keywords: Activities of daily living (ADLs), Activity recognition, Deep learning, Healthcare, Smartphone sensors data, Elderly people, Transfer Learning.

Contents

DEDICATION	i
ACKNOWLEDGMENT	ii
ABSTRACT	iii
LIST OF FIGURES	v
LIST OF TABLES	vi
LIST OF ABBREVIATIONS	vii
1 Introduction and Motivation	1
1.1 Goals and Objectives	2
1.2 Motivation	3
1.3 Problem Statement.....	3
1.4 Thesis Structure	4
2 Literature Review	5
2.1 Traditional Methods.....	5
2.2 Deep Learning based Models.....	9
3 Methodology	16
3.1 Dataset Collection.....	16
3.2 Data Segmentation	18
3.3 Data Pre-processing	19
3.3.1 Short-Time Fourier Transform (STFT)	20
3.3.2 Synthetic Minority Over-sampling Technique (SMOTE)	22
3.4 Classification Models.....	23
3.4.1 Long Short-Term Memory (LSTM).....	23
3.4.2 Convolutional Neural Network (CNN).....	25
3.4.3 EfficientNet_B0	26
3.4.4 ViT (Vision) Transformer	28
3.5 Model Evaluation.....	29
4 Experimentation & Results	31
4.1 Experimentation Setup.....	31
4.2 Classification Results.....	31
4.2.1 Results on direct Application of LSTM Models	32
4.2.2 Results of CNN and Transfer Learning Models.....	37
4.2.3 Results of Transfer Learning Models.....	42
4.2.4 Vision Transformer (ViT).....	47
5 Conclusion & Future Work	53
REFERENCES	

List of Figures

3.1	Proposed Framework.....	16
3.2	Fig: (a) Smartphone placement locations (b) Classified ADLs.....	17
3.3	Time-series data segmentation using MATLAB Signal Labeller tool	19
3.4	Butterworth Filter application on signal data.....	20
3.5	Generated Spectrograms using STFT for Different Activities.....	21
3.6	Data Balancing using SMOTE Technique.....	23
3.7	Proposed architecture of LSTM model	24
3.8	Flow Diagram of ViT Transformer	29
4.1	LSTM model Learning curve on Bag Location	32
4.2	LSTM model Learning curve on Belly Location	32
4.3	LSTM model Learning curve on Hand Location	33
4.4	LSTM model Learning curve on Thigh Location	33
4.5	LSTM Confusion Matrix Results on all locations.....	34
4.6	LSTM model results (a) Bag, (b) Belly, (c) Hand, (d) Thigh.....	36
4.7	CNN model Learning curve on Bag Location	37
4.8	CNN model Learning curve on Belly Location	37
4.9	CNN model Learning curve on Hand Location	38
4.10	CNN model Learning curve on Thigh Location	38
4.11	CNN Confusion Matrix Results	39
4.12	CNN model results (a) Bag, (b) Belly, (c) Hand, (d) Thigh.....	41
4.13	EfficientNet_B0 model Learning curve on Bag Location.....	42
4.14	EfficientNet_B0 model Learning curve on Belly Location.....	42
4.15	EfficientNet_B0 model Learning curve on Hand Location.....	43
4.16	EfficientNet_B0 model Learning curve on Thigh Location.....	43
4.17	EfficientNet_B0 Confusion Matrix Results	44
4.18	EfficientNet_B0 model results (a) Bag, (b) Belly, (c) Hand, (d) Thigh.....	45
4.19	Learning curve on Bag Location.....	47
4.20	Learning curve on Belly Location.....	47
4.21	Learning curve on Hand Location.....	48
4.22	Learning curve on Thigh Location.....	48
4.23	Precision, Recall, and F1-Score Curves in ViT Transformer Evaluation.....	49
4.24	ViT Transformer Confusion Matrix Results	50

List of Tables

2.1	Summary of Recent Work on Human Activity Recognition.....	15
3.1	Summary of Activities of Daily Livings.....	18
3.2	Detailed Summary of CNN Model.....	26
1.1	Generic Confusion Matrix	34
1.2	Performance Metrics of LSTM Model	35
1.3	Performance Metrics of CNN Model	40
1.4	Performance Metrics of EfficientNet_B0 Model	46
1.5	Performance Metrics of ViT Transformer	51
1.6	Comparison of Performance Metrics.....	52

LIST OF ABBREVIATIONS

ADL	Activities of daily living
HAR	Human Activity Recognition
STFT	Short-Time Fourier Transform
SMOTE	Synthetic Minority Over-sampling Technique
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
KNN	K-Nearest Neighbour
MCC	Matthews Correlation Coefficient
QDA	Quadratic Discriminant Analysis
GB	Gradient Boost
MLP	Multi-layer Perceptron
NB	Naïve Bayes
LWRT	Light Weight Reinforced Thermoplastic
GBDT	Gradient Boosted Decision Tree
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
RF	Random Forest

Chapter 1

Introduction and Motivation

Activity recognition is an advanced field that focuses on detecting daily activities performed by individuals using time series records of sensors. Significant progress has been made in technological connectivity over the last ten years, including cloud computing, edge computing, sensors, and the Internet of Things (IoT). Additionally, affordable and simple to integrate into mobile and immobile devices, sensors support current ADL analysis research. The swift rise of affordable smartphones, which are equipped with Inertial Measurement Units (IMU) like accelerometers and gyroscopes, has spurred the creation of various applications in areas such as healthcare, elder care, biometrics, sports analytics, personal fitness tracking, and security and surveillance [1].

The gyroscope and accelerometer sensors can capture detailed motion data. Accelerometers detect changes in velocity along three axes (X, Y, and Z). This allows the device to measure linear acceleration, such as the force of gravity or movement in a specific direction. Gyroscopes measure the rate of rotation around three axes (pitch, roll, and yaw). This enables the detection of angular velocity and rotational changes [2]. Combined with accelerometer, gyroscope capture more accurate and precise motion sensing. The captured data is then fed into a specialized network known as the Smartphone Location Recognition (SLR) Network. This network processes the sensor data to recognize and interpret the smartphone location or the users daily activities i.e. walking, jumping, sitting, and hitting etc.

ADL recognition encompasses two types of activities: simple and complex [3], [4]. Simple activities include sitting, walking, standing, running, etc. Difficult activities include specific activities and specific transitions, such as standing up, getting into a car, sitting still (for 10

seconds), and then walking out. There is little research on these two tasks. In machine learning, correctly identifying ADLs using field data is a multi-time, discrete challenge that falls within the scope of learning. Previous studies have focused on traditional methods such as support vector machine (SVM), random forests, and XGBoost. Traditional algorithms have a number of issues, not the least of which being their time-consuming manual implementation and need for complex architecture. Secondly, mostly studied had considered only one location data for processing e.g. waist or wrist. This work clipped to develop sophisticated DL models for accurately classifying both daily routine activities using time-series data compiled from built-in sensors in smartphones placed at multi-locations on the body (bag, belly, hand, and thigh). This work is conducted in collaboration with the Korea Bootcamp Co Ltd, where we have collected custom data from Korean elderly performing different ADLs. We have proposed a novel approach to convert signals into images using STFT technique and tackle the class imbalancing problem by using top-notch method SMOTE with detailed data augmentation. The proposed model achieves high accuracy in recognizing a diverse set of activities, thereby made contributions in the advancement of activity recognition technology and its applications in improving the well-being of elderly people.

1.1 Goals and Objectives

The primary goal of this study is to build an effective deep learning model for recognizing multi-locations smartphone-based ADLs of elderly people. The specific objectives are:

- Develop and implement state-of-the-art deep learning models, to accurately identify Activities of Daily Life (ADLs) from smartphone sensor data collected from multiple locations using a custom dataset.
- Create a comprehensive benchmark for smartphone-based activity recognition, evaluating model performance across diverse activities, sensor locations, and demographic groups, using standard metrics such as accuracy, precision, recall, F1-score, and computational efficiency.
- Explore and optimize deep learning models for computational efficiency and memory

usage, utilizing techniques such as model compression and efficient architecture design to enable deployment on resource-constrained devices like smartphones without compromising performance.

1.2 Motivation

Smartphone-based activity recognition offers a non-intrusive, cost-effective solution that makes use of ubiquitous devices most people already own. Several factors drive this research:

- The global increase in the elderly population of South Korea necessitates innovative solutions to support their independent living and health monitoring.
- Accurate recognition of ADLs can help in timely detection of abnormal activities or health issues, allowing for prompt interventions.
- Recent progress in deep learning and sensor technology provides an opportunity to develop sophisticated models capable of handling the variability and complexity of real-world sensor data.
- Despite advancements in activity recognition, challenges such as signal noise, class imbalance, and the need for computationally efficient models persist. This study helps to address these gaps and contribute to the field with robust, high-performing solutions.

1.3 Problem Statement

The core problem addressed in this research is the accurate and efficient ADLs using time series custom data, especially for elderly people of South Korea. Our research uses a unique custom dataset that distinguishes itself from previous studies, as no prior deep learning work has been applied to it. While most existing research focuses on activity recognition using data from a single smartphone location, such as the waist or wrist, our dataset is more comprehensive, encompassing multiple smartphone locations including the bag, belly, hand, and

thigh. This multi-location approach presents additional complexities, particularly the challenges of dealing with noise in raw sensor signals and the issues arising from imbalanced datasets. These factors can significantly impact the accuracy of activity recognition models. Therefore, to address these challenges and harness the full potential of our dataset, there is a pressing need to develop and optimize deep learning models that are specifically tailored to manage the intricacies of multi-location data and improve overall model performance.

1.4 Thesis Structure

The general layout of the thesis report is as follows:

Chapter 2: This chapter offers a thorough analysis of the existing research on ADLs or HAR. It examines a number of studies, including their targeted problem, proposed solution, and results. The fundamental objective of literature review is to provide an overview and assessment of the currently suggested approaches related to HAR.

Chapter 3: Discusses the proposed framework for reliable classification of daily living activities.

Chapter 4: Explain the experimentation and results obtained and their discussion

Chapter 5: concludes the study by making some suggestions for possible future research that is not included in this study but may be investigated in the future.

Chapter 2

Literature Review

This section is a detailed but a crucial effort to review the existing methodologies and pinpoint their limitations. The following is the literature which focused on different traditional machine learning & deep learning techniques, their challenges in HAR problem. According to the methodology used by the authors, the works in the literature can be classified into two types:

1. Traditional Machine Learning Methods

Decision Tree, Random Forest, K-Nearest Neighbors, Logistic regression, Support Vector Machine, Ensemble etc.

2. Deep Learning based Methods

Custom Convolutional Neural Networks, Plain Residual Neural Networks, transfer learning models like MobileNet, Inception etc.

2.1 Traditional Methods

Accurately recognizing activities of daily living (ADLs) in indoor environments is essential for smart homes, healthcare monitoring, and assistive technologies. It is highlighted how crucial feature selection and high-quality data are to enhancing model performance [5]. The dataset used is the CASAS Kyoto sensor dataset, collected from 20 volunteers performing five activities. Classification models tested include: Classification Via Regression, Random

SubSpace, Bagging, RF, J48, REP Tree, and IBK (Instance-Based Learning). These techniques were evaluated based on performance metrics like false positive rate (FPR), accuracy, recall, and ROC area.

In [6] the new SNU-Ag-Fall&ADL used for classification of realistic fall scenarios in controlled environments. An inertial sensor on subjects' waists measured two acceleration signals and one angular velocity signal during simulated falls and ADLs. Three machine learning classifiers were used: kNN, SVM, and ANN. Their performance was evaluated with metrics like ROC AUC Score, Accuracy, and MCC. The ANN achieved the highest classification rate, while the SVM was reliable even with unbalanced data, indicating its suitability for this task.

The complexity of human activities, including concurrent activities, complicates accurate recognition. The study [7] propose an EDA method for HAR. Motion signal data is collected from smartphone sensors, preprocessed to reduce noises. A 0.3Hz low-pass filter isolates the gravity component. Data is divided into frames using a sliding window approach (1s or 2.56s) with overlapping frames. Features extracted from these frames include time domain (mean, standard deviation) and frequency domain (information entropy, spectral energy). Stochastic Neighbor Embedding (SNE) is used for dimensionality reduction to visualize patterns. The feature vectors are classified using GridSearchCV and LinearSVC algorithms, achieving a high classification accuracy of 96.56%.

With the growing elderly population, effective monitoring systems for independent living are needed. To improve care and safety, the study [8] propose a novel ensemble classification algorithm integrating ML & DL techniques, utilizing data from ubiquitous smartphones. The study employs various classifiers (RF, KNN, Logistic Regression, MLP, SVM, QDA, Decision Trees, CNN, and LSTM) and three ensemble methods: Stacking, Blending, Bagging. Features like skew, minimum, maximum, and average percentile are used to train classifiers. Data is collected from smartphone sensors (e.g., accelerometer, gyroscope) during various activities. The ensemble algorithm achieved a 98% classification accuracy, outperforming traditional classifiers and effectively detecting multiple postures.

The study [9] proposes a ML-based fall detection application that discriminates falls from random Activities of Daily Living (ADLs), ensuring high sensitivity to actual falls while minimizing false positives. The system is rotation-invariant, functioning effectively regardless of the wearable device's orientation (wrist or neck). They collected a large dataset over

400-days representing daily-life activities of older adults, including ambiguous ADLs and simulated falls. A preprocessing system extracts ambiguous ADLs from the recorded data. The study uses nine machine learning classifiers, including gradient boosting, random forest, and AdaBoost, implemented with the scikit-learn Python library. Classifiers are configured with default settings unless specified otherwise. Performance is evaluated based on sensitivity, specificity, accuracy, and the average number of false positives per day. The system achieved 100% sensitivity, successfully detecting all actual falls during the evaluation.

The study [10] discusses challenges in human activity recognition for older adults, focusing on the potency of ML techniques in real-world contexts. The reliance on multiple sensors raises hardware costs and privacy concerns. The Aruba CASAS dataset, containing sensor events from a smart home, was used. Sixteen features were extracted from each 30-second time window. The dataset was shuffled, stratified, and split into training (70%) and testing (30%) sets, with under sampling and oversampling techniques applied for class balance. Various machine learning algorithms were evaluated, including SVM with RBF kernel, Naïve Bayes, RF, XGBoost, Logistic Regression, KNN, and MLP. Grid search optimized hyperparameters for KNN, SVM, and Random Forest. Specific RF parameters included 300 trees, maximum depth of 30, two samples are needed in order to split an internal node, whereas one sample is needed in order to form a leaf node. The proposed techniques achieved 97% accuracy with the best model, surpassing other approaches using the same dataset. Under a realistic scenario with only ten motion sensors and an additional "Other" class for raw sensor events, the model achieved 89% accuracy.

The study [11] proposes a robust solution to accurately predict a wide range of activities using a single wrist-worn accelerometer. They introduce a local weighted machine learning approach that combines local weighting with the RF algorithm to improve activity recognition accuracy by utilizing and locally weighting neighboring data points during prediction. The study uses the "PAAL ADL Accelerometry dataset," which includes single wrist-worn data from an accelerometer capturing 24 activities from 52 participants. A two-step approach for preprocessing reduces noise, including a sliding window technique to increase sample size. Time & frequency domain features are extracted to represent complex human motions. The local weighting scheme enhances prediction accuracy by considering nearby data points, achieving high accuracy in activity recognition and demon-

strating effectiveness in handling diverse activities.

The local weighting mechanism mitigates dataset variation effects, making the model more robust, especially with limited samples. However, the authors note that the no. of neighbors (k value) requires parameter tuning, and the LWRF can become computationally complex with more samples and features. The study [12] proposes a combined conductive fabric-based suspender system integrating ML techniques for improved HAR, aiming to enhance wearability, reduce noise, and improve accuracy while addressing power consumption and washing durability. The data from three sensors was normalized and segmented. To extract time & frequency domain features dimensionality reduction techniques were applied. Eight classifiers, utilizing machine learning and deep learning techniques, were trained on a dataset split into training (70%) and testing (30%) subsets. The models were evaluated on unseen test data, effectively recognizing various activity patterns. Imbalanced datasets can hinder classification models, affecting sensitivity and specificity. A novel two-stage strategy for HAR is proposed by the paper [13], which involves coarse-level and fine-level categorization for free-style data gathering that represents real-world activities. A wearable accelerometer was used to record everyday activities through the free-form collection of data. Activities were categorized as "with exercise" or "without exercise" using a decision tree model that included the 5-top features. Correctly classified samples then used to train deep learning models for specific activities within these broad categories. Two deep learning models were proposed and evaluated using accuracy, sensitivity, and specificity metrics. Down sampling addressed data imbalance. The dataset, collected from a single participant over 795.86 hours, recorded activities realistically. The two-stage method showed remarkable performance with accuracies for specific activities: Desk-working: 93.53%, Driving: 93.52%, Walking: 85.70%, Downstairs: 98.11%, and Upstairs: 85.91%.

Accurate prediction of the Barthel Index (BI) is crucial for understanding changes in patients' abilities to perform ADL after therapy. The study [14] uses data from 3419 patients (1575 neurological, 1844 orthopedic) at San Raffaele Hospital in Rome (2015-2018), focusing on individual, social, and clinical records to evaluate BI at admission and discharge. Machine learning models analyzed include RF, MLP, AdaBoost, SVR, DT, and NNR. Data features were transformed for compatibility, and 16 sets of features were selected for impact analysis on BI prediction. A leave-one-out cross-validation method validated model training outcomes. The Tree-structured Parzen Estimator (TPE) algorithm optimized hyperparame-

ters, i.e. the number and maximum depth of estimators for RF, the number of estimators for AdaBoost, and the batch size and number of iterations for MLP, preventing overfitting by halting training when loss increased. RF, MLP, and AdaBoost models had the lowest prediction errors, with MLP noted for its feasibility.

Several authors used different signal processing and classification techniques. Processing the signals: The very-first step is to preprocess the signal by performing operations like reducing background noise, removing irregularly captured and poorly-captured signals etc. The downfall of all these previous methods is that informative and high-level features have to be extracted before feeding it into the model for classification purposes. As you can see, extraction of these types' features requires working with a series of tools for feature selection and reduction is a time-consuming task.

2.2 Deep Learning based Models

Within the data science field, deep learning is a new trend for multi-classification and is a major step ahead of traditional machine learning methods in that it tries to learn important features from the image data on its own. Researches have proven that the implementation of deep learning is time saving and also enhances model performance.

Whether deep learning models are gaining attention but focusing on accurate HAQ model to classify human activities for elderly are still a complex challenge. [15] integrate sensor-based dataset HAR70+ with active learning strategies to train deep Learning models. Recursive Feature Elimination and PCA is used for feature selection and extraction to increase model performance. Different algorithms including LSTM is tested on dataset. The LSTM architecture consisted of 64 units and two dense layers, with 'relu' and 'softmax' activation functions. The model is optimized with the 'Adam' optimizer and 'categorical_crossentropy' loss function for multiclass classification problem, with accuracy monitored during training. For accurate model performance cross-validation techniques and standard evaluation metrics are used. Across three tries of the experimentation, the models show different results of evaluation parameters. By comparing the results, the LSTM model performed the best, with an accuracy of 0.9845.

Many deep learning models for HAR overlook behavioral patterns in different environments.

Using smartwatches, [16] suggests a deep neural network with convolutional layers, residual networks, and a squeeze-and-excite mechanism for HAR in older adults. The method makes use of three datasets that are separated into simple (SHA) and complicated (CHA) activities: WISM-HARB, UT-Smoke, and UT-complicated. Walking and sitting are examples of repetitive motions in SHA, whereas typing and clapping are examples of non-repetitive hand gestures in CHA. The framework uses both nonoverlapping and overlapping temporal frames for data collecting, preprocessing, feature representation, and model training and testing.

A convolutional block, five residual-SE blocks, global average pooling, batch normalization, ReLU activation, and squeeze-and-excitation layers are all included in the ResNet-SE architecture. Categorical cross-entropy and the Adam optimizer are used to train the model, and early halting is implemented in case of validation loss. Based on the WISDM-HARB, UT-Smoke, and UT-Complex datasets, the ResNet-SE network achieved accuracy rates of 94.91%, 98.75%, and 97.73%, respectively.

Ground robot vision faces issues like occlusions and limited view compared to stationary cameras. [17] proposed solution is a real-time 1D LSTM model for multi-person activity recognition in elderly group exercise. Tested on 16 datasets, including a Routine Exercise Dataset (RED) with 14,440 samples, the model uses frames from video, feature extraction, and data cleaning. They split the data into 90% training and 10% testing. Validation was done using stratified K-Fold. The Adam optimizer with a learning rate set to 0.00003 and 32-batch size, and a 'ReduceLROnPlateau' scheduling strategy, were used. The model employs pose estimation, YOLOv3 with Darknet-53, and MOSSE tracker, feeding features into an MLSTM network. The system is designed for deployment on the Pepper robot using ROS. The method achieved accuracies of 99.27% to 99.90% on various daily activity datasets and 99.61% to 98.88% on exercise datasets, outperforming existing methods.

Due to smartphones' flexibility, sensor data acquisition is affected by position and orientation changes, reducing activity recognition accuracy. To address this, the Extended Kalman Filter (EKF) and Accelerometer Linear Gaussian Principal Component Analysis (ALGPCA) preprocessing methods are proposed in [18] for an efficient TCIANet model for human activity recognition. The time series data is divided into windows, split 7:3 for training and testing classifiers. The Adam Optimizer is used with a learning rate set to 0.001 over 50 epochs. The dataset, collected using the Sensor Logger app on Xiaomi MIX4 and Huawei P20 smartphones, recorded at 100Hz with 50% overlap.

Experiments compared traditional machine learning methods and deep learning methods (e.g., DeepConvLSTM, AttnSense, HDL, DLSTM), using cross-validation to evaluate the algorithms' generalization. TCIANet achieved the highest accuracy in activity recognition, with 93.25% using raw data and 98.93% after EKF-ALGPCA preprocessing. TCIANet also demonstrated high accuracy on public datasets, achieving 99.68% on UCL-HAR, 99.01% on WISDM, and 98.58% on MHEALTH.

The 24-hour activity classification behaviors from raw accelerometer data is also challenging due to the variability in human movements. To address this, the AccNet24 framework was developed in [19], utilizing deep learning techniques to classify activities with high accuracy. Data was obtained from the Capture-24 dataset. The raw acceleration data was converted into signal images. Each signal image was resized to 224×224 pixels. Deep features were extracted using the ResNet101 model, using pre-trained ImageNet dataset. The output from the ResNet101 final pooling layer, a feature vector of size 2048, was changed into a 100-dimensional feature space using Reconstruction ICA (RICA) and classified with a Bidirectional Long Short-Term Memory (BiLSTM) network. The dataset was divided into training, validation, and test sets, with AccNet24 achieving impressive accuracies of 96.9% on validation and 98.2% on test sets.

In human activity recognition (HAR) in real-life environments there is also gap between theoretical data in HAR and its practical implication in everyday life [20]. The use of deep learning algorithms, specifically CNN and LSTM models can fill this gap for the classification of activities depends on data collected from smartphone sensors in real-life settings. Implementing deep learning algorithms, particularly DS-CNN and Bi-LSTM, and combining them into hybrid models. The real-life HAR dataset captured from four sensors: accelerometer, gyroscope, magnetometer, and GPS, collected from participants using their personal smartphones. The results obtained show that the hybrid models combining DS-CNN and LSTM achieved a peak accuracy of 94.80% on the dataset used, which is an improvement over previous traditional machine learning techniques that had lower accuracy rates.

Feature engineering is considered crucial in traditional human activity recognition methods [21], [22]. Post-stroke patients often struggle with daily activities due to motor impairments. A novel pipeline integrating Continuous Wavelet Transform (CWT) for feature extraction with Transfer Learning (TL) models, specifically ResNet architectures, and

a fine-tuned SVM classifier. The dataset consisted of EEG signals from five healthy subjects, split into training, validation, and test sets in a 60:20:20 ratio. The SVM classifier was optimized using grid search and five-fold cross-validation, evaluating hyperparameters such as: Gamma: 0.01, 0.1, 1, 10, 100, Kernel Types: Linear, RBF, Polynomial, Sigmoid, Regularization (C): 0.01, 0.1, 1, 10, 100 and Degree: 2, 3. The best model used a polynomial kernel with a quadratic degree and gamma and regularization values of 0.01. The ResNet152 V2-SVM pipeline achieved 100% classification accuracy across all datasets.

The study [23] introduces "Eldo-care," a solution for elderly and disabled individuals with neurological conditions affecting emotional and cognitive processing. By integrating EEG signals and Kinect sensor data, the system monitors patients' psycho-neurological states. EEG data, collected with the BrainTech Traveler using a 10–20 system, is cleaned with a 6th order Chebyshev II filter. An autoencoder extracts features, and a CNN with transfer learning classifies the data, achieving over 95% accuracy.

[24] propose a 1D-CNN model for time series data from smartphone sensors, focusing on accelerometer and gyroscope readings. The raw sensor data was preprocessed through normalization and segmentation. The 1D CNN, designed for time series, learns spatial hierarchies of features through convolutional layers. The models were trained with softmax activation and categorical cross-entropy loss, and evaluated on Recall, Precision, and F1 score. Recognizing activities involves understanding temporal relationships between frames, a challenge not effectively captured by many methods. To address this, [25] proposes a deep learning approach with the use of video data from public sources and applied preprocessing techniques such as filtering, shrinking frames, and normalization. Benchmark datasets UCF50 (50 action categories) and HMDB51 (51 action categories) were employed. Various models, including CNN, LSTM, ConvLSTM, and LRCN, were trained, with a pre-trained CNN used for feature extraction and Fuzzy Logic to extract keyframes. The ConvLSTM model achieved 82.00% accuracy on UCF50 and 68.00% on HMDB51.

The study [26] highlights the need to distinguish between various activities, including hand-oriented and general movements. Using the WISDM-HARB dataset from UCI, which includes data from 51 participants performing 18 activities with accelerometers and gyroscopes at 20Hz. The difficulty of effectively categorizing complex upper limb movements from pre-movement EEG signals is addressed in this work, which pertains to brain-computer interfaces (BCIs). A unique method combining spectral feature extraction with cutting-edge

machine learning techniques is proposed in the article [27]. To eliminate noise, they employed a 50Hz notch filter and a 4th-order zero-phase Butterworth filter (0.3Hz to 70Hz). To isolate EEG data from aberrations like eye blinks and muscle movements, sophisticated methods including Independent Component Analysis (ICA) and Second-Order Blind Identification (SOBI) were used. Spectral characteristics were extracted using STFT. Twelve healthy people's EEG signals were included in the sample. A 64-channel device was used to record EEG data at a sampling rate of 512 Hz. A 3-second baseline and a 5-second pre- movement visualization phase preceded each trial.

The paper addresses the decrease in HAR accuracy due to random changes in smartphone orientation and position. The authors in [28] used the UCL-HAR dataset and a custom dataset with varied orientations and positions during daily activities. They introduced EKF-ALGPCA (Extended Kalman Filter and Adaptive Linear Generalized Principal Component Analysis) to preprocess and align accelerometer data, mitigating orientation and position effects. Trained with the Adam Optimizer at a 0.001 learning rate over 50 epochs on a Py-Torch framework using an RTX3060 GPU, the model achieved a 99.68% accuracy on the UCL-HAR dataset.

[29] proposes a dynamic active learning approach to address these challenges, tested on synthetic datasets generated using Gaussian Mixture Models (GMMs), the USC-HAD health-care dataset, and the UCI-HAR dataset. The approach focuses on activity discovery using novelty checks and affinity propagation-based clustering to identify new activities. It selects the most informative samples for manual annotation based on uncertainty, diversity, and representativeness.

A windowing technique extracts activity indicators from raw sensor data, with a 15-dimensional feature set including both frequency and time domain features, like zero-crossings and mean values, showing superior activity recognition accuracy.

The study [30] proposes a structured prediction strategy using probabilistic graphical models (PGMs) within a sequence-labeling framework to improve activity recognition. By modeling relationships between poses and actions, it enhances human activity understanding. Tested on the Florence 3-D, CAD-60, and UT-Kinect datasets, the approach efficiently handles variability in movements and performs well with smaller training data compared to LSTM models.

To address temporal dynamics of human actions, [31] proposes a CNN architecture that pro-

cesses short video clips (approximately 2 seconds) to classify human activities. This architecture combines a 3D convolutional layer, which captures short-term spatial and temporal features from consecutive frames, with an LSTM layer, which learns long-term dependencies. The architecture was trained on the KARD, CAD-60, and MSR Daily Activity datasets. Performance was evaluated using confusion matrices, precision, and recall metrics. The design minimizes network size and training time compared to other state-of-the-art methods, making it suitable for hardware-constrained environments. The KARD dataset achieved full recognition of all 18 classes, the CAD-60 dataset correctly classified all 12 classes, and the MSR Daily Activity dataset yielded a classification accuracy of 95.6%.

The study [32] introduces an unsupervised domain transfer method to adapt a pre-trained activity classifier for use at a new target body location. Using the PAMAP2 and MHEALTH datasets, they split the data into 30:20:50. Three experiments were conducted 1) Supervised Source Model to evaluate the source-domain model's performance 2) Unsupervised Target Model to assess the model adapted to the target domain without labeled data and 3) Comparison of Models to adapt the model's performance against the original source model. Performance is measured using accuracy, precision, recall, and F1 scores, calculated for each activity class as a one-vs-rest problem.

The study [33] introduces a novel method for classifying human activities based on movement features derived from changes in joint distances using the Euclidean distance formula and employing CNN. Two public datasets are utilized in the study 1) Florence 3D Action Dataset contains nine activities with 215 video segments captured using a Kinect camera, totaling 4016 frames. 2) UTKinect-Action3D Dataset. Various window sizes are tested, with a size of 16 yielding the best performance. These features are then formatted for the CNN model. The data is split using an 80/20 validation approach. The CNN processes the input with three-dimensional filters, and the model is trained through supervised learning. The model achieved 94.08% accuracy on the Florence 3D dataset and 93.18% accuracy with a loss value of 0.1964 on the UTKinect-Action3D dataset.

The study [34] introduces the 19NonSens dataset, collected from 12 subjects using e-Shoes and a Samsung Gear G2 smartwatch, performing 19 activities. Data preprocessing included filtering, segmentation, and normalization. SensCapsNet, a Capsule Network for wearable sensor data, achieved 80% accuracy.

Table 2.1. Summary of Recent Work on Human Activity Recognition

Ref.	Year	Dataset	Activities	Models	Results	Strengths	Limitations
[5]	2021	CASAS Kyoto	Various daily activities	Multiple classifiers	High accuracy for regression models	Robust experimental setup	Potential overfitting issues
[6]	2022	SisFall open access dataset	Falls and daily activities	SVM, KNN, ANN	Excellent performance with ANN	Validation across diverse subjects	Lack of analysis on misclassification cases
[7]	2021	Postural Transitions from UCI	Basic postural transitions	SVM, Grid-SearchCV	Strong ROC-AUC results	Well-balanced data handling	Potential data bias not fully addressed
[8]	2021	Posture Detection dataset	Basic postures	QDA, SVM, CNN, Decision Tree	High accuracy reported	Use of hybrid feature selection techniques	Limited details on performance metrics
[9]	2021	FallAIID and RealAct	Fall detection	Multiple classifiers	Moderate to high accuracy	Device orientation invariant methodology	Lack of clarity on achieving perfect accuracy
[10]	2023	Aruba CASAS	Daily home activities	Various classifiers	Good performance overall	Realistic experimental scenarios	Short window extraction may overlook activity nuances
[11]	2022	PAAL ADL Accelerometry dataset	Activities of daily living	Multiple classifiers	Satisfactory accuracy	Comprehensive preprocessing and evaluation	Manual feature extraction limits model flexibility
[12]	2023	Body-worn suspender data	Mixed daily activities	Various classifiers including LSTM	High sensitivity reported	Enhanced feature extraction techniques	Focus on a specific wearable limits generalizability
[13]	2023	Wearable accelerometer sensor	Office and movement-related activities	Decision Tree, ModelNEX	High performance across models	Effective imbalance handling	Decision tree susceptible to overfitting
[14]	2024	San Raffaele Hospital in Rome	Healthcare-related activities	Various models including stacking	Strong accuracy for HAR	Effective hyperparameter optimization	Lacks real-time prediction capability

Chapter 3

Methodology

The proposed method for classifying activities of daily living consists of five phases: dataset collection, data segmentation, data preprocessing, classification models, and performance evaluation. The flow of these main processes are shown in fig 3.1.

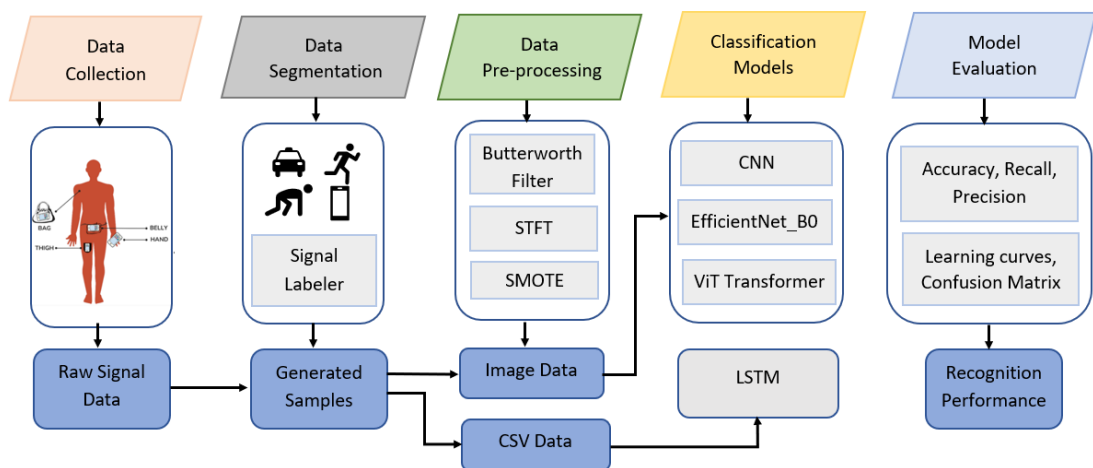


Figure 3.1: Proposed Framework

3.1 Dataset Collection

The dataset consists of total 19 South Korean subjects: 12 females and 7 males, with an average age of 60 years. Adults' subjects (SA) age is between 18 and 60 years and elderly

subjects (SE) are older than 60 years. Subjects have an average height of 158.24 cm and an average weight of 58.26 kg. Data was collected at 100 Hz sampling rate from LM-G900N smartphone model placed at four different body locations: in-hand, in a thigh pocket, on the belly (inside a bag), and carrying a purse on hand as shown in Figure 3.2.

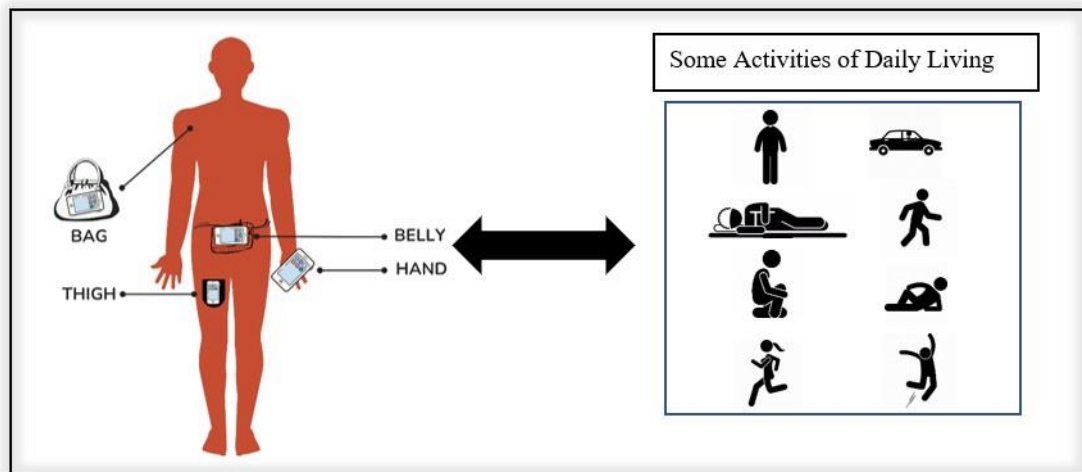


Figure 3.2: Fig: (a) Smartphone placement locations (b) Classified ADLs

The accelerometer sensor is designed to operate within a range of $\pm 4g$, which enables it to detect and measure acceleration forces that are up to four times greater than the force of gravity in both positive and negative directions. This broad range of detection is crucial for accurately capturing various intensities of movement, from subtle shifts to more vigorous motions. Meanwhile, the gyroscope sensor, with its measurement range of ± 500 degrees per second, is adept at capturing angular velocity, detecting rotational movements in both clockwise and counterclockwise directions with precision. This capability is essential for tracking rapid rotational dynamics, such as those involved in sudden changes in orientation or quick spins. The combination of these signal readings ensures accurate and comprehensive data collection, capturing different types of human activities. The dataset includes a total of 15 different activities, each carefully chosen to represent a diverse array of human motions, ranging from simple actions like walking and standing to more complex movements such as running and jumping. The detailed description of each activity mentioned in table 3.1.

Table 3.1. Summary of Activities of Daily Livings

Signal ID	Activities	No. of Trials	Total Duration
D01	Walking - slowly/quickly	2	100s
D02	Jogging - Slowly/Quickly	2	100s
D03	Jumping on the same place	2	5s
D04	Walk upstairs/downstairs - Slowly/Quickly	2	25s
D05	Standing, bending/without bending knees and getting up	4	12s
D06	Sit in a half height chair, wait a moment (5secs), and up slowly/quickly	4	12s
D07	Sitting a moment, trying to get up, and collapse into a chair	2	12s
D08	Sit in a Sofa, wait a moment (5secs), and up slowly/quickly	4	12s
D09	Sit on the ground, wait a moment (5secs), and up slowly/quickly	4	12s
D10	Sitting a moment on the floor, lying slowly/quickly, wait a moment, and sit again	4	12s
D11	Being on one's back, change to lateral position, wait a moment, and change to one's back	2	12s
D12	Walking (4-5 steps), make a stop, bend down to pick up something and continue walking (4-5 steps)	2	12s
D13	SP/Person Touched or hit accidentally pole/another person	2	12s
D14	Gently jump without falling (1 jump only) (trying to reach a high object)	2	12s
D15	Standing, get into a car, remain seated (10secs), and get out of the car	2	15s

3.2 Data Segmentation

Data segmentation is a technique for dividing a dataset into smaller segments/chunks based on specific criteria or features. This technique is commonly used to improve the analysis, processing, and understanding of complex datasets. In this study, we have segmented our time-series data using MATLAB Signal Labeler tool. This tool facilitates the manual labeling of the sensor signals captured from four different locations on the body: Bag, Belly, Hand, and Thigh. The segmentation process is shown in figure 3.3 .

Each signal is color-coded for clarity: blue for Bag, red for Belly, yellow for Hand, and purple for Thigh. The signals are plotted against the sample index, illustrating the variations in accelerometer readings (Accel_x, Accel_y, Accel_z) over time. The segments represent different activities (labeled from D01 to D05, including the specific activity instances) that were performed during data collection. The segmentation ensures that each activity is distinctly identified within the continuous time-series data, allowing for precise activity classification

during subsequent analysis. Next, we automated the extraction of these segments by loading the labeling information and identified all relevant signals files. For each file, read the data into a table and iterated through each labeled region of interest (ROI). This process ensured that each activity’s data was systematically separated, facilitating subsequent analysis and classification tasks.

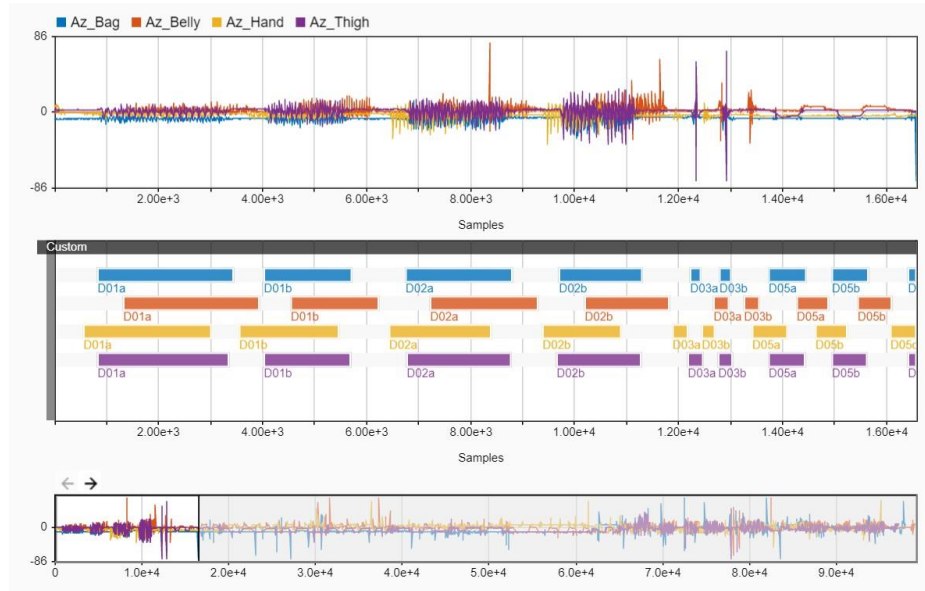


Figure 3.3: Time-series data segmentation using MATLAB Signal Labeller tool

3.3 Data Pre-processing

Preprocessing is an important step to clean and convert raw data into an appropriate format for model training. In this study, we used a Butterworth low-pass filter to preprocess time-series data from human activities. A Butterworth filter is a kind of electronic filter which permits low-frequency signals to pass while decreasing the amplitude of high-frequency signals. We designed the filter with 6-Hz cutoff frequency and 100-Hz sampling frequency, using an 8th-order filter. Each sensor 6-channel was individually filtered to reduce high-frequency noise and artifacts, ensuring the data retained the relevant low-frequency components associated with different human activities. Along with noise removal we also normalize the filtered data using MinMaxScaler, which normalizes the datapoints to a particular range of 0 to 1, improving the performance of subsequent analysis and learning models. This approach

enhances the signal-to-noise ratio and provides a clearer representation by removing noise from edges and within patterns, facilitating more accurate activity recognition as shown in figure 3.4.

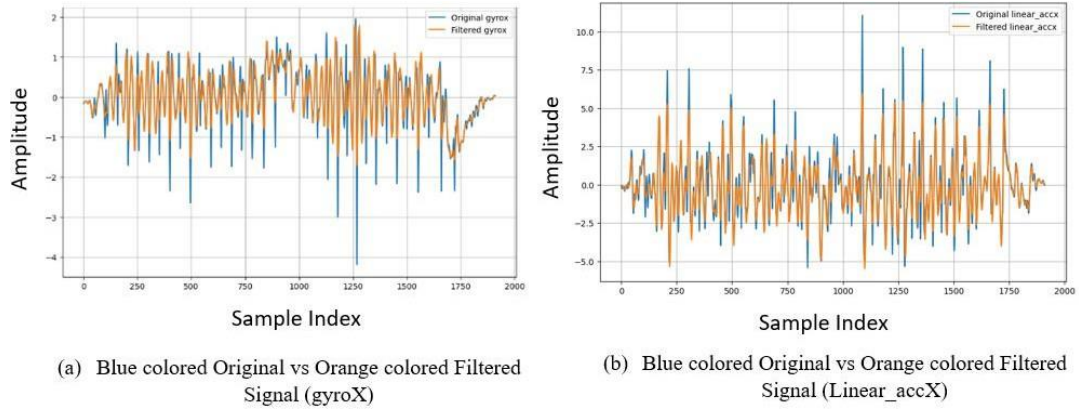


Figure 3.4: Butterworth Filter application on signal data

3.3.1 Short-Time Fourier Transform (STFT)

STFT is a technique widely used in signal processing that provides a qualitative representation of a signal based on frequency shift. Unlike the Fourier transform, which provides fixed-frequency analysis, STFT divides the signal into short periods and uses a Fourier transform on each section to create a series of spectrograms. This transformation from a time-domain signal to a frequency-domain spectrogram provides detailed information about how the frequency components evolve over time. This method involves the selection of certain parameters, such as the size of the imaging area and overlap, to ensure accuracy and consistency. In order to apply STFT on our data we have set some parameter according to our data. The sampling frequency (f_s) has 100 Hz value, which is appropriate for the frequency range of our used SP. We used a Hamming window function to minimize spectral leakage, with the window size (n_{perseg}) dynamically adjusted to be the smaller value between 256 samples and the length of the signal, ensuring the STFT was adaptable to varying signal lengths. The overlap between consecutive windows, denoted as $n_{overlap}$, was set to half of the window size. This approach ensures a compromise between time and frequency resolution. By over-lapping windows, we reduce the risk of losing important transient information and improve

frequency resolution. This balance helps in capturing both temporal and spectral characteristics effectively.

Additionally, we specified a number of FFT points as 256 to control the resolution of the frequency bins. We detrended each segment using a constant detrend method to remove any linear trend from the signals, making the frequency components more distinguishable. The STFT computation is set to return a one-sided spectrum, appropriate for real-valued signals. Boundary effects are managed by zero-padding the signals, ensuring the entire signal is analyzed without edge artifacts. The axis parameter is set to -1 to indicate that the STFT should be computed along the last axis of our input data. Using STFT technique, we have generated a total of 24,948 images for all locations. The generated spectrograms for different activities have been shown in Figure 3.5.

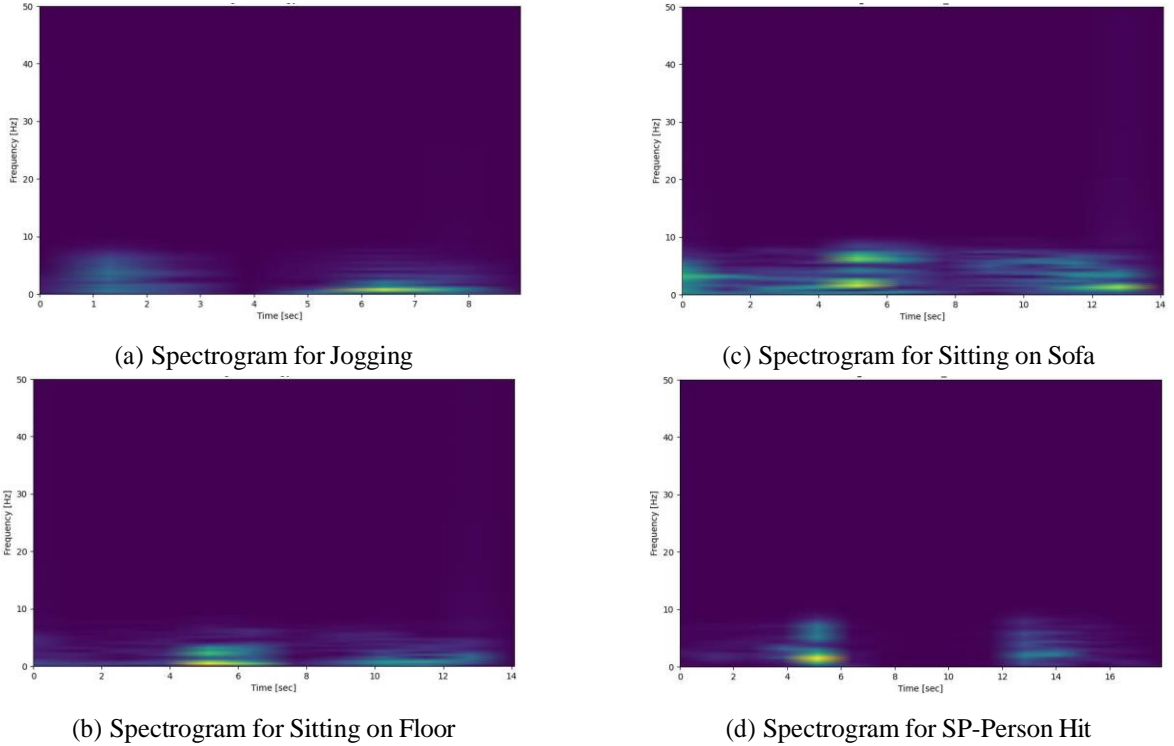


Figure 3.5: Generated Spectrograms using STFT for Different Activities

3.3.2 Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is a method used to resolve class inequalities in data. Through interaction, it can construct a synthetic model of the minority class through the current minority sample, thus constructing the entire model of the minority class and improving the performance of the model. The issue of not handling imbalanced datasets is that mostly the learning methods ignore the minority class, leading to poor performance on it, even though performance on the minority class is often the most critical aspect.

Our dataset contains activities with varying numbers of trials: some activities have 4 trials (e.g., signal ID D05 for Standing, Bending/Without Bending Knees, and Getting Up), while others have only 2 trials (e.g., signal ID D01 for Walking Slowly and Walking Quickly). The classes which have more trails contain more samples as compare to those classes with less trails. This results in class imbalance. To address this class imbalance issue, we employ the SMOTE approach on our dataset.

Before application of SMOTE, we divided our images data into training, validation, and testing sets with ratio the 70:20:10. SMOTE is then applied on training set to transform images into feature vectors. SMOTE generated synthetic samples by interpolating between existing examples in feature space, which requires the data to be in numerical form. For SMOTE to function, we convert our textual image path into numerical format. This is done using a pretrained VGG16 model. VGG16 is a popular deep learning model trained on ImageNet, and we use it for feature extraction from images by removing the top classification layer and using the output from the average pooling layer.

For the efficiency of SMOTE application images were resize into 224x224 pixels, normalized the pixel values, and then use the VGG16 model to extract features. It extracts low-level features from initial convolutional layers. These features include simple structures such as edges, gradients, contours, simple texture patterns, temporal evolution and frequency characteristics of the signals in the spectrogram. Based on these extracted features it generates synthetic samples by interpolating between feature vectors of existing samples from the minority class. It chooses a sample image from the minority class, on the basis of its k-nearest neighbors, and originate new samples with the connected line segments to the original sample to its identified neighbors, corresponding to the labels. Unique filenames are generated for the synthetic images using UUIDs to avoid conflicts.

Data distribution changes before and after the application of SMOTE can be seen in Figure 3.6

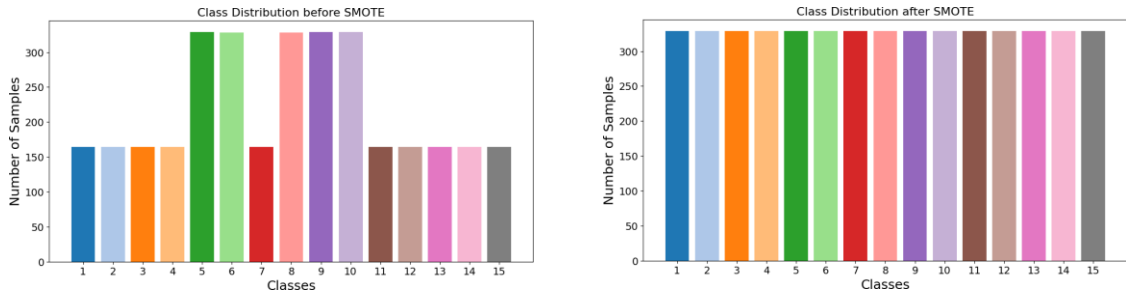


Figure 3.6: Data Balancing using SMOTE Technique

3.4 Classification Models

3.4.1 Long Short-Term Memory (LSTM)

LSTM is a special kind of neural network built to handle sequences of data. They use deep mechanisms to learn important features over long periods and avoid issues that can cause learning problems in traditional models. Because of this, LSTMs are good choice for tasks for predicting future trends, classify object, and recognizing speech.

In this study, we used LSTM deep learning model to classify activities from multi-location signals. To design LSTM model, we apply it directly on preprocessed MinMax scaled CSV files. In order to train the model, we converted our time-series data into sequences. This was the essential step because LSTM models are designed to handle sequential data, and converting the data into the appropriate format enables the model to learn from the temporal dependencies within the data. The sequence length is set to 50. For each data frame, a sliding window approach is used to create sequences: the function slides a window of sequence length over the time-series data, extracting sequences of consecutive time steps with the corresponding label.

Next, we convert these sequenced data into 80:20 split for training LSTM. In order to handle the accurate labels, we use One-hot encoding to transforms the labels into a binary matrix representation. In our dataset, we have 15 activities (labels from 1, 2, 3, 4, 5,6, . . . , 15), each label is converted into a 15-element binary vector where the position corresponding to

the class index is marked with a 1, and all other positions are marked with 0. This step is beneficial because the Neural networks, particularly those used for classification tasks, often perform better with one-hot encoded labels as it helps in calculating the loss and optimizing the model more effectively. For building an LSTM neural network model we used TensorFlow's Keras API. The first layer consists of 50 units with backward processing, which allows the LSTM process to obtain all output sequences for each input. The input shape parameters are defined as (50, 6) for 50-time steps and 6 features. To avoid overfitting, we included a dropout layer with a 0.2 (20%) drop rate. Since it is the last LSTM layer in the stack, the second LSTM layer, which has 50 units as well, does not return sequences. To further reduce overfitting, another dropout layer with a 0.2 dropout rate is added. The last layer of the model is a dense layer that uses the softmax activation function and has 15 classes, the result is converted into probability distributions for several categories, enabling the model to generate predictions using 32,365 parameters. The model is trained using the Adam optimizer with a learning rate of 0.001 and the categorical cross-entropy loss function. . We tried batch sizes of 16, 32, and 64 during training 30 times, and finally found that batch size 64 performed best. The suggested LSTM model's overall architecture is shown in figure 3.7.

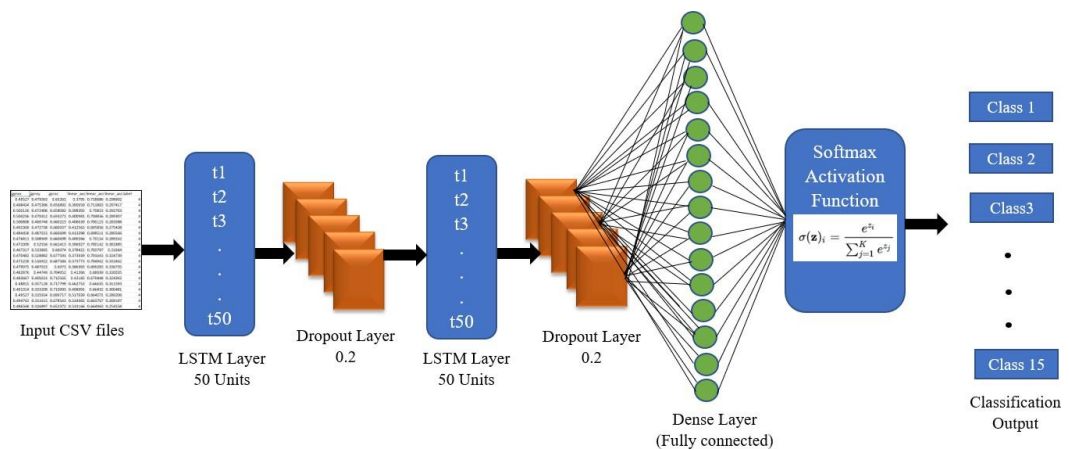


Figure 3.7: Proposed architecture of LSTM model

3.4.2 Convolutional Neural Network (CNN)

CNN is a network of deep neural networks created specially to handle grid-like input, like images. They are suitable for a variety of computational tasks, such as segmentation, object identification, and image classification, since they can learn hierarchical representations quickly and effectively. In this work, we present a CNN architecture for the classification of STFT-generated spectrogram images that define various ADLs. The proposed model is capable of learning and recognizing patterns in the displayed image, facilitating the proper classification of ADLs.

Each layer in our suggested model has a specific function in the extraction and classification of features. Images with dimensions of (256, 256, 3)—with image height and width set to 256, 256, and 3 channels—are accepted by the input layer. With 32 filters and a kernel size of (3, 3), the first convolutional layer uses the Rectified Linear Unit (ReLU) activation function to produce 32 feature maps from the input image, with an emphasis on low-level features. Max pooling layer (MaxPooling2D Layer 1) replaces this layer by using a pool size of (2, 2), which reduces the spatial dimensions (height and width) by a factor of 2 while maintaining the most noticeable features.

Conv2D Layer 2, the second convolutional layer, uses 64 filters with a kernel size of (3, 3) in addition to the activation function of ReLU to extract 64 feature maps from the output of the prior layer, allowing for the capture of higher-level features. By summing the retrieved features, an extra max pooling layer (MaxPooling2D Layer 2) with a pool size of (2, 2) further decreases the spatial dimensions.

Even higher-level characteristics are captured by the third convolutional layer (Conv2D Layer 3), which consists of 128 filters with a kernel size of (3, 3) and extracts 128 feature maps using the ReLU activation function. The third max pooling layer (MaxPooling2D Layer 3) reduces the spatial dimensions and gets the feature maps ready for the fully connected layers by having a pool size of (2, 2).

The convolutional and pooling layers' 2D feature maps are modified into a 1D feature vector by the flatten layer. A fully connected layer (Dense Layer 1) with 128 units and a ReLU activation function comes next. The purpose of this layer is to facilitate the final classification by learning how to merge the characteristics retrieved into a compact representation. With a softmax activation function that provides the probabilities for each class, the output layer's

fifteen units (matching to the number of classes in our dataset) allow for multi-classification. The dataset is split into three major parts: a training set (70% of the data: 4935 images), a validation set (20% of the data: 821 images), and a testing set (10% of the data: 456 images). Proposed model has total 14,840,911 trainable parameters. A 32-batch size on 20-epochs are used for training model. Training data have batches which are input into the model during each epoch, and the weights are adjusted depending on the gradients of the loss function Sparse Categorical Crossentropy relative to the model parameters.

The Adam optimizer is used to assemble the model, and it adjusts the learning rate by 0.001 during training to ensure effective convergence. The EarlyStopping callback, which ignores the validation loss and stops training if the loss has not improved for five consecutive epochs, is used to prevent overfitting. It also returns the optimal weights that were seen throughout training.

Table 3.2. Detailed Summary of CNN Model

Layer (type)	Output Shape	Param#
conv2d_12 (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d_12 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_13 (Conv2D)	(None, 125, 125, 64)	18,496
max_pooling2d_13 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_14 (Conv2D)	(None, 60, 60, 128)	73,856
max_pooling2d_14 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten_4 (Flatten)	(None, 115200)	0
dense_8 (Dense)	(None, 128)	14,745,728
dense_9 (Dense)	(None, 15)	1,935

3.4.3 EfficientNet_B0

EfficientNet is inherited from convolutional neural networks designed for image classification tasks. Introduced by Google Research, EfficientNet models works on a compound scaling method that uniformly scales the depth, width, and resolution of the network to achieve better performance with fewer parameters.

The three scaling coefficients are:

$$\begin{aligned}
\text{Depth: } d &= \alpha\varphi \\
\text{Width: } w &= \beta\varphi \\
\text{Resolution: } r &= \gamma\varphi \\
\text{Subject to: } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\
\alpha, \beta, \gamma &\geq 1
\end{aligned}$$

$$\text{Scale factor} = \text{Depth Scaling} \times \text{Width Scaling} \times \text{Resolution Scaling}$$

EfficientNet_B0 is the baseline model of the EfficientNet family, and it serves as the starting point for the larger EfficientNet models. We have proposed the EfficientNet_B0 architecture with different layers. First Stem Layer is a single convolutional layer set a kernel size of 3x3 and a stride of 2. It has MBConv Blocks stands for Mobile Inverted Residual Bottleneck Convolution. Each MBConv block consists of Depth wise Separable Convolutions which reduces computational complexity by separating the convolution operation into depth wise and pointwise convolutions. We also set model with Squeeze-and-Excitation Layers. It's a lightweight mechanism to adapt recalibrating channel-wise feature responses. We have also created residual Connections that allow gradients to flow through the network more easily. There are two final layers: Global Average Pooling (GAP) which reduces the spatial dimensions of the feature map to a single value per channel and fully Connected dense Layer, which have 15 output neurons equal as per classes in our dataset.

To propagate the data via the model, the images are preprocessed with a standard transform pipeline: resizing to 224x224 pixels, converting to tensor format, and normalizing using the mean and standard deviation of the pre-trained ImageNet. ImageNet is widely known for its annual competition, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which has significantly influenced the development of deep learning algorithms. The dataset split into 70:20:10 i.e. training, validation, and test sets. In order to load the data, we use PyTorch ImageFolder utility to automatically organizes images based on their directory structure. Data loaders is created for each dataset split with a batch size of 32, enabling efficient batch processing during training and evaluation.

In order to get the best training model, we fine-tuned the EfficientNet_B0 model, which was pre-trained on the ImageNet dataset, by replacing its final classifier layer with a new fully connected layer fitted to the number of classes in our dataset i.e. 15. For the training process,

we use the cross-entropy loss function with the Adam optimizer, and tried different learning rates i.e. 0.1, 0.001, and 0.0001. Additionally, a learning rate scheduler is used to reduce the learning rate by a factor of 0.1 every seven epochs, helping in fine-tuning the model's weights during later stages of training. We achieved the best results with 0.001 learning rate. The training procedure spanned multiple epochs. During the training phase, the model weights were updated based on the loss calculated from the training data. To reduce the overfitting, early stopping mechanism is used. The validation phase evaluated the model's performance on unseen data, preventing overfitting and ensuring generalization. Key metrics such as training and validation losses, along with accuracy, were recorded and analyzed across epochs. The EfficientNet_B0 model demonstrated its capability by effectively learning and classifying the spectrogram images, with detailed monitoring of loss and accuracy metrics guiding the training process.

3.4.4 ViT (Vision) Transformer

The Vision Transformer (ViT) is another model for deep learning, which work for image classification that implements the transformer architecture, designed for natural language processing (NLP), to process and classify images [35]. In our proposed model, the input image is first resized to 224x224 pixels and divided into fixed-size patches (16x16 pixels), each of which is flattened into a one-dimensional vector. These vectors are then linearly projected into a higher-dimensional space, resulting in a sequence of embedded patch vectors akin to NLP tokens. To preserve spatial information lost during patch flattening, positional embeddings are added to each patch, making sure the model maintains the spatial relationships within patches. This sequence of positionally-embedded patches is fed into a standard transformer encoder composed of multiple layers of multi-head self-attention and feed-forward neural networks. The self-attention process enables the model to weigh the importance of each patch relative to others, capturing global dependencies and context across the entire image. A special classification token ([CLS]) is vision to the sequence of each patch embeddings, and the output corresponding to this token is used as an aggregate representation of the input image. This representation is passed through a feed-forward neural network (classification head) to forecast the class label.

We used the pre-trained ViT model with vit-base-patch16-224 from the Hugging Face Trans-

formers library, which is fine-tuned for our specific classification task with 15 output classes. Images undergo preprocessing steps including resizing, normalization, and tensor conversion. The datasets for training, validation, and testing are structured, and data loaders facilitate 32-batch processing. The model is trained using an Adam optimizer with a 0.0001 learning rate and cross-entropy loss, fitted for multi-class classification tasks. During training, the model parameters are updated through backpropagation, and performance metrics such as loss and accuracy are tracked for both training and validation sets. This way the transformer architecture’s ability to model long-range dependencies and contextual information enables ViT to effectively learn and represent complex features in images. The overall flow of ViT Transformer is shown in fig 3.8

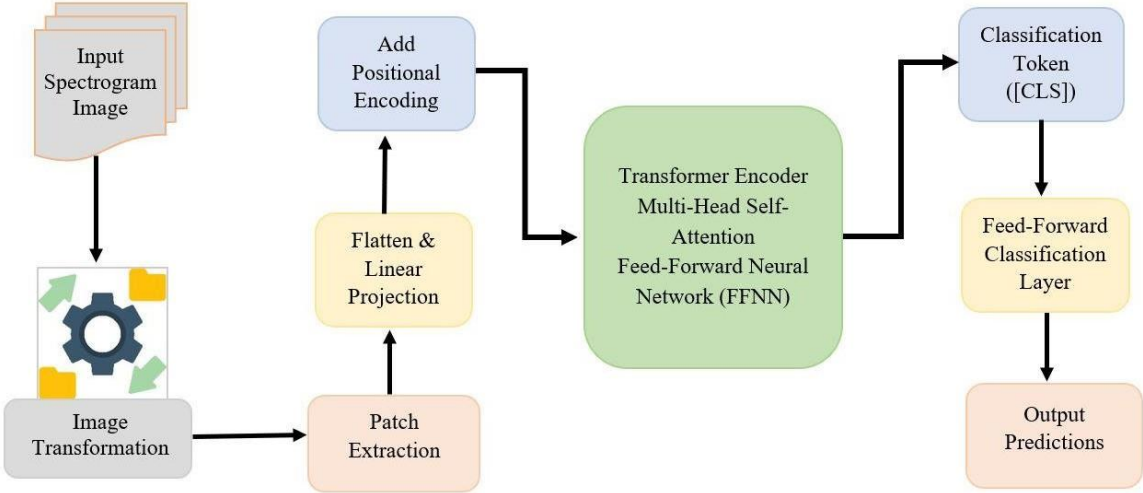


Figure 3.8: Flow Diagram of ViT Transformer

3.5 Model Evaluation

To evaluate the trained model, four performance metrics are employed: accuracy, precision, recall and F1-score. Precision measures the proportion of positive predictions that are actually correct, while recall assesses the proportion of actual positives that were correctly identified. The F1-score, which is the harmonic mean of precision and recall, was used to

combine these metrics. The formulas for these metrics are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP (True Positive) is the count of correctly predicted positive activities.
- TN (True Negative) is the count of correctly predicted negative activities.
- FP (False Positive) counts the number of negative examples incorrectly predicted as positive.
- FN (False Negative) counts the number of positive examples incorrectly predicted as negative.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For model evaluation, we also analyze the results using learning curves, confusion matrices, and precision-recall curves. These plots give additional information into the model's performance:

- Learning Curves: These plots show how the performance of model (such as accuracy or loss) changes with training and validation over epochs. They help identify issues like overfitting or underfitting.
- Confusion Matrix: It displays the number of true positives, true negatives, false positives, and false negatives, providing a detailed performance of model across different classes.
- Precision-Recall Curves: These curves plot precision against recall for various threshold values, offering a visualization of balance between precision & recall and highlighting the functionality of model on different classification thresholds.

Chapter 4

Experimentation & Results

4.1 Experimentation Setup

For the model implementation we used Kaggle Notebooks with Keras and TensorFlow, utilising a GPU backend on Google Compute Engine, which provided 29 GB of RAM and

147.4 GB of disk space. The code was developed in Python 3, using several packages such as NumPy, pandas, and scikit-learn etc.

4.2 Classification Results

This section presents our detailed results for the classification of smartphone-based time series data collected from four different locations: bag, belly, hand, and thigh. We have passed various preprocessing techniques, including filters, SMOTE, and Short-Time Fourier Transform (STFT), to reach to this stage for enhancing the data quality. Our proposed methodology was consisting of two different model applications:

1. Direct Application of LSTM Models: We applied our proposed LSTM model directly to the signal's CSV data segmented via signal labeler tool.
2. Conversion and Application of CNN and Transfer Learning Models: We converted the signal data into STFT spectrograms and then applied CNN and transfer learning-based models.

4.2.1 Results on direct Application of LSTM Models

This section describes the detailed output of applying the LSTM model to the signal data collected from four different locations: bag, belly, hand, and thigh. To see the performance of the model we used learning curves, confusion matrices, and key classification metrics. Each location's results are analyzed in detail to understand the strength of the applied model and potential space for improvement. Below figures shows the learning curves for each location with training and validation accuracy and loss over epochs.

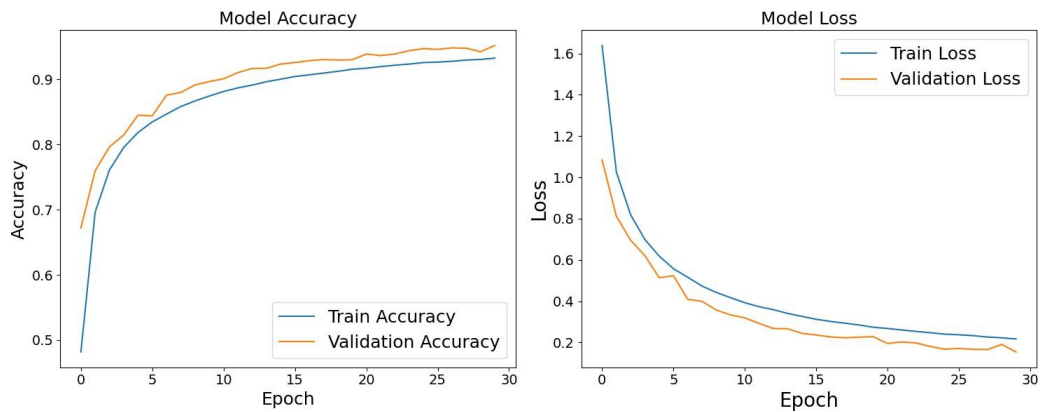


Figure 4.1: LSTM model Learning curve on Bag Location

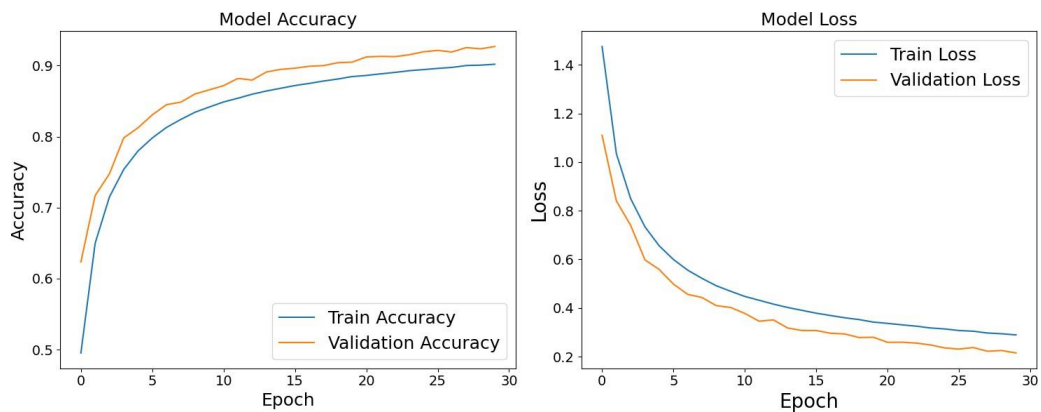


Figure 4.2: LSTM model Learning curve on Belly Location

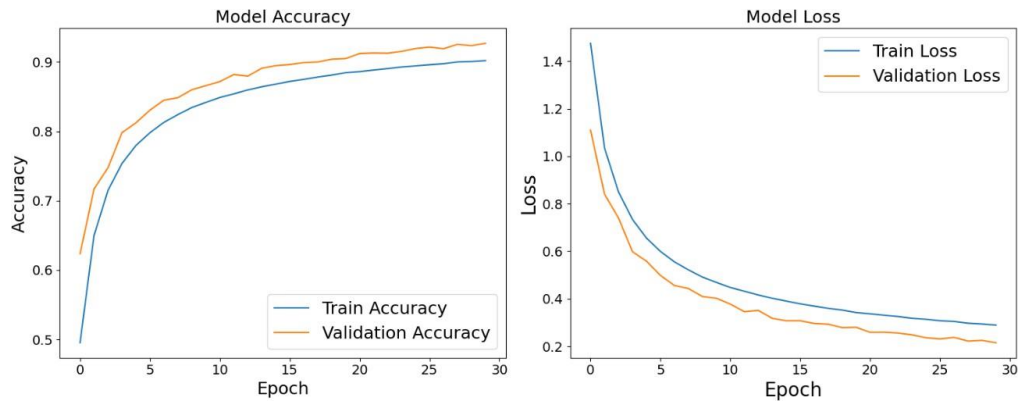


Figure 4.3: LSTM model Learning curve on Hand Location

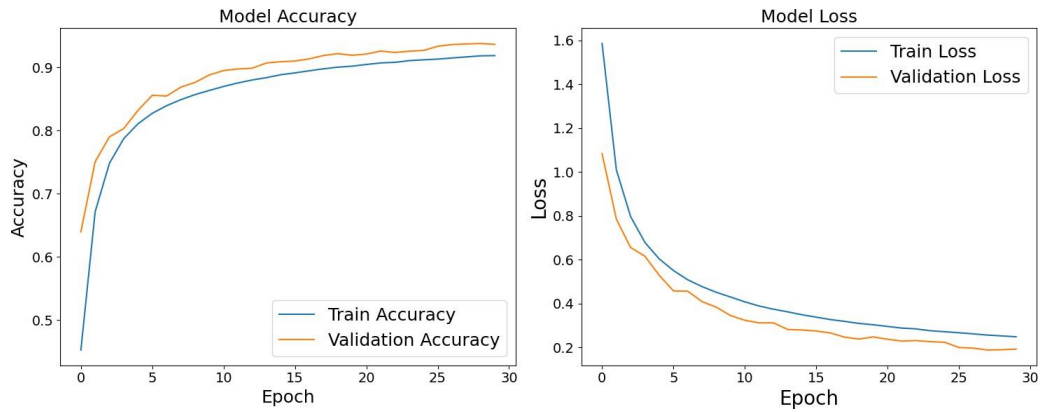
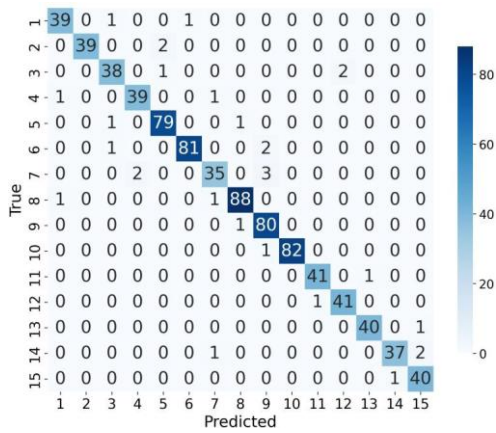


Figure 4.4: LSTM model Learning curve on Thigh Location

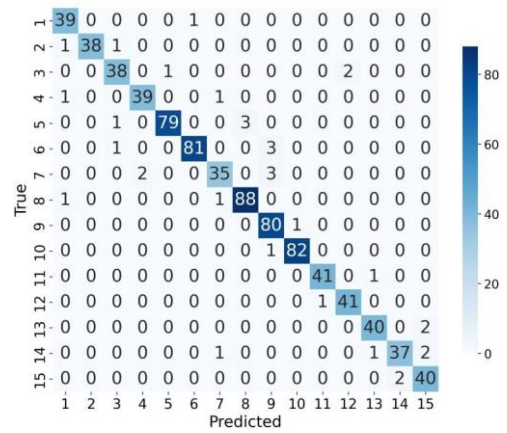
The learning curves showing steady decrease in both training and validation losses, alongside an increase in accuracy, proving effective model learning over 30-epochs across four locations: Bag, Belly, Hand, and Thigh. Despite differences in signals data due to rotations and carrying positions, accuracy curves stabilize around 0.75 for Hand and Belly and 0.85 for Bag and Thigh. Training loss converges near 0.2, while validation loss settles around 0.4. To further assess model performance, confusion matrices for each location are shown in figure 4.5, with the 15-classification problem represented in a 15x15 (rowsxcolumns). In the matrix, TP_i denotes the correctly classified samples for Class i , and FP_{ij} represents Class i incorrectly classified as Class j .

Table 4.1. Generic Confusion Matrix

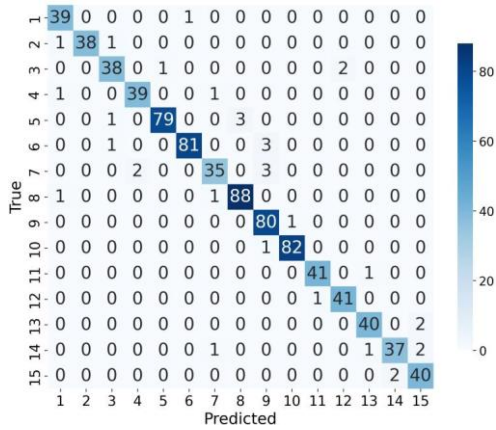
	Predicted Class 1	Predicted Class 2	...	Predicted Class 15
Actual Class 1	TP1	FP12	...	FP15
Actual Class 2	FP21	TP2	...	FP25
...
Actual Class 15	FP151	FP152	...	TP15



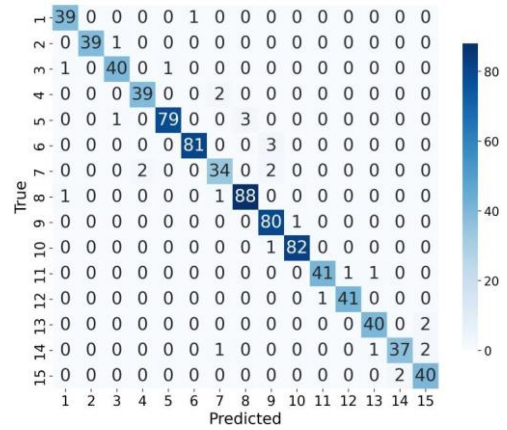
(a) Bag



(c) Hand



(b) Belly



(d) Thigh

Figure 4.5: LSTM Confusion Matrix Results on all locations

The confusion matrices represent the model achieves higher accuracy in distinguishing certain classes while facing challenges with others. The diagonal shows are correctly classified samples, the upper and the lower part shows the misclassified samples. The diagonal numbers are correctly classified activities, but there are certain activities on which model results misclassification. For example, in the bag location, the model shows confusion between Class 1 (e.g., Walking -slowly/quickly) and Class 3 (e.g., Jumping on the same place), likely due to the similarity in signal patterns generated by these activities when the sensor is placed in the bag, which may also be subject to more movements. The results show that for the testing set, the classification performance varies across different locations. Our proposed model correctly classified 94% of the samples at bag location, 92% of the samples correctly classified at belly location, 93% correctly classified activities at hand and 94% at Thigh. A comparison across locations shows that this confusion is less evident in the bag and thigh location, where the sensor is more directly involved in capturing arm and leg movements, providing clearer distinctions between these activities. Conversely, in the belly location, the confusion matrix is less favorable, possibly due to the sensor’s position being less optimal for capturing different activity patterns, leading to more overlap between classes. For our 15-classification problem overall performance metrics is shown in table 4.2.

Table 4.2. Performance Metrics of LSTM Model

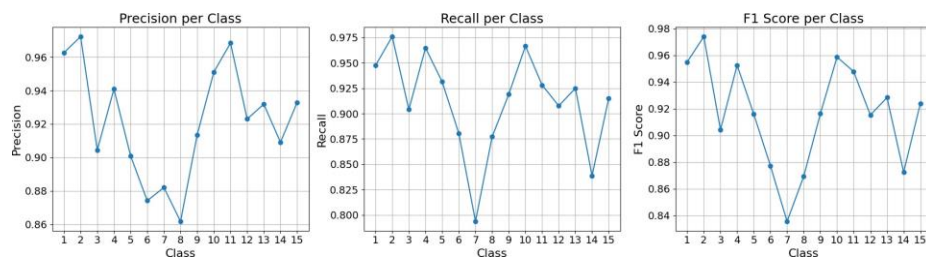
Location	Accuracy	Precision	Recall	F1-Score
Bag	94%	0.92	0.91	0.92
Belly	92%	0.92	0.91	0.91
Hand	93%	0.92	0.91	0.91
Thigh	94%	0.94	0.93	0.93

For looking the model’s performance at a more granular level, we also find the values of the precision, recall, and F1-score for individual class across the four sensor locations. The plot at x-axis has number of classes and y-axis shows values as shown in figure 4.6. Based on our analysis, it is evident that the performance at Bag Location 7 is suboptimal compared to other locations. The model is experiencing a higher rate of misclassification with samples from this activity. Specifically, the model has incorrectly classified this activity as class 4 on two occasions and as class 9 on three occasions. Additionally, there is an instance where

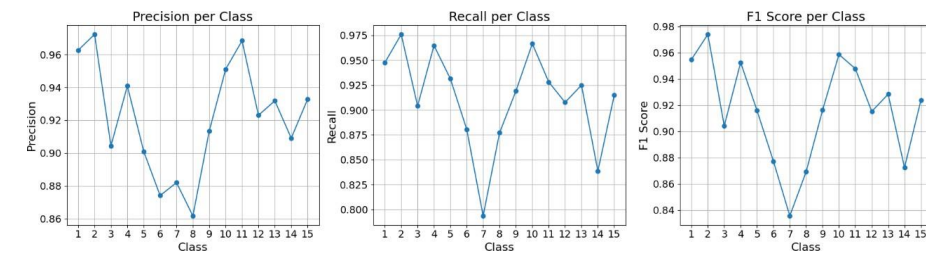
the model predicted class 8 as class 7. These misclassifications suggest that the model is struggling to accurately distinguish samples from Activity 7.



(a) Individual Class Precision, recall, F1-score points



(b) Individual Class Precision, recall, F1-score points



(c) Individual Class Precision, recall, F1-score points



(d) Individual Class Precision, recall, F1-score points

Figure 4.6: LSTM model results (a) Bag, (b) Belly, (c) Hand, (d) Thigh

4.2.2 Results of CNN and Transfer Learning Models

This section presents the results our CNN model and Transfer Learning models to STFT images. This approach provides a comparative perspective to our initial method, focusing on the effectiveness of these advanced models in handling image-based data across four different locations: bag, belly, hand, and thigh. We analyze the performance of these models through learning curves, confusion matrices, and accuracy metrics.

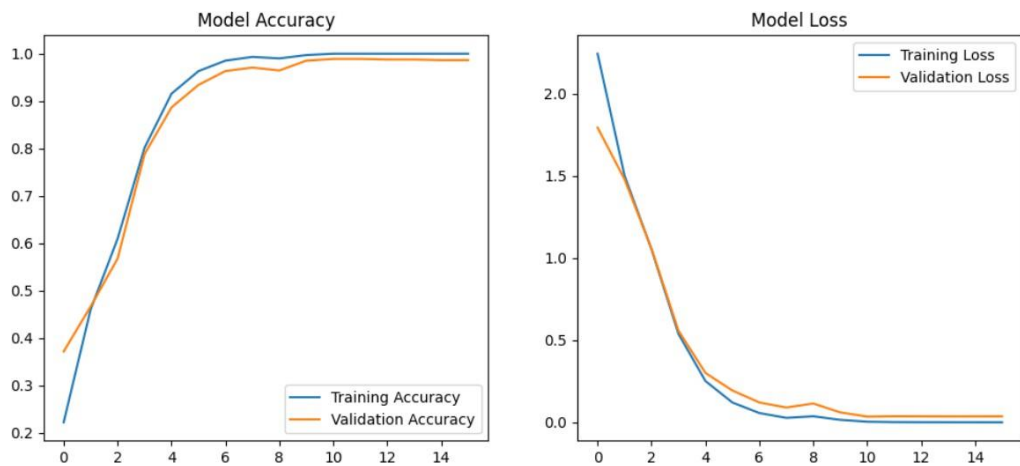


Figure 4.7: CNN model Learning curve on Bag Location

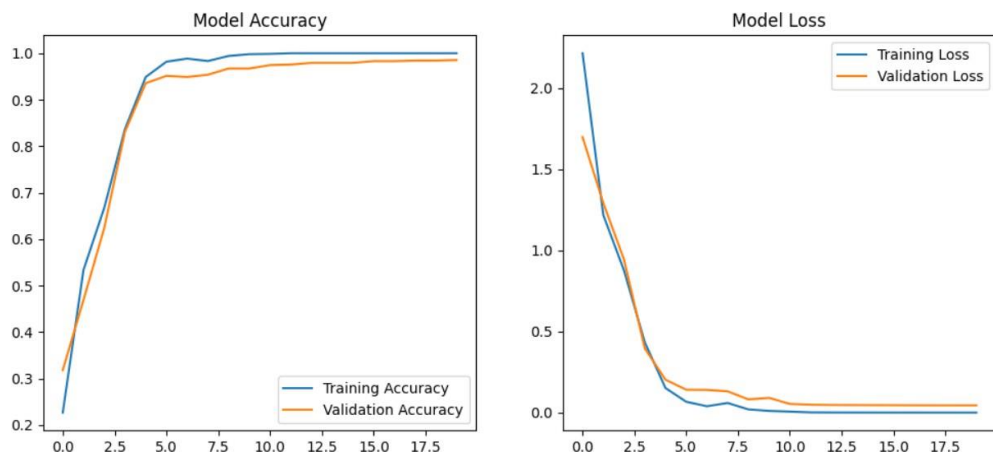


Figure 4.8: CNN model Learning curve on Belly Location

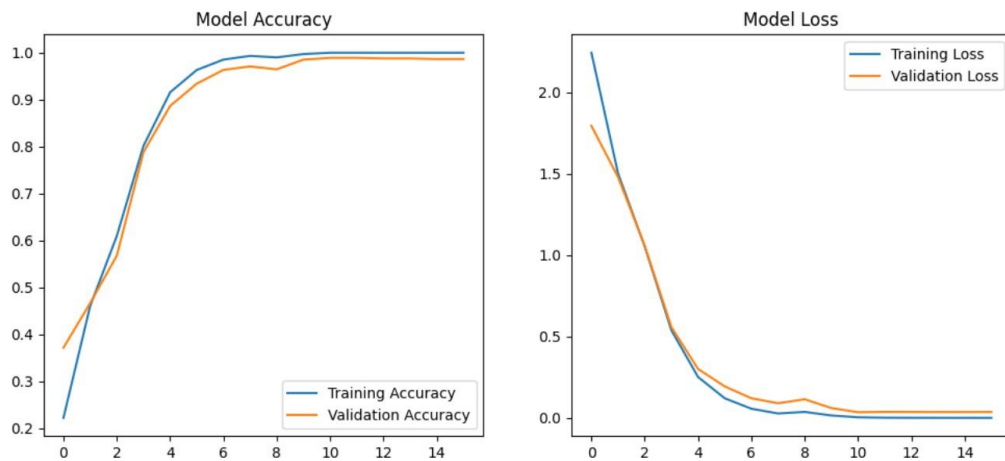


Figure 4.9: CNN model Learning curve on Hand Location

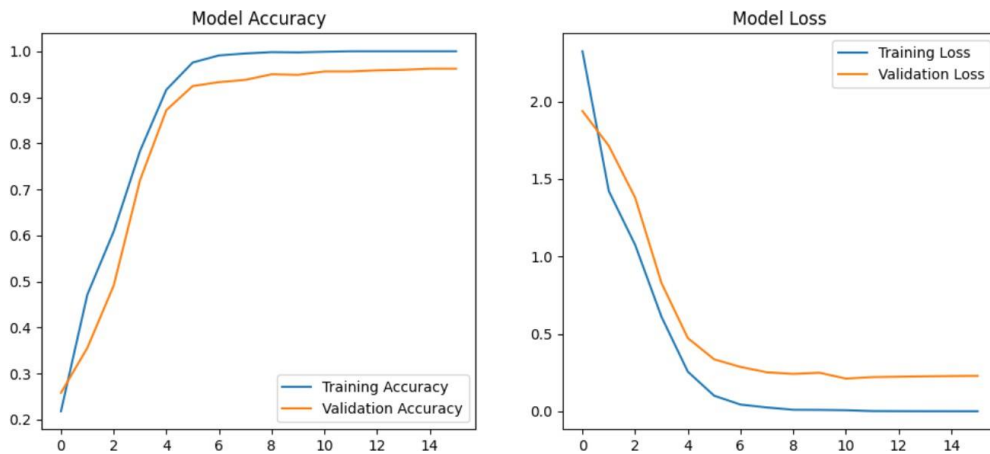


Figure 4.10: CNN model Learning curve on Thigh Location

The learning curves indicate that the CNN model is performing well on images data, with both training and testing losses consistently decreasing and accuracies increasing, suggesting convergence towards a well-optimized state. Notably, the gap between training accuracy and training loss is minimal, with their curves closely aligned in the bag, belly, and hand plots, reflecting high model accuracy. However, in the thigh position, while the accuracy graph converges at around 0.9, the gap between validation and training accuracy widens. Similarly, in the loss graph, convergence occurs at approximately 0.6 with an increased gap. This is due

to the different data distribution compared to other locations, leading to increased variability in validation performance.

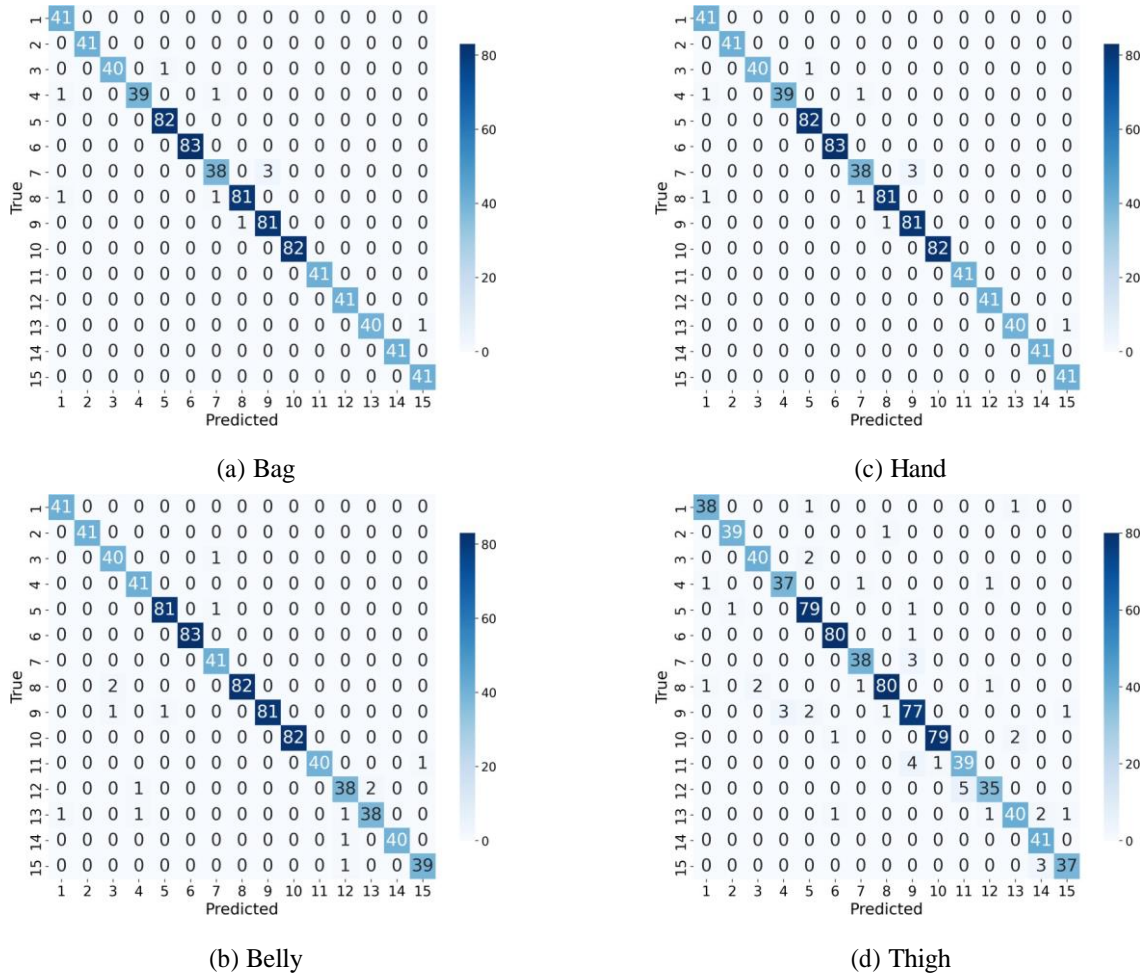


Figure 4.11: CNN Confusion Matrix Results

The confusion matrices show the distribution of misclassified and correctly classified classes by the CNN model. For the 'Bag,' 'Belly,' and 'Hand' datasets, the model exhibits high accuracy, reflecting its strong performance on the testing data from these locations. However, the 'thigh' dataset shows a few more misclassified classes due to variations in patterns, yet it still achieves commendable results on the testing images data.

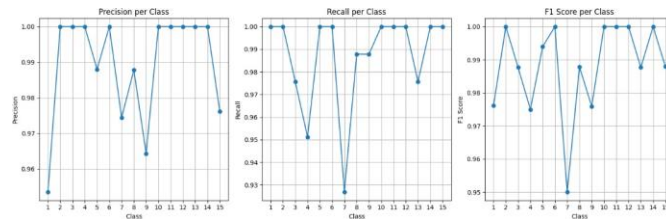
Table 4.3. Performance Metrics of CNN Model

Location	Accuracy	Precision	Recall	F1-Score
Bag	98%	0.98	0.98	0.98
Belly	98%	0.98	0.91	0.91
Hand	98%	0.98	0.91	0.91
Thigh	95%	0.94	0.93	0.93

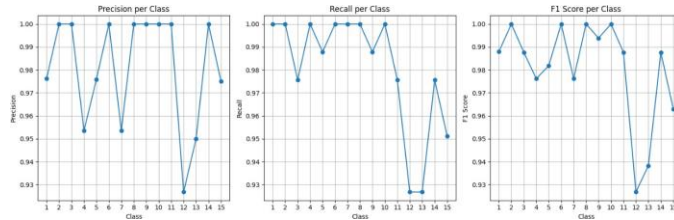
For the bag, belly, and hand locations, the model achieved high accuracies (98%) and strong precision, recall, and F1-scores. For the testing set of 821 samples, our proposed model correctly classified 98% of the sample images, for the Belly location, the accuracy is also high at 98%, with 2% of the samples being misclassified. Similarly, the Hand and Thigh location model correctly classified 98% and 95% sample images. Our results proved that the model is performing well on testing images of all locations. There are certain reasons, the bag, belly, and hand locations generally exhibit more stable and consistent movement patterns, which the model more easily learn and distinguished. But, comparing with the thigh location, it observed 3 points lower accuracy and performance metrics on our CNN-based smartphone activity recognition model. The reason is sensor placement on the thigh is likely to encounter greater variability in movement patterns compared to other locations. This increased variability introduces significant challenges for the model to distinguish between different activities accurately. Additionally, the signal quality from the thigh location can be compromised by factors such as loose attachment, clothing interference, or varying sensor positions, leading to degraded signal quality and subsequently lower accuracy. Also, certain activities may inherently produce less distinguishable patterns when recorded from the thigh, as subtle leg movements may not generate sufficiently distinctive signals. But overall the model has satisfactory results on spectrogram images.

Upon reviewing the performance across various activities, we observe some issues. Activity 7, which involves sitting for a moment, attempting to stand, and then collapsing into a chair, is misclassified three times. It is being incorrectly identified as "Sit on the ground, wait a moment (5 seconds), and get up slowly/quickly. Also, the model is encountering difficulties with accurately classifying the activity "Walking (4-5 steps), making a stop, bending down to pick up something, and then continuing to walk (4-5 steps)." It is frequently mis-classified as either "Walk upstairs/downstairs - Slowly/Quickly" or "SP/Person Touched or

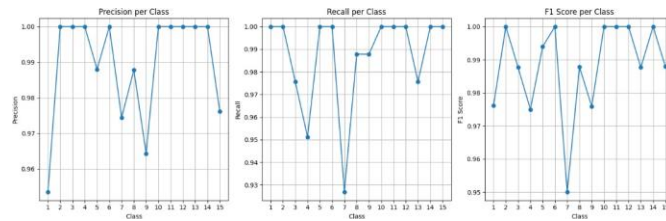
hit accidentally by a pole/another person. The misclassifications likely stem from overlapping motion patterns between activities and inadequate feature differentiation. For example, sitting down and standing up might have motion patterns that closely resemble those of sitting on the ground and getting up. Similarly, walking with brief stops and bends might have overlapping features with walking upstairs/downstairs or accidental touches.



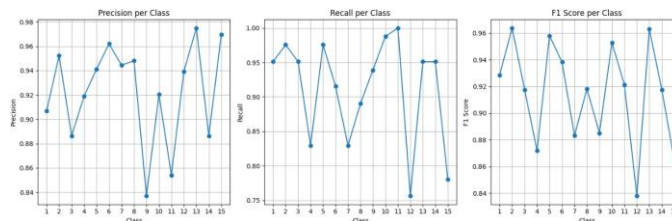
(a) Individual Class Precision, recall, F1-score points



(b) Individual Class Precision, recall, F1-score points



(c) Individual Class Precision, recall, F1-score points



(d) Individual Class Precision, recall, F1-score points

Figure 4.12: CNN model results (a) Bag, (b) Belly, (c) Hand, (d) Thigh

4.2.3 Results of Transfer Learning Models

Our proposed EfficientNet_B0 model shown a remarkable balance between accuracy and computational efficiency, making it an excellent choice for our activity recognition task. By integrating the potential of transfer learning, this model was trained on filtered images that were preprocessed using STFT to extract relevant features. Additionally, we used the SMOTE to handle class imbalance, ensuring that the model was subjected to a well-balanced dataset. The learning curves generated during the process of training proved the model's robust learning capability. Below figures shows that the training and validation loss steadily decreased over time, indicating that the model capably captured the core patterns in the images due to applied preprocessing techniques.

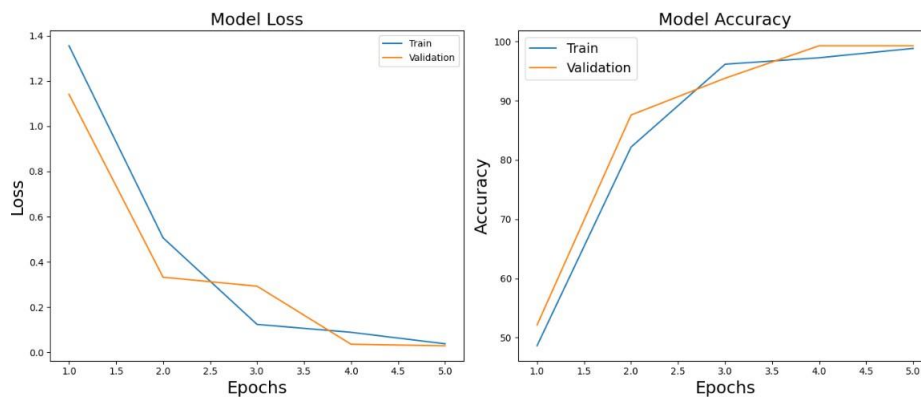


Figure 4.13: EfficientNet_B0 model Learning curve on Bag Location

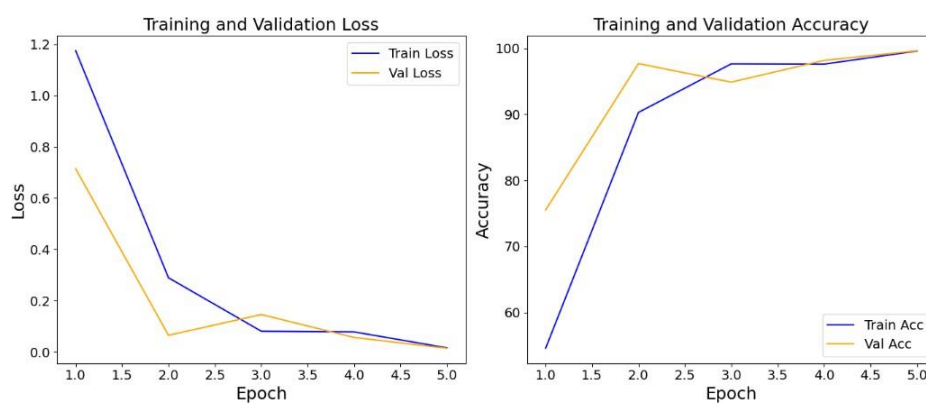


Figure 4.14: EfficientNet_B0 model Learning curve on Belly Location

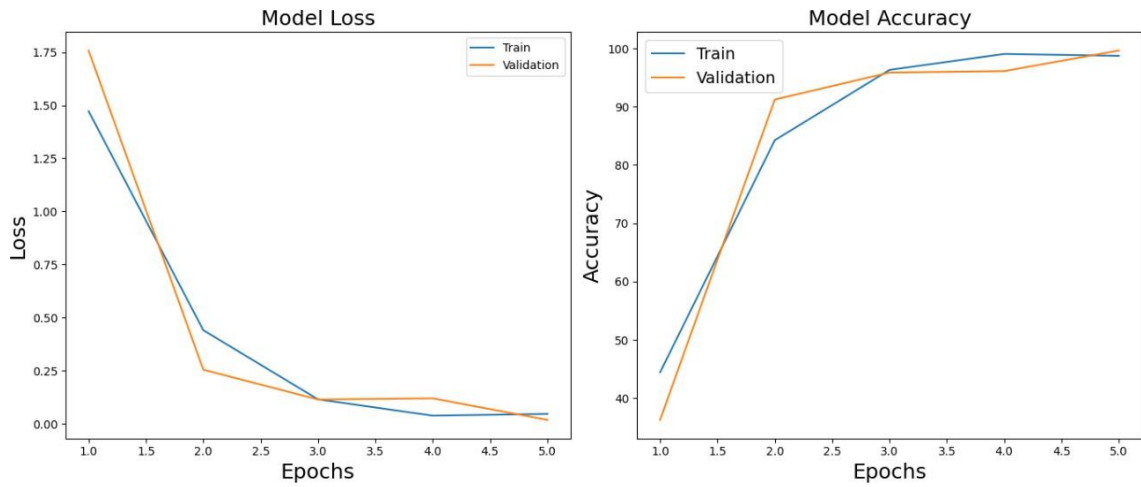


Figure 4.15: EfficientNet_B0 model Learning curve on Hand Location

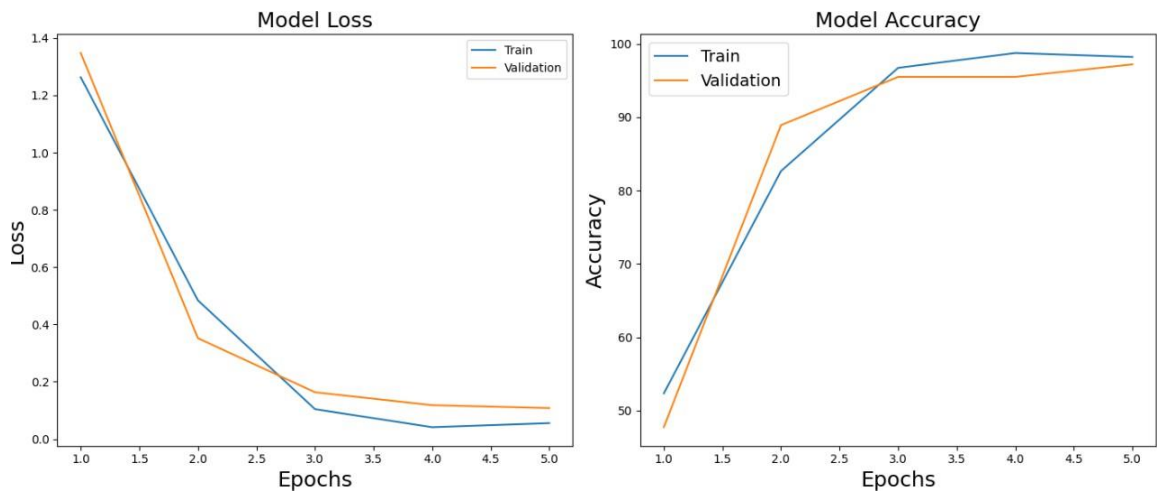


Figure 4.16: EfficientNet_B0 model Learning curve on Thigh Location

Although there is a slight fluctuation observed in both the training and validation curves at certain points, which is a common occurrence in deep learning training processes. As the model learns, it might encounter mini-batches with different characteristics or difficulty levels, causing temporary spikes or drops in loss and accuracy. Secondly, we used dynamic learning rate schedule, the fluctuations occur when the learning rate changes. For instance, when the learning rate decreases after a certain number of epochs, the model shows some

adjustment period where the loss or accuracy momentarily fluctuates before stabilizing. The confusion matrices, presented in figure 4.17, provide more detailed points of the classification outcomes of proposed model across all classes. The high diagonal values across the matrices suggest that the EfficientNet_B0 model accurately identified the majority of the activities, with only a few misclassifications in classes with overlapping features. This result underscores the model’s capacity to distinguish between complex activities, a critical aspect of smartphone-based activity recognition.

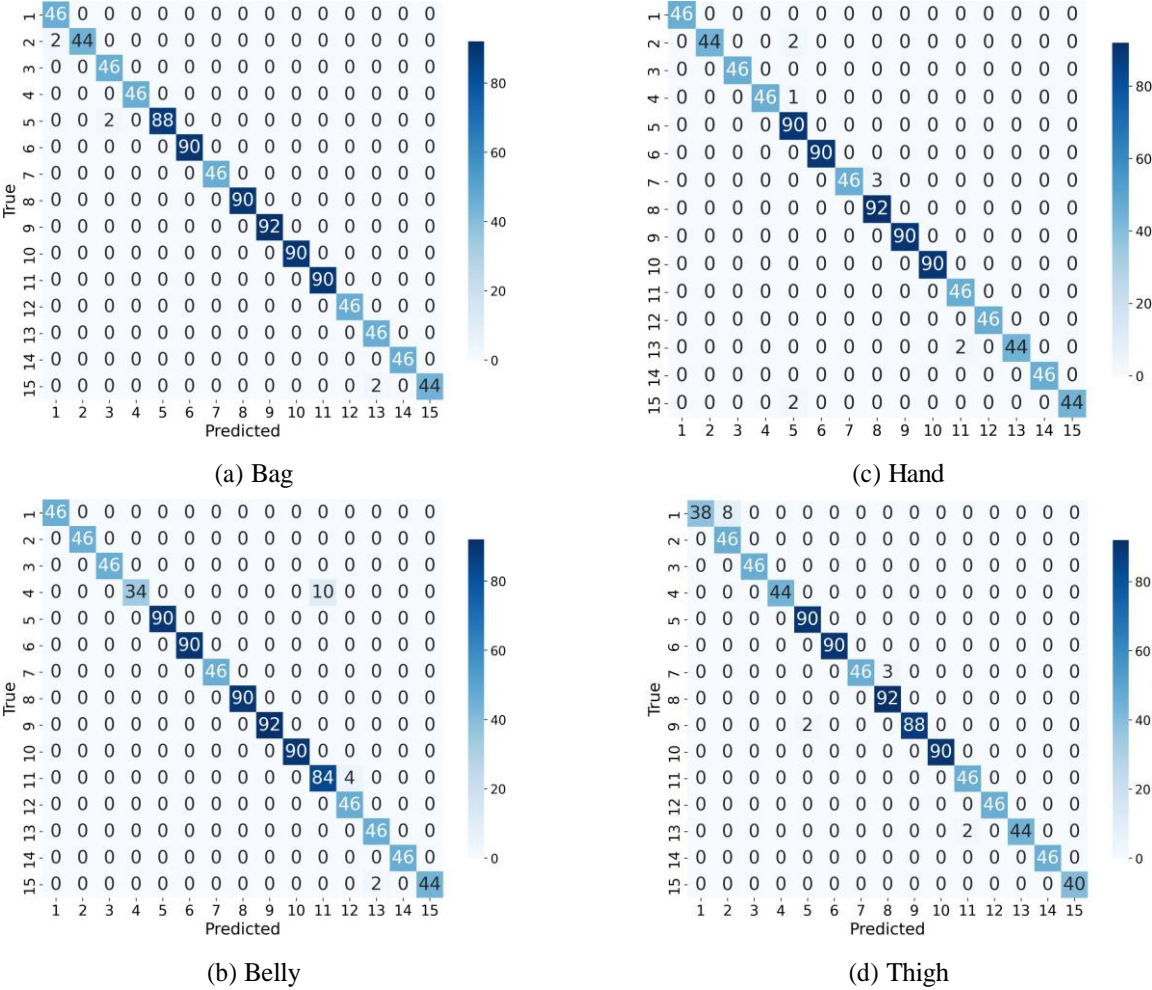
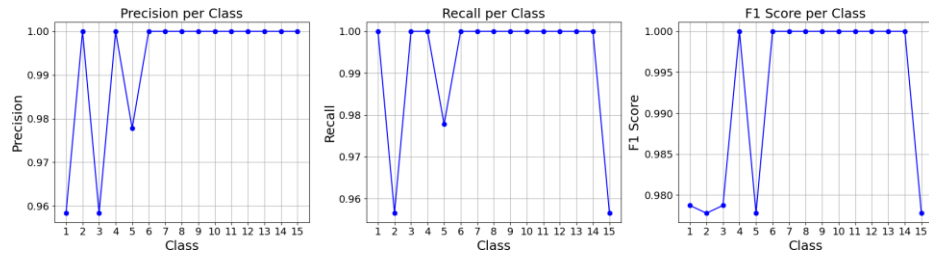


Figure 4.17: EfficientNet_B0 Confusion Matrix Results

For a deeper analysis of the model’s classification results across different locations, below tables provides a comprehensive overview of the accuracies, precision, recall, and F1-scores

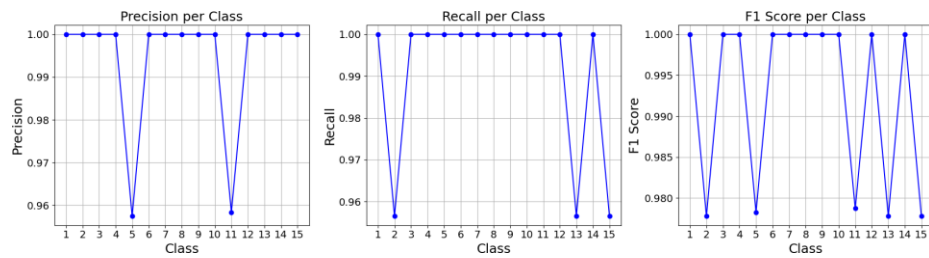
for each location, further illustrating each model’s ability to achieve high performance metrics, confirming its successful application to the datasets.



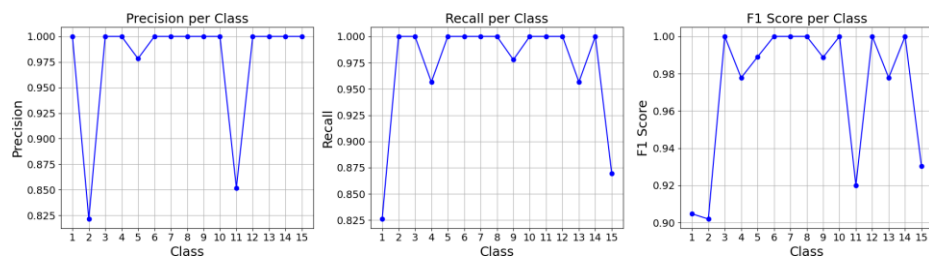
(a) Individual Class Precision, recall, F1-score points



(b) Individual Class Precision, recall, F1-score points



(c) Individual Class Precision, recall, F1-score points



(d) Individual Class Precision, recall, F1-score points

Figure 4.18: EfficientNet_B0 model results (a) Bag, (b) Belly, (c) Hand, (d) Thigh

Although, there is a drastic drop in precision, recall, and F1-score for specific classes likely result from the model’s difficulty in distinguishing between activities with similar features, such as "walking" and "running." The reason can be some activities may naturally be more ambiguous or have edge cases where they are difficult to classify correctly. For instance, activities performed in unusual ways or combined with other activities could lead to such misclassifications. But those are less in number.

Table 4.4. Performance Metrics of EfficientNet_B0 Model

Location	Accuracy	Precision	Recall	F1-Score
Bag	99%	0.99	0.99	0.99
Belly	98%	0.98	0.98	0.98
Hand	99%	0.99	0.99	0.99
Thigh	97%	0.98	0.98	0.98

EfficientNet_B0 achieved impressive accuracy and consistency across various sensor placements: 99% accuracy with excellent precision, recall, and F1-score (0.99) for Bag and Hand locations. The model performed slightly lower at the Belly (98%) and Thigh (97%) locations, still maintaining strong precision and recall (0.98). These results show the model’s robust generalization due to a unique compound scaling method that adjusts the network’s width (total channels), depth (total layers), and input resolution simultaneously and proportionally.

Our model employs a balanced scaling methodology that optimizes learning efficiency while mitigating the risks of overfitting and underfitting. Empirical evidence indicates that EfficientNet_B0 demonstrates superior feature extraction capabilities, effectively distinguishing between low-level textures and high-level semantics across diverse sensor data modalities, in contrast to traditional CNNs. The robust generalization performance of EfficientNet_B0 across these variations underscores its suitability for Human Activity Recognition (HAR) applications.

4.2.4 Vision Transformer (ViT)

The performance of the Vision Transformer (ViT) model was evaluated across multiple sensor locations, with the analysis focusing on the learning dynamics and predictive accuracy. The results, visualized through learning curves and precision-recall plots, provide a deep understanding of the model to learn and generalize across different tasks. The learning curves indicate a rapid decrease in loss during the initial epochs, with both training and validation losses stabilizing after around 4 epochs with some fluctuation like efficientNet_B0 model due to learning rate scheduler. This suggests that the model quickly adapts to the task.

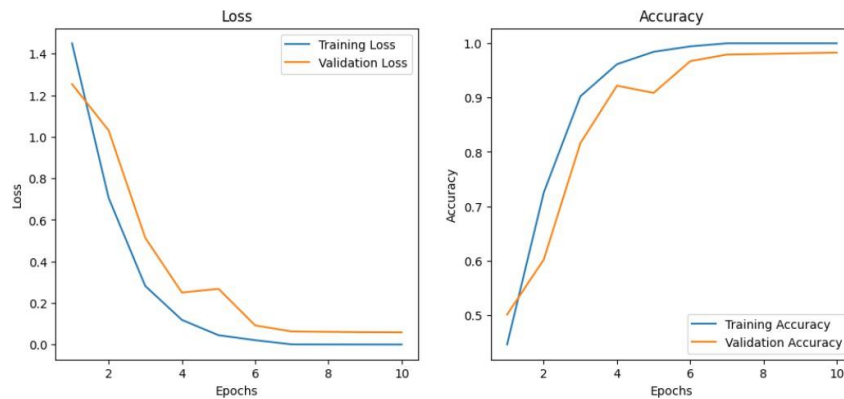


Figure 4.19: Learning curve on Bag Location

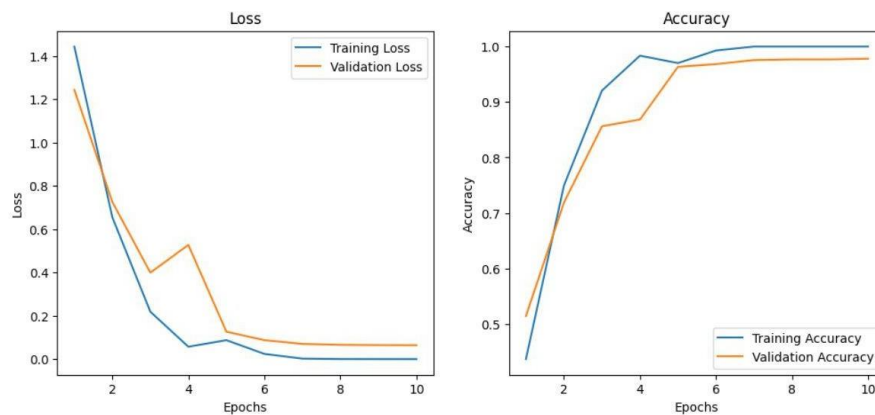


Figure 4.20: Learning curve on Belly Location

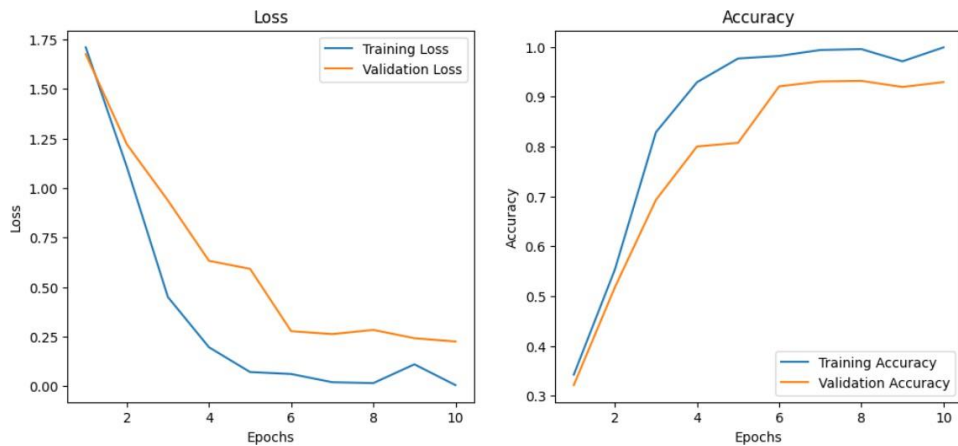


Figure 4.21: Learning curve on Hand Location

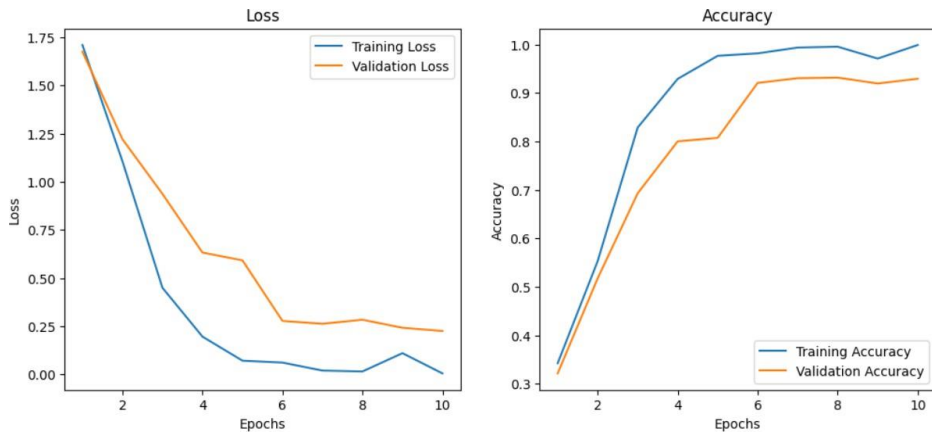


Figure 4.22: Learning curve on Thigh Location

Furthermore, the precision, recall, and F1 scores for each class were analyzed to assess the performance of the Vision Transformer (ViT) model on the all four location. Overall the model performs well across most classes, maintaining high precision, recall, and F1 scores for the majority of activities. However, a notable decrease in performance is observed in thigh location for certain classes, particularly for classes 2 and 13. This is be due to the complexity or similarity of these activities with others, leading to misclassification. Despite this, the model still shows strong performance across most classes, indicating its robustness in recognizing a wide range of activities.

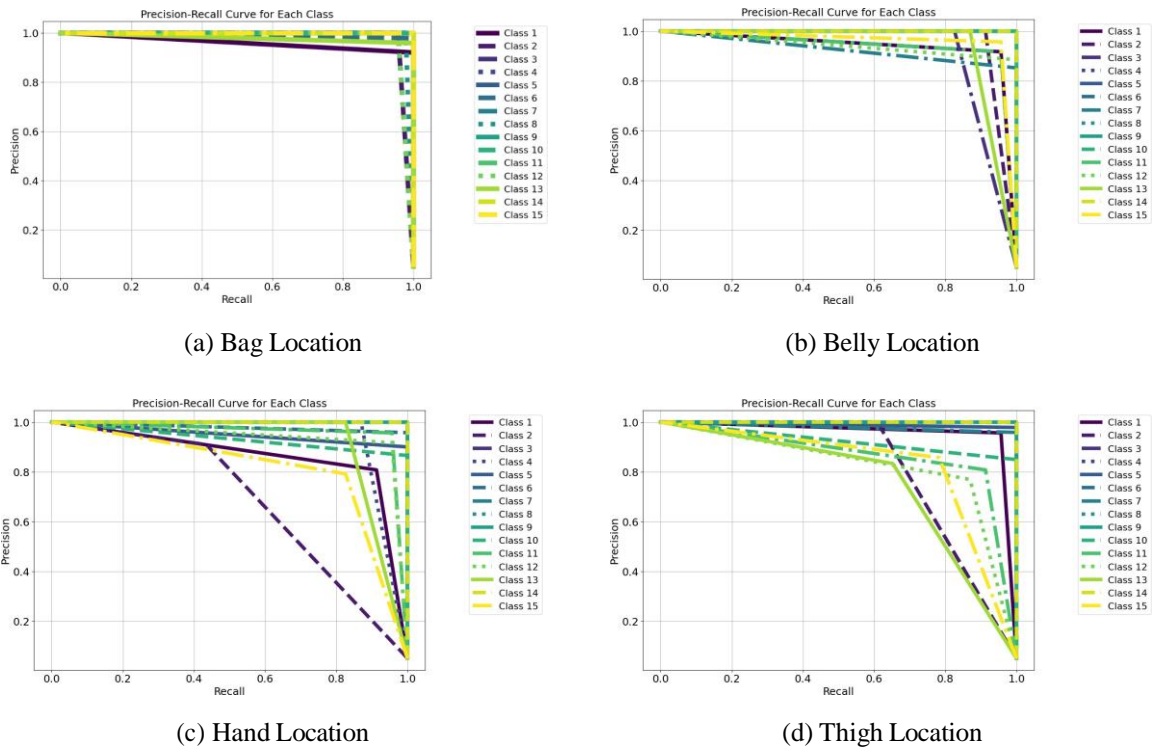


Figure 4.23: Precision, Recall, and F1-Score Curves in ViT Transformer Evaluation

Furthermore, to measure the performance of the Vision Transformer (ViT) model, the confusion matrix offers an in-depth analysis of the model's predictions, showcasing both accurate classifications and instances where the model made errors in identifying activities. By analyzing the confusion matrix, we can gain deeper insights into the specific challenges faced by the model, such as identifying classes that are frequently confused with one another. In the thigh location, we are getting high misclassified classes as compared to other, that is the reason for comprising accuracy. The diagonal entries show the number of correct predictions for each class. For example, in confusion matrix at thigh location, class 1 has 22 correct predictions, class 2 has 14, and so on. Off-Diagonal values represent misclassifications, where the model predicted an incorrect class. For example, 1 instance of class 1 was misclassified as class 5, and 8 instances of class 2 were misclassified as class 10. Classes such as 6, 7, 8, 9, and 10 have high correct predictions (45, 23, 46, 45, 45 respectively), indicating strong performance in these categories. However, there are noticeable misclassifications in classes like 2, 11, 12, and 15. Same goes

for other locations, like in hand confusion matrix (c) class 1 has 21 correct predictions, class 2 has 10, and so on. There are some misclassifications like class 2 was misclassified 5 times as class 1, 4 times as class 5, 3 times as class 10, and 1 time as class 12. Overall classes such as 5, 6, 7, 8, 9, and 10 performed well with 45, 45, 23, 46, 45, and 45 correct predictions respectively, indicating strong performance in these categories.

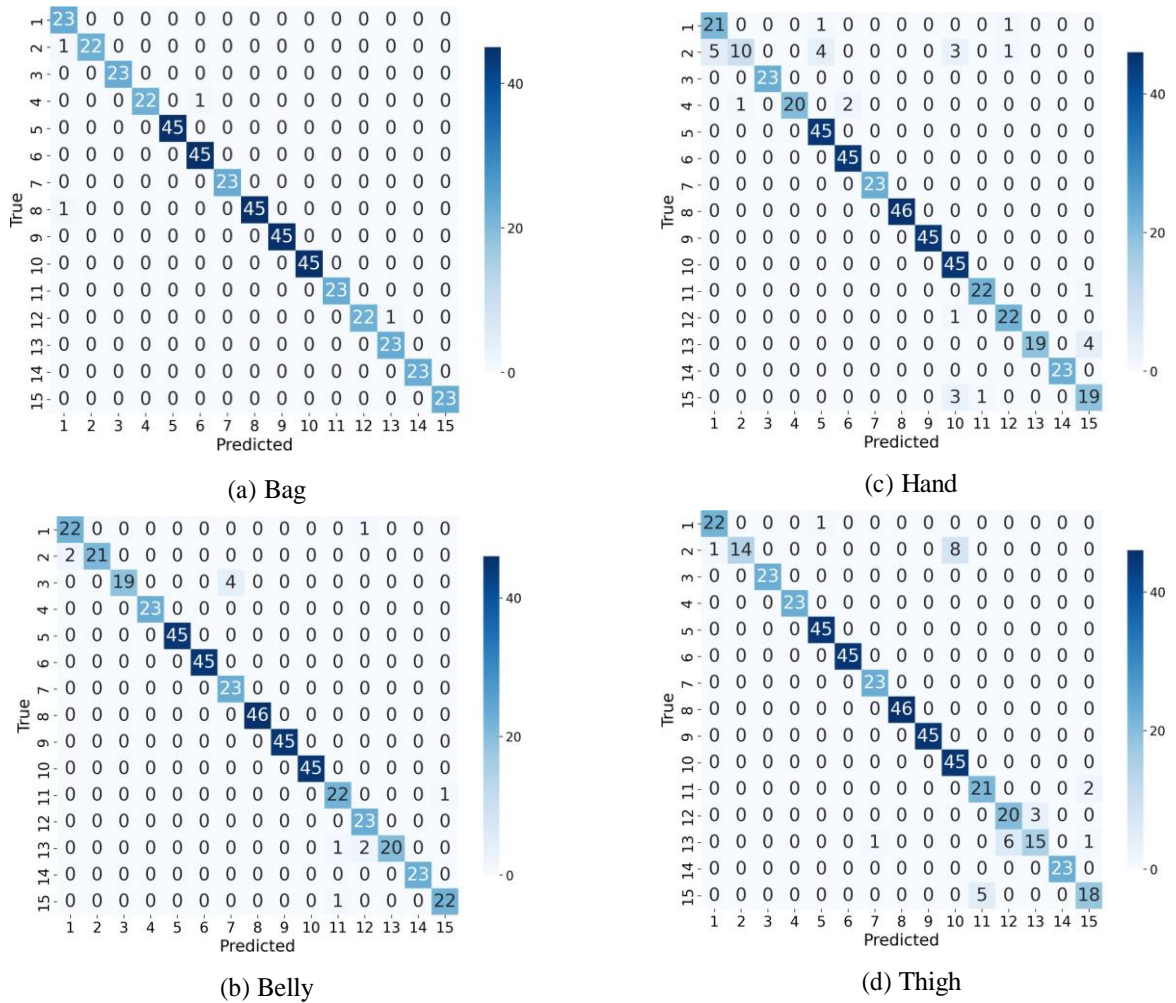


Figure 4.24: ViT Transformer Confusion Matrix Results

Table 4.5. Performance Metrics of ViT Transformer

Location	Accuracy	Precision	Recall	F1-Score
Bag	99%	0.99	0.99	0.99
Belly	97%	0.97	0.97	0.97
Hand	99%	0.99	0.99	0.99
Thigh	94%	0.94	0.94	0.94

Based on the performance tables for the LSTM, CNN, and transfer learning models applied to our custom activity recognition dataset, it is evident that the transfer learning models consistently outperform the other approaches across key metrics such as accuracy, precision, recall, and F1-score. It proved that the pre-trained models used in transfer learning are better at capturing the underlying patterns, enabling the data to generalize better to new and unseen data. Secondly, the feature extraction capabilities of transfer learning models are enhanced by integrating learned features from previous tasks. These features are more robust and transferable, leading to better performance in our activity recognition task.

For further testing our proposed methodology, we also applied it to the publicly available Human Activity Recognition Trondheim (HARTH) dataset. This dataset is a professionally annotated collection featuring data from 22 subjects, each equipped with two 3-axial accelerometers worn on the right thigh and lower back for approximately 2 hours in a free-living environment. The dataset provides a rich resource for the development and benchmarking of machine learning models aimed at precise human activity recognition (HAR) in real-world settings.

A prior study [36] utilized this dataset with traditional machine learning techniques, specifically Support Vector Machines (SVM), achieving an F1 Score of 0.81, Precision of 0.79, and Recall of 0.85 when analyzing the data from the thigh sensor. To assess the effectiveness of our proposed methodology, we applied three advanced algorithms: Vision Transformer (ViT), EfficientNet_B0, and Convolutional Neural Networks (CNN). The ViT model achieved an impressive Accuracy of 98%, F1 Score of 0.9809, Precision of 0.9821, and Recall of 0.9808. The EfficientNet_B0 model outperformed with an Accuracy of 99%, F1 Score of 0.99, Precision of 0.99, and Recall of 0.99, while the CNN model demonstrated

robust performance with an Accuracy of 97%, F1 Score of 0.97, Precision of 0.97, and Recall of 0.97.

The superior performance of our models can be attributed to their ability to capture complex patterns and relationships within the data, which traditional models like SVM might overlook. By leveraging the power of deep learning architectures, particularly those designed for handling high-dimensional and sequential data, our methodology more accurately captures the nuances of human activity, leading to significantly improved accuracy and generalization in real-world scenarios.

Table 4.6. Comparison of Performance Metrics

Reference	Location	Models/Techniques	Accuracy	F1 Score	Precision	Recall
Logacjov et al. [36]	Thigh	ML models SVM	-	0.81	0.79	0.85
Proposed Models	Thigh	ViT Transformer	98%	0.98	0.98	0.98
		EfficientNet_B0	99%	0.99	0.99	0.99
		CNN	97%	0.97	0.97	0.97

Summarizing the overall results, the efficientNet_B0 and Vision Transformer (ViT) models performed well across different sensor locations in our custom dataset, though they struggled with certain activities, particularly at the thigh sensor. This performance drop may be due to the complexity or similarity of some activities, which the models found challenging to distinguish. Environmental factors were not considered, affecting classification accuracy for activities like D15, involving complex actions and interactions. Despite these issues, the models showed robustness, and the use of transfer learning reduced training time and resources, making them practical for real-time activity recognition systems.

To further validate our methodology, we applied the same models to the HARTH dataset, a professionally annotated dataset featuring data from multiple subjects in a free-living environment. The results showed that both ViT and EfficientNet_B0 achieved impressive accuracy and F1 scores, outperforming traditional machine learning models like SVM. The EfficientNet_B0 model, in particular, achieved near-perfect performance, demonstrating the efficacy of transfer learning techniques in handling high-dimensional and sequential data.

Chapter 5

Conclusion & Future Work

As the global population continues to age, with the number of individuals aged 60 and older projected to reach 1.4 billion by 2030 and 2.1 billion by 2050, the need for effective health monitoring and quality of life enhancement becomes increasingly critical. According to the World Health Organization (WHO), this demographic shift underscores the importance of innovative solutions for maintaining and improving the health of the elderly. Accurate activity recognition, facilitated by advances in technology, plays a pivotal role in achieving these goals.

Prior studies have predominantly concentrated on activity recognition from single-body locations, such as the waist or wrist. This narrow focus leaves a significant gap in understanding how activity patterns vary across multiple body locations. Furthermore, there is a paucity of research exploring the application of transfer learning techniques in this domain. This oversight limits the potential to use pre-trained models and adapt them to our specific dataset, which spans multiple smartphone locations and diverse activities.

Smartphones, equipped with built-in sensors such as accelerometers and gyroscopes, have emerged as valuable tools for real-world data collection. Their widespread use provides a practical means of gathering activity-related data, which can be integrated to enhance health monitoring. Our study, in collaboration with Bootcamp Co Ltd, specifically targets the elderly population in South Korea due to increasing aging problem there. We have developed a custom dataset that encompasses activity data from multiple smartphone locations—Bag, Belly, Hand, and Thigh—addressing a notable research gap. Our custom dataset, which includes data recorded at 100 Hz, captures a diverse range of ADLs performed by 19 South

Korean participants. This dataset, featuring data from four distinct body locations and including 15 different ADLs, is unique in its comprehensive coverage and the absence of previous deep learning applications.

The study introduces and evaluates deep learning models designed to classify these activities accurately. The proposed models, including LSTM, CNN, EfficientNet_B0 and Vision Transformer (ViT), demonstrate significant performance improvements over existing methods. The findings highlight that EfficientNet_B0 outperformed in classifying ADLs and offer a comprehensive benchmark for smartphone-based activity recognition.

Looking ahead, future work will involve broadening the dataset to include younger age groups and integrating exercise activities alongside ADLs. This will help create a more comprehensive model and allow for a better understanding of how activity recognition patterns differ across various age demographics and improve the model's performance in more diverse populations. Including exercise activities alongside the existing Activities of Daily Living (ADLs) in our dataset will provide a more holistic view of physical activity. Exercise activities often involve different movement patterns and intensities compared to routine ADLs. By integrating these activities, we can enhance the model's ability to recognize and classify a broader spectrum of physical behaviors, leading to more accurate and useful in- sights for health monitoring.

References

- [1] S. K. Tripathy, R. Singh, R. Srivastava, A. K. Bhoi, and S. K. Satapathy, *Advances in Human Activity Detection and Recognition (HADR) Systems*, 1st ed., ser. Synthesis Lectures on Computer Science. Cham: Springer Cham, 2024.
- [2] A. Barua, X. Jiang, and D. Fuller, “The effectiveness of simple heuristic features in sensor orientation and placement problems in human activity recognition using a single smartphone accelerometer,” *BioMedical Engineering OnLine*, vol. 23, no. 1, p. 21, February 2024.
- [3] T.-Y. Ren, L.-H. Yang, C. Nugent, F.-F. Ye, N. Irvine, and J. Liu, “Extended belief rule base model with novel rule generation for sensor-based human activity recognition under big data,” in *Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022)*, J. Bravo, S. Ochoa, and J. Favela, Eds. Cham: Springer International Publishing, 2023.
- [4] A. Logacjov, S. Herland, A. Ustad, and K. Bach, “Selfpab: Large-scale pre-training on accelerometer data for human activity recognition,” *Applied Intelligence*, vol. 54, no. 6, pp. 4545–4563, March 2024.
- [5] G.-R. Johanna, A.-C. Paola Patricia, O.-B. Alvaro Agustín, S.-B. Eydy del Carmen, U.-T. Miguel, D. la Hoz-Franco Emiro, D.-M. Jorge Luis, B. Shariq Aziz, and M. Diego, “Predictive model for the identification of activities of daily living (adl) in indoor environments using classification techniques based on machine learning,” *Procedia Computer Science*, vol. 191, pp. 361–366, 2021.
- [6] H. Son, J. W. Lim, S. Park, B. Park, J. Han, H. B. Kim, M. C. Lee, K.-J. Jang, G. Kim, and J. H. Chung, “A machine learning approach for the classification of falls and activ-

- ities of daily living in agricultural workers,” *IEEE Access*, vol. 10, pp. 77 418–77 431, 2022.
- [7] W. Kong, L. He, and H. Wang, “Exploratory data analysis of human activity recognition based on smart phone,” *IEEE Access*, vol. 9, pp. 73 355–73 364, 2021.
- [8] S. Liaqat, K. Dashtipour, S. A. Shah, A. Rizwan, A. A. Alotaibi, T. Althobaiti, K. Arshad, K. Assaleh, and N. Ramzan, “Novel ensemble algorithm for multiple activity recognition in elderly people exploiting ubiquitous sensing devices,” *IEEE Sensors Journal*, vol. 21, no. 16, pp. 18 214–18 221, 2021.
- [9] M. Saleh, M. Abbas, J. Prud’Homm, D. Somme, and R. Le Bouquin Jeannès, “A reliable fall detection system based on analyzing the physical activities of older adults living in long-term care facilities,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 2587–2594, 2021.
- [10] A. E. Nieto-Vallejo, C. A. Parra-Rodriguez, and O. Ramirez-Perez, “Classification of activities of daily living for older adults using machine learning and fixed time windowing technique,” *IEEE Sensors Journal*, vol. 23, no. 24, pp. 31 513–31 522, 2023.
- [11] T. Aşuroğlu, “Complex human activity recognition using a local weighted approach,” *IEEE Access*, vol. 10, pp. 101 207–101 219, 2022.
- [12] N. Mani, P. Haridoss, and B. George, “Evaluation of a combined conductive fabric-based suspender system and machine learning approach for human activity recognition,” *IEEE Open Journal of Instrumentation and Measurement*, vol. 2, pp. 1–10, 2023.
- [13] T. Li, X. Zhou, K. Kokubun, and W. Chen, “Human activity recognition in free living using a wearable sensor: A two-stage approach,” in *2023 12th International Conference on Awareness Science and Technology (iCAST)*, 2023, pp. 142–146.
- [14] P.-H. Lin, P.-H. Kuo, and K.-L. Chen, “Developmental prediction of poststroke patients in activities of daily living by using tree-structured parzen estimator–optimized stacking ensemble approaches,” *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 5, pp. 2745–2758, 2024.

- [15] S. Abbas, G. A. Sampedro, S. Alsubai, S. Ojo, A. S. Almadhor, A. A. Hejaili, and L. Strazovska, "Advancing healthcare and elderly activity recognition: Active machine and deep learning for fine-grained heterogeneity activity recognition," *IEEE Access*, vol. 12, pp. 44 949–44 959, 2024.
- [16] S. Mekruksavanich, A. Jitpattanakul, K. Sitthithakerngkiet, P. Youplao, and P. Yupapin, "Resnet-se: Channel attention-based deep residual network for complex activity recognition using wrist-worn wearable sensors," *IEEE Access*, vol. 10, pp. 51 142–51 154, 2022.
- [17] S. H. H. Shah, A. S. T. Karlsen, M. Solberg, and I. A. Hameed, "An efficient and lightweight multiperson activity recognition framework for robot-assisted healthcare applications," *Expert Systems with Applications*, vol. 241, p. 122482, 2024.
- [18] K. Zhang, Q. Wang, X. Meng, and J. Wang, "A human activity recognition scheme using mobile smartphones based on varying orientations and positions," *IEEE Sensors Journal*, vol. 24, no. 10, pp. 17 127–17 139, 2024.
- [19] V. Farrahi, U. Muhammad, M. Rostami, and M. Oussalah, "Accnet24: A deep learning framework for classifying 24-hour activity behaviours from wrist-worn accelerometer data under free-living environments," *International Journal of Medical Informatics*, vol. 172, p. 105004, 2023.
- [20] L. Zhang, J. Yu, Z. Gao, and Q. Ni, "A multi-channel hybrid deep learning framework for multi-sensor fusion enabled human activity recognition," *Alexandria Engineering Journal*, vol. 91, pp. 472–485, 2024.
- [21] J. L. Mahendra Kumar, M. Rashid, R. M. Musa, M. A. Mohd Razman, N. Sulaiman, R. Jailani, and A. P.P. Abdul Majeed, "The classification of eeg-based wink signals: A cwt-transfer learning pipeline," *ICT Express*, vol. 7, no. 4, pp. 421–425, 2021.
- [22] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco, and M. R. Luaces, "Deep learning models for real-life human activity recognition from smartphone sensor data," *Internet of Things*, vol. 24, p. 100925, 2023.

- [23] S. Das, A. Adhikary, A. A. Laghari, and S. Mitra, "Eldo-care: Eeg with kinect sensor based telehealthcare for the disabled and the elderly," *Neuroscience Informatics*, vol. 3, no. 2, p. 100130, 2023.
- [24] M. K. A. Ramesh, R. G. S. Prem, R. A. A., and D. M. Gopinath, "1d convolution approach to human activity recognition using sensor data and comparison with machine learning algorithms," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 130–143, 2021.
- [25] M. A. Uddin, M. A. Talukder, M. S. Uzzaman, C. Debnath, M. Chanda, S. Paul, M. M. Islam, A. Khraisat, A. Alazab, and S. Aryal, "Deep learning-based human activity recognition using cnn, convlstm, and lrcn," *International Journal of Cognitive Computing in Engineering*, vol. 5, pp. 259–268, 2024.
- [26] R. K. Athota and D. Sumathi, "Human activity recognition based on hybrid learning algorithm for wearable sensor data," *Measurement: Sensors*, vol. 24, p. 100512, 2022.
- [27] E. Azam, A. Hassan, M. A. Basit Malik, and I. K. Niazi, "Classification of complicated upper limb movements from pre-movement eeg signals using stft and spectral characteristics," in *2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS)*, 2023, pp. 386–391.
- [28] S. Gupta, "Deep learning based human activity recognition (har) using wearable sensor data," *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100046, 2021.
- [29] H. Bi, M. Perello-Nieto, R. Santos-Rodriguez, and P. Flach, "Human activity recognition based on dynamic active learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 922–934, 2021.
- [30] M. M. Arzani, M. Fathy, A. A. Azirani, and E. Adeli, "Switching structured prediction for simple and complex human activity recognition," *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 5859–5870, 2021.

- [31] Y. A. Andrade-Ambriz, S. Ledesma, M.-A. Ibarra-Manzano, M. I. Oros-Flores, and D.-L. Almanza-Ojeda, "Human activity recognition using temporal convolutional neural network architecture," *Expert Systems with Applications*, vol. 191, p. 116287, 2022.
- [32] C. Hegde, G. Wen, and L. C. Price, "Activity classification using unsupervised domain transfer from body worn sensors," *Smart Health*, vol. 30, p. 100431, 2023.
- [33] E. S. Rahayu, E. M. Yuniarno, I. K. E. Purnama, and M. H. Purnomo, "Human activity classification using deep learning based on 3d motion feature," *Machine Learning with Applications*, vol. 12, p. 100461, 2023.
- [34] C. Pham, S. Nguyen-Thai, H. Tran-Quang, S. Tran, H. Vu, T.-H. Tran, and T.-L. Le, "Senscapsnet: Deep neural network for non-obtrusive sensing based human activity recognition," *IEEE Access*, vol. 8, pp. 86 934–86 946, 2020.
- [35] H. Han, H. Zeng, L. Kuang, X. Han, and H. Xue, "A human activity recognition method based on vision transformer," *Scientific Reports*, vol. 14, no. 1, p. 15310, July 2024.
- [36] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "Harth: A human activity recognition dataset for machine learning," *Sensors (Basel, Switzerland)*, vol. 21, 2021.