University of Miyazaki

Doctoral Dissertation

A Study on Real-time Elderly Monitoring and Behavior Analysis Using Stereo Depth Camera at the Elderly Care Center

June 2024

Interdisciplinary Graduate School of Agriculture and Engineering

Department of Materials and Informatics

YE HTET

宮崎大学大学院

博士学位論文

深度カメラによる高齢者施設等での高齢者の行動解析およびリアルタイムモニタリングに関する研究

2024年6月

宮崎大学大学院農学工学総合研究科

物質・情報工学専攻

イエ　テ

Contents

Chapter 1. Overall Introduction 1

1.1. Research Background 2

1.2. Research Objectives 5

1.3. Main Contributions 5

1.4. Thesis Organization Structure 5

Chapter 2. Literature Review 7

2.1. Elderly Health Supporting Technologies 8

2.1.1. Smart Aging Technologies 8

2.1.2. Indoor Elderly Monitoring Systems 9

2.1.3. Vision Sensor-based Action Recognition Systems 11

2.2. Action Recognition for Elderly Activity Monitoring 12

2.2.1. Transition-aware Action Recognition 12

2.2.2. HMM-based Action Recognition 13

2.2.3. DL-based Action Recognition 15

2.3. Conclusion 19

Chapter 3. Depth Data Acquisition and Processing 20

3.1. Depth Data Acquisition 21

3.1.1. Data Collection Overview 21

3.1.2. Principle of Stereo Depth Cameras 22

3.1.3. Data Collection Protocol 23

3.1.4. Camera Setting and Room Environment 23

3.1.5. Recorded Data Information 25

3.2. Depth Data Processing 25

3.2.1. Depth Preprocessing 25

3.2.2. Depth Image Colorization 26

3.2.2.1. Inverse Colorization 27

3.2.2.2. Depth Image Recovery from Colorization 28

3.3. Visualization of the Elderly’s Common Daily Actions 29

Chapter 4. Person Detection 31

4.1. Introduction 32

4.2. YOLO Detector 32

4.2.1. Model Selection 32

4.2.2. Dataset Preparation 33

4.3. Evaluation Metrics 34

4.4. Comparison Result 34

4.5. Bounding Box Recovery 35

4.6. Conclusion 36

Chapter 5. Action Recognition 37

5.1. Introduction 38

5.2. Evaluation Metrics 38

5.3. HOG Features-based Recognition 41

5.3.1. Feature Extraction 42

5.3.1.1. Depth Motion Appearance 42

5.3.1.2. Depth Motion History 43

5.3.1.3. Histogram of Oriented Gradients 44

5.3.2. Recognition Model 44

5.3.2.1. Support Vector Machine 44

5.3.2.2. Hybrid Hidden Markov Models 45

5.3.2.3. Discussion 57

5.4. Temporal Features-based Recognition 58

5.4.1. Person Segmentation 58

5.4.2. Experimental Results and Discussion 61

5.5. Spatiotemporal Features-based CRNN Integration 62

5.5.1. MotionCRNN Action Recognition Model 63

5.5.2. Experimental Result Analysis 65

5.5.2.1. Dataset Preparation 65

5.5.2.2. Training and Validation Datasets 65

5.5.2.3. Testing Dataset 67

5.5.2.4. Comparison of Experiments 74

5.6. Graphical User Interface (GUI) 78

5.7. Conclusion 80

Chapter 6. Overall Conclusion and Future Research 81

6.1. Research Summary and Realization 82

6.2. Limitations 83

6.3. Future Research 84

References 86

List of Publications 95

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Thi Thi Zin, Graduate School of Engineering, for her invaluable guidance and support throughout my studies. Her expertise and encouragement have been instrumental in shaping this work.

I am also deeply grateful to my co-supervisors, Professor Mitsuhiro Yokota and Professor Hiroki Tamura, Graduate School of Engineering, and Professor Aya Nishiwaki, Graduate School of Agriculture, for their insightful feedback and support during my research.

My sincere appreciation extends to Professor Pyke Tin, my advisor, who is a Distinguished Professor with the Graduate School of Engineering, and a Visiting Professor with the International Relations Center, for his guidance and encouragement throughout my academic journey.

I am particularly grateful to Dr. Etsuo Chosa, Director of the University of Miyazaki Hospital, for his collaboration and contributions to this research. Furthermore, I would like to express my appreciation to Dr. Kazuhiro Kondo, former Director, and Dr. Shinji Watanabe, current Director of Miyazaki City Tano Hospital, for their invaluable advice and support in collecting the data from elder care facilities in real-world settings. Their support was contributory in making this research possible.

I would also like to express my sincere appreciation to Professor Kunihito Yamamori, Faculty of Engineering, a member of the dissertation committee, for his invaluable advice and suggestions that significantly improved the quality of my thesis.

My deepest gratitude goes to the Support for Pioneering Research Initiated by the Next Generation (SPRING) Program by the Japan Science and Technology Agency (JST), the IUCHI Foundation, Japan Student Services Organization (JASSO), and Japan Educational Exchanges and Services (JEES) for their generous scholarship support, which made my study in Japan.

I would also like to express my gratitude to the University of Miyazaki and its helpful professors and staff for providing a stimulating academic environment.

Besides, I would like to express my appreciation to all the members of my laboratory for their cooperation and support during my time as a student.

This work would not have been possible without the support of all the individuals mentioned above.

Last but not least, I would like to express my deepest gratitude to my family, my parents, and my brother, for their unwavering love, and constant support throughout my academic journey. This work is dedicated to them.

Preface

The global rise in the elderly population is posing challenges to healthcare systems due to labor shortages in caregiving facilities. This necessitates innovative solutions for elderly care services. Smart aging technologies such as robotic companions and digital home gadgets offer a solution by improving the elderly's quality of life and assisting caregivers. However, limitations in data privacy, real-time processing, and reliability often hinder the effectiveness of existing technologies. Among these, privacy concerns are a major barrier to ensuring user trust and ethical implementation. Therefore, this study proposes a more effective approach for smart aging that prioritizes data privacy and real-time processing capability. Our main goal is to support the elderly's well-being and assist their caregivers. To achieve this, we developed an activity monitoring and behavior analysis system for the elderly and designed a user-friendly interface for caregivers.

The proposed methodology, with the primary objective of implementing a real-time privacy-preserving activity monitoring system for elderly people, involves a visual monitoring process utilizing stereo depth cameras to continuously analyze the activities of the residents. Data were collected from real-world environments with the participation of elderly individuals. This study focuses on analyzing common daily actions of the elderly, including sitting, standing, lying, and using a wheelchair. Given the vulnerability of elderly individuals, we also focus on transition states (in-between actions such as changing from sitting to standing), which are crucial for assessing balance issues and potential risks.

This thesis is organized as follows:

In **Chapter 1**, the overall research background and the objective of the study are established. This thesis presents a deep learning-based monitoring system for the elderly that can recognize not only the common daily activities of the elderly but also the transition states between actions which is an important factor for reducing the risk of indoor incidents. This chapter addresses the overall introduction of this thesis. The objectives, contributions, and overall flow of this thesis are also described in this chapter.

In **Chapter 2**, the relevant research areas are delved deeper into, reviewing existing smart aging technologies, exploring indoor elderly monitoring systems utilizing sensors and cameras, and examining prior research on elderly action recognition. Additionally, the chapter presents a comparative analysis of various action recognition techniques, including transition-aware action recognition, action recognition based on the Hidden Markov Model (HMM), and Deep Learning (DL)-based approaches.

Preface

In **Chapter** 3, two key aspects are encompassed: depth data acquisition and subsequent processing. The first section details how real-world data was collected from the elderly care facilities. The focus lies on capturing the daily routines of elderly participants while ensuring their privacy. The second section elaborates on the steps to refine the raw depth data captured by the stereo depth cameras. This process aims to enhance data quality by addressing noise, inconsistencies, and missing values within the data.

In **Chapter 4**, the essential role of person detection in elderly monitoring systems is explored using computer vision techniques. It delves into the You Only Look Once (YOLOv5) detector and compares the model's performance. The evaluation process defines various metrics and utilizes two data-splitting strategies to identify the effectiveness of the YOLOv5 model for our specific application.

In **Chapter 5**, the proposed models for elderly action recognition are explored. It explains how these models utilize spatial and temporal features extracted from the person's movement with three main approaches. In the first approach, motion appearance and history features are extracted from the depth image sequences and represented using a Histogram of Oriented Gradients (HOG) descriptor. These HOG feature vectors are classified using single Machine Learning (ML) algorithms and those combined with the stochastic Hidden Markov Model (HMM) in the recognition process. In the second approach, straightforward temporal-dependent features are extracted from the sequence of segmented person masks, and a Support Vector Machine (SVM) is used for classification. In the third approach, spatiotemporal features are extracted automatically using Convolutional Recurrent Neural Networks (CRNN). The system achieved robust transition state recognition by leveraging the motion information derived from body posture changes (inspired by the second approach) with CRNN.

In **Chapter 6**, the proposed research is summarized, the overall effectiveness of the proposed system is described, and finally concluded by outlining its potential contributions and future directions.

Keywords: *smart aging, elderly care, elderly activity monitoring, stereo depth cameras, person detection, real-time elderly action recognition, transition state recognition, artificial intelligence, computer vision, deep learning, machine learning, motion information, temporal-dependent features, spatiotemporal features, YOLOv5, hidden markov model, histogram of oriented gradients, support vector machine, GUI*

List of Figures

1.1 Research Concepts of Smart Elderly Monitoring 3

1.2 Illustration of Transition States between Sitting and Standing Actions 4

1.3 Overall System Flow 6

3.1 Overview of Data Collection Process 21

3.2 Stereo Depth Cameras used in the Proposed Study 22

3.3 Output Images from the Depth Camera and the Data Storage Format 22

3.4 Illustration of Room Environment at the Care Center 24

3.5 Illustration of Room Environment at the Hospital 24

3.6 Depth Preprocessing Pipeline 26

3.7 Hue Color Bar (Represented with Degree Values) 27

3.8 Sample Hue Colorized Image 28

3.9 Comparison of Original Depth Image and Recovery Depth Image 29

3.10 Comparison of Raw Depth Image and Preprocessed Colorized Image 29

3.11 Visualization of Each Daily Action of the Elderly Person 30

4.1 Illustration of Rectangular Boxes around an Elderly Person 32

4.2 Principle of YOLOv5 Detector 33

4.3 Data Splitting Strategies 33

4.4 Illustration of Bounding Box Recovery Process 36

5.1 Flowchart of Action Labelling 39

5.2 Sample Evaluation for “Transition State” Label 40

5.3 Overview of HOG Features-based Action Recognition 41

5.4 Person Silhouette Cropping, Resizing, and Depth Recovery 42

5.5 Architecture of Feature Extraction 43

5.6 HMM Model Structure 46

5.7 Heatmap Visualization after Training with Baum-Welch Algorithm 49

5.8 HMM Prediction on Testing Image Sequence 50

5.9 Examples of the Common False Action Recognition 57

5.10 Overview of Temporal Features-based Action Recognition 58

5.11 Proposed Person Detection and Mask Extraction 59

5.12 Binarized Padding Image with a Fixed Size 59

5.13 Histogram of Transition Duration 60

5.14 Comparison of the Transition State and Primitive Action 60

5.15 Overview of Spatiotemporal Features-based CRNN Integration 62

5.16 Transition State Recognition using CRNN Architectures 62

5.17 Process of MotionCRNN Action Recognition Model 63

5.18 Calculation of Motion Image 63

5.19 CRNN Architecture 64

5.20 Confusion Matrix for Training and Validation Datasets 66

5.21 Visual Representation of a Sample Action Recognition Result 68

5.22 Illustration of Majority Voting Decisions 70

5.23 Visualization of Action Recognition with Refinements 71

5.24 Confusion Matrix for Testing Sequences 73

5.25 Graphical User Interface 79

6.1 Future Research 85

List of Tables

3.1 Details of Data Collection Protocol 23

3.2 Details of Specific Data Information 25

4.1 Evaluation Metrics for Person Detection 34

4.2 Experimental Results for YOLOv5l Model 35

5.1 Evaluation Metrics for Action Recognition (for “Transition State” Label) 39

5.2 Experimental Results for SVM Recognition 45

5.3 Training Datasets for Calculating HMM Emission Probability Distribution *B* 47

5.4 Confusion Matrix of HMM Prediction for Training Dataset (MeanHOG + HMM) 50

5.5 Confusion Matrix of HMM Prediction for Training Dataset (a) k-NN + HMM, (b) SVM + HMM 53

5.6 Comparison of Three Methods 54

5.7 Accuracy for Each Specific Action Testing with the (SVM + HMM) Method 54

5.8 Accuracy for Each Testing Sequence (SVM + HMM) 55

5.9 Comparison between the Proposed Methods and the Previous Works 56

5.10 Performance Evaluation 61

5.11 Performance Evaluation on Training and Validation Datasets 66

5.12 Testing Dataset Information 67

5.13 Performance Evaluation on Testing Dataset 72

5.14 Transition State Recognition Performance 73

5.15 Impact of Refinements on Transition State Recognition 74

5.16 Overall Accuracy and Processing Time Comparison (EfficientNet Variants) 75

5.17 System Comparison 77

5.18 Performance Evaluation on Extended Dataset 78

Abbreviation Table in Alphabetical Order

Abbreviation	Full Term
4IR	Fourth Industrial Revolution
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
CSV	Comma-Separated Value
CV	Computer Vision
DB-LSTM	Deep Bidirectional LSTM
DIP	Digital Image Processing
DL	Deep Learning
DMA	Depth Motion Appearance
DMH	Depth Motion History
FN	False Negative
FP	False Positive
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
HMM	Hidden Markov Model
IoT	Internet of Things
IoU	Intersection over Union
k-NN	k-Nearest Neighbors
LOOCV	Leave-One-Out Cross-Validation
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
MotionCRNN	Motion-based Convolutional Recurrent Neural Network
MSE	Mean Squared Error
ML	Machine Learning
RGBD	Red, Green, Blue plus Depth
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ReT	Recurrent Transformer Neural Network
R-CNN	Regions with Convolutional Neural Network
SAM	Segment Anything Model
STD-TA	Standard Deviation Trend Analysis
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
ViT	Vision Transformer Neural Network
YOLO	You Look Only Once

Chapter 1

Overall Introduction

This thesis presents a deep learning-based video monitoring system for the elderly that can recognize not only the common daily activities of the elderly but also the transition states between actions which is an important factor for reducing the risk of indoor accidents. This proposed system analyzes the motion patterns of the human region extracted from the depth camera images and is intended for implementing the user-demanded vision-based elderly monitoring system. This chapter addresses the overall introduction of this thesis. The objectives, contributions, and overall flow of this thesis are also described in this chapter.

1.1 Research Background

Nowadays, the increase in the aging population has become a huge global issue for human society. The global population aged 65 and above is rapidly growing in many of the developed and developing countries around the world, which places significant demand for healthcare and nursing care services [1]. The improved modern medical technology is one of the key drivers for increasing the life expectancy at birth, and improved survival rates among the elderly. According to the 2019 Revision of the World Population Prospects which has been published by the United Nations [2], one in six people (16%) will be over age 65 by 2050, up from one in eleven (9%) in 2019. Therefore, the modern innovation technology for creating a more age-friendly world becomes an urgent and high-demand requirement. In addition, the global community should take specific actions to improve the health and well-being of the elderly and to develop supportive environments with independent living styles.

With the current rates of aging in the population, the support in health care services for the elderly is an increasing concern. Declining mobility and health are common issues associated with aging that significantly affect the quality of life and independence of the elderly [3]. Due to low fertility rates in recent decades, most of the elderly have only a couple of family members and usually plan to go to the elderly care centers or independently live alone. Consequently, the elder care facilities play a vital role in ensuring the safety of the elderly in case the elderly reach a situation in which they cannot live on their own. Luckily, due to the numerous assistive technologies, now we can implement various kinds of healthcare systems for supporting the elderly associated with growing older.

One promising solution involves understanding the well-being of the elderly through the concept of ‘smart aging’ [4]. Smart aging can be defined as an innovative approach that enables the elderly population to live freely, securely, comfortably, healthily, and happily [5]. Although there are various ways to facilitate smart aging for the elderly, the utilization of modern technologies has increased in recent years, by leveraging advanced software and hardware technologies. Assisted living [6, 7] and healthcare monitoring [8] are among the approaches that are aimed at helping elderly individuals with independent living and smarter aging.

Some of the research concepts in the context of smart aging for elderly monitoring are shown in **Fig. 1.1**. There are two main approaches for smart elderly monitoring technologies: sensor-based and camera-based methods ranging from facial recognition to location tracking. However, existing sensor-based technologies often rely on physical sensors that need to be placed in the environment or require intrusive wearing, which can be inconvenient and limit the mobility of elderly individuals. On the other hand, privacy concerns also arise with certain camera-based monitoring methods to ensure user trust and ethical implementation. Hence, this study proposes a more effective approach to smart aging through indoor activity monitoring of the elderly by pushing the boundaries of these existing limitations.

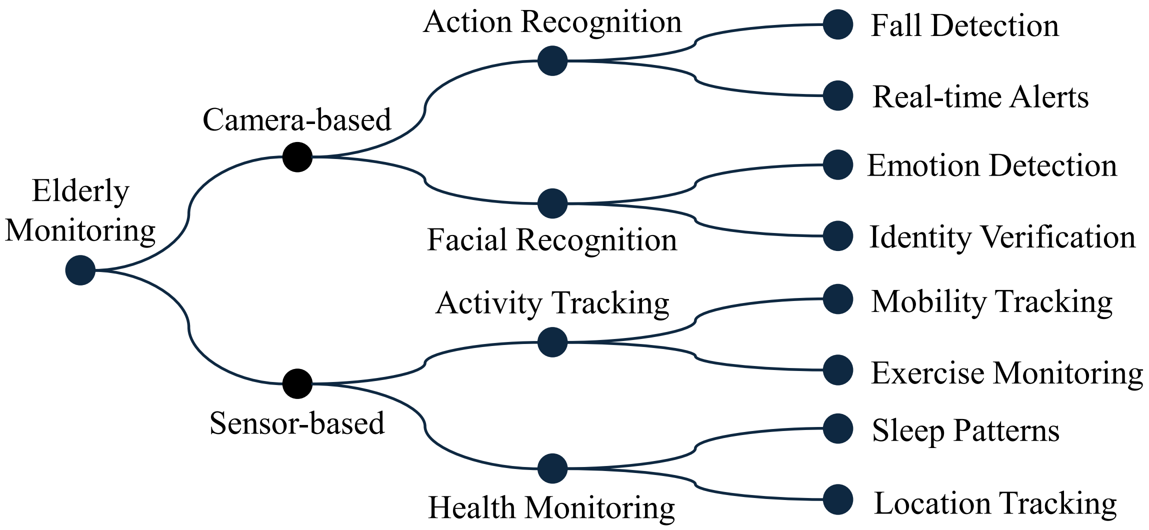


Fig. 1.1. Research Concepts of Smart Elderly Monitoring

This study aims to develop an activity monitoring system for the elderly in indoor settings using stereo depth cameras. To achieve robust performance, modern technologies such as Deep Learning (DL), Machine Learning (ML), and Artificial Intelligence (AI), a cornerstone of the Fourth Industrial Revolution (4IR or Industry 4.0), are applied in this study. Notably, the DL approach is particularly well-suited to this study in comparison with the Internet of Things (IoT). DL architectures, with their multi-layered structure, are adept at handling complex relationships within the raw data, leading to more accurate recognition of the intended actions. On the other hand, IoT generally focuses on connecting devices and sensors for collecting and sharing data. Although depth cameras can be integrated into IoT systems, real-time action recognition often requires additional processing and analysis. Therefore, the proposed system offers several advantages over other existing traditional methods. It eliminates the need for wearable devices or sensors that may interfere with the elderly by allowing easy camera installation within the room, which is cost-effective, can achieve robust performance, and preserves privacy by utilizing depth data rather than color images.

To ensure the practicality of the system, data were collected from real-world environments, including a care center and hospital, with the participation of elderly individuals.

This study focuses on recognizing the seven common daily indoor actions such as seated in a wheelchair, standing, sitting on the bed, lying on the bed, and the transition states between these actions, being outside the room and receiving assistance. Among them, transition states can denote the changes in body position and movement from one specific action to another (e.g., changing from sitting to standing), and are crucial elements for the daily monitoring routines of the elderly. During transition states, elderly individuals may experience feelings of exhaustion due to the need for body balance or may be concerned about falling due to weakened physical conditions as illustrated in **Fig. 1.2**. Hence, recognizing the transition states is important for the elderly health monitoring and can address the challenges associated with impaired mobility and balance, thereby promoting the overall safety of elderly individuals.

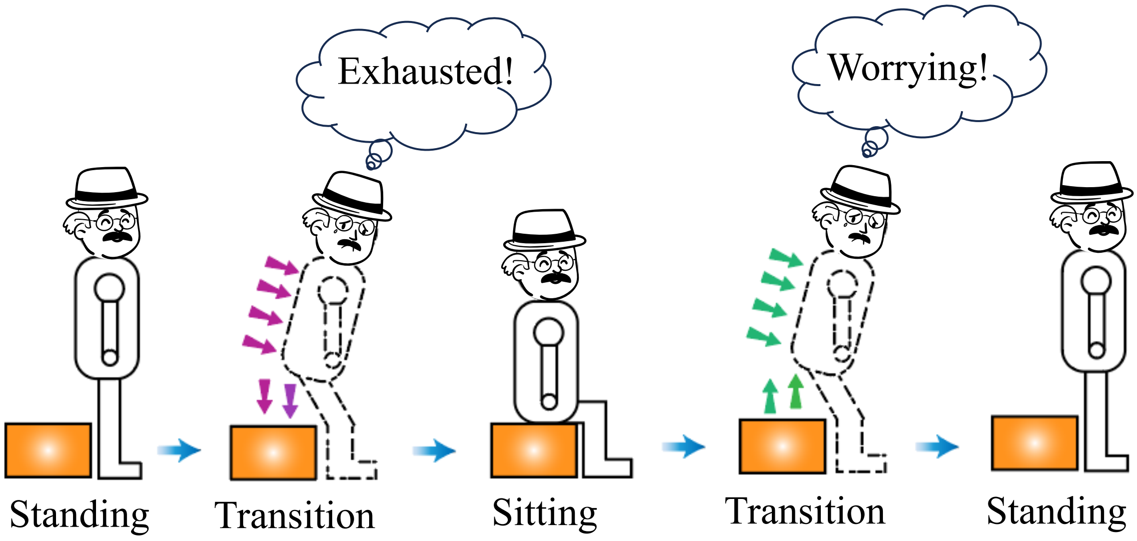


Fig. 1.2. Illustration of Transition States between Sitting and Standing Actions

In caregiving and assisting facilities, the elderly typically rely on the caregivers for continuous support and monitoring. However, caregivers are unable to provide constant monitoring because there is an unbalance between the ratio of the elderly and caregivers. The proposed system aims to reduce the workload of caregivers while supporting the automatic monitoring of the elderly's well-being. Therefore, prioritizing the interaction between the system and caregivers is more important than interaction with the elderly. To achieve this, a user-friendly Graphical User Interface (GUI) was designed and implemented in the proposed system to assist caregivers and provide a convenient environment for the elderly and seniors. The results obtained from the analysis of the proposed model will be described on the GUI and can be shared with caregivers, family members, and healthcare providers, enabling comprehensive monitoring and potentially leading to early interventions.

1.2 Research Objectives

The main objective of this study is to support the well-being of the elderly and assist the caregivers in reducing both the mental and physical load. To achieve this, this study developed an activity monitoring and behavior analysis system for the elderly by utilizing stereo depth cameras and designed a user-friendly interface for the caregivers.

Along with the implementation of this research, the following cutting-edge technologies have been applied:

(1) AI and ML techniques emphasizing DL algorithms,

(2) Advanced Digital Image Processing (DIP) and Computer Vision (CV) techniques,

(3) Hidden Markov Models (HMM) and sequential analysis.

Moreover, the reliability of this research is validated by using self-collected real-life data in real-world scenarios.

1.3 Main Contributions

The main contributions of the study are as follows:

(1) **Depth cameras for elderly monitoring**: Explore the application of stereo depth cameras for privacy-preserving, real-time indoor action recognition for the elderly in the environment.

(2) **Transition state recognition**: Identify transition states from primitive actions of elderly residents using spatiotemporal features.

(3) **Hybrid HMM combinations**: Assess how well combining HMM and ML models classify actions in real time for continuous monitoring.

(4) **Convolutional Recurrent Neural Network (CRNN) integration**: Leverage motion information derived from the body posture changes with CRNN, achieving robust transition state recognition.

(5) **Validate reliability**: Evaluate system reliability using real-world elderly datasets.

1.4 Thesis Organization Structure

The overall system flow is shown in **Fig. 1.3**. There are three main parts included in the proposed system: depth data acquisition and processing, person detection, and action recognition. The data are collected using stereo depth cameras and the results are displayed in the user interface. This thesis is organized according to the overall system flow.

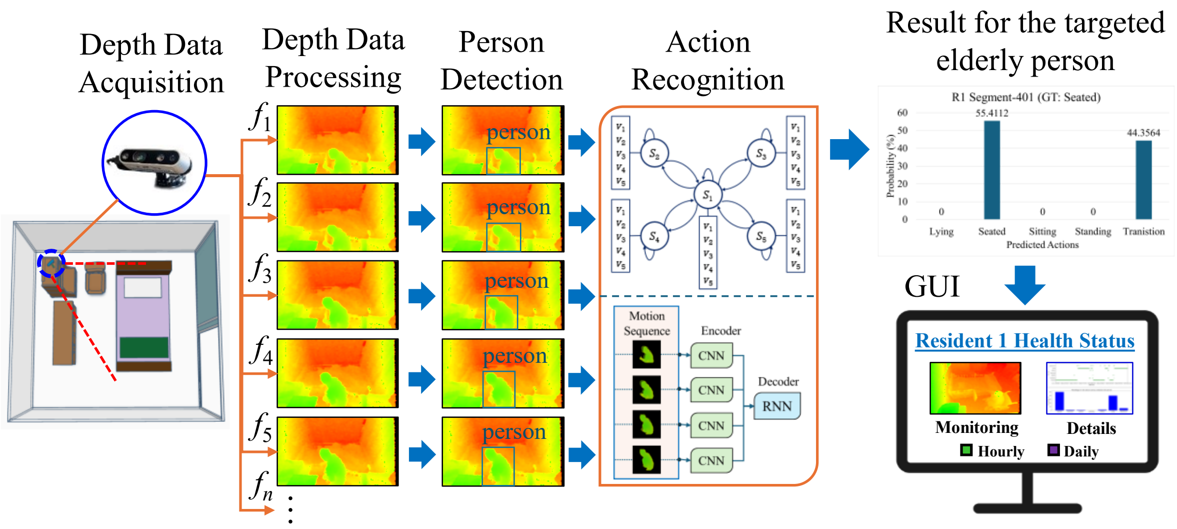


Fig. 1.3. Overall System Flow

Firstly, **Chapter 1** establishes the overall research background and the objective of the study. Then, **Chapter 2** delves deeper into relevant research areas, reviewing existing smart aging technologies, exploring indoor elderly monitoring systems utilizing various sensors and cameras, and examining prior research on elderly action recognition. After that, **Chapter 3** encompasses two key aspects: depth data acquisition and subsequent processing which are crucial for analyzing the activities of elderly individuals. The first sub-section presented detailed information on how real-world data was collected from the elderly care facilities. The second sub-section elaborates on the steps taken to refine the raw depth data captured by the depth cameras. Subsequently, **Chapter 4** explores the essential role of person detection in elderly monitoring systems using computer vision techniques. It delves into the You Only Look Once (YOLO) detector and compares the model's performance. Then, **Chapter 5** explores the proposed models for elderly action recognition and explains how these models utilize spatial and temporal features extracted from the person's movement with three main approaches. Finally, **Chapter 6** summarizes the research, discusses the overall effectiveness and limitations of the proposed system, and concludes by outlining its potential contributions and future directions.

Chapter 2

Literature Review

This chapter conducts a comprehensive literature review of elderly health supporting technologies. It explores existing smart aging technologies, delves into indoor elderly monitoring systems using various sensors and cameras, and examines prior research on elderly action recognition. Additionally, the chapter presents a comparative analysis of various action recognition techniques, including transition-aware action recognition (Section 2.2.1), action recognition based on the Hidden Markov Model (HMM) (Section 2.2.2), and Deep Learning (DL)-based approaches (Section 2.2.3).

2.1 Elderly Health Supporting Technologies

As the population ages, there is a growing need for technologies that can support independent living for older adults [9]. Many modern assistive technologies, such as ambient assisted living systems and smart homes, incorporate action recognition to improve elderly care [10]. Action recognition allows these systems to monitor and analyze the daily activities of elderly individuals, enabling features such as prompting and warning systems, health monitoring, and support for people with dementia [11]. Researchers in industry and academia have built numerous systems for the elderly using wearable sensors (accelerometers and gyroscopic sensors) [12], ambient sensors (motion, radar, object pressure, and floor vibration sensors) [13], and vision sensors [14]. The proposed system relies on vision sensors (cameras) rather than wearable sensors to maximize the comfort of those living in the care center. The following subsection explores recent trends in smart aging technologies, with a particular focus on how these trends are utilized in vision-based action recognition systems for indoor elderly monitoring.

2.1.1 Smart Aging Technologies

Smart aging technologies offer a wide range of innovative solutions to support elderly people in their daily lives and promote aging. These solutions encompass smart home products, gadgets, wearable devices, remote monitoring systems, and Internet of Things (IoT)-enabled healthcare applications [15-17]. These include functions such as fall detection, electronic fences, temperature monitoring, and sleep monitoring. For example, a smart wearable device based on IoT has been designed to monitor physiological parameters in real time and provide remote access to the elderly's health status [18]. On the other hand, public entities deploy and operate smart mobility technologies to improve mobility and independence for older adults, while reducing operating costs [19]. Similarly, smart grid technology has been developed to provide useful information on the activities of daily living and monitor the short and long-term health of elderly individuals [20]. Owing to advancements in technology, Artificial Intelligence (AI) has played a crucial role in developing smart aging systems to personalize healthcare for the elderly. For instance, AI tools such as Machine Learning (ML) and DL models are used to develop solutions that improve the quality of life and autonomy and reduce caregiver burden [21-24].

However, challenges arise in the implementation of personalized healthcare using smart aging technologies which include the potential disruption of existing care systems, technological literacy gaps, and privacy concerns due to constant monitoring [25-27]. Moreover, the security vulnerabilities in IoT systems [28] and ethical considerations in AI must be addressed carefully. For instance, co-adaptation between technology and the elderly is crucial for user satisfaction and long-term adoption [29]. Therefore, a person-centered approach and sufficient governance are necessary to ensure generalizability, transparency, and effectiveness in implementing smart aging technologies.

Overall, smart aging technologies offer promising solutions for aging and enhancing the well-being of the elderly. Addressing security vulnerabilities, ethical considerations, and implementation challenges is crucial for successful adoption and impact. The future of smart aging technologies is bright, with the potential to revolutionize the way we care for older adults. However, it is important to ensure that these technologies are developed and used in a way that is ethical and respectful of the needs and preferences of older adults.

Motivated by this, this study addresses data privacy concerns in smart aging through indoor elderly activity monitoring using stereo depth cameras. This practical system, developed and evaluated for easy adoption in real-world environments, utilizes data collected from a care center and hospital with the participation of elderly individuals. As an ethical consideration, a waiver of written informed consent was obtained from all participants, and the data acquisition protocol received ethical approval for the experiment. Some related systems for indoor elderly activity monitoring are explained in the next subsection.

2.1.2 Indoor Elderly Monitoring Systems

Elderly monitoring refers to an indoor system designed to process data related to the daily activities of the elderly, collected from sensors or cameras. It provides information concerning health conditions and behavioral status to aid in understanding the well-being of the elderly. A recent study introduced a system for activity monitoring that utilized wearable sensor data and environment-independent fingerprints generated from Wi-Fi channel state information using a hybrid DL model [30]. This system aimed to enhance the independence of the elderly and visually impaired individuals, achieving an accuracy of 99% in experiments conducted on two public datasets featuring various activities. However, sensor-based systems sometimes face challenges, including noisy data affecting accuracy, unreliable readings owing to sensor placements, and the need for sophisticated data collection and processing. Additionally, they often require frequent charging, causing inconvenience for the elderly who may forget to use them.

In contrast, camera-based systems are particularly attractive due to their non-invasive nature, aligning well with the principles of smart aging to promote user comfort and freedom for older adults. Cameras offer a broader field of view, enabling monitoring of multiple activities using one device within a room or area. Importantly, they can serve multiple purposes beyond action recognition, including fall detection, medication monitoring, and remote communication. However, they also present challenges such as privacy concerns and limitations in environments with poor lighting or clutter.

Depth cameras offer several distinct advantages over traditional RGB cameras. Whereas regular cameras capture 2D information, depth cameras provide 3D depth data, revealing the distance between the objects and the camera sensor [31]. Thus, depth data offers privacy advantages because they capture distance information in the form of a 3D point cloud, without recording facial details or other identifiable features. Moreover, depth cameras perform well under low-light conditions, where regular cameras struggle, making them suitable for monitoring various indoor environments with limited lighting. Depth cameras do have limitations such as limited sensing distance and low resolution [32]. Despite these limitations, advancements in the technology originally developed for gaming, automotive, and medical fields have led to their increasing application in elderly care and smart homes.

Several studies have explored the use of depth cameras to monitor the elderly by analyzing their activity patterns [33-36]. For example, a non-invasive sleep monitoring system was developed using a 3D depth camera (Microsoft Kinect II) [33] with the aim of long-term monitoring of sleep behaviors in seniors. Another study utilized depth-video-based methods for human activity recognition in indoor environments [34] and achieved efficient and robust results by experimenting with three publicly available depth datasets. In addition, a framework for fall detection that utilizes both accelerometer data and depth maps from a Kinect sensor was proposed [35], demonstrating high performance in differentiating falls from other daily activities. The experiment was conducted on a public fall detection dataset and achieved a high performance. Furthermore, a solution was proposed that solely utilizes depth information from RGB plus depth (RGBD) cameras to monitor the elderly within indoor living spaces [36], enabling remote monitoring by family members and caregivers to understand their behavior and take appropriate action when needed.

Through a review of previous studies, it is evident that various categories are included for elderly monitoring purposes, such as sleep monitoring, fall detection, remote monitoring, and activity recognition. However, many of these systems rely on public datasets or performance datasets demonstrated by young people rather than testing actual elderly data. In addition, the camera view in most datasets is typically located in front of a person, which may be uncomfortable or impractical in real-world scenarios. By leveraging the advantages of depth cameras and collecting real-world data from elderly residents in care centers, this study proposes a system for 24-hour monitoring and real-time action recognition processing, addressing limitations identified in previous research.

2.1.3 Vision Sensor-based Action Recognition Systems

Building on the advantages of camera-based systems, vision sensor-based action recognition utilizes computer vision and image processing techniques to analyze video sequences and understand a subject's activities. This technology has become a major area of research in recent years. For instance, the following studies were conducted on human action recognition using various types of vision sensors. The authors in [14] designed a monitoring and action recognition system by exploiting modern image processing techniques and RGB cameras. They trained the detection model in their system using the faster Regions with Convolutional Neural Network features (R-CNN) by focusing on the ‘person’ class to locate the person. Again, the action recognition model was trained by the integration of two-stream inflated 3D ConvNet and deep human action recognition models. The authors introduced a new dataset with a large number of samples to balance the action samples and designed a client-side, web-app interface for monitoring people. Another study [37] emphasized a method for real-time human action classification using a single RGB camera, which can also be integrated into a mobile robot platform. To extract skeletal joints from RGB data, the authors combined OpenPose and 3D-baseline libraries and then used a CNN to identify the activities.

As more and more technologies emerge to assist older adults, researchers should consider the effect of health-related technologies on the people being monitored. Most people want to keep their health information private, and they also worry about how such information could be used against them. According to surveys collected by the authors in [38], older adults have positive opinions of assistive technologies but rarely accept systems that use cameras because of privacy concerns. To overcome this attitude, the use of depth cameras became more common for their advantages from a privacy perspective. By measuring distances between the camera and objects, depth data can be used for action recognition without using images that could be used to identify individuals. Furthermore, depth cameras can be used at night without needing additional light. By using color with depth data, some authors have proposed a cloud-based approach [39] that recognizes human activities without compromising privacy. In this approach, researchers collect one motion-history image generated from color data, three depth-motion maps extracted from depth data, and then use deep Convolutional Neural Network (CNN) for the recognition process.

Likewise, we also used a depth camera in our previous work [40], which introduced a real-time action recognition system that helps prevent accidents and supports the well-being of residents in care centers. In [40], we extracted both appearance-based depth features and distance-based features, extending the system described in [41] to recognize actions using the automatic rounding method. As another approach, [42] proposed a skeleton-based system for recognizing human activities for monitoring the elderly. They used Minkowski and cosine distances between 3D joint features for the recognition process, by characterizing the spatiotemporal components of a human activity sequence. The authors of [43] used 3D point clouds for action recognition by only processing depth maps. They developed a descriptor based on the histogram of oriented principal components for 3D action recognition. The researchers used this descriptor to determine the spatiotemporal key points in 3D point cloud sequences. In contrast to previous studies, our method relies on the depth map features of a stereo depth camera, and actions are recognized based on these depth images.

2.2 Action Recognition for Elderly Activity Monitoring

Action recognition for elderly activity monitoring involves identifying both primitive actions (e.g., sitting, standing, seated in the wheelchair, and lying down) and transition states that might indicate potential risks. These transition states could include falls, abnormal activities, or attempts to perform actions that could lead to harm. The following subsections explore different approaches to recognizing actions and transitions using vision sensor data.

2.2.1 Transition-aware Action Recognition

Transitions between actions are often disregarded in traditional action recognition due to their short duration compared with full actions. However, failing to account for transitions can negatively impact the performance of recognition systems [44]. Hence, the real-time detection of transitions between actions remains a challenging but valuable area of research, particularly for continuous monitoring of human daily activities [45, 46].

Several studies have explored various approaches to transition-aware action recognition, demonstrating its effectiveness in real-world scenarios. These approaches often leverage sensor data or video features to identify transition states.

For instance, real-time ML-based methods have been employed for automatic segmentation and recognition of continuous human daily action by integrating change point detection algorithms with smart home action recognition [47, 48]. In another study, a transition-aware context network was proposed [49] to distinguish transition states. The network comprised two components: a temporal context detector to extract long-term context information and a transition-aware classifier to classify actions and transition states. Utilizing spatiotemporal features, the network achieved a competitive performance and significantly outperformed state-of-the-art methods on the untrimmed UCF101 dataset. Moreover, CNN models were utilized to recognize transition actions, and the effectiveness of the approach was demonstrated through experiments with fuzzy logic [50].

Other innovative approaches focus on incorporating realistic human motion into the transition recognition process [51, 52]. For example, one approach emphasizes the importance of natural leg movements during transitions [51]. Another study proposed an algorithm based on Standard Deviation Trend Analysis (STD-TA) of sensor data for recognizing transition states [52]. Additionally, smartphone-based systems have been developed for transition recognition [53].

The related studies mentioned above share a common approach of utilizing time-series or spatiotemporal features to identify transition states from other actions, although they employ different classifier models. Building on these concepts, the system in this study utilizes spatiotemporal features extracted from the body movements of the elderly to distinguish between transition states and primitive actions as well as among specific actions.

2.2.2 HMM-based Action Recognition

HMMs are a type of statistical model that extends the Markov process by including hidden states along with visible states. They are widely used in various detection and recognition systems, particularly for recognizing activities or sequences of events [54].

When applied to action recognition using sensor data, HMMs offer a powerful approach for modeling sequential activity patterns. For instance, the authors in [55] proposed a two-stage continuous HMM approach to recognizing human activities from temporal streams of sensory data (collected by accelerometer and gyroscope on a smartphone). The first level of HMM separated stationary and moving activities, while the second level separated data into their corresponding activity classes. Likewise, other research [56] has employed a two-layer HMM to build an activity recognition model using sensor data, but that differs from the model in the work of [55]. In the first layer, location information obtained from the sensors was used to classify activity groups; in the second layer, individual activities in each group were classified. Then, they applied the Viterbi algorithm to their HMM to infer the activities. The activity recognition model in [57] established a Hierarchical HMM to detect ongoing activity by monitoring a live stream of sensor events. Their method also included two phases, but only the first phase used HMM. In this method, data streams were segmented according to the start and end points of activity patterns.

For systems relying on vision sensor data, the studies in [58, 59] proposed HMM-based automatic fall detection systems with image processing techniques by utilizing RGB and RGB-D cameras, respectively. In [60], HMM was used as a decision-making process for differentiating abnormal (falling) from normal sequential states for a given person. The system made this decision by observing the six possible feature values which were defined according to the distance between the centroid of the person's silhouette and the associated virtual ground point, the shape's area, and the person's aspect ratio. The HMM model was then developed by defining feature thresholds and calculating emission probabilities. On the other hand, [61] created an HMM model to detect and distinguish falling events from the other eight activities of the person. The observation symbols of their model were the vertical position of the center of mass, the vertical speed, and the standard deviation of all the points belonging to the person. In another study [62], a continuous HMM was used for human action recognition from the image data. The authors explicitly modeled the HMM using a temporal correlation between human postures, described using a Histogram of Oriented Gradients (HOG) for shape encoding, and a Histogram of Optical Flow (HOF) for motion encoding. Their HMM made continuous observations, modeling the probability distribution in each state by a mixture of Gaussians. Their experimental results showed that the continuous HMM outperformed recognition systems using a Support Vector Machine (SVM) based on spatiotemporal interest points. In another study [61], an HMM was developed for a human activity recognition system in which the discrete symbols for HMM were generated by mapping into code words from estimated body joint-angle features. The HMM was trained for each activity, and the activities were then recognized using the trained models. Meanwhile, the authors in [62] and [63] had driven the development of Fisherposes for view-invariant action recognition using 3D skeleton data collected using a Kinect sensor. In [62], an HMM was used to characterize the temporal transition between body states in each action, and in [63], an HMM was used to classify actions into an input series of poses.

In the current work, we developed an HMM model for elderly action recognition that uses space-time features to obtain observation symbols. Furthermore, we compared the results of various models which combine HMM with other ML classification models.

2.2.3 DL-based Action Recognition

Research on spatiotemporal feature extraction and action recognition has explored traditional methods [64, 65], which often rely on handcrafted features, and DL techniques [66-71]. DL models, such as CNNs and Recurrent Neural Networks (RNNs), have emerged as powerful tools for action recognition due to their ability to automatically learn complex features directly from data, reducing the need for manual feature engineering. CNNs are adept at capturing spatial features from video frames, while RNNs excel at managing temporal dependencies by processing feature sequences over time. Integrating CNNs and RNNs for spatiotemporal feature extraction offers advantages in terms of accuracy and efficiency, as proven in existing literature. Therefore, the proposed system uses CNNs to encode spatial features and RNNs to decode temporal dependencies. These components were then fused and built into a single-model hybrid architecture for action recognition.

Several studies have investigated the application of CNNs and RNNs in action recognition. For example, one approach proposed recognizing human actions from videos using a combination of deep CNN and multi-layered RNN, specifically Long Short-Term Memory (LSTM) units [66]. CNNs extract features from individual video frames, whereas LSTMs are a type of RNN that can effectively capture long-term dependencies within sequences, making them suitable for processing the sequence of extracted features to capture temporal information. In their approach, different GoogLeNet architectures were used to extract various features from images. The extracted features were then converted into sequences and fed into multi-layered LSTMs. Finally, a softmax regression classifier categorizes the videos based on processed features. Notably, the network architecture utilizes both residual and inception blocks to handle convergence during the training process. Experiments showed that this approach, particularly the combination of multi-layered LSTMs with the Inception_Residual model, improved the evaluation performance.

Another study proposed a novel architecture using CNNs and RNNs for action recognition [67]. The approach incorporated separate layers to capture spatial and temporal information. In the first stage, that is, feature extraction, they utilized an improved p-non-local operation within a deep CNN. This operation effectively captures long-range dependencies within video data. In the second stage, class prediction, they introduced a novel technique called fusion keyless attention. This technique, combined with a forward and backward bidirectional LSTM network, allows the model to learn the sequential nature of the data, that is, how actions unfold over time. Their experiments on two datasets demonstrated that this model outperformed the traditional models.

To improve action recognition performance, researchers have explored transfer learning by leveraging pre-trained models that have already learned powerful feature representations from large datasets [68]. Their approach utilized two separate CNNs, one for analyzing spatial information from RGB images and another for capturing motion information through optical flow. Both CNNs leveraged pre-trained models for efficient feature extraction. They further investigated combining the spatial and temporal features extracted by separate CNNs. This involved employing various CNN-RNN architectures, where CNNs (ResNet101, GoogleNet, and VGG16) act as encoders to extract features and RNN variants (LSTM, Bi-directional LSTM, Gated Recurrent Unit (GRU), and Bi-directional GRU) act as decoders to handle the sequential nature of video data. The researchers proposed six additional aggregation networks after generating the individual models (one motion CNN model, three spatial CNN models, and twelve CNN-RNN fusion models). These networks used a technique called Average Fusion to combine the outputs from the spatial and temporal CNNs, as well as CNN-RNNs. This was aimed at further improving the overall action recognition performance.

Another approach utilized a Deep Bidirectional LSTM (DB-LSTM) network for action recognition in long videos [69]. The method combines a CNN for feature extraction and a DB-LSTM to handle the sequential nature of video data. To reduce computational complexity and capture representative motion patterns, the approach extracts spatial features from every sixth frame of the video using a pre-trained CNN model (AlexNet). A deep DB-LSTM network then processes the extracted features. By stacking multiple layers in both the forward and backward directions, the DB-LSTM learns long-term dependencies within the video sequence, making it suitable for analyzing longer videos. Experiments showed that this approach achieved state-of-the-art performance on the UCF-101, HMDB51, and YouTube action video datasets, outperforming other recent techniques.

Recent research has addressed the limitations of DL-based action recognition, particularly regarding computational efficiency, scalability, and accuracy for real-time applications. One promising approach involves lightweight architectures and transformer neural networks. Transformer neural networks, a relatively new DL architecture, can process sequences directly, without relying on recurrent connections, potentially reducing computational complexity. Additionally, they can effectively capture long-range dependencies within video data, potentially improving recognition accuracy. These techniques aim to address challenges such as high computational demands by offering reduced model size and faster processing. For example, a recent study proposed Vision and Recurrent Transformer Neural Networks (ViT-ReT) for human action recognition in videos [70]. The framework combined a Vision Transformer (ViT) for efficient feature extraction and a Recurrent Transformer (ReT) to model the temporal information within a video sequence. Researchers compared ViT-ReT with traditional CNN and RNN-based approaches on several benchmark datasets. Their findings demonstrated that ViT-ReT achieved a significant speedup compared with the baseline method (ResNet50-LSTM) while maintaining comparable accuracy. Furthermore, ViT-ReT outperformed the state-of-the-art methods in terms of both accuracy and processing speed, making it suitable for resource-constrained and real-time activity recognition applications.

To address the challenges of real-time action recognition, the proposed system prioritizes efficiency while maintaining accuracy. While previous approaches have explored combining CNNs and RNNs for action recognition by adjusting parameters and scaling architectures, these models can be computationally expensive for real-time applications, especially when processing spatial features from every frame in a long video sequence. The proposed system aims to achieve real-time performance through two key strategies. First, the person in the image is segmented, and then spatial features are extracted from this segmented region rather than from the entire image, significantly reducing processing time. In addition, motion information is incorporated from two consecutive frames for the CNN to extract features, enhancing the model’s capability to capture motion information. Furthermore, the encoder-decoder architecture combining CNN and RNN leverages a lightweight network design inspired by previous works.

In many cases, the results from DL models can benefit from post-processing techniques like majority voting and reasoning, which are crucial for handling uncertainties and inconsistencies in real-world applications. These techniques can help to refine the predictions from DL models and improve their overall accuracy and reliability.

Several related studies have applied majority voting decisions and conditional reasoning to action recognition predictions. For example, a sliding window approach was used in combination with majority voting on skeleton data to achieve online human action recognition using spatiotemporal graph CNNs [72]. In this approach, the video sequence is segmented into overlapping windows, predictions are made for each window. The final action recognition result is determined by applying a majority vote across the predictions from all windows. This approach demonstrates the high performance and efficiency of the majority-voting approach. Similarly, a model was developed to predict four different actions using majority voting for gameplay [73]. The findings indicated that majority voting yielded more accurate predictions with 92.59% accuracy, exceeding the peak accuracy value of individual pre-trained models. Subsequently, a model blending technique [74] was developed using majority voting in an ensemble of DenseNet-201 and ResNet-50 for melanoma classification. This method displayed satisfactory results, demonstrating the influence of majority voting decisions.

Regarding reasoning, one study [75] improved the performance of action recognition by modeling causal relationships based on preconditions and effects. The suggested cycle-reasoning model demonstrated improved action recognition performance through efficient reasoning about preconditions and effects. Additionally, an action reasoning framework [76] that uses prior knowledge was proposed to explain the semantic-level observations of video state changes. The experimental results indicated an improvement in recognition using this reasoning approach.

Motivated by the effectiveness of majority voting and reasoning in refining action recognition results, the proposed approach incorporates both techniques. Specifically, it utilizes majority voting to refine the prediction results and a specific type of conditional reasoning to address the potential over-segmentation of transition states. Unlike previous works that applied majority voting on individual frames, the proposed method leverages sequential-based majority voting decisions and reasoning to reduce over-segmentation in transition states. This approach aims to enhance the accuracy and robustness of the system for effective recognition of transition states.

2.3 Conclusion

This chapter focused on a literature review of action recognition techniques for elderly health supporting technologies, providing a foundation for the approaches to be presented in the next chapters. It began by reviewing traditional methods based on HMMs for both vision sensor data and skeletal data. It then discussed the advantages of DL for action recognition, highlighting its ability to automatically learn complex features from video data. CNNs were identified as effective for capturing spatial features from video frames, while RNNs were shown to be adept at managing temporal dependencies within video sequences. Finally, the chapter discussed how combining deep learning models with reasoning techniques like majority voting and conditional reasoning can improve the robustness and accuracy of action recognition systems.

Building upon this foundation, the proposed action recognition process will delve into the first approach: hybrid HMMs combined with ML classifiers. This will explore how HMMs can be integrated with various ML classifiers to leverage the strengths of both techniques. Then, another approach will be presented by focusing on leveraging motion information with Convolutional Recurrent Neural Networks (CRNNs) for action recognition. This will detail the architecture of our CRNN model and how it extracts motion features to improve recognition accuracy.

Chapter 3

Depth Data Acquisition and Processing

This chapter delves into the process of acquiring and processing depth data crucial for analyzing the daily activities of elderly individuals. It encompasses two key aspects: depth data acquisition and subsequent processing. The first section details how real-world data was collected from the elderly care facilities. Here, the focus lies on capturing the daily routines of elderly participants while ensuring their privacy. The second section elaborates on the steps taken to refine the raw depth data captured by the stereo cameras. This processing aims to enhance data quality by addressing inherent noise, inconsistencies, and missing values within the data.

3.1 Depth Data Acquisition

This section describes the process of acquiring depth data for the development of our system. Real-world data from an elderly care center and a hospital were utilized, focusing on capturing the daily activities of elderly participants. Stereo depth cameras were employed to ensure user privacy by solely recording depth information. Details regarding data collection procedures, camera settings, room environments at each location, and the specific information of the recorded data are provided in the following subsections.

3.1.1 Data Collection Overview

Real-world data were collected from the actual environment of the elderly care facilities in Miyazaki City, Miyazaki Prefecture, Japan. To ensure user privacy, we utilized stereo depth cameras instead of RGB color cameras that could reveal participants' identities. Depth images, which provide distance information between objects and the sensor, were used throughout the development process. This resilience to lighting variations ensures consistent and reliable action recognition, regardless of shadows or color changes. To enhance visualization for subsequent analysis, the retrieved depth images were colorized using the hue colorization method, which will be explained in **Section 3.2.2**. **Fig. 3.1** illustrates the overview of the data collection process, including a sample depth image.

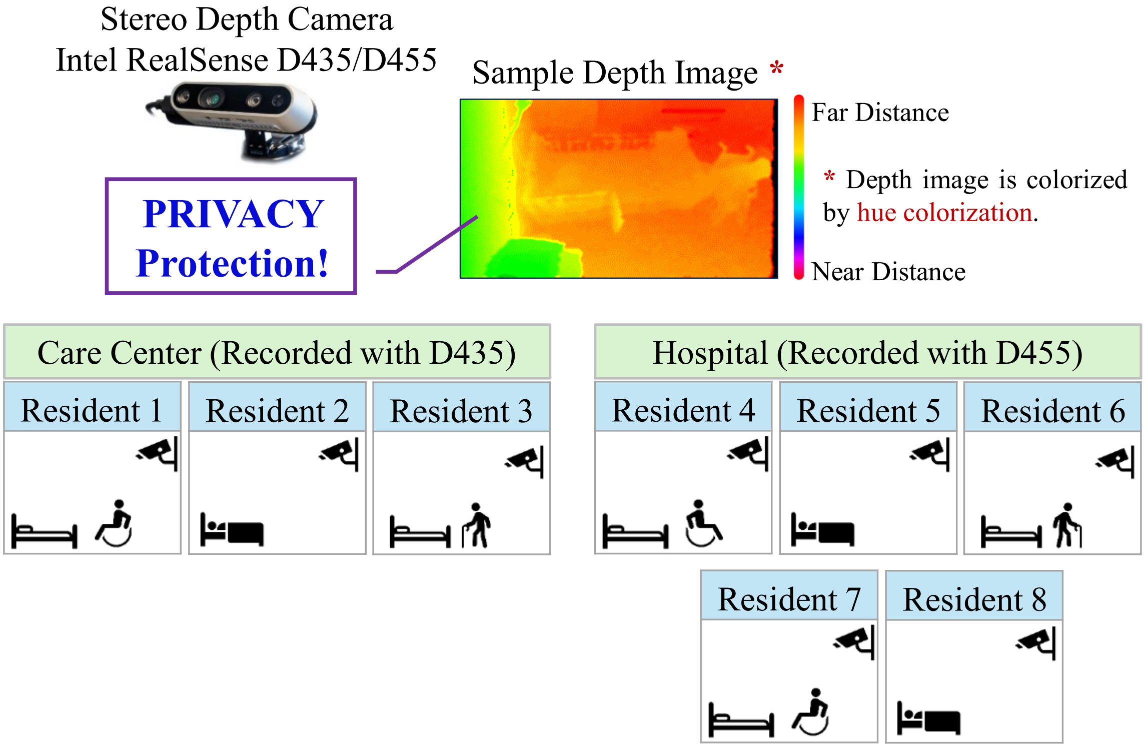
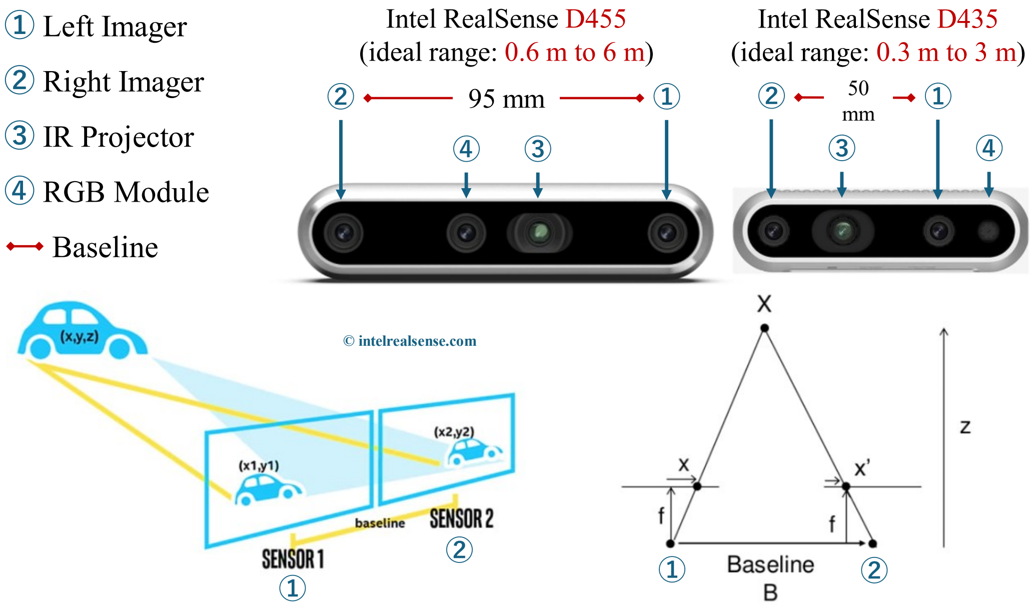


Fig. 3.1. Overview of Data Collection Process

3.1.2 Principle of Stereo Depth Cameras

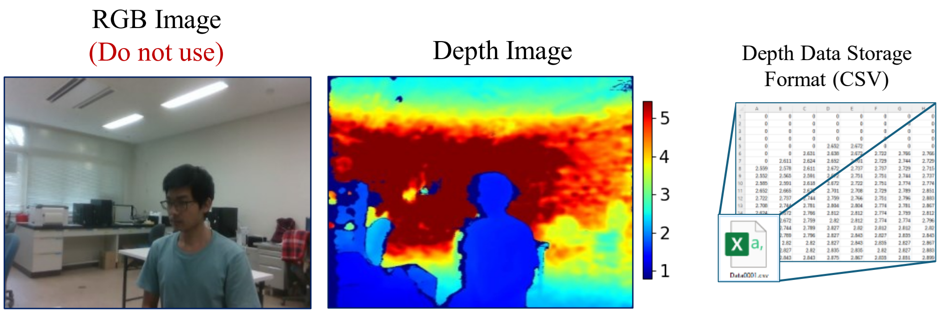
Stereo depth cameras, such as the Intel RealSense D435 (baseline 50 mm) used at the care center and the D455 (baseline 95 mm) used at the hospital, capture depth information using two imagers, one for the left and right viewpoint. The distance between these imagers, called the baseline as shown in **Fig. 3.2**, determines the camera's depth range. A larger baseline allows for capturing objects at greater distances. In our case, the D455 offers a range of 0.6 m to 6 m, while the D435 covers 0.3 m to 3 m.

The cameras utilize the baseline, focal length, and disparity values to calculate depth values for each pixel in the captured image. This information is initially stored as raw floating-point data representing the distance from the camera to the object. For easier data handling, these values are then converted and saved in a Comma-Separated Value (CSV) file format, as shown in **Fig. 3.3**.



© intelrealsense.com

Fig. 3.2. Stereo Depth Cameras used in the Proposed Study



❖ Both images are captured by the D435 camera.

Fig. 3.3. Output Images from the Depth Camera and the Data Storage Format

3.1.3 Data Collection Protocol

Eight elderly residents (three from the care center and five from the hospital), each in a separate room, participated in our study. Detailed information is provided in **Table 3.1**. All participants were over 65 years old and diagnosed with cognitive decline or frailty.

To ensure informed consent, we followed ethical protocols approved by the Ethics Committee of the University of Miyazaki, Japan (protocol code O-0451, dated January 28, 2019, and protocol code O-1449, dated November 20, 2023). Participants were provided with a thorough explanation of the study and their written consent was obtained. For data collection, stereo depth cameras captured depth information at a resolution of 320×180 pixels and a frame rate of 5fps.

Table 3.1. Details of Data Collection Protocol

Participants	1) Three elderly residents from the Care Center, 2) Five elderly residents from the Hospital
Selection Criteria	1) Aged older than 65 years, 2) Diagnosed with cognitive decline or frailty, 3) Have been informed about participation, 4) Have voluntarily provided written consent.
Data Information	1) Collected depth images for continuous 24 hours, 2) Recorded with 320×180 pixel resolution at 5fps.
Protocol Code	1) O-0451 on January 28, 2019 (Care Center) 2) O-1449 on November 20, 2023 (Hospital)

3.1.4 Camera Setting and Room Environment

Stereo depth cameras were installed in each participant's room to capture their daily activities. Mini PCs processed and recorded the data, storing it on external HDDs. To prevent accidental interactions with the cameras, they were strategically positioned above the curtain beside the bed and angled downward 45° toward the bed inside the room. The distance from the depth camera to the bed was maintained at 2.5 meters, with the depth camera mounted 2.1 meters above the ground.

As shown in **Fig. 3.4** and **Fig. 3.5**, the core camera setup was consistent across both the care center and hospital, with some variations in camera positions and environments reflected in the sample depth images included in the figures. During the recording, only depth data (distance information) were captured to preserve participants' privacy, with color images intentionally omitted.

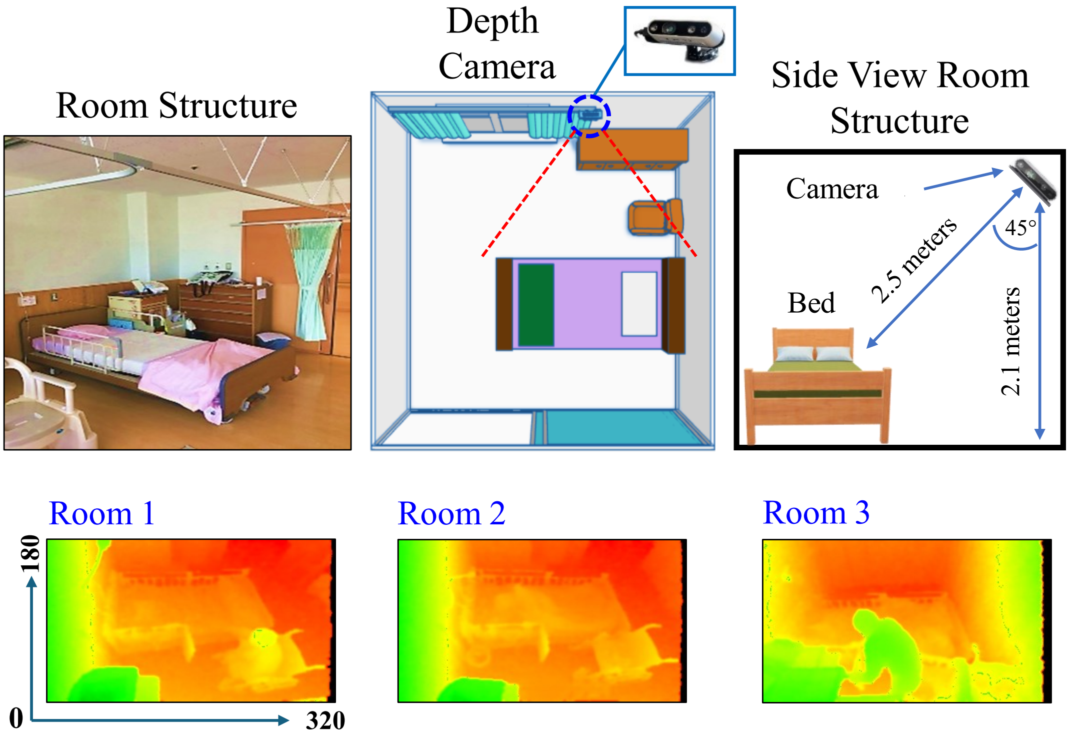


Fig. 3.4. Illustration of Room Environment at the Care Center

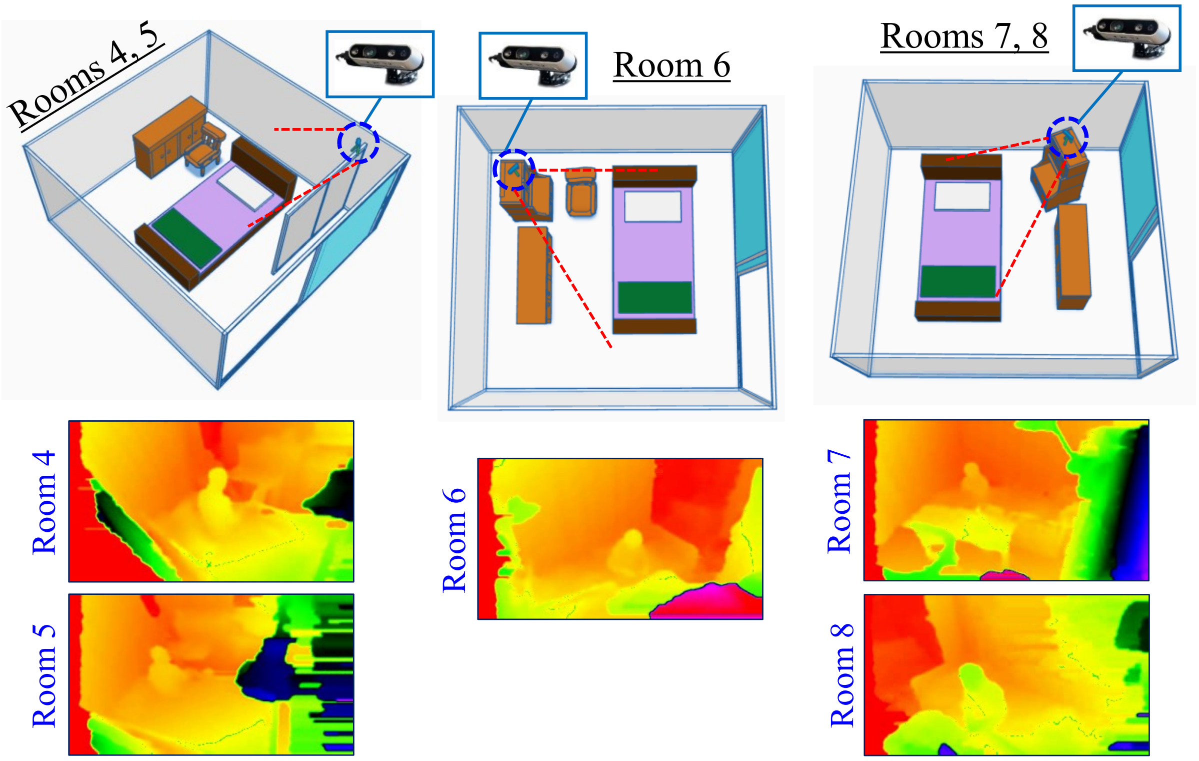


Fig. 3.5. Illustration of Room Environment at the Hospital

3.1.5 Recorded Data Information

Table 3.2 details the specific information regarding the recorded data. It is important to note that the data at the care center were collected in 2019, while the hospital data collection was conducted in 2024.

Table 3.2. Details of Specific Data Information

Room ID		Dates (yyyy/mm/dd_hh:mm)		Duration (hours)
		Start Time	End Time	
Care Center	1	2019/10/12_10:15	2019/10/13_04:14	18
		2019/10/18_11:34	2019/10/24_13:21	146
	2	2019/10/25_11:45	2019/10/28_06:53	67
	3	2019/10/12_11:10	2019/10/13_05:10	18
		2019/10/18_11:25	2019/10/22_20:33	105
		2019/10/25_12:00	2019/10/28_07:22	68
Hospital	4	2024/01/05_15:35	2024/01/09_13:20	94
	5	2024/01/26_14:20	2024/01/29_13:30	71
	6	2024/01/31_14:35	2024/02/07_13:15	167
	7	2024/03/25_16:10	2024/03/28_13:15	69
	8	2024/03/28_15:00	2024/04/01_13:30	94

3.2 Depth Data Processing

This section details the processing steps applied to the raw depth data captured by the stereo depth cameras. The primary goal of this processing is to improve data quality by addressing noise, inconsistencies, and missing values. The processing pipeline involves two main steps: depth preprocessing and depth image colorization. The following subsections provide a detailed explanation of each processing step.

3.2.1 Depth Preprocessing

The raw depth data captured by the stereo depth camera exhibited inherent noise and inconsistencies, leading to fluctuations in the depth measurements. To address these issues and improve data quality, a series of preprocessing steps were applied sequentially.

Fig. 3.6 illustrates the overall depth preprocessing pipeline. The first step involved hole filling, a process that addresses gaps or missing pixel values (black pixels) in the depth image. The “filling-from-left” method was employed, strategically chosen because the camera’s reference point is the left camera and shadows often appear on the left background. This method fills the gaps by referencing valid depth values from the leftmost pixel column and progressing rightwards.

Following hole filling, the depth image was converted into a disparity image using the formula in **Eq. (3.1)** that considers the camera’s focal length f in pixels and the baseline b between its imagers in meters. Disparity represents the depth information in a different format.

Next, bilateral spatial filtering was applied to the disparity image. This filtering technique effectively smooths the data while preserving edges, further reducing noise and inconsistencies. The filtered disparity image was then converted back into depth space using the reciprocal of **Eq. (3.1)**.

Depth thresholding was then implemented to define the valid depth range within the captured scene. In our case, a minimum depth of 0.3 meters and a maximum depth of 6 meters were chosen. This range encompasses the entire room, considering the camera-to-bed distances of 2.5 meters. Finally, another round of hole filling was applied to the processed depth frame to ensure data consistency.

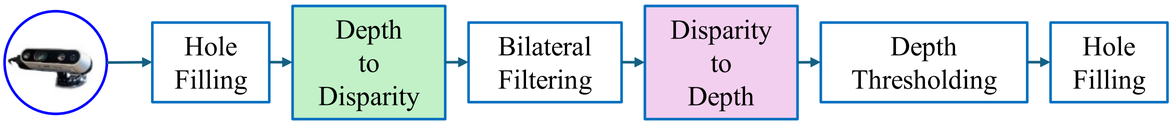


Fig. 3.6. Depth Preprocessing Pipeline

$$\text{Disparity} = \frac{f \times b}{\text{Depth}} \tag{3.1}$$

3.2.2 Depth Image Colorization

Following depth processing, the resulting depth image underwent colorization to enhance visualization and facilitate subsequent analysis. Hue color space, known for its ability to represent a wide range of colors with minimal black or white extremes, was chosen for this process.

As shown in **Fig. 3.7**, hue color space is used for the colorization process. This color space has six scales in both directions of RGB channels and can thus be denoted as having 1529 discrete ranks, or approximately 10.5 bits [77]. Moreover, as one of the colors in the hue color space is always 255, the colorized images will not be too dark. This facilitates visualization but also transforms the depth image into a format compatible with most object detection algorithms, which typically operate on RGB color channels. This conversion acts as a bridge between the depth data and object detection tools, enabling more efficient and effective analysis of the captured information.

The colorization process of depth image and recovery of depth image from colorization is mainly described in the following subsections.

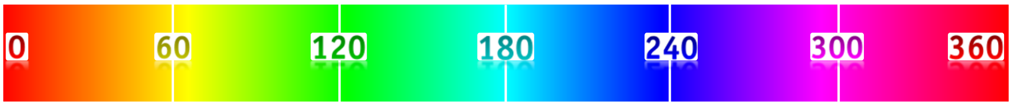


Fig. 3.7. Hue Color Bar (Represented with Degree Values)

3.2.2.1 Inverse Colorization

This proposed system employs a technique called inverse colorization for depth image colorization. Here, “inverse” refers to the use of disparity values, which are the reciprocals of depth values, for the colorization process instead of the original depth values themselves. **Eq. (3.2)** to **Eq. (3.6)** from the reference paper [77] are utilized for this inverse colorization. These equations involve parameters; *d* (depth value), *disp* (disparity value), d_{min} and d_{max} (manually chosen minimum and maximum depth values), and p_r, p_g, and p_b (colorized pixel values). **Fig. 3.8** showcases a sample image generated through this hue-based inverse colorization process, depicting an elderly person sitting on a bed.

$$disp = \frac{1}{d}, \quad disp_{max} = \frac{1}{d_{min}}, \quad disp_{min} = \frac{1}{d_{max}} \tag{3.2}$$

$$d_{normal} = \frac{disp - disp_{min}}{disp_{max} - disp_{min}} * 1529 \tag{3.3}$$

$$p_r = \begin{cases} 255, & \text{if} \quad 0 \le d_{normal} \le 255 \cup d_{normal} > 1275 \\ 255 - d_{normal}, & \text{if} \quad 255 < d_{normal} \le 510 \\ 0, & \text{if} \quad 510 < d_{normal} \le 1020 \\ d_{normal} - 1020, & \text{if} \quad 1020 < d_{normal} \le 1275 \end{cases} \tag{3.4}$$

$$p_g = \begin{cases} d_{normal}, & \text{if} \quad 0 \le d_{normal} \le 255 \\ 255, & \text{if} \quad 255 < d_{normal} \le 510 \\ 765 - d_{normal}, & \text{if} \quad 510 < d_{normal} \le 765 \\ 0, & \text{if} \quad 765 < d_{normal} \le 1529 \end{cases} \tag{3.5}$$

$$p_b = \begin{cases} 0, & \text{if} \quad 0 \le d_{normal} \le 765 \\ d_{normal} - 765, & \text{if} \quad 765 < d_{normal} \le 1020 \\ 255, & \text{if} \quad 1020 < d_{normal} \le 1275 \\ 1275 - d_{normal}, & \text{if} \quad d_{normal} > 1275 \end{cases} \tag{3.6}$$

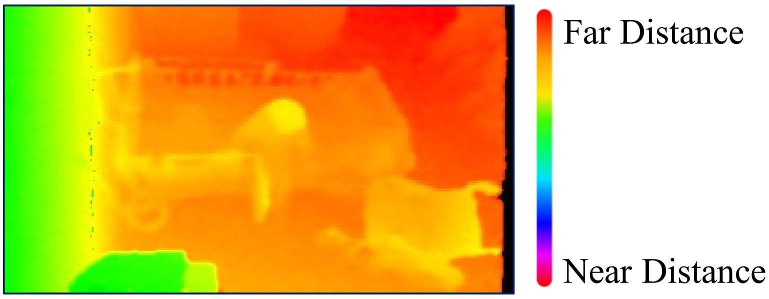


Fig. 3.8. Sample Hue Colorized Image

3.2.2.2 Depth Image Recovery from Colorization

Following depth image colorization, which allows for person detection using color information, it becomes necessary to recover the original depth values for further analysis, particularly feature extraction. **Eq. (3.7)** and **Eq. (3.8)** from the reference [77] are employed for this depth recovery process, essentially reversing the colorization step, and reconstructing the depth map from the colorized image channels. The formulas involve the parameters like p_{rr}, p_{rg}, and p_{rg} (colorized pixel values), $d_{rnormal}$ (recovered normal depth value), and $d_{recovery}$ (restored depth value).

$$d_{rnormal} = \begin{cases} p_{rg} - p_{rb}, & \text{if} \quad p_{rr} \geq p_{rg} \cap p_{rr} \geq p_{rb} \cap p_{rg} \geq p_{rb} \\ p_{rg} - p_{rb} + 1529, & \text{if} \quad p_{rr} \geq p_{rg} \cap p_{rr} \geq p_{rb} \cap p_{rg} < p_{rb} \\ p_{rb} - p_{rr} + 510, & \text{if} \quad p_{rg} \geq p_{rr} \cap p_{rg} \geq p_{rb} \\ p_{rr} - p_{rg} + 1020, & \text{if} \quad p_{rb} \geq p_{rg} \cap p_{rb} \geq p_{rr} \end{cases} \tag{3.7}$$

$$d_{recovery} = \frac{1529}{1529 disp_{min} + \left(disp_{max} - disp_{min}\right) d_{rnormal}} \tag{3.8}$$

Fig. 3.9 shows the results of colorization (center) and depth recovery (rightmost) from the original depth image (leftmost). The original depth image is a Viridis color-mapped image, converted from depth values and not suitable for direct visualization. To assess the quality of the depth recovery, the Mean Squared Error (MSE) was calculated between the original and recovered depth images by using **Eq. (3.9)** in which *m* and *n* are the image width and height, *I(i,j)* and *K(i,j)* are the coordinates of the original and recovered images. A lower MSE indicates greater similarity between the two images. Our experiment determined that the average MSE error for 100 image pairs is 0.05. **Fig. 3.10** illustrates the comparison between the original raw depth image in grayscale (left) and the preprocessed hue-colorized image (right).

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[I(i,j) - K(i,j) \right]^2 \tag{3.9}$$

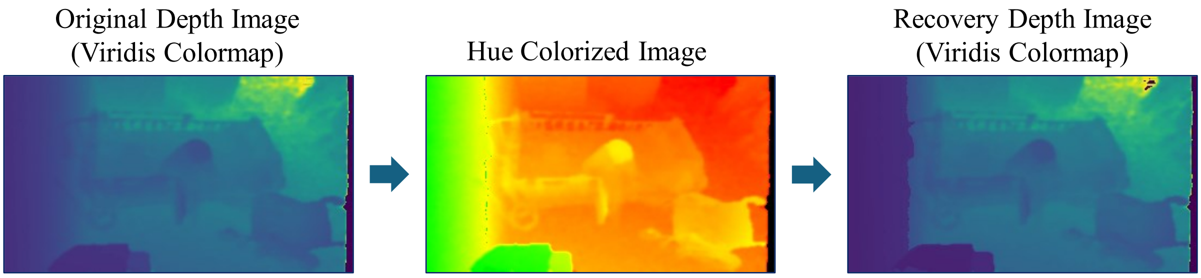


Fig. 3.9. Comparison of Original Depth Image and Recovery Depth Image

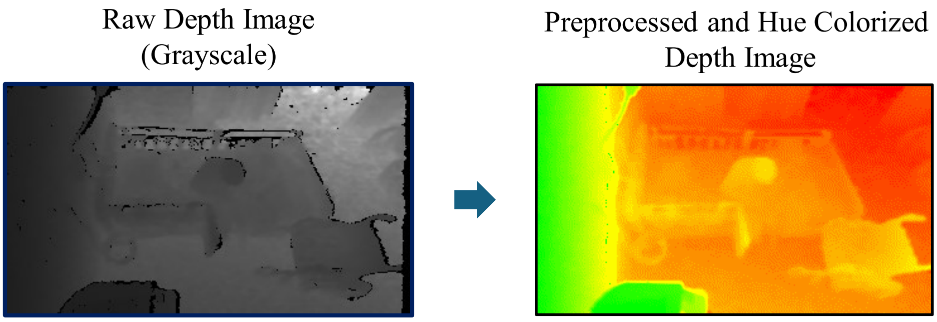
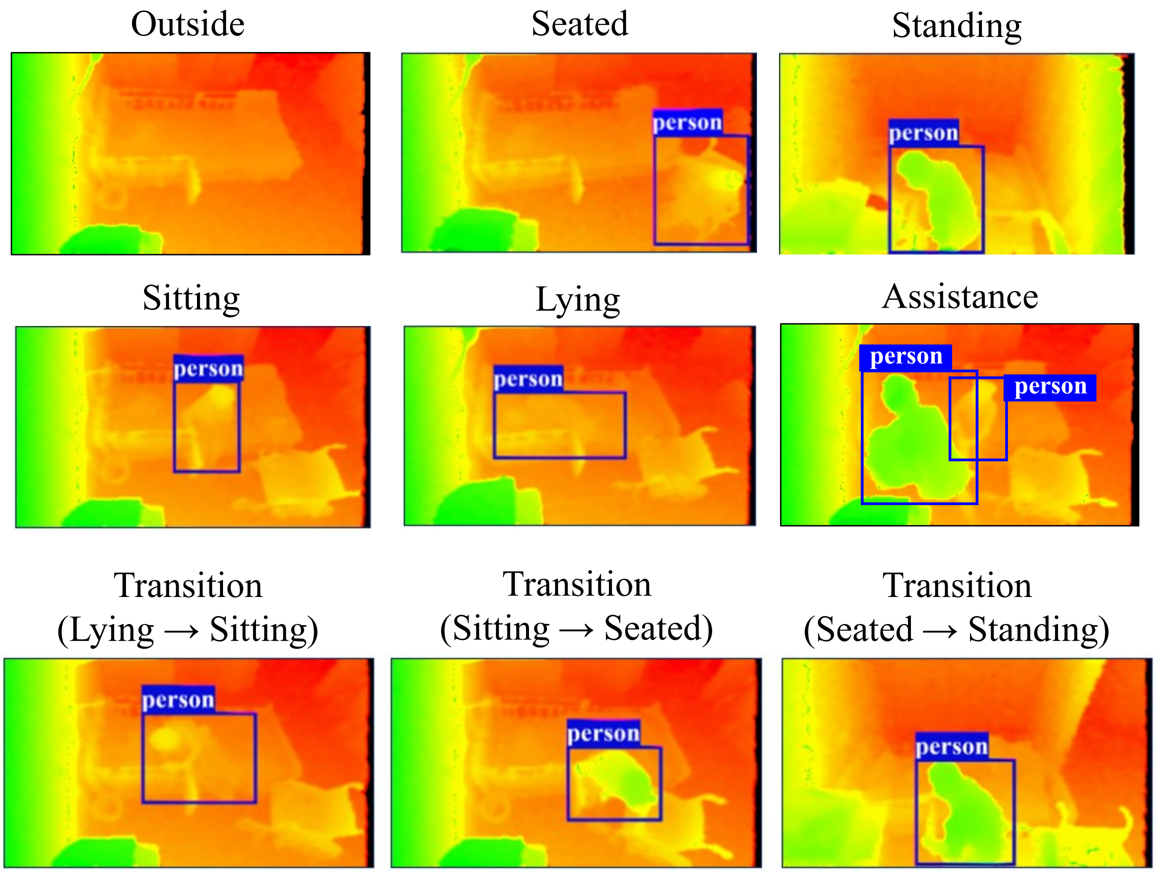


Fig. 3.10. Comparison of Raw Depth Image and Preprocessed Colorized Image

3.3 Visualization of the Elderly’s Common Daily Actions

During the recording period, elderly residents engaged in a set of common daily activities, including “seated in the wheelchair,” “standing,” “sitting on the bed,” and “lying on the bed,” along with transition states between these actions. Notably, the rooms were typically occupied by a single elderly person, primarily during sleep or rest periods. At other times, the person would be marked as “outside” the room. Another significant state captured was “receiving assistance,” typically when a caregiver entered the room to provide aid. Activities like folding clothes were not considered core components of daily routines.

The colorization approach significantly improved the visual representation of depth images, enhancing the interpretation of each action and state as shown in **Fig. 3.11**. In the figure, “outside” indicates the absence of a person within the camera’s view. Others represent specific actions when a person is present: “seated” (in a wheelchair), “standing” (upright posture), “sitting” (on the bed), and “lying” (down for rest). “Transition” signifies a change between actions, and “assistance” indicates support from a healthcare provider.



Outside

Seated

Standing

Sitting

Lying

Assistance

Transition (Lying → Sitting)

Transition (Sitting → Seated)

Transition (Seated → Standing)

Fig. 3.11. Visualization of Each Daily Action of the Elderly Person

Chapter 4

Person Detection

This chapter explores the crucial role of person detection in elderly monitoring systems using computer vision techniques. It delves into the You Only Look Once (YOLO) detector and compares the model's performance. The evaluation process defines various metrics and utilizes two data-splitting strategies to identify the effectiveness of the YOLO model for our specific application. Additionally, we introduce a bounding box recovery approach to address challenges in detecting residents covered by blankets, aiming to improve the overall accuracy of the person detection system.

4.1 Introduction

Person detection, a computer vision technique that identifies people in images and videos, plays a vital role in elderly monitoring systems. By automatically locating individuals within their living spaces, these systems can monitor activity levels, detect falls and emergencies, and ensure overall well-being. Accurate person detection, especially in dynamic environments, is crucial to minimize false alarms and missed detections. Furthermore, a reliable detection algorithm is necessary to obtain bounding box coordinates (rectangular boxes) around the targeted person, as illustrated in **Fig. 4.1**.

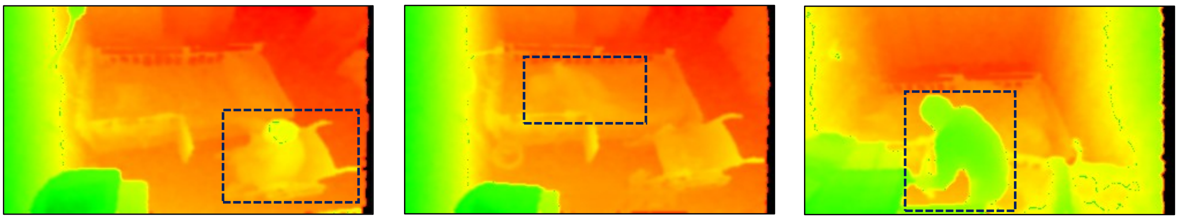


Fig. 4.1. Illustration of Rectangular Boxes around an Elderly Person

4.2 YOLO Detector

This section explores a YOLO detector for person detection in our elderly monitoring system. This model analyzes entire images at once for real-time applications, balancing speed, accuracy, and ease of use. We focus on a large-sized pre-trained version considering our use of colorized depth images. Finally, we discuss the evaluation process using two data-splitting strategies to identify the optimal YOLO model for our specific needs.

4.2.1 Model Selection

This study investigates a large-sized YOLO model (YOLOv5l) [78] for person detection in elderly monitoring systems. The principle of the model is analyzing the entire image at once with a single neural network to predict bounding boxes and object class probabilities as shown in **Fig. 4.2**. This characteristic makes the YOLO model attractive for real-time applications like elderly monitoring due to their balanced performance in terms of speed, accuracy, and ease of use.

Additionally, considering our use of colorized images (converted from depth data) that lack the typical texture information, we will evaluate the model to determine the most suitable option for our elderly monitoring system. The experiment will involve feeding these colorized depth images into the chosen YOLOv5l detector to generate bounding boxes around the person of interest.

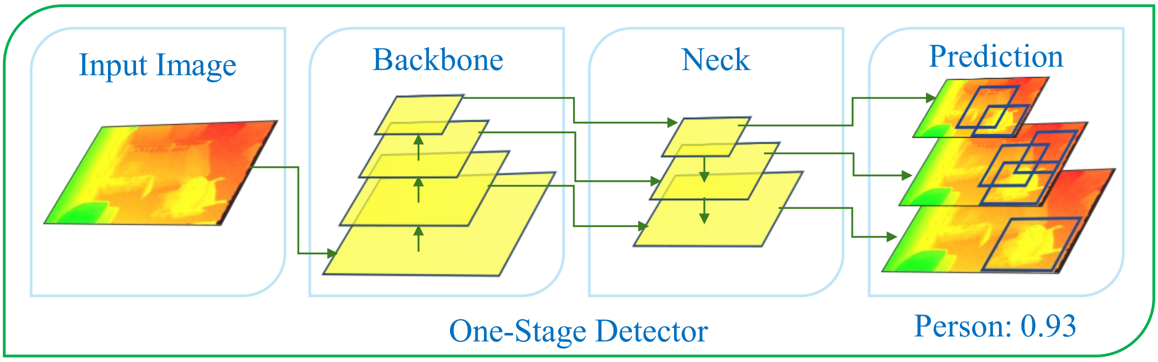
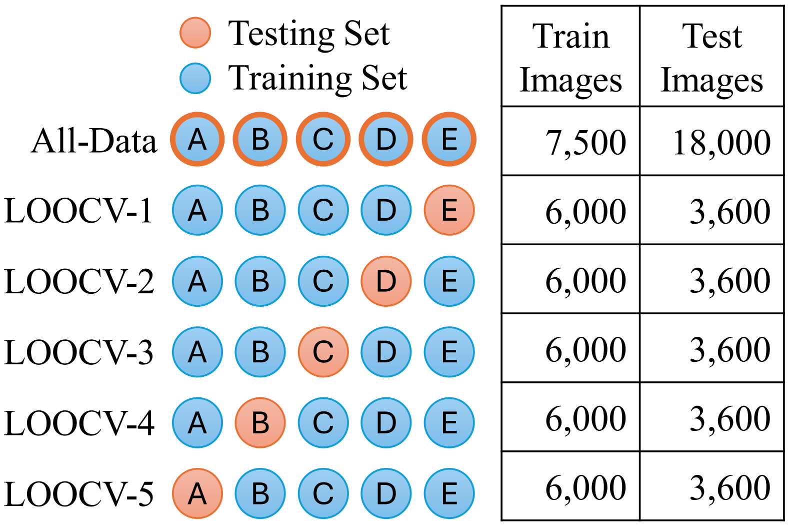


Fig. 4.2. Principle of YOLOv5 Detector

4.2.2 Dataset Preparation

We evaluated the performance of the person detection model using two training and testing data split strategies: All-Data training and Leave-One-Out Cross-Validation (LOOCV) [79]. Data splitting strategies for training and testing are illustrated in **Fig. 4.3**. For evaluation, five elderly data from the hospital are used.

In the All-Data training approach, we utilize the combined data from all five rooms for training/validation and testing. However, to prevent overfitting, we ensure the data used for testing comes from different dates and times compared to the training/validation data. On the other hand, the LOOCV approach involves dividing the data into five sets, one for each room. In each iteration, data from one room is used for testing, while the remaining four rooms' data are combined for training. This process is repeated five times, with each room serving as the testing set once. Finally, the average results across all five iterations are calculated to assess the model's generalizability. PyTorch served as the software development framework for the training and testing processes.



🟠 Testing Set
🔵 Training Set

						Train Images	Test Images
All-Data	🔵🟠 A	🔵🟠 B	🔵🟠 C	🔵🟠 D	🔵🟠 E	7,500	18,000
LOOCV-1	🔵 A	🔵 B	🔵 C	🔵 D	🟠 E	6,000	3,600
LOOCV-2	🔵 A	🔵 B	🔵 C	🟠 D	🔵 E	6,000	3,600
LOOCV-3	🔵 A	🔵 B	🟠 C	🔵 D	🔵 E	6,000	3,600
LOOCV-4	🔵 A	🟠 B	🔵 C	🔵 D	🔵 E	6,000	3,600
LOOCV-5	🟠 A	🔵 B	🔵 C	🔵 D	🔵 E	6,000	3,600

Fig. 4.3. Data Splitting Strategies

4.3 Evaluation Metrics

To determine whether the developed algorithm was reliable, different metrics were employed to assess the performance of the models. For person detection, the focus was on the detection model's ability to correctly identify people (elderly in this context) and distinguish them from the background in colorized images. The two key metrics employed were "precision" and "recall" [80]. "Precision" measures the accuracy of positive detections whereas "recall" assesses the model's ability to capture all relevant detections. Additionally, the localization accuracy of the detection model was evaluated using "mAP@50" and "mAP@50-95" metrics where mAP represents the mean Average Precision [80]. These metrics assess how precisely the model locates people within the images. They are derived from the Intersection over Union (IoU), which measures the overlap between a predicted bounding box (the model's estimate of person location) and a ground-truth bounding box (actual location). "mAP@50" was calculated at an IoU threshold of 0.5, indicating a 50% overlap between predicted and actual bounding boxes. "mAP@50-95", calculated across varying IoU thresholds (0.5 to 0.95), indicates consistent accuracy even with stricter overlap requirements. **Table 4.1** details the focus and desired outcome (higher values) for each evaluation metric.

Table 4.1. Evaluation Metrics for Person Detection

Metric	Focus	Higher Value Indicates
Precision	Correctly identified elderly people	Fewer false positives (background clutter identified as elderly)
Recall	Capturing all actual elderly people	Fewer missed detections (actual elderly people not identified)
mAP @50	Precise location of elderly people	Better overlap between predicted and ground-truth bounding boxes (IoU $\geq$ 0.5)
mAP @50-95	Accurate elderly location across varying overlap thresholds	Consistent performance even with stricter overlap requirements (0.5 $\leq$ IoU $\leq$ 0.95)

4.4 Comparison Result

We evaluated the YOLOv5l using LOOCV-Average and All-Data training approaches. The experimental results are shown in **Table 4.2**. The experimental results indicated that the model achieved the overall performance for LOOCV-Average approach with an average precision exceeding 88% and recall exceeding 72%, demonstrating strong generalization ability to unseen data. It also achieved a mAP@50 of 80%, indicating accurate person localization. On the other hand, the model performed well in All-Data training, with precision and mAP@50 exceeding 98%. The inference time per image is about 19 milliseconds for both methods. Overall, YOLOv5l achieved a balance between accuracy and speed for person detection in our elderly monitoring system.

Table 4.2. Experimental Results for YOLOv5l Model

Training Method	Evaluation Metrics (%)				Inference Time (milliseconds per image)
	Precision	Recall	mAP@50	mAP@50-95	
LOOCV-Average	88.3	72.5	80.0	45.5	19.2
All-Data	98.1	97.5	98.5	72.4	19.4

4.5 Bounding Box Recovery

After analyzing the results of person detection, it was observed that the most challenging detection scenario involved residents lying on beds covered by blankets. The blankets masked the depth information, making it difficult to distinguish the person from the bed (i.e., depth values become uniform). This resulted in missed detections, negatively impacting the recall rate (percentage of the elderly correctly identified). Despite utilizing numerous training annotations, some frames presented these types of missed detections.

To address this issue, a bounding box recovery approach was implemented as shown in **Fig. 4.4**. If the person is not detected in the current frame, the algorithm checks the previous frame for a detection. If a person was detected previously, the bounding box coordinates from the previous frame are used in the current frame. Next, a frame difference is calculated within the defined bounding box. This involves counting the number of pixels with intensity changes compared to the previous frame. If the total intensity change is less than 30% of the bounding box area, it suggests minimal movement, similar to the previous frame. In such cases, the algorithm replicates the previous bounding box, crops the image based on that box, and continues processing. Conversely, if the intensity change exceeds 30%, significant movement is detected, and no bounding box is assigned to the current frame.

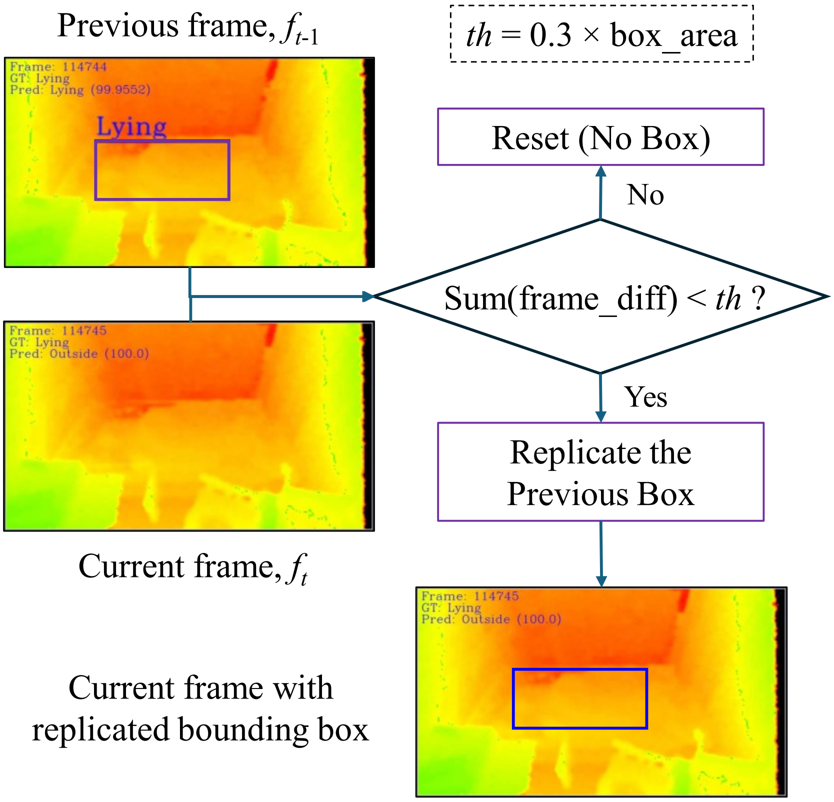


Fig. 4.4. Illustration of Bounding Box Recovery Process

4.6 Conclusion

This chapter compares the performance of YOLOv5l person detection algorithms for elderly monitoring. Evaluating the real-world hospital data revealed that it achieved a promising overall balance between accuracy and speed for person detection within our system. While utilizing colorized depth images, which lack the typical texture information of natural images, it demonstrated a favorable trade-off in the LOOCV approach, achieving an average precision above 88%, recall exceeding 72%, and mAP@50 of 80%. In the All-Data training approach, both precision and mAP@50 both surpassed 98%.

Furthermore, the implemented bounding box recovery technique addressed a specific challenge: residents lying on beds covered by blankets. This approach improved the recall rate by utilizing information from previous frames when encountering such scenarios with limited depth information.

Chapter 5

Action Recognition

This chapter explores the proposed models for elderly action recognition. It explains how these models utilize spatial and temporal features extracted from the person's movement with three main approaches. In the first approach, we extract motion appearance and motion history features from the depth image sequences and represent them using a Histogram of Oriented Gradients (HOG) descriptor. These HOG feature vectors are classified using single Machine Learning (ML) algorithms and those combined with the stochastic Hidden Markov Model (HMM) in the recognition process. In the second approach, we extract straightforward temporal-dependent features from the sequence of segmented person masks, and a Support Vector Machine (SVM) is used for classification. In the third approach, we extract spatiotemporal features automatically using Convolutional Recurrent Neural Networks (CRNN). This approach incorporates motion information derived from body posture changes, inspired by the second approach, to achieve robust transition state recognition using CRNN. This chapter addresses the introduction, the implementation details, and the experimental results.

5.1 Introduction

Action recognition, the process of identifying human actions in videos or image sequences, plays a vital role in elderly monitoring systems. It is an active field of research, with applications using various sensor technologies like wearables, ambient sensors, and vision sensors. The proposed system used a stereo depth camera to recognize actions at care facilities for senior citizens. This system can not only reduce workloads for caregivers by automating the monitoring process but also provide useful and insightful information to support care decisions.

By recognizing activities like walking, sitting, or eating, these systems can provide valuable insights into the health, independence, and well-being of the elderly. This allows for early identification of potential problems and timely interventions to support their independence and quality of life. Our focus is on action recognition for elderly patients residing in elder care facilities, where residents may require varying levels of assistance and may exhibit changes in their daily routines. We utilize real-world data captured during their daily activities. Common actions include "seated in the wheelchair," "standing," "sitting on the bed," and "lying on the bed." We also consider transitional states like "sitting to lying down," "standing to sitting," and so forth. There's typically only one person per room, with "outside" indicating the resident's absence. Additionally, we capture "receiving assistance" when a caregiver enters the room. Activities like folding clothes or eating are not considered as they are less frequent during our recording period. Hence, our system prioritizes accurate recognition between four actions (seated, standing, sitting, lying) and three states (transition, outside, receiving assistance) for effective monitoring. The flowchart of processing for these actions is described in **Fig. 5.1**. Actions can be generally described as a sequence of images and thus a sequence of images is used to extract features to recognize various actions. To do that, spatial and temporal features are extracted from the sequences of images in the proposed system.

5.2 Evaluation Metrics

To evaluate how well the model performed in recognizing actions, especially for elderly care applications, we used several metrics common in multi-class classification tasks. These include "accuracy," "precision," "recall," and "F1-score." "Accuracy" represents the overall percentage of actions, including transition states, classified correctly. However, for elderly care applications, accurate recognition of transition states is crucial for timely monitoring and intervention. Therefore, we also consider metrics like "precision," "recall," and "F1-score" to provide a more detailed picture of the model's performance in recognizing these critical states. "Precision" focuses on how good the model is at identifying specific actions without any mistakes. "Recall" measures how well the model catches all instances of a particular action, avoiding misses, which is crucial for not missing activities like "lying on the bed for extended periods." "F1-score" combines precision and recall into a single score, giving a balanced view of both. **Table 5.1** details the specific focus and ideal values for each metric when applied to recognizing transition states.

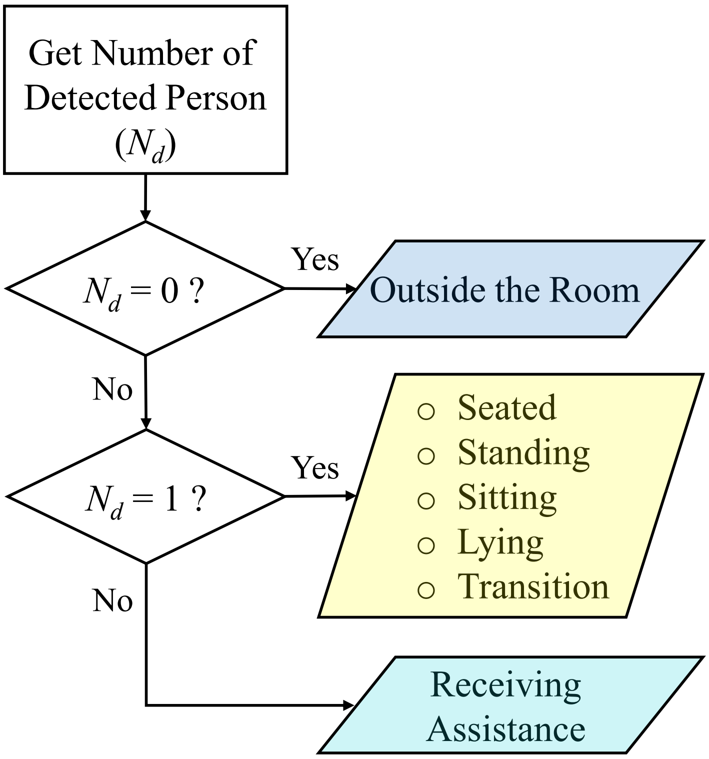
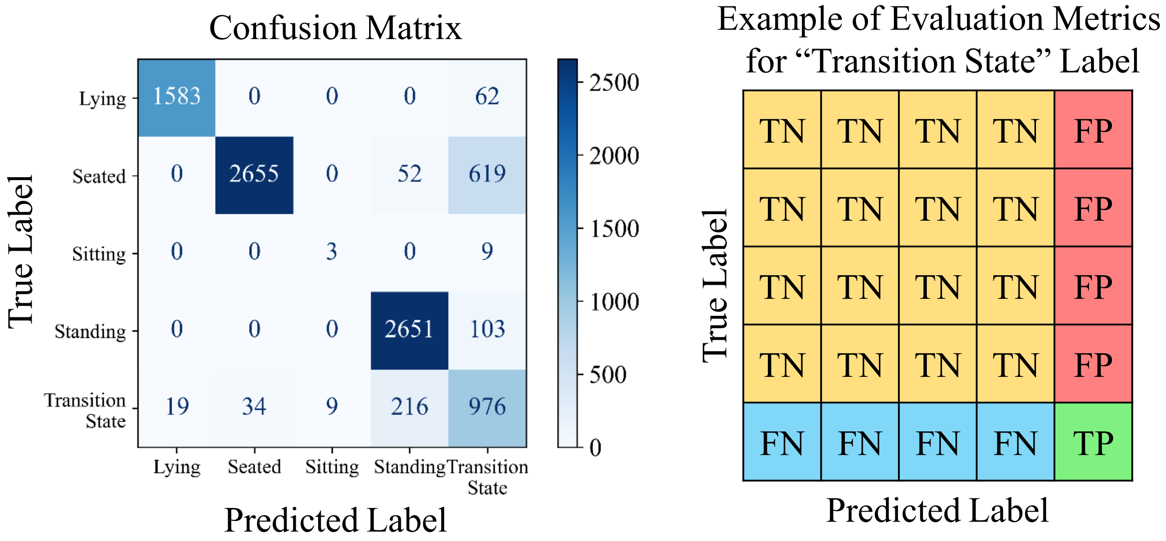


Fig. 5.1. Flowchart of Action Labelling

Table 5.1. Evaluation Metrics for Action Recognition (for "Transition State" Label)

Metric	Focus On	Higher Values Indicates
Accuracy	Overall correct classifications	More actions classified correctly out of the total
Precision	Specific "transition state" recognition	Fewer incorrect classifications of other actions as "transition states"
Recall	Capturing all "transition states"	Fewer actual "transition state" instances missed
F1-score	Balanced "transition state" performance	Minimizes confusion between "transition states" and other classes

Fig. 5.2 shows an example of calculating these metrics for the "transition state" label in which True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are marked in the right matrix according to the left confusion matrix. The resulting percentages for each metric are described in **Eq. (5.1)** to **Eq. (5.8)**. Overall, achieving high values for all these metrics in action recognition tasks, especially for critical actions and transitions in elderly care applications, signifies a well-performing algorithm.



True Label \ Predicted Label	Lying	Seated	Sitting	Standing	Transition State
Lying	1583	0	0	0	62
Seated	0	2655	0	52	619
Sitting	0	0	3	0	9
Standing	0	0	0	2651	103
Transition State	19	34	9	216	976

TN	TN	TN	TN	FP
TN	TN	TN	TN	FP
TN	TN	TN	TN	FP
TN	TN	TN	TN	FP
FN	FN	FN	FN	TP

Fig. 5.2. Sample Evaluation for "Transition State" Label

$$\mathrm{TP} = 976 \tag{5.1}$$

$$\mathrm{FP} = 103 + 9 + 619 + 62 = 793 \tag{5.2}$$

$$\mathrm{TN} = 2651 + 3 + 52 + 2655 + 1583 = 6944 \tag{5.3}$$

$$\mathrm{FN} = 216 + 9 + 34 + 19 = 278 \tag{5.4}$$

$$\begin{aligned}\text{Accuracy} &= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \\ &= 0.8809 = 88.09\%\end{aligned} \tag{5.5}$$

$$\begin{aligned}\text{Precision} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \\ &= 0.5517 = 55.17\%\end{aligned} \tag{5.6}$$

$$\begin{aligned}\text{Recall} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \\ &= 0.7783 = 77.83\%\end{aligned} \tag{5.7}$$

$$\begin{aligned}\text{F1-score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 0.6470 = 64.70\%\end{aligned} \tag{5.8}$$

5.3 HOG Features-based Recognition

HOG features are commonly used descriptors in computer vision for object detection and action recognition. HOG features capture the local gradient information within an image, which can be helpful for distinguishing between different human poses and actions. **Fig. 5.3** illustrates the overview of the HOG features-based action recognition approach. First, a You Only Look Once (YOLOv5) object detector trained with custom data is used to identify people in the video frames. The bounding boxes around the detected persons are then used to extract HOG features from the corresponding image regions. These extracted HOG feature vectors represent the spatial information of the person's pose within the image.

The extracted HOG feature vectors are then used for action recognition. We explore two approaches:

- **Single ML algorithm**: In this approach, we utilize an SVM algorithm to classify the HOG feature vectors into different action categories. This allows us to assess the effectiveness of individual ML algorithms for real-time action recognition.
- **HMM with ML classifiers**: Here, we combine the ML classifiers with a stochastic HMM for action recognition. HMMs are powerful for modeling sequential data, which can be beneficial for capturing the temporal dynamics of human actions. The HOG features provide spatial information about the person's pose at each frame, while the HMM helps in recognizing the sequence of poses that constitute an action.

We evaluate the proposed HOG features-based recognition approach using data collected from three elderly residents at the care center.

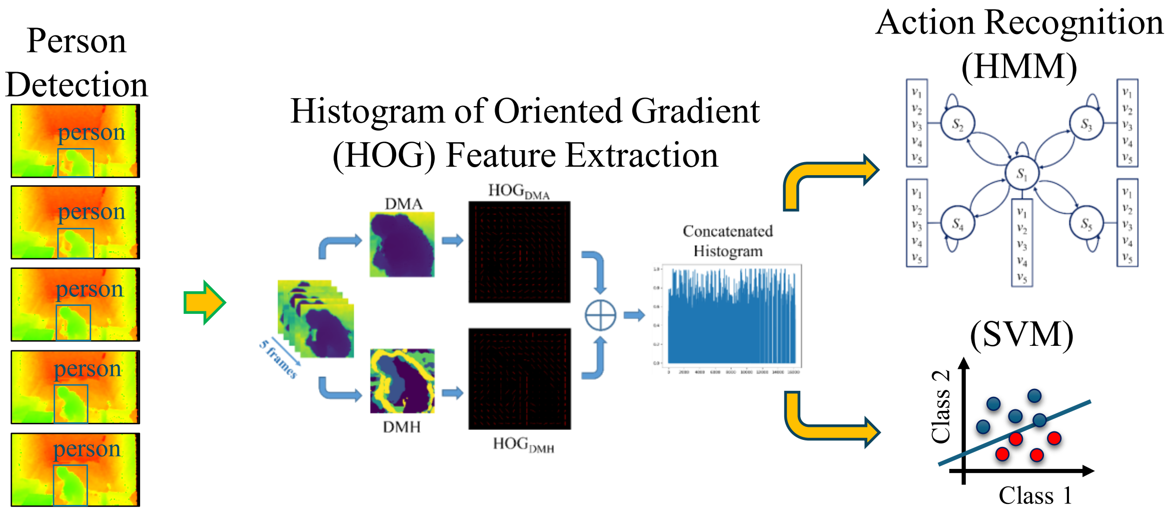


Fig. 5.3. Overview of HOG Features-based Action Recognition

5.3.1 Feature Extraction

After identifying the person using YOLOv5, we obtain bounding boxes of various sizes. These bounding boxes are cropped from the original image and then resized to a fixed size (128×128) for consistency. We intend to extract features from the depth image sequences in this approach. For this purpose, depth maps are then recovered from the resized images as shown in **Fig. 5.4**. The formula used for the depth recovery was described in **Eq. (3.7)** and **Eq. (3.8)** (refer to **Chapter 3** for details).

The architecture of feature extraction is shown in **Fig. 5.5**. Unlike using single frames, this approach utilizes an action sequence of five consecutive bounding boxes for feature extraction. To capture both spatial and temporal information, the Depth Motion Appearance (DMA) and the Depth Motion History (DMH) features are applied [81]. The DMA captures the overall shape and appearance of the person within the sequence. The DMH captures the temporal information of depth motion throughout the action sequence. After obtaining these two action representation maps, the HOG is used as a feature descriptor to summarize the extracted information. Details on calculating DMA and DMH features are provided in the following subsections.

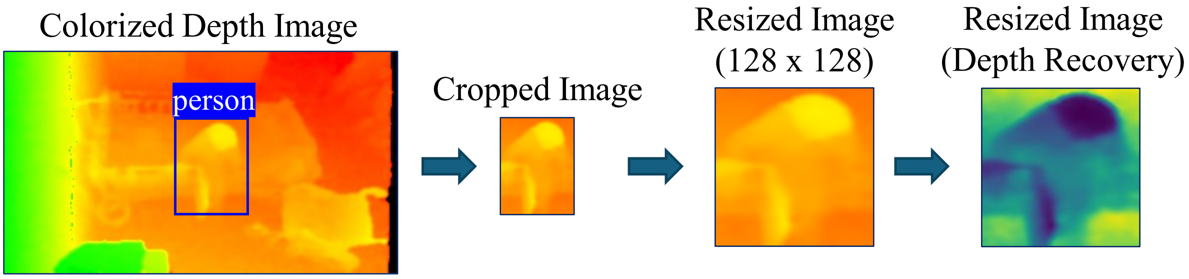


Fig. 5.4. Person Silhouette Cropping, Resizing, and Depth Recovery

5.3.1.1 Depth Motion Appearance

The DMA captures the overall shape and appearance of the person's movement throughout the action sequence. It achieves this by creating a 3D representation that combines all the depth images in the sequence [81]. **Eq. (5.9)** shows the calculation for DMA, where the *$D_t(i, j)$* represents a depth value at a specific pixel location *(i, j)* in the t^{th} depth image, and *$DMA_t(i, j)$* represents the corresponding depth value in the resulting DMA image. In essence, DMA provides a way to extract information about the person's appearance and overall 3D shape during the action.

$$DMA_t(i,j) = \begin{cases} D_t(i,j) & \text{if } DMA_{t-1}(i,j) = 0, \\ min(D_t(i,j), DMA_{t-1}(i,j)) & \text{otherwise.} \end{cases} \quad \textbf{(5.9)}$$

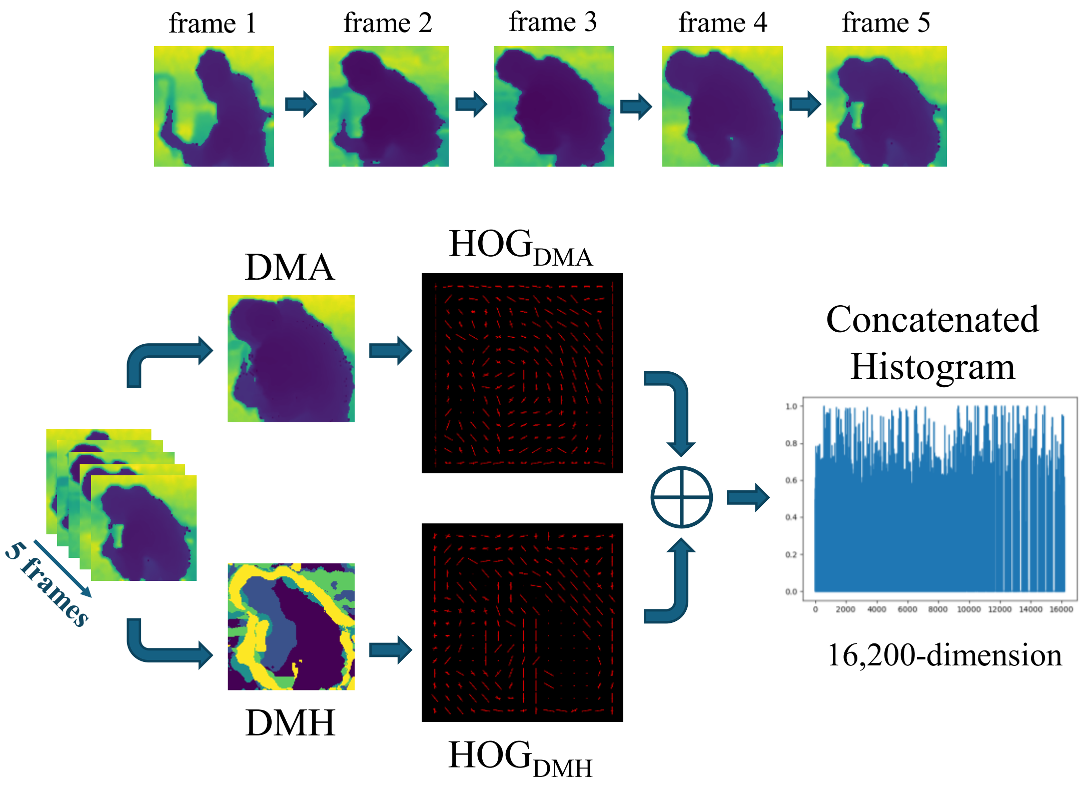


Fig. 5.5. Architecture of Feature Extraction

5.3.1.2 Depth Motion History

The DMH complements the DMA by capturing the temporal aspect of the action sequence. While DMA focuses on the overall appearance of the motion, DMH helps us understand the actual movements performed over time [81]. **Eq. (5.10)** shows the calculation for DMH, where *DMH*$_t$*(i, j)* represents the historical depth motion value at a specific pixel location *(i, j)* in the DMH image, τ defines the time interval considered for the history, and δ is a threshold value for depth differences between consecutive depth maps. The DMH builds upon the concept of Motion History Image (MHI) typically used in 2D videos. However, DMH incorporates depth information to capture changes in depth over time. This allows DMH to account for situations where direction changes in body movements might be obscured in a standard MHI. By combining DMA and DMH, we obtain a more comprehensive representation of the action sequence for recognition.

$$DMH_t(i,j) = \begin{cases} \tau & \text{if } |D_t(i,j) - D_{t-1}(i,j)| > \delta, \\ max(DMH_{t-1}(i,j) - 1,0) & \text{otherwise.} \end{cases} \quad \textbf{(5.10)}$$

5.3.1.3 Histogram of Oriented Gradients

Once we have obtained the two action representation maps (DMA for appearance and DMH for temporal information), we use HOG to describe the local features within these maps. HOG essentially analyzes the intensity gradients in different directions within small image regions. For each map (DMA and DMH), the image is divided into small grids of 8×8 pixels (cells). Within each cell, the gradients are calculated in 9 different directions. To account for variations in lighting, a normalization step is applied using a slightly larger block size (2×2 cells). By performing this analysis on all the cells within a map, we obtain a HOG descriptor with a dimension of 8,100. Finally, the HOG descriptors from both DMA and DMH maps are combined (concatenated) to create a single 16,200-dimensional feature vector that represents the entire action sequence. This feature vector will be used for classification in the next step.

5.3.2 Recognition Model

This section explores two approaches for action recognition using the extracted features: utilizing an SVM to classify HOG feature vectors in the first approach and integrating HMM with ML classifiers for classification in the second approach. The experimental results and discussion for each approach are also described in this section.

5.3.2.1 Support Vector Machine

A linear SVM is used here to classify various actions based on depth map features. For each action, we extract depth information from five consecutive frames and convert them into HOG descriptors. These descriptors are then fed into the trained SVM for action recognition.

To evaluate the performance of the SVM, we use a leave-one-out cross-validation approach. Data from Room 1 are used to train the SVM, while Rooms 2 and 3 provide testing data. Three image sequences of varying lengths are randomly generated from each of these rooms. All sequences are annotated as the first frame signifying the elderly entering the room and the last frame signifying leaving. The SVM attempts to recognize actions every five frames within each testing sequence, and its results are compared with the pre-defined labels (ground truth) to calculate accuracy.

The detailed results for each sequence are presented in confusion matrices as shown in **Table 5.2**, where the values represent the number of frames classified for each action. The results are encouraging. The average accuracy for the three testing sequences from Room 2 is 95.3%, and for Room 3, it is 88.7%, leading to an overall average accuracy of 92% across all six sequences. This demonstrates that the system achieves reasonable action recognition rates even with random sequences of varying lengths. However, it is important to note that some frames are misclassified as “Transition states” due to similarities in appearance and motion between certain activities. Therefore, we explore another technique to improve the differentiation of these transition states.

Table 5.2. Experimental Results for SVM Recognition

Actual Actions *	Predicted Actions											
	Room 2-1 (Accuracy: **94**%)						Room 3-1 (Accuracy: **97**%)					
	Tr	*Se*	*St*	*Si*	*Ly*	*A*	*Tr*	*Se*	*St*	*Si*	*Ly*	*A*
Tr	**27**	2	0	40	0	0	**19**	0	0	41	1	0
Se	10	**57**	0	15	0	0	0	**0**	0	5	0	1
St	0	0	**0**	0	0	0	0	0	**0**	0	0	0
Si	0	0	0	**120**	0	0	0	0	0	**29**	0	0
Ly	146	5	0	0	**3,622**	1	17	0	0	0	**3,489**	0
A	2	0	0	5	3	**9**	14	25	0	5	0	**46**
	Room 2-2 (Accuracy: **98**%)						Room 3-2 (Accuracy: **94**%)					
Tr	**0**	0	0	0	0	0	**20**	0	0	2	1	0
Se	0	**0**	0	0	0	0	5	**0**	0	0	0	1
St	0	0	**0**	0	0	0	0	0	**0**	0	0	0
Si	0	0	0	**0**	0	0	1	0	0	**78**	0	0
Ly	976	5	0	0	**6,605**	0	15	0	0	0	**1,379**	0
A	29	13	0	10	0	**20**	39	15	0	20	0	**54**
	Room 2-3 (Accuracy: **94**%)						Room 3-3 (Accuracy: **75**%)					
Tr	**5**	0	0	49	0	0	**315**	5	12	0	5	0
Se	0	**0**	0	0	0	0	39	**57**	0	0	0	0
St	0	0	**0**	0	0	0	62	30	**33**	0	0	0
Si	0	0	0	**826**	0	2	0	0	0	**0**	0	0
Ly	0	0	0	0	**0**	0	163	0	0	0	**1,203**	0
A	5	0	0	0	0	**38**	106	33	15	45	67	**135**

* *Tr*: Transitions, *Se*: Seated, *St*: Standing, *Si*: Sitting, *Ly*: Lying, *A*: Assistance

5.3.2.2 Hybrid Hidden Markov Models

The second approach to HOG features-based action recognition uses HMM combined with ML algorithms. This hybrid approach leverages the strengths of both techniques. Here, we will develop hybrid HMMs that integrate the classification outputs from the ML algorithms. This section will delve deeper into the details of the HMMs, including: (1) defining the number of hidden states in the model, which represent different actions or activities, (2) calculating the probabilities of transitioning between states and observing specific features given a particular state, and (3) outlining the steps involves in using HMM for prediction of actions.

5.3.2.2.1 Action Definitions

This section defines the set of actions we aim to recognize using the HMMs. The experiment focuses on five distinct actions: 'Transition', 'Seated in the wheelchair', 'Standing', 'Sitting', and 'Lying'. For each action, depth features are extracted from five consecutive frames and converted into HOG descriptors, similar to the SVM approach. Since HMMs require sequences of observations, the classified actions are transformed into observation sequences. These observation sequences are then fed into the HMM to predict the most probable sequence of actions occurring in the entire sequence of features.

5.3.2.2.2 HMM Parameters and Probability Measures

The HMM structure employed in this system is visualized in **Fig. 5.6**. The model is characterized by two parameters and three probability measures.

- **Number of States (*S*)**: This corresponds to the number of actions the system can recognize. Here, we have five states (S_1 to S_5) representing the five defined actions.
- **Number of Observation Symbols per State (*V*)**: Each state can emit a specific observation symbol based on the HOG feature classification results from the ML algorithms. Therefore, we define five observation symbols (v_1 to v_5) per state, corresponding to the five possible actions.

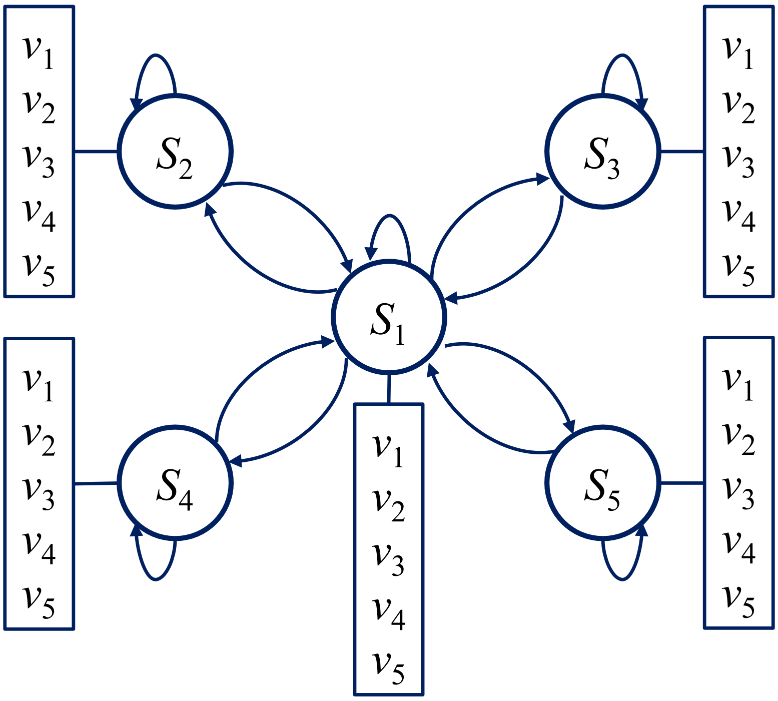


Fig. 5.6. HMM Model Structure

Then, the three key probability measures describe the HMM's behavior.

- **State Transition Probability Distribution (*A*)**: This describes the probability of transitioning from one action state to another. It is calculated by analyzing a long training sequence and creating a co-occurrence matrix that captures how frequently actions transition from one to another. The formula is described in **Eq. (5.11)** where a_{ij} is the transition from one state to another and q_t is the state at time t.

$$A = \{a_{ij}\}, \quad a_{ij} = \mathrm{P}[q_t = S_j \mid q_{t-1} = S_i] \tag{5.11}$$

- **Emission Probability Distribution (*B*)**: This represents the probability of observing a particular feature (HOG descriptor) given the current action state. To calculate this, two training datasets with equal samples for each action are used as shown in **Table 5.3**. HOG features are extracted from five consecutive frames in each sequence, and the calculation process is further explained in **Algorithm 1**. The formula is described in **Eq. (5.12)** where $b_j(k)$ is the occurrence probability in each state j and O_t is the observation symbol at time t.

$$B = \{b_j(k)\}, \quad b_j(k) = \mathrm{P}[O_t = v_k \mid q_t = S_j] \tag{5.12}$$

- **Initial State Distribution (*π*)**: This describes the probability of starting in each action state at the beginning of a sequence. Here, we assume an equal probability for each action as the starting point. The formula is described in **Eq. (5.13)** where π_i is the initial probability for each state i. The probability we used in our approach is described in **Eq. (5.14)**. Then, λ represents the complete HMM model as in **Eq. (5.15)**.

$$\pi = \{\pi_i\}, \quad \pi_i = \mathrm{P}[q_1 = S_i] \;\; \forall \;\; 1 \le i, j, k \le 5 \tag{5.13}$$

$$\pi = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix} \tag{5.14}$$

$$\lambda = (A, B, \pi) \tag{5.15}$$

Table 5.3. Training Datasets for Calculating HMM Emission Probability Distribution *B*

Action	Transition	Seated	Standing	Sitting	Lying
State	S_1	S_2	S_3	S_4	S_5
Number of Sequences in Dataset-*X*	900	900	900	900	900
Number of Sequences in Dataset-*Y*	100	100	100	100	100

Algorithm 1. Calculation of *B* by Computing Mean HOGs

```
Input: Dataset-X, Dataset-Y
Output: B: emission probability distribution
1.  Function MeanHOG_HMM (Dataset-X, Dataset-Y):
2.      Calculate mean HOG feature vectors for each state of Dataset-X and set as
        $M_1H$, $M_2H$, $M_3H$, $M_4H$, and $M_5H$.
3.      Perform the following steps for each state of Dataset-Y.
4.          Assign labels to all 100 input HOGs using Eq. (5.16) to Eq. (5.18).
```

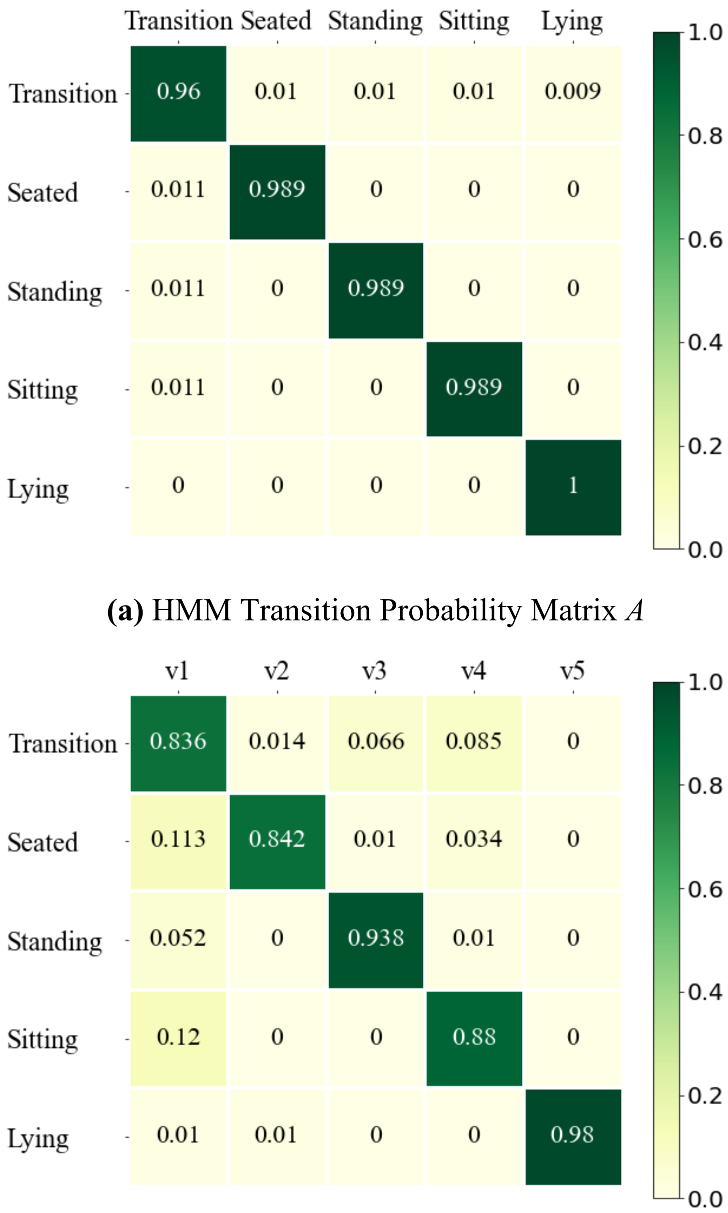
$$d(IH, MH) = \sqrt{\sum_{i=1}^{n}(IH_i - MH_i)^2}, \; n = 16200 \tag{5.16}$$

$$k = \arg\min_{1 \le j \le 5}[d(IH, M_jH)] \tag{5.17}$$

$$\text{assign label} = v_k, \quad k \in \{1,2,3,4,5\} \tag{5.18}$$

```
            where $d(IH, MH)$ is the Euclidean distance between input HOG $IH$ and
            mean HOG $MH$ and $n$ is the length of each HOG feature vector.
5.          Calculate the length (magnitude) of each labeled HOG.
6.          Compute the normal distribution for each HOG length.
7.          Sum normal distributions which have the same labels.
8.          Normalize five normal distributions by dividing each one with a
            summation of all normal distributions.
9.      Return B
10. End Function
```

The Baum-Welch Algorithm [82] is used to train the HMM based on the calculated *A* and *B* probabilities. During training, the HOG features from sequences in Dataset-*Y* are transformed into observation sequences, and the trained *A* and *B* probabilities are visualized using heatmaps in **Fig. 5.7**. These heatmaps ensure that the probabilities for each row (action state) sum to one. Finally, by combining the initial state distribution *π* with the trained *A* and *B* probabilities, we obtained the complete HMM model parameter set. The Viterbi Algorithm is then applied to predict the most likely sequence of hidden states (actions) for a given observation sequence.



(a) HMM Transition Probability Matrix *A*

(b) HMM Emission Probability Matrix *B*

Fig. 5.7. Heatmap Visualization after Training with Baum-Welch Algorithm

5.3.2.2.3 Procedures for HMM Prediction

This section describes how the HMM predicts action sequences for unseen data. **Fig. 5.8** illustrates the prediction process using a short one-minute video sequence (60 frames) processed at 1 frame per second (1fps). HOG features capturing depth appearance and motion are extracted every five frames, resulting in 12 feature vectors. Each feature vector is converted into an observation symbol based on the action it represents. To match the original number of frames, each symbol is duplicated five times, leading to 60 observed symbols.

These 60 observed symbols are then fed into the HMM. The Viterbi Algorithm is used to predict the most likely sequence of hidden states (actions) for the entire sequence. The predicted hidden states are then compared one-on-one with the ground truth labels (predefined actions) to calculate accuracy. For instance, if 50 states out of 60 in the example sequence of **Fig. 5.8** were correctly predicted, the accuracy would be 83.33%.

Following this prediction process, the HMM achieves an accuracy of 91% on the training dataset (Dataset-*Y*). The detailed results for these predictions, including how often each action was correctly classified and misclassified, are presented in the confusion matrix shown in **Table 5.4**.

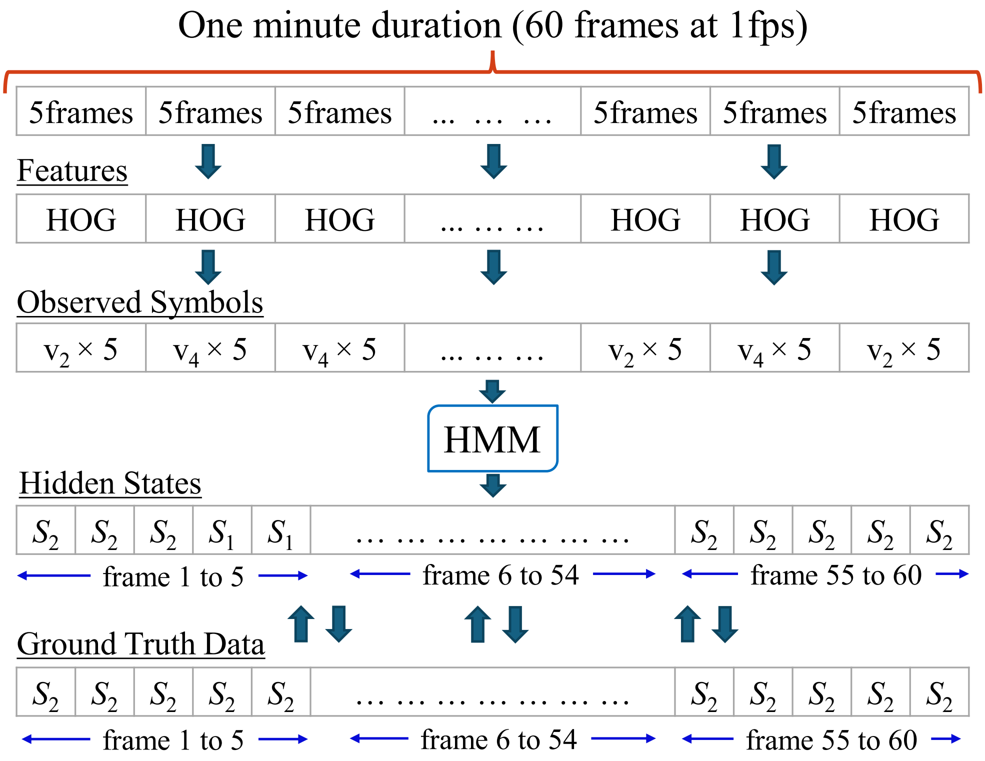


Fig. 5.8. HMM Prediction on Testing Image Sequence

Table 5.4. Confusion Matrix of HMM Prediction for Training Dataset (MeanHOG + HMM)

Actual Actions	Predicted Actions				
	Transition	Seated	Standing	Sitting	Lying
Transition	**90**	1	7	2	0
Seated	14	**86**	0	0	0
Standing	7	0	**93**	0	0
Sitting	14	0	0	**86**	0
Lying	2	0	0	0	**98**

5.3.2.2.4 Alternative HMM Combinations

We also explored alternative approaches for calculating the emission probability distribution B within the HMM. These alternatives aim to improve the accuracy of the HMM predictions. The transition probability matrix A and initial distribution matrix π remain unchanged from the previous approach. However, the procedure for calculating B in **Algorithm 1** is modified.

- **Mean HOG and Euclidean Distance**: This method calculates the average HOG feature vector (mean HOG) for each action class as described in **Algorithm 1**. Sequences from Dataset-Y are then assigned labels based on the Euclidean distance between the input HOG and the mean HOG of each class.
- **k-Nearest Neighbors (k-NN)**: This method employs the k-NN algorithm to classify each HOG feature vector based on the most frequent action label among its k nearest neighbors in the training data. The resulting action labels are used to calculate the emission probabilities. The calculation process is described in **Algorithm 2**.

Algorithm 2. Calculation of B by Computing k-NN

```
Input: Dataset-$X$, Dataset-$Y$
Output: $B$: emission probability distribution
Function k-NN_HMM (Dataset-$X$, Dataset-$Y$):
    Train the k-NN model using HOG features from Dataset-$X$ and divide it into
    five classes such as $C_1$, $C_2$, $C_3$, $C_4$, and $C_5$.
    Perform the following steps for each state of Dataset-$Y$.
        Assign labels to all 100 input HOGs using Eq. (5.19) and Eq. (5.20).
                    $k = kNNpred(IH)$                                   (5.19)
                    $\text{assign label} = v_k, \quad k \in \{1,2,3,4,5\}$   (5.20)
        where $kNNpred(IH)$ is k-NN prediction on HOGs.
        Calculate the length (magnitude) of each labeled HOG.
        Compute the normal distribution for each HOG length.
        Sum normal distributions which have the same labels.
        Normalize five normal distributions by dividing each one with a
        summation of all normal distributions.
    Return $B$
End Function
```

- **SVM**: This method utilizes an SVM to classify each HOG feature vector into one of the action classes. The classified action labels are then used to calculate the emission probabilities. The calculation process is described in **Algorithm 3**.

Algorithm 3. Calculation of *B* by Computing SVM

```
Input: Dataset-$X$, Dataset-$Y$
Output: $B$: emission probability distribution
Function SVM_HMM (Dataset-$X$, Dataset-$Y$):
    Train the SVM model using HOG features from Dataset-$X$ and divide it into
    five classes such as $C_1$, $C_2$, $C_3$, $C_4$, and $C_5$.
    Perform the following steps for each state of Dataset-$Y$.
        Assign labels to all 100 input HOGs using Eq. (5.21) and Eq. (5.22).
            $k = SVMpred(IH)$                                        (5.21)
            $\text{assign label} = v_k, \quad k \in \{1,2,3,4,5\}$     (5.22)
        where SVMpred($IH$) is SVM prediction on HOGs.
        Calculate the length (magnitude) of each labeled HOG.
        Compute the normal distribution for each HOG length.
        Sum normal distributions which have the same labels.
        Normalize five normal distributions by dividing each one with a
        summation of all normal distributions.
    Return $B$
End Function
```

The same training datasets (Dataset-*X* and Dataset-*Y*) used previously are employed for these alternative methods. New HMM models are trained using the Baum-Welch algorithm with the emission probabilities calculated from each method (k-NN and SVM). Finally, the HMM predictions are performed using the Viterbi Algorithm.

Both alternative approaches, (k-NN + HMM) and (SVM + HMM), achieve a significantly higher accuracy of 99% on the training dataset compared to the original method (91%). The detailed results of these predictions are shown in **Table 5.5**.

Table 5.5. Confusion Matrix of HMM Prediction Results Tested on the Training Dataset

(a) k-NN + HMM					
Actual Actions	Predicted Actions				
	Transition	Seated	Standing	Sitting	Lying
Transition	**100**	0	0	0	0
Seated	1	**99**	0	0	0
Standing	0	0	**100**	0	0
Sitting	1	0	0	**99**	0
Lying	1	0	0	0	**99**
(b) SVM + HMM					
Actual Actions	Predicted Actions				
	Transition	Seated	Standing	Sitting	Lying
Transition	**100**	0	0	0	0
Seated	0	**100**	0	0	0
Standing	1	0	**99**	0	0
Sitting	2	0	0	**98**	0
Lying	0	0	0	0	**100**

5.3.2.2.5 Experimental Results

This section evaluates the system's performance using real-world data from three separate rooms of the elder care center. Each testing sequence was trimmed to capture the resident entering the room (first frame) and leaving (last frame). The HMM recognition was performed once per minute throughout each testing sequence. The predicted actions were then compared with ground truth labels to calculate accuracy.

Table 5.6 compares the three proposed methods that combine HMMs with different classification algorithms. The average accuracy for each method in each room is presented. The results show that the combination of SVM classification and HMM for calculating the emission probability matrix achieved the highest overall accuracy of 84.04% across all testing sequences.

Table 5.7 details the recognition accuracy rate for each specific action, using the (SVM + HMM) method in each testing room, in which the "Transition" label has the lowest accuracy rate with an average of 64.14%.

Table 5.6. Comparison of Three Methods

Room ID	Total Sequences	Average Accuracy for All Sequences After Testing with Three Methods (%)		
		Mean + HMM	k-NN + HMM	SVM + HMM
1	22	87.05	95.19	90.28
2	10	74.83	89.01	81.37
3	17	79.41	57.56	80.48
Average Accuracy		80.43	80.59	**84.04**

Table 5.7. Accuracy for Each Specific Action Testing with the (SVM + HMM) Method

Room ID	Accuracy (%)					
	Transition	Seated	Standing	Sitting	Lying	Overall
1	63.37	95.45	97.24	95.41	91.65	90.28
2	54.93	75.09	-	98.83	74.61	81.37
3	74.13	48.68	59.96	83.87	91.89	80.48
Average	**64.14**	73.07	78.60	92.70	86.05	84.04

Table 5.8 further details the action recognition results for each sequence in each room using the (SVM + HMM) method. These tables provide information about the recorded data from all three rooms, including the sequence of duration, number of frames, start date and time, and processing time. The sequences are presented from shortest to longest duration. Notably, the average accuracy for Room 1 sequences was 90.28%, while Rooms 2 and 3 achieved 81.37% and 80.48%, respectively. It is important to note that the testing process was conducted on a powerful machine with a 64-bit Core i9 processor, 64GB RAM, and an NVIDIA GeForce RTX 3090 GPU. These results from **Table 5.8** demonstrate that the proposed system can handle real-time action recognition on continuous, long-duration video sequences.

For a broader comparison, **Table 5.9** presents the performance of our HMM methods against related methods from previous works. Different approaches to input data and methods were used in these prior studies, so the comparison included only the number of actions each approach can recognize. The results indicate that our proposed approach achieves higher recognition accuracy compared to these other methods. However, the accuracy may vary according to the dataset scales of each approach.

Table 5.8. Accuracy for Each Testing Sequence (SVM + HMM)

(a) Room 1 Sequences

Sequence	Duration	Total Frame (1fps)	Start Time (yyyy/mm/dd_hh:mm)	Accuracy (%)	Processing Time
1	3 min	168	2019/10/21_07:19	86.31	1 min
2	7 min	433	2019/10/24_08:44	87.76	4 min
3	9 min	530	2019/10/21_18:25	85.66	4 min
4	20 min	1,230	2019/10/19_12:47	72.11	11 min
5	28 min	1,684	2019/10/21_17:37	91.75	15 min
6	36 min	2,192	2019/10/22_17:16	94.30	20 min
7	37 min	2,235	2019/10/19_13:32	96.06	21 min
8	1 hr 05 min	3,876	2019/10/24_07:27	83.18	35 min
9	1 hr 51 min	6,673	2019/10/24_11:34	86.68	1 hr
10	1 hr 52 min	6,687	2019/10/19_08:12	95.10	1 hr 02 min
11	2 hr 36 min	9,347	2019/10/21_11:54	90.46	1 hr 23 min
12	2 hr 44 min	9,865	2019/10/20_11:44	97.77	1 hr 31 min
13	2 hr 48 min	10,063	2019/10/22_11:19	85.44	1 hr 33 min
14	3 hr 02 min	10,927	2019/10/18_12:08	99.35	1 hr 40 min
15	3 hr 15 min	11,688	2019/10/23_11:38	96.89	1 hr 47 min
16	7 hr 20 min	26,412	2019/10/12_20:54	96.41	4 hr 03 min
17	10 hr 59 min	39,535	2019/10/20_18:57	86.78	6 hr 06 min
18	11 hr 10 min	40,158	2019/10/21_18:40	86.38	6 hr 14 min
19	11 hr 30 min	41,407	2019/10/22_18:02	92.67	6 hr 25 min
20	11 hr 43 min	42,234	2019/10/18_17:58	91.69	6 hr 54 min
21	12 hr 11 min	43,842	2019/10/23_17:05	93.37	7 hr 22 min
22	12 hr 36 min	45,366	2019/10/19_17:51	90.11	7 hr 59 min
Average Accuracy				90.28	-

(b) Room 2 Sequences

Sequence	Duration	Total Frame (1fps)	Start Time (yyyy/mm/dd_hh:mm)	Accuracy (%)	Processing Time
1	6 min	362	2019/10/26_07:20	67.96	3 min
2	18 min	1,078	2019/10/26_07:32	92.12	11 min
3	41 min	2,455	2019/10/28_04:45	93.93	24 min
4	1 hr 10 min	4,214	2019/10/26_06:02	92.05	43 min
5	2 hr 10 min	7,808	2019/10/27_04:35	83.03	1 hr 19 min
6	2 hr 22 min	8,494	2019/10/26_09:54	94.59	1 hr 26 min
7	2 hr 28 min	9,488	2019/10/25_11:07	77.18	1 hr 35 min

8	11 hr 21 min	40,846	2019/10/25_17:01	53.04	6 hr 53 min
9	11 hr 44 min	42,214	2019/10/26_15:10	63.57	7 hr 33 min
10	12 hr 11 min	43,896	2019/10/27_14:02	96.22	7 hr 24 min
Average Accuracy				81.37	-

(c) Room 3 Sequences

Sequence	Duration	Total Frame (1fps)	Start Time (yyyy/mm/dd_hh:mm)	Accuracy (%)	Processing Time
1	3 min	171	2019/10/26_07:56	84.80	2 min
2	25 min	1,511	2019/10/27_04:24	72.34	16 min
3	27 min	1,630	2019/10/12_12:09	93.56	17 min
4	32 min	1,941	2019/10/26_07:23	80.94	21 min
5	39 min	2,325	2019/10/22_19:16	81.29	23 min
6	52 min	3,146	2019/10/25_12:21	69.52	33 min
7	1 hr 02 min	3,697	2019/10/12_13:47	93.70	38 min
8	1 hr 25 min	5,112	2019/10/27_06:12	64.10	55 min
9	2 hr 01 min	7,240	2019/10/28_05:11	79.93	1 hr 17 min
10	10 hr 25 min	38,085	2019/10/18_19:45	79.95	6 hr 22 min
11	10 hr 43 min	38,551	2019/10/20_19:23	77.13	6 hr 34 min
12	11 hr 17 min	40,608	2019/10/12_17:54	93.84	6 hr 50 min
13	11 hr 23 min	40,971	2019/10/26_16:48	71.05	5 hr 51 min
14	11 hr 41 min	42,092	2019/10/19_19:33	73.67	7 hr 19 min
15	11 hr 44 min	42,263	2019/10/21_18:32	70.23	7 hr 25 min
16	11 hr 54 min	42,823	2019/10/27_15:47	92.31	7 hr 11 min
17	11 hr 55 min	42,882	2019/10/25_17:39	89.82	7 hr 23 min
Average Accuracy				80.48	-

Table 5.9. Comparison between the Proposed Methods and the Previous Works

Approach	Method	No. of Actions	Accuracy (%)
RGB Images	CNN [37]	15	71.00
Skeleton	Random Forest [42]	20	70.00
Sensor Data	Two-Layer HMM [56]	13	74.85
Sensor Data	Hierarchical HMM [57]	12	65.20
Depth Images (Proposed Method)	SVM	5	73.22
	Mean HOG + HMM	5	80.43
	k-NN + HMM	5	80.59
	SVM + HMM	5	**84.04**

5.3.2.3 Discussion

This section discusses the key findings and limitations of the explored approaches for HOG features-based action recognition. Our experiments evaluated three methods that combine HMMs with different classification algorithms. The results revealed that the combination of SVM classification and HMM for calculating the emission probability matrix achieved the highest overall accuracy. This suggests that SVM effectively classifies the extracted features (HOG descriptors) for action recognition within the HMM framework. Moreover, it also significantly outperformed the recognition that used SVM alone.

The findings of this study warrant careful consideration due to limitations in the data collection process. The data used for training and testing was collected from only three rooms in a single care center. This limited scope might affect the generalizability of the system to different environments.

Despite promising results, the method encounters some recognition challenges. As shown in **Table 5.7**, it sometimes misclassifies actions as "Transition", especially in the following scenarios:

- **Situation 1**: When a person lies on the bed in an unusual position.
- **Situation 2**: When the person is not visible due to positioning or objects in the frame.
- **Situation 3**: When two actions share similar movement patterns.

Some examples of common false recognition in these situations are shown in **Fig. 5.9**. Another interesting area for further research would be to examine the 'Transition' action in more detail and analyze the high-risk transitional states from one action to another. Hence, these recognition errors are investigated in the next approach to improve the performance of the system.

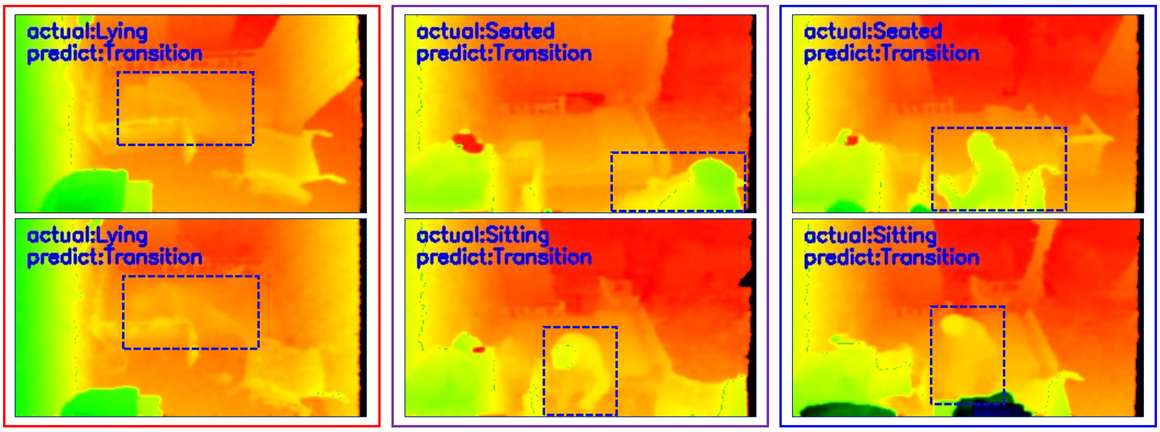


Fig. 5.9. Examples of the Common False Action Recognitions in Situation 1 (left two images), Situation 2 (middle two images), and Situation 3 (right two images)

5.4 Temporal Features-based Recognition

As discussed earlier (in the introduction section), we hypothesized that a person's movements would increase during transitions between actions (e.g., standing up from a chair) compared to periods of stable posture while performing basic actions (e.g., sitting). This suggests that analyzing body movement over a short period could be useful for classifying whether the person is in a transition state or not. Since this classification depends on analyzing movement over time, the extracted features are referred to as temporal-dependent features.

This section explores a complementary approach that leverages temporal features for action recognition. Here, we extract these features from consecutive binary mask images and combine the pixel values at the same coordinates to create a representative image. **Fig. 5.10** provides an overview of this approach. To evaluate its effectiveness, we applied this method to data collected from three elderly residents at the care center.

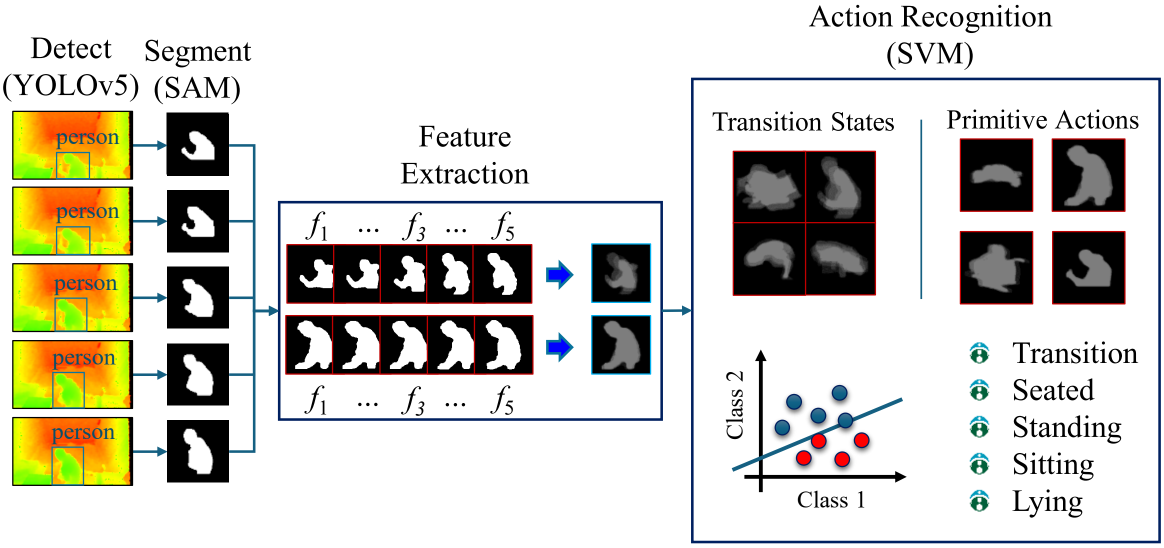


Fig. 5.10. Overview of Temporal Features-based Action Recognition

5.4.1 Person Segmentation

Extracting a clean representation of the person in each frame is crucial for analyzing their movements during action transitions. While the person detector provides bounding boxes that include the person along with background information and other objects, we need to isolate the person for further analysis.

This section describes the approach used for person segmentation. Here, we leverage the Segment Anything Model (SAM) [83] developed by Meta AI. Unlike YOLOv5's instance segmentation model [84], which requires pre-existing mask annotations for training, SAM offers a powerful alternative. SAM excels at various segmentation tasks without additional training data (mask annotations). This makes it ideal for our scenario where creating custom training masks would be time-consuming. Besides, trained on a massive dataset, SAM generates high-quality person masks from various prompts like bounding boxes used in this experiment as shown in **Fig. 5.11**. The process flow is that the bounding box coordinates obtained from YOLOv5 are fed as prompts to SAM first. Then, SAM generates person masks for each frame. The obtained masks are converted into binary images for further processing.

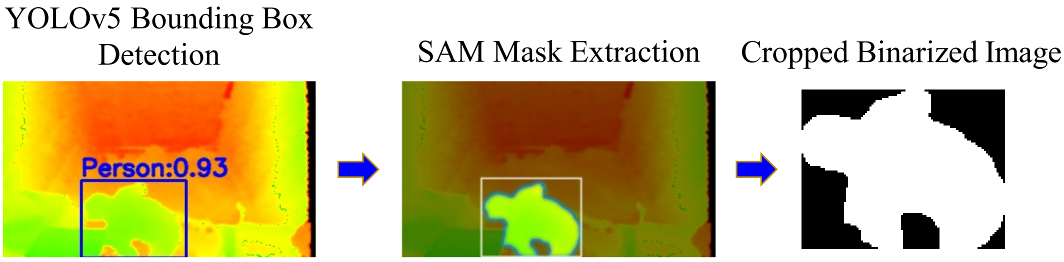


Fig. 5.11. Proposed Person Detection and Mask Extraction

Since the bounding boxes may have different sizes, the extracted masks are padded with black pixels to create uniform images (144×144 pixels) as shown in **Fig. 5.12**. The size was chosen based on the analysis of bounding box sizes from the person detection step.

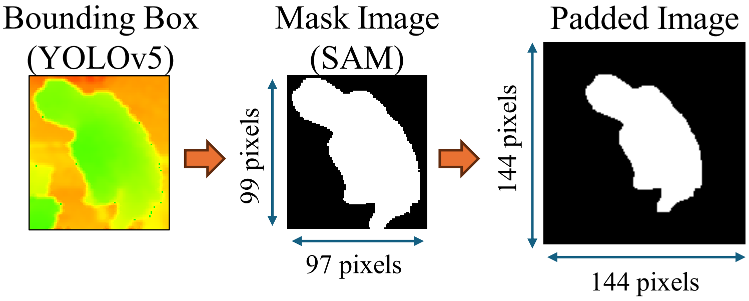


Fig. 5.12. Binarized Padding Image with a Fixed Size

Our approach focuses on analyzing sequences of images rather than individual frames. This emphasis stems from the understanding that transitions between actions involve more movement than stable postures during specific actions.

To determine the ideal duration for analyzing movement and classifying transitions, we conducted an in-depth analysis of ground-truth transition durations in the three experimental rooms. This analysis revealed that 3 seconds was the most frequent transition duration. The histogram of the transition durations shown in **Fig. 5.13** supports the selection of an optimal duration between 2 and 12 seconds. Considering the trade-off between delayed recognition for longer durations and potential inaccuracies for shorter durations, a duration range of 3-5 seconds was deemed suitable.

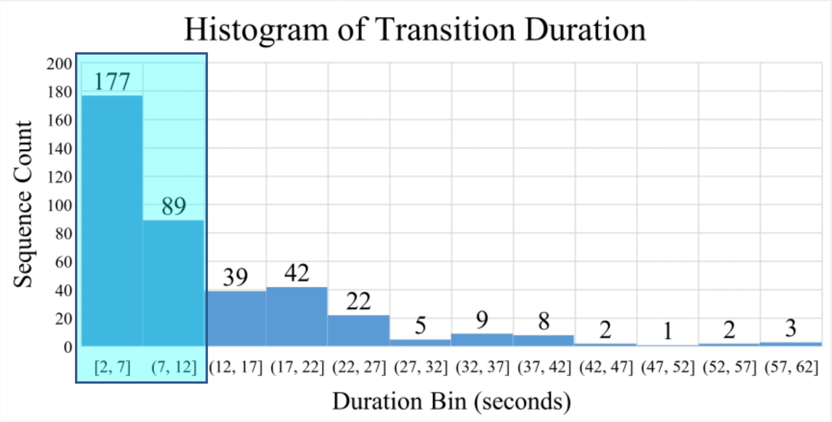


Fig. 5. 13. Histogram of Transition Duration

Given the processing rate of 1fps, a 5-second duration was chosen for robust action recognition. This translates to analyzing five consecutive frames within each sequence. To process a continuous video stream, a sliding window method with a window size of 5 frames and a stride of 1 frame is employed (similar to the ground-truth labeling process). This approach ensures continuous analysis throughout the video sequence. By combining person segmentation and analyzing sequences of frames, we can effectively capture the body movement information necessary to identify transition states.

After comparing the representative images, the differences can be seen as expected in **Fig. 5.14** in which the representative images are converted to grayscale by multiplying the pixel values with 25 for visualization purposes. However, the original combined pixel values (minimum of 0 and maximum of 5) are used for further processing by converting 2D images into 1D feature vectors. Then, these extracted feature vectors are classified using the SVM classifier.

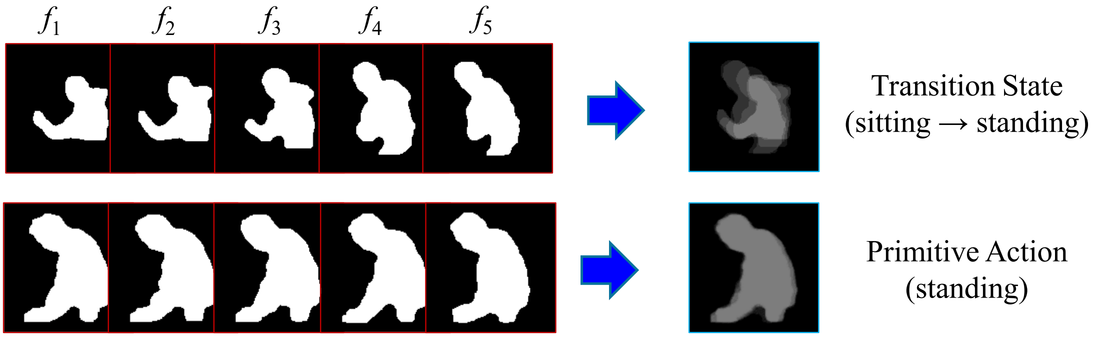


Fig. 5.14. Comparison of the Transition State and Primitive Action

5.4.2 Experimental Results and Discussion

To evaluate the effectiveness of the temporal features-based approach for transition state recognition, we conducted experiments using data collected from three rooms in the care center. Data was collected from all three rooms to ensure a balanced representation of different environments. We selected 30-minute video sequences from each room for testing. These sequences were chosen to include a significant number of transition states. Each room sequence focused on capturing specific transitions that were prevalent in that environment:

- **Room 1**: Standing ↔ Seated in Wheelchair
- **Room 2**: Sitting on Bed ↔ Lying on Bed
- **Room 3**: Standing ↔ Sitting on Bed

Table 5.10 summarizes the model performance on four evaluation metrics. Compared to previous approaches relying on HOG features, this method achieved a significantly higher accuracy rate for recognizing transition states. This result suggests that analyzing motion information derived from body posture changes is effective in distinguishing transitions from primitive actions.

Encouraged by the success of this approach, we explored a new method that leverages motion information within DL networks. This approach aims to achieve not only robust transition state recognition but also automate feature extraction and action recognition within a single framework.

Table 5.10. Performance Evaluation

Metric	Room ID	Action					Average (%)
		Transition	Seated	Standing	Sitting	Lying	
Accuracy (%)	1	90.22	97.94	95.78	99.78	96.61	96.01
	2	97.27	-	-	99.00	98.16	98.14
	3	82.28	-	96.06	82.06	-	86.80
Precision (%)	1	51.50	98.70	94.51	60.00	93.22	79.59
	2	61.64	-	-	100.0	98.54	86.73
	3	48.97	-	26.03	99.56	-	58.19
Recall (%)	1	74.46	97.33	91.82	60.00	49.11	74.54
	2	68.18	-	-	93.94	99.16	87.09
	3	77.02	-	52.78	78.14	-	69.31
F1-score (%)	1	60.89	98.01	93.15	60.00	64.33	75.28
	2	64.76	-	-	96.88	98.85	86.83
	3	59.97	-	34.87	87.56	-	60.80

5.5 Spatiotemporal Features-based CRNN Integration

Building upon the success of the previous approach, this section explores a method that integrates motion information with a CRNN for transition state recognition. This approach aims to not only achieve robust recognition but also automate feature extraction and action recognition within a single framework. The proposed approach is illustrated in **Fig. 5.15**. **Fig. 5.16** visually demonstrates the hypothesis behind using CRNN for transition state recognition. We evaluated this method using data from six elderly residents, collected from both the care center and the hospital setting, to ensure broader applicability.

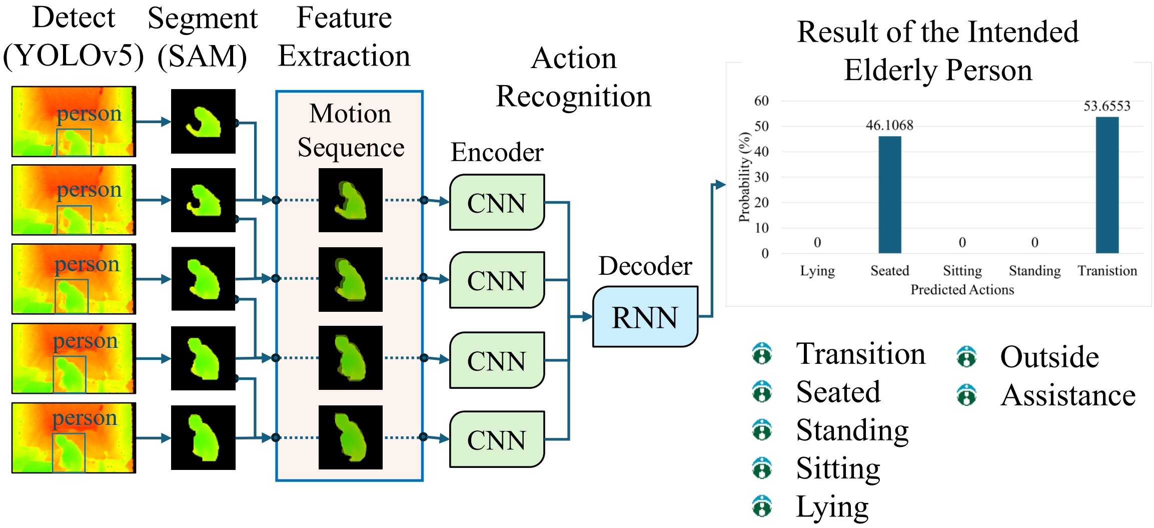


Fig. 5.15. Overview of Spatiotemporal Features-based CRNN Integration

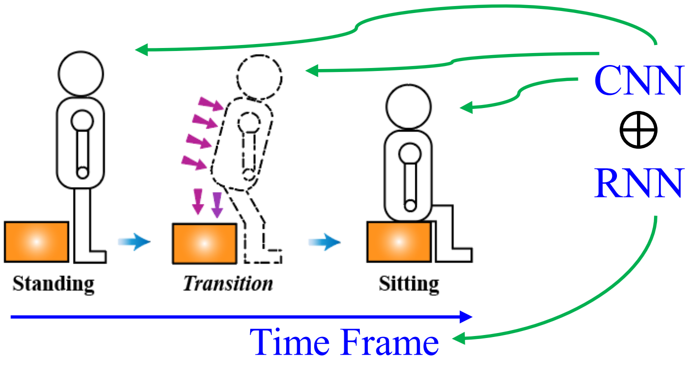


Fig. 5.16. Transition State Recognition using CRNN Architectures

5.5.1 MotionCRNN Action Recognition Model

This subsection delves into the core component of the proposed method: the MotionCRNN action recognition model. This model integrates Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to extract spatiotemporal features for robust action recognition, particularly focusing on transition states. Here, the model not only recognizes actions but also automates feature extraction, streamlining the overall process. The process flow of MotionCRNN-based action recognition is shown in **Fig. 5.17**.

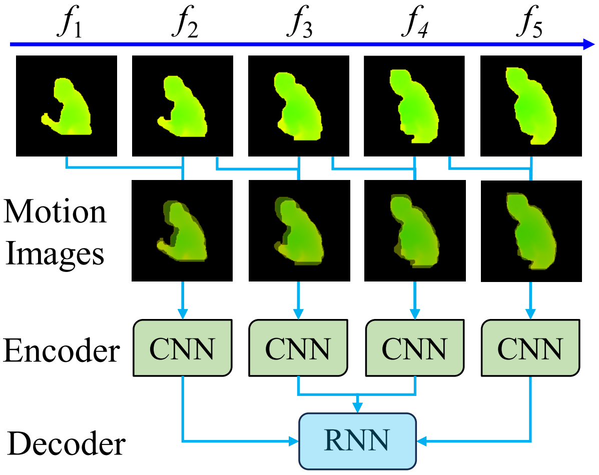
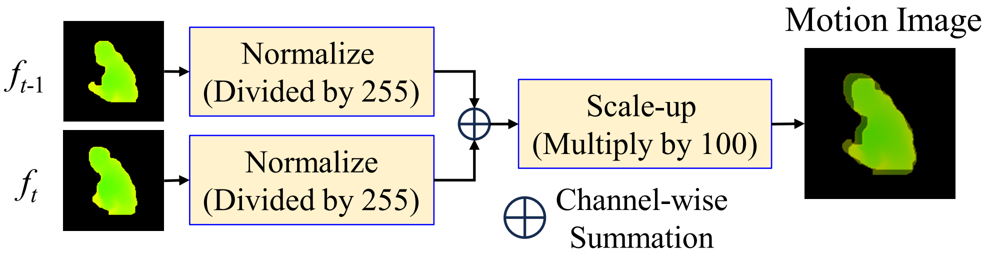


Fig. 5.17. Process of MotionCRNN Action Recognition Model

The model follows a multi-step process:

- **Motion image calculation**: Motion information is captured by calculating the difference between consecutive segmented person mask images as shown in **Fig. 5.18**. This creates a motion image sequence for each video frame.



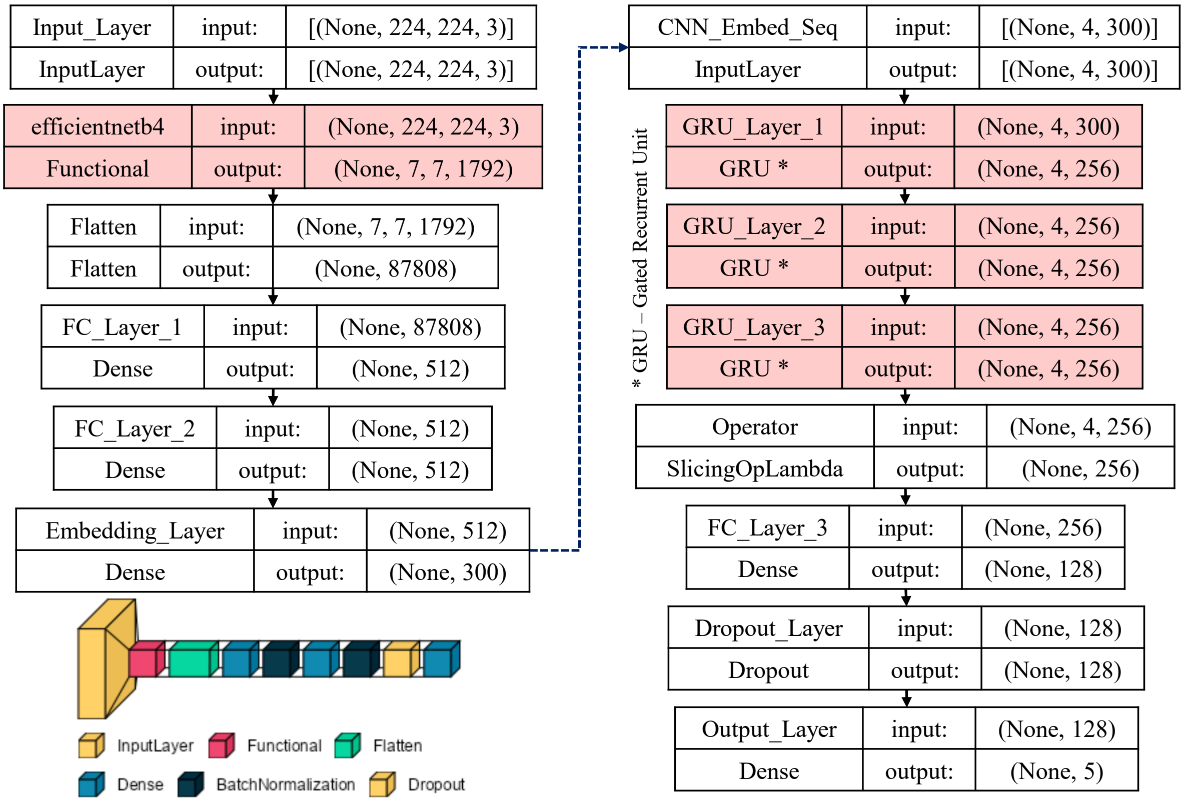
Motion Image
$f_{t\text{-}1}$
f_t
Normalize (Divided by 255)
Normalize (Divided by 255)
Scale-up (Multiply by 100)
Channel-wise Summation

Fig. 5.18. Calculation of Motion Image

- **Spatial feature extraction with CNN**: Transfer learning is applied to a pre-trained EfficientNetB4 [85] architecture to extract spatial features from the motion image sequences. EfficientNetB4 is chosen for its efficiency and lightweight pre-trained weights.

- **Temporal feature extraction with RNN**: Gated Recurrent Unit (GRU) layers [86] are employed within the RNN component to capture temporal dependencies in the motion information across consecutive frames. GRUs are specifically chosen for their effectiveness in handling these dependencies compared to Long Short-Term Memory (LSTM) networks or basic RNNs.

The specific model architectures of the CNN encoder and the RNN decoder are shown in **Fig. 5.19**.



Input_Layer	input:	[(None, 224, 224, 3)]
InputLayer	output:	[(None, 224, 224, 3)]
efficientnetb4	input:	(None, 224, 224, 3)
Functional	output:	(None, 7, 7, 1792)
Flatten	input:	(None, 7, 7, 1792)
Flatten	output:	(None, 87808)
FC_Layer_1	input:	(None, 87808)
Dense	output:	(None, 512)
FC_Layer_2	input:	(None, 512)
Dense	output:	(None, 512)
Embedding_Layer	input:	(None, 512)
Dense	output:	(None, 300)

CNN_Embed_Seq	input:	[(None, 4, 300)]
InputLayer	output:	[(None, 4, 300)]
GRU_Layer_1	input:	(None, 4, 300)
GRU *	output:	(None, 4, 256)
GRU_Layer_2	input:	(None, 4, 256)
GRU *	output:	(None, 4, 256)
GRU_Layer_3	input:	(None, 4, 256)
GRU *	output:	(None, 4, 256)
Operator	input:	(None, 4, 256)
SlicingOpLambda	output:	(None, 256)
FC_Layer_3	input:	(None, 256)
Dense	output:	(None, 128)
Dropout_Layer	input:	(None, 128)
Dropout	output:	(None, 128)
Output_Layer	input:	(None, 128)
Dense	output:	(None, 5)

(a) **(b)**

Fig. 5.19. CRNN Architecture: (a) CNN Encoder, (b) RNN Decoder

- **CNN encoder**: The CNN encoder leverages transfer learning from EfficientNetB4. The last Fully Connected (FC) layer is removed and replaced with two hidden FC layers with batch normalization, Rectified Linear Unit (ReLU) activation, and a dropout layer to prevent overfitting. An additional FC layer is added for feature embedding.
- **RNN decoder**: The RNN decoder consists of three unidirectional GRU layers, a dropout layer, and a final FC layer for action classification.
- **Feature fusion and classification**: The extracted spatial and temporal features from the CNN and RNN are combined and fed into the final classification layer to predict the action being performed (including transition states).

This strategic integration of CNNs and RNNs within the MotionCRNN framework allows for the effective extraction of both spatial and temporal features from the motion image sequences. This combined approach contributes to robust and accurate action recognition, particularly for identifying transition states between actions.

5.5.2 Experimental Result Analysis

In this section, the dataset preparation process, evaluation performance for each process on different datasets, analysis of the results, and refining process are described. Furthermore, various comparisons are performed to find the optimal solution.

5.5.2.1 Dataset Preparation

In the proposed system, trainable DL algorithms played a pivotal role in the implementation, encompassing the fusion of YOLOv5-SAM for person detection and segmentation, as well as EfficientNet and GRU for MotionCRNN-based action recognition. Diverse datasets were carefully prepared to facilitate the training of these algorithms. For the experiment, data from all three rooms of the care center were used firstly, emphasizing the inclusion of a varied dataset. Dataset preparation involves the standard practice of splitting data into training, validation, and testing datasets. Importantly, the data used for each dataset did not overlap, thereby ensuring that the data used for training were distinct from those included in the validation and testing datasets. Specific datasets for each stage of the process are described in detail in the following sections.

5.5.2.2 Training and Validation Datasets

For action recognition, multiple sequences of five consecutive images each were selected to train the MotionCRNN. These short sequences were collected from the three rooms to ensure a balanced dataset. In total, 13,600 sequences were chosen, with 70% (9,520 sequences) designated for training, and the remaining 30% (4,080 sequences) assigned to the validation dataset. These datasets were selected and organized to include diverse situations and ensure robust training of the respective algorithms.

The proposed system utilized the EfficientNetB4 pre-trained weight for CNN transfer learning and employed unidirectional GRUs for recurrent decision-making. In training the MotionCRNN, the cross-entropy loss function was calculated once every epoch for the CNN-RNN integration output, and the Adam optimizer was applied with the default learning rate

(0.001). The model was trained for 20 epochs with a batch size of 64. The performance evaluation of the training and validation datasets are presented in **Table 5.11**. The confusion matrix for each dataset is shown in **Fig. 5.20**. Emphasizing the transition state class, it is evident that the training process performed well, achieving over 99% accuracy for all evaluation metrics on both the training and validation datasets. However, some false and missing predictions persisted as can be seen in the confusion matrices, indicating areas for potential improvement, despite the overall strong performance in training for recognizing transition states.

Table 5.11. Performance Evaluation on Training and Validation Datasets

Metric	Dataset	Action				
		Transition	Seated	Standing	Sitting	Lying
Accuracy (%)	Training	99.60	99.96	99.83	99.94	99.85
	Validation	99.41	99.90	99.71	99.85	99.95
Precision (%)	Training	99.89	99.66	98.92	99.75	98.84
	Validation	99.70	99.22	98.26	99.41	99.61
Recall (%)	Training	99.31	100.0	99.75	99.75	100.0
	Validation	99.12	100.0	99.41	99.41	100.0
F1-score (%)	Training	99.60	99.83	99.33	99.75	99.42
	Validation	99.41	99.61	98.83	99.41	99.80

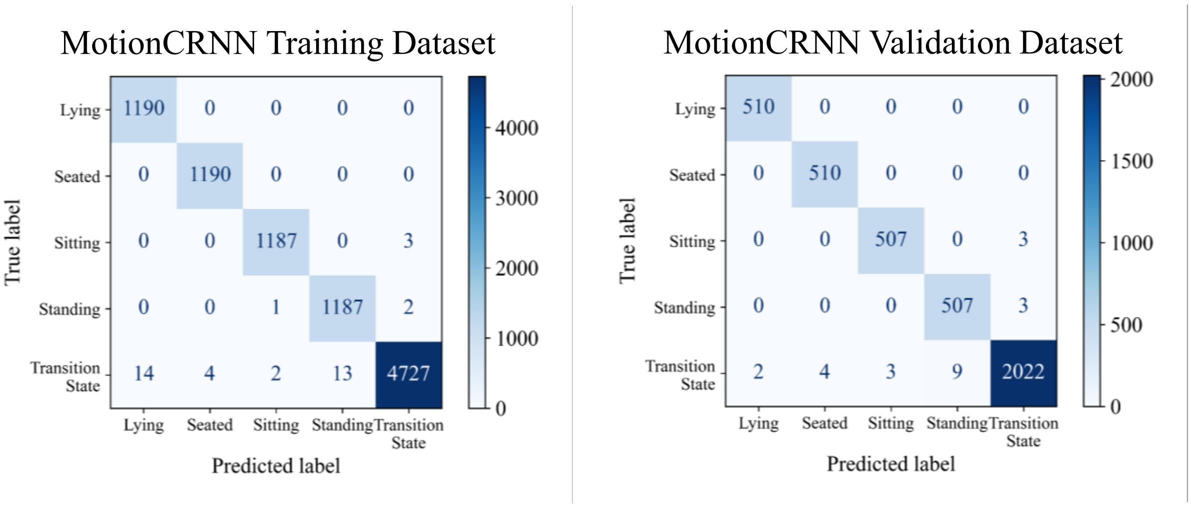


Fig. 5.20. Confusion Matrix for Training and Validation Datasets

5.5.2.3 Testing Dataset

The proposed algorithm was tested on the recorded datasets described in **Table 3.2**. However, because of the time-consuming and intensive nature of the ground-truth labeling task, only one 15-hour duration (approximately) random long sequence from each room was selected for performance evaluation. The sequences comprised up to 54,800 frames at 1fps and were recorded during both the daytime and nighttime. Detailed information on the selected testing data is presented in **Table 5.12**, where the actions included in each sequence are also described. Specifically, these testing data were selected to include the significant transition states observed in each room. For example, the Room 1 sequence highlighted transition states from seated in the wheelchair to standing and vice versa, whereas the Room 2 sequence covered a considerable number of transition states from sitting to lying down and vice versa. Finally, the Room 3 sequence mostly featured transition states from sitting to standing and vice versa, which are actions frequently performed by elderly residents. Reminding of the approach, a sliding window method was employed with a window size of 5 and a stride of 1 to process the long sequence.

Table 5.12. Testing Dataset Information

Room ID	Date and Time		Duration	Number of Frames	Included Action *
	Start Time	End Time			
1	2019/10/12 10:15:00	2019/10/13 00:39:00	14 hr 24 min	51,840	*A, L, O, Se, St, Tr*
2	2019/10/25 11:50:00	2019/10/26 02:50:00	15 hr	54,000	*A, L, O, Se, Si, Tr*
3	2019/10/12 11:10:00	2019/10/13 01:34:00	14 hr 24 min	51,840	*A, L, O, Se, Si, Tr*

**A*: Assistance, *L*: Lying, *O*: Outside, *Se*: Seated, *St*: Standing, *Si*: Sitting, *Tr*: Transition states

A visual representation of a sample 10-minute duration action recognition result from the Room 1 testing sequence is illustrated in **Fig. 5.21**, in which the top one is the scatter plot and the bottom two are the bar chart representations of ground truth and predictions across each time frame. By observing the visualization in this sample result, it can be seen that the person was in a transition state between standing and seated in the wheelchair frequently within the 10-minute duration. However, some results indicate the presence of over-segmentation errors, particularly during the transition state, as represented by the red color in **Fig. 5.21**. This issue arises because the model relies on Top-1 accuracy, assigning the most probable action for each prediction. To address this problem, a sequential-based majority voting decision and condition reasoning for transition states were implemented.

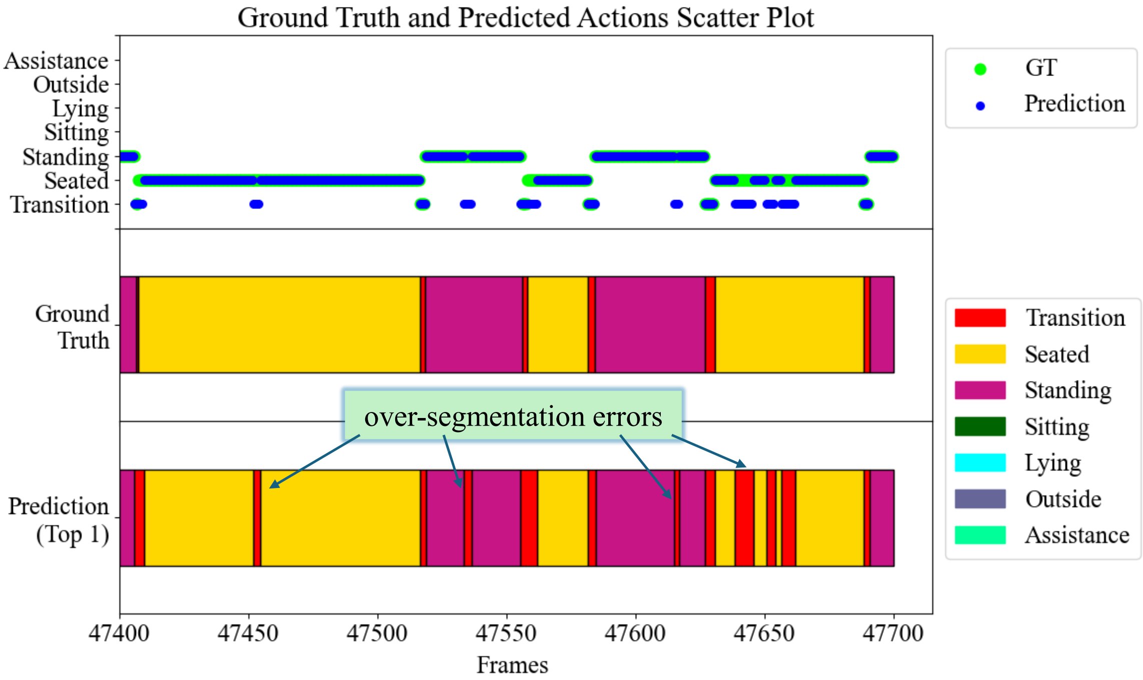
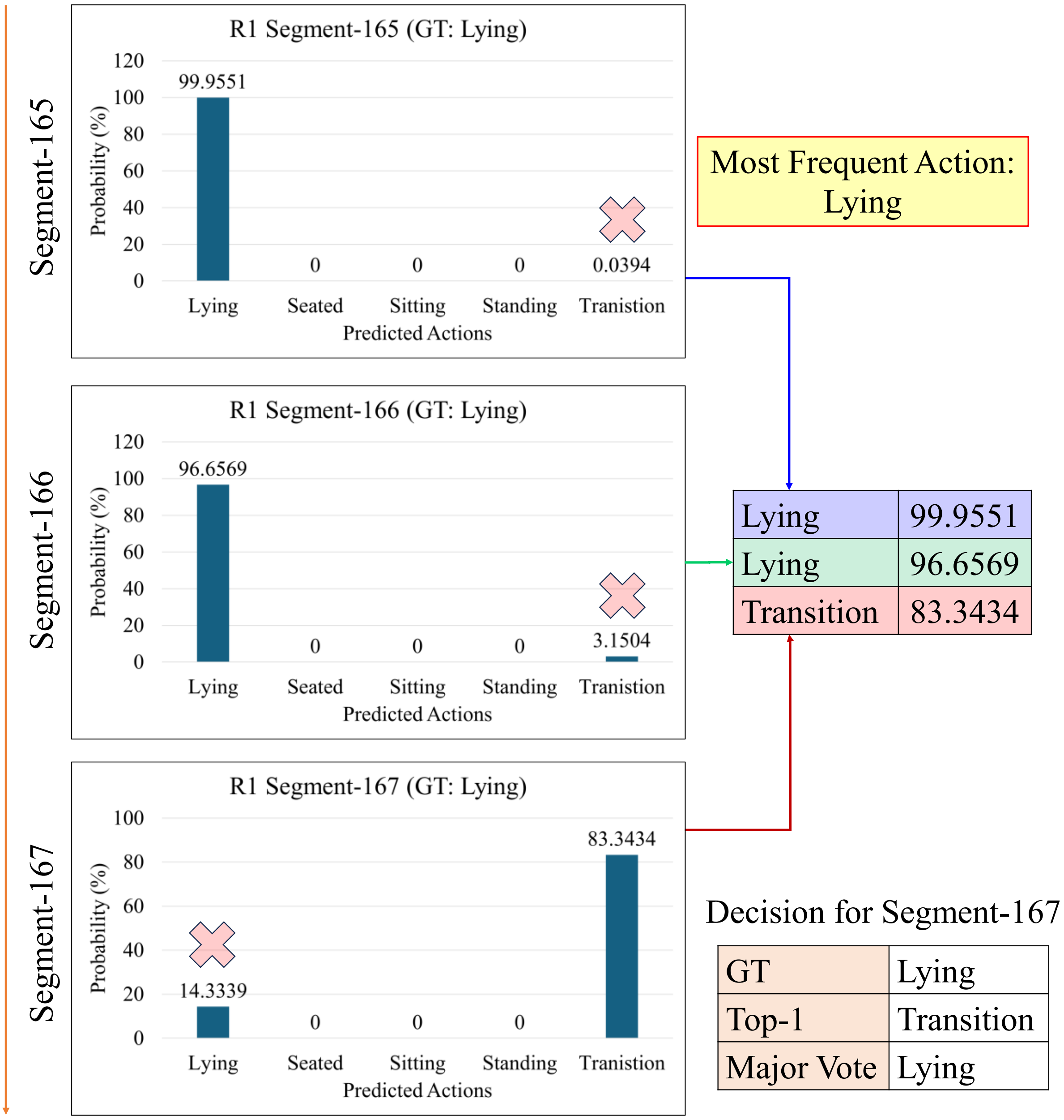


Fig. 5.21. Visual Representation of a Sample Action Recognition Result

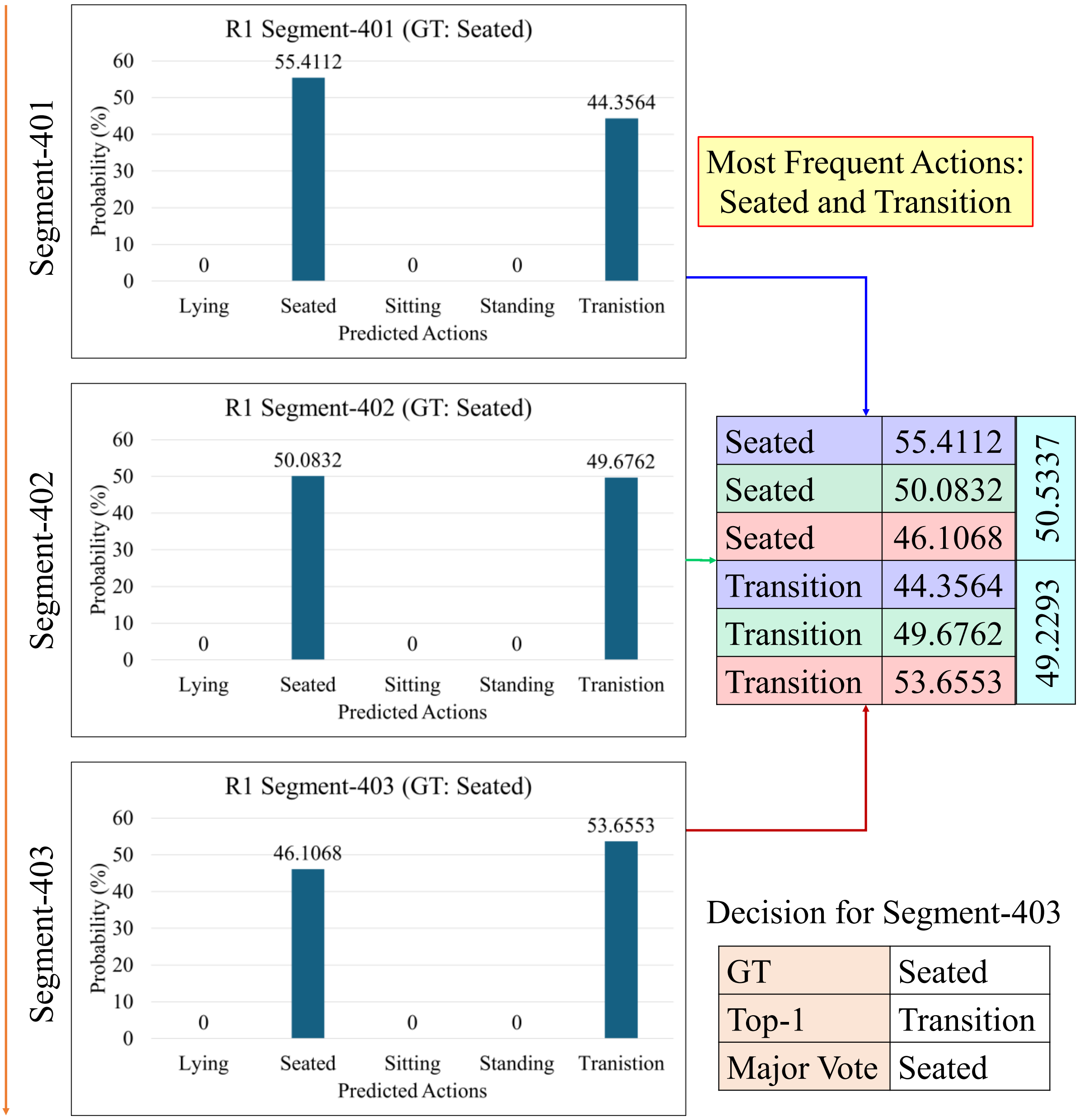
To implement the majority voting decision, Top-2 predicted labels were utilized, where "Top-2" refers to the two most probable predictions among the five prediction probabilities from the model. An illustration of the majority-voting decision is shown in **Fig. 5.22**. For instance, in **Fig. 5.22 (a)**, to determine the predicted label for Segment-167 (bottom graph), the previous two segments (Segments 166 and 165) were considered, and the Top-2 predicted labels for each segment were checked. The small probability values were then removed using a threshold of 20 and the remaining probability values and labels were examined. In this example, two labels were identified as "lying" and one label as a "transition state." Hence, the most frequent action was determined as "lying" for Segment-167. It is evident that the ground-truth label was "lying," and the majority voting decision also indicated "lying," which achieved a better result than the Top-1 label, which was a "transition state." However, there were conditions in which the thresholded values resulted in the same number of labels, as shown in **Fig. 5.22 (b)**. In such cases, the average probability values for each label were obtained and the decision was determined as the label with the highest average probability value. For this example, the "seated" label has the highest probability of 50.53%. Hence, even though the Top-1 label was a "transition state," majority voting correctly identified the action as "seated."



Lying	99.9551
Lying	96.6569
Transition	83.3434

GT	Lying
Top-1	Transition
Major Vote	Lying

(a) Decision for Segment-167



Seated	55.4112	50.5337
Seated	50.0832	
Seated	46.1068	
Transition	44.3564	49.2293
Transition	49.6762	
Transition	53.6553	

GT	Seated
Top-1	Transition
Major Vote	Seated

(b) Decision for Segment-403

Fig. 5.22. Illustration of Majority Voting Decisions

In this experiment, a transition state was generally defined as a state that changes from one action to another. However, even after applying majority voting, there were instances of false predictions as transition states throughout the long sequence. This problem occurred because of the model's lack of reasoning capabilities. Applying reasoning to the predictions of DL models is crucial for real-world effectiveness. Hence, to enhance the recognition results, a reasoning step was introduced that specified that a transition state should not occur between the same specific actions. In cases in which this condition occurred, the system refined the results after a certain period (1 hour in this experiment) by replacing the predicted transition states with specific actions before or after the transition state. This approach aimed to improve the accuracy and reliability of the recognition results by integrating conditional reasoning into the prediction process.

A comparison between the Top-1 and the final refined results is shown in **Fig. 5.23**. Upon checking the visualization, it is evident that the two refined approaches (sequential-based majority voting and transition state reasoning) smoothened the action recognition results and reduced the over-segmentation errors among the predicted actions. By examining the resulting visualization, users can make decisions regarding the actions of their intended residents regarding health monitoring. An example of decision-making for the results in **Fig. 5.23** could be: “Resident A is observed standing for a while, then transitioning to being seated in the wheelchair within 10 minutes. During this period, there are frequent transitions between seated and standing positions.” Such insights allow caregivers to understand residents’ activities over time, facilitating informed decision-making and appropriate interventions, as needed.

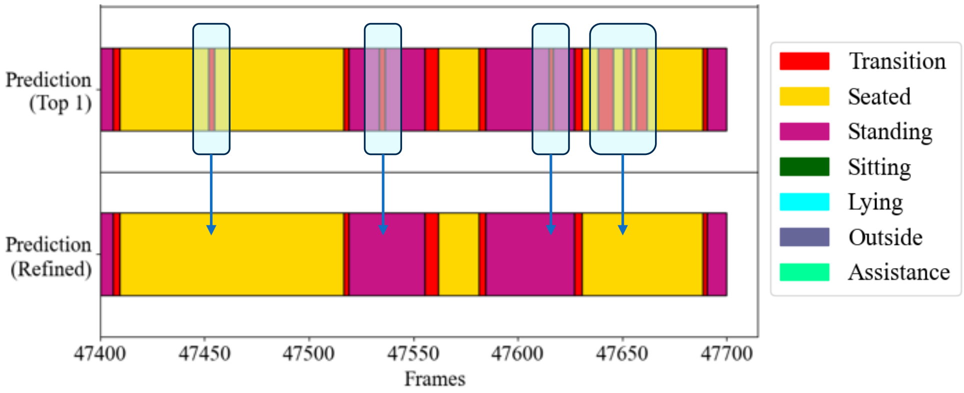


Fig. 5.23. Visualization of Action Recognition with Refinements

The action recognition performance on the evaluation metrics and confusion matrix after the refinements are shown in **Table 5.13** and **Fig. 5.24**. In all testing sequences, the “outside” state was included, which is easy to identify even after person detection. Therefore, the evaluation was performed after excluding the “outside” state; however, it was included in the confusion matrix. In all three testing sequences, although the performance was promising, there was still some confusion between the actions. According to the result analysis, some false recognition cases were identified, primarily attributed to occlusion, low-quality segmented person masks, and misalignment of the transition states.

- **Occlusion**: When an elderly resident is blocked from the camera view by a nurse or caregiver, the system only detects the caregiver and predicts their actions instead. This leads to false detections between “assistance” and other actions due to the person detection model mistaking the situation for a single person.

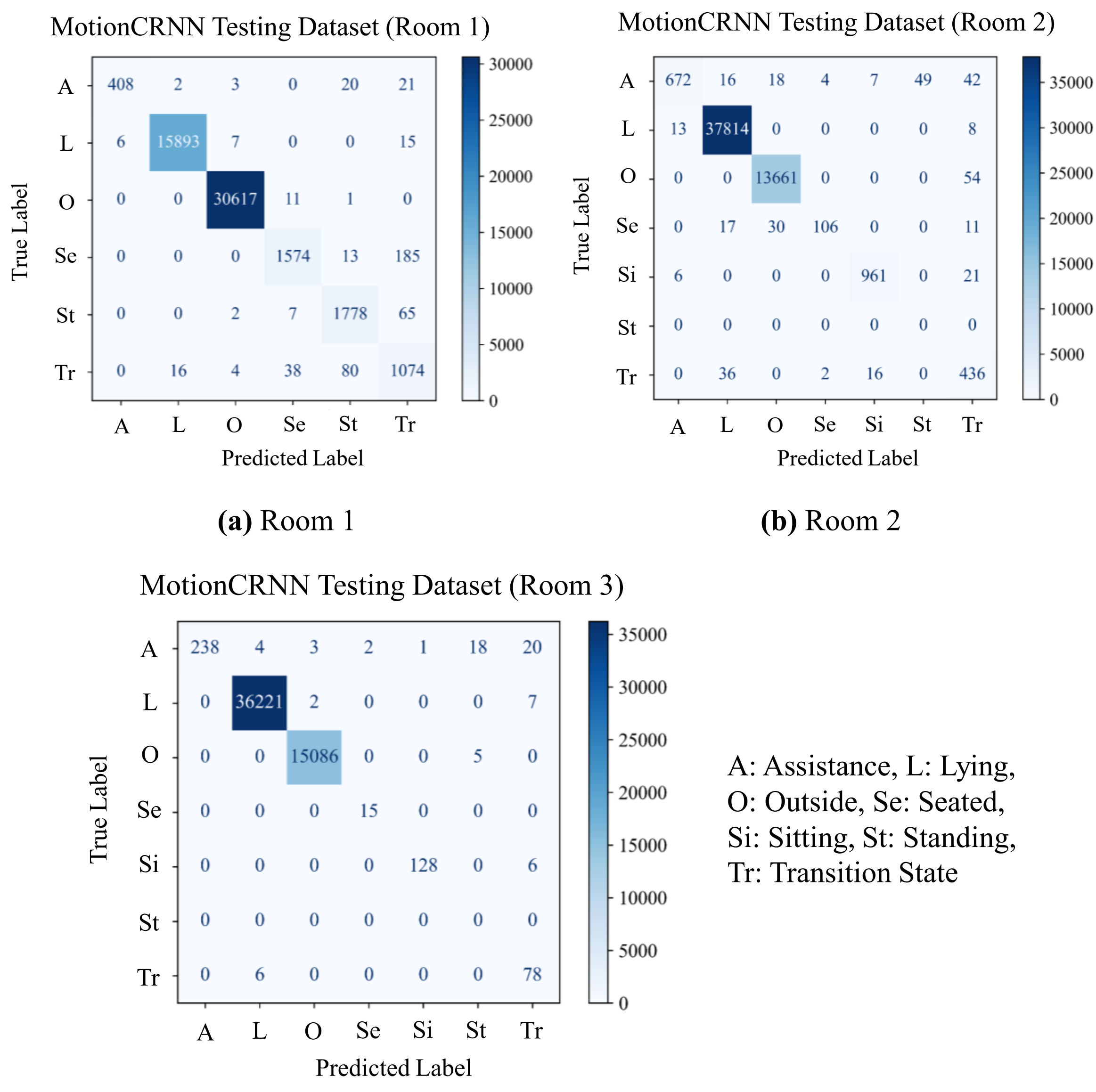
- **Inaccurate person masks**: Confusion between specific actions (seated in the wheelchair, sitting, standing, and lying down) occurred because of the inaccuracy of the extracted person masks segmented from the person segmentation process.
- **Misaligned transition states**: Discrepancies between the ground truth labels for transition states and the model's predictions can occur. This misalignment can result in predicted transitions appearing at different times compared to the actual events.

These challenges contribute to lower precision and recall values in the evaluation metrics.

Table 5.14 highlights the results after refinements, specifically focusing on transition state recognition and excluding the "outside" state. Although there are still some areas for improvement, the experimental results are promising, highlighting the key contribution of this study. MotionCRNN with result refinement achieved an average accuracy of 99.19% and an average F1-score of 83.39%, demonstrating its effectiveness in differentiating transition states from other specific actions.

Table 5.13. Performance Evaluation on Testing Dataset

Metric	Room ID	Action						Average (%)
		Transition	Seated	Standing	Sitting	Lying	Assistance	
Accuracy (%)	1	98.02	98.85	99.13	-	99.82	99.77	99.12
	2	99.66	99.92	-	99.88	99.78	99.78	99.80
	3	99.89	99.99	-	99.98	99.95	99.93	99.95
Precision (%)	1	78.97	97.22	94.02	-	99.89	98.55	93.73
	2	84.17	94.64	-	97.66	99.82	97.25	94.71
	3	70.27	88.24	-	99.22	99.97	100.0	91.54
Recall (%)	1	88.91	88.83	96.11	-	99.87	90.47	92.84
	2	88.98	79.10	-	97.27	99.94	90.69	91.20
	3	92.86	100.0	-	95.52	99.98	89.81	95.63
F1-score (%)	1	83.65	92.84	95.05	-	99.88	94.34	93.15
	2	86.51	86.18	-	97.46	99.88	93.86	92.78
	3	80.00	93.75	-	97.33	99.97	94.63	93.14



(a) Room 1

(b) Room 2

(c) Room 3

Fig. 5.24. Confusion Matrix for Testing Sequences

Table 5.14. Transition State Recognition Performance

Action	Room ID	Evaluation Metrics (%)			
		Accuracy	Precision	Recall	F1-score
Transition	1	98.02	78.97	88.91	83.65
	2	99.66	84.17	88.98	86.51
	3	99.89	70.27	92.86	80.00
	Avg	99.19	77.80	90.25	83.39

5.5.2.4 Comparison of Experiments

To comprehensively evaluate the MotionCRNN model's performance, various comparative experiments were conducted, exploring the impact of different factors. The following subsections will detail the results of these comparisons.

5.5.2.4.1 Impact of Refinement on Action Recognition

Table 5.15 compares the overall recognition performance, particularly for transition states, before and after incorporating various refinements. These refinements include bounding box recovery, majority voting, and reasoning for transition states. The results indicate that while the recall rate for all three rooms decreased slightly, improvements in precision and F1-score rates were evident, especially in Rooms 2 and 3, where they increased by up to 65.98% compared with the results before refinements. Among the refinement processes, conditional reasoning for transition states had the most significant impact on increased recognition rates.

Table 5.15. Impact of Refinements on Transition State Recognition

Metric	Room ID	Transition State Recognition		
		Before Refinement	After Refinement	Impact
Precision (%)	1	65.26	78.97	+13.71
	2	18.19	84.17	+65.98
	3	14.83	70.27	+55.44
Recall (%)	1	92.05	88.91	-3.14
	2	91.84	88.98	-2.86
	3	97.62	92.86	-4.76
F1-score (%)	1	76.37	83.64	+7.27
	2	30.36	86.51	+56.15
	3	25.75	80.00	+54.25

5.5.2.4.2 Processing Time Analysis

The testing process achieved real-time performance (processing time less than video duration) for 1-hour video sequences captured at 1fps. This efficiency is maintained even with initial depth data processing and colorization. While increasing the frame rate to 2.5fps resulted in a processing time of 1.5 hours, no significant improvement in accuracy was observed. This suggests that the 1fps frame rate offers a good balance between processing speed and accuracy for real-time applications.

5.5.2.4.3 CNN Base Model Comparison

Furthermore, various EfficientNet architectures were tested to determine whether the model could be enhanced by changing its base model. Four model variants were used for comparison: one EfficientNetB4 and three EfficientNetV2 models [87] (V2L, V2M, and V2S), which were tested in three testing rooms. A comparison of the overall accuracy and processing time of each variant is presented in **Table 5.16**, where the accuracy was calculated for all classes with and without including the "outside". Average processing time is based on a 1-hour duration sequence.

The results indicate that while there were no significant differences in overall accuracy between the models, processing time varied considerably. EfficientNetB4 offered the best balance between accuracy and processing efficiency. For example, on the Room 1 dataset, EfficientNetB4 processed a 1-hour sequence in an average of 28.14 minutes.

These findings suggest that while alternative EfficientNet architectures may not significantly impact accuracy, they can influence processing time. This highlights the importance of considering the trade-off between accuracy and computational efficiency when selecting a CNN model for real-time applications.

Table 5.16. Overall Accuracy and Processing Time Comparison (EfficientNet Variants)

EfficientNet Model	Room ID	Overall Accuracy (%)		Average Processing Time
		All Classes	Excluding "Outside"	
B4	1	99.04	97.79	28 min
	2	99.35	99.51	28 min
	3	99.86	99.88	28 min
V2L	1	98.67	96.85	37 min
	2	99.20	99.29	37 min
	3	99.90	99.88	37 min
V2M	1	98.59	96.73	34 min
	2	99.10	99.17	33 min
	3	99.90	99.88	30 min
V2S	1	98.84	97.32	29 min
	2	99.26	99.37	29 min
	3	99.89	99.87	29 min

5.5.2.4.4 System Comparison

The proposed MotionCRNN model was evaluated against related works for recognizing the daily activities of elderly individuals. While all these systems share the same goal, they utilize different technologies, data types, and recognition models. **Table 5.17** presents a comparison of the proposed system with recent studies employing distinct methodologies. Key factors considered include:

- **Input data type**: This highlights whether the system uses depth data or other modalities like RGB images or wearable sensors.
- **Real-world data**: This indicates if the system was evaluated on real-world datasets captured in practical settings or public datasets.
- **Transition state recognition**: This emphasizes whether the system recognizes transitions between activities, which is crucial for elderly monitoring.
- **Recognition model architecture**: This specifies the type of model used for activity recognition.
- **Privacy preservation**: This highlights how the system addresses privacy concerns, such as using depth data which avoids capturing personal details.

The comparison results indicate that the proposed MotionCRNN achieved an average accuracy of 99.42% for recognizing seven actions, outperforming the previous (SVM + HMM) method which has a recognition rate of 84.04%. Both are tested on three elderly datasets from the care center. This approach prioritized both privacy by utilizing depth data and real-world reliability through the use of real-world data, which is one of the contributions of this work. In addition, it captured the crucial transition states vital for elderly monitoring.

While another sensor-based approach achieved transition-aware recognition [52], its accuracy was limited to 80%. Notably, while state-of-the-art hybrid DL recognition models [30], [70], [71] obtained high accuracy, they were not considered for the application with real-time processing, privacy concerns, or transition state recognition. However, it is remarkable that although most of the other systems used public datasets, this study used custom real data; thus, this difference in data sources can influence the reported accuracy levels.

Table 5.17.

(a) System Comparison - 1

Related Work (Year)	Data Type	Real Data	Total Action	Transition Awareness
[30] (2023)	Sensor	✗	6	✗
[52] (2020)	Sensor	✓	9	✓
[70] (2023)	RGB	✗	101	✗
[71] (2020)	RGBD	✗	27	✗
Proposed Method	Depth	✓	7	✓

(b) System Comparison - 2

Related Work (Year)	Recognition Model	Real-Time	Privacy-Preserving	Average Accuracy (%)
[30] (2023)	CNN-LSTM	N/A	✗	99.00
[52] (2020)	STD-TA	N/A	✗	80.00
[70] (2023)	Vit-ReT	✓	✗	94.70
[71] (2020)	Deep CNN	N/A	✓	87.21
Proposed Method	SVM + HMM	✓	✓	84.04
	MotionCRNN			99.42

5.5.2.4.5 Extended Testing on Different Datasets

To assess the model's ability to adapt to different environments, it was tested on data from a hospital setting. This environment featured significantly different camera positions and structures compared to the care center used for training. Three elderly datasets from the hospital are used for evaluation. While a smaller dataset of 600 new sequences was used for initial action recognition training, transfer learning enabled the effective evaluation of three 1-hour testing sequences (3,600 frames each) from the new environment. The results presented in **Table 5.18**, ranging from 84.83% to 99.22% overall accuracy rates, demonstrate the potential of the system as a foundational model that can adapt to diverse settings with minimal additional data requirements. This highlights another key strength of this study.

Table 5.18. Performance Evaluation on Extended Dataset

Room ID	Date and Time		Included Action [*]	Overall Accuracy (%)
	Start Time	End Time		
4	2024/01/06 10:45:00	2024/01/06 11:45:00	*A, L, St, Tr*	99.22
5	2024/01/27 07:02:00	2024/01/27 08:02:00	*A, L, St, Si, Tr*	84.83
6	2024/01/31 19:56:00	2024/01/31 20:56:00	*A, L, O, Si, Tr*	94.89

[*] *A*: Assistance, *L*: Lying, *O*: Outside, *St*: Standing, *Si*: Sitting, *Tr*: Transition states

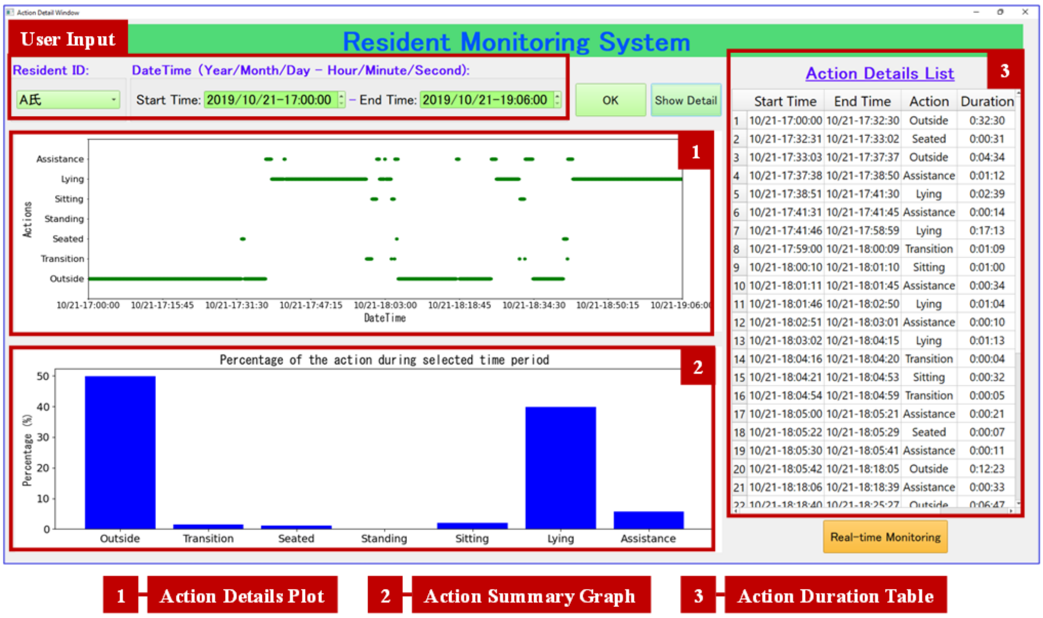
5.5.2.4.6 Discussion

A unique contribution of this experiment is the application of MotionCRNN to image sequences for action recognition. This approach incorporated motion information into a hybrid CNN-RNN architecture, which is valuable for identifying transition states that rely heavily on movement patterns. The system achieved a high accuracy of 99.42% in recognizing not only the transition states but also various specific actions in real time. Through experimentation, the model architecture and parameters were optimized, further refining the results with sequential-based majority voting, and condition reasoning to enhance action recognition performance.

5.6 Graphical User Interface (GUI)

A GUI specifically designed for end users, including family members and health caregivers, was developed to facilitate the real-time monitoring of elderly individuals and access detailed action information captured by the proposed action recognition system. The GUI consisted of two main windows, as shown in **Fig. 5.25**. The first window is the action detail window, where users can select the name or ID of the elderly resident they wish to monitor and input the desired start and end times to view the detailed information. The GUI then displays the recognized actions of the selected resident on a scatter plot, providing a second-by-second representation. Additionally, a bar chart summarizes the actions performed during the specified time frame. For a more comprehensive view of continuous actions, users can refer to a table that lists the specific durations of the consecutive actions. The GUI design ensures that end users and healthcare providers can easily access insightful information within a single window.

The second window in the GUI is a real-time monitoring window. Similar to the action detail window, users can input relevant information to either re-play or monitor the actions of elderly residents in real time. This feature allows users to validate the accuracy of previously captured action details, thereby providing reassurance and confidence in the system's performance. In summary, this GUI serves as a comprehensive tool for caregivers to monitor the elderly in real time, access detailed action information, and interact with the analytics and recognition processes of the system.



Resident ID: A氏

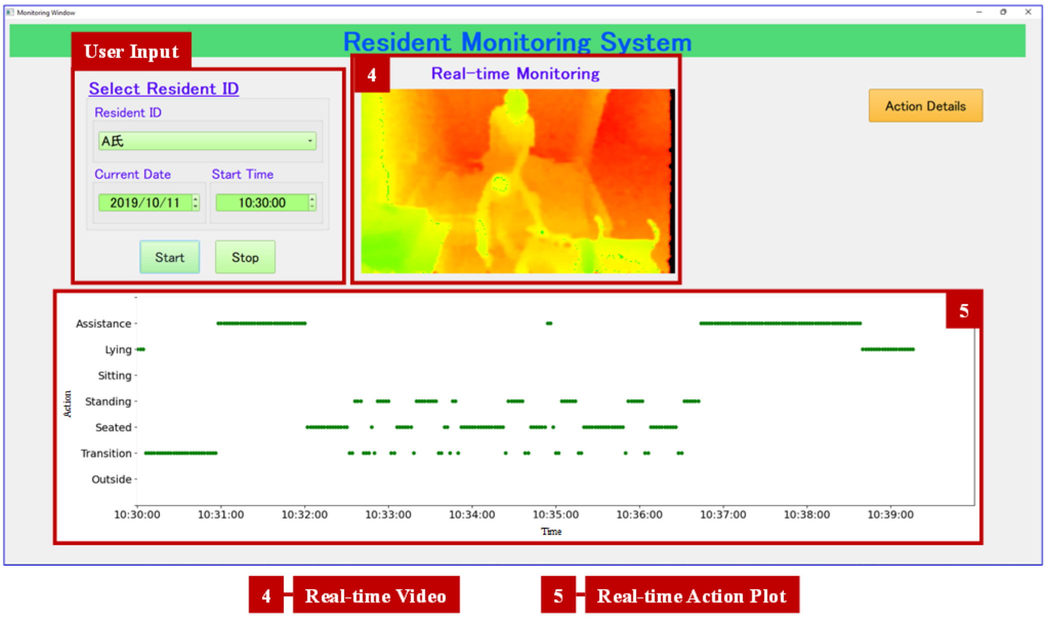
DateTime (Year/Month/Day - Hour/Minute/Second):

Start Time: 2019/10/21-17:00:00 - End Time: 2019/10/21-19:06:00

Action Details List

	Start Time	End Time	Action	Duration
1	10/21-17:00:00	10/21-17:32:30	Outside	0:32:30
2	10/21-17:32:31	10/21-17:33:02	Seated	0:00:31
3	10/21-17:33:03	10/21-17:37:37	Outside	0:04:34
4	10/21-17:37:38	10/21-17:38:50	Assistance	0:01:12
5	10/21-17:38:51	10/21-17:41:30	Lying	0:02:39
6	10/21-17:41:31	10/21-17:41:45	Assistance	0:00:14
7	10/21-17:41:46	10/21-17:58:59	Lying	0:17:13
8	10/21-17:59:00	10/21-18:00:09	Transition	0:01:09
9	10/21-18:00:10	10/21-18:01:10	Sitting	0:01:00
10	10/21-18:01:11	10/21-18:01:45	Assistance	0:00:34
11	10/21-18:01:46	10/21-18:02:50	Lying	0:01:04
12	10/21-18:02:51	10/21-18:03:01	Assistance	0:00:10
13	10/21-18:03:02	10/21-18:04:15	Lying	0:01:13
14	10/21-18:04:16	10/21-18:04:20	Transition	0:00:04
15	10/21-18:04:21	10/21-18:04:53	Sitting	0:00:32
16	10/21-18:04:54	10/21-18:04:59	Transition	0:00:05
17	10/21-18:05:00	10/21-18:05:21	Assistance	0:00:21
18	10/21-18:05:22	10/21-18:05:29	Seated	0:00:07
19	10/21-18:05:30	10/21-18:05:41	Assistance	0:00:11
20	10/21-18:05:42	10/21-18:18:05	Outside	0:12:23
21	10/21-18:18:06	10/21-18:18:39	Assistance	0:00:33
22	10/21-18:18:40	10/21-18:25:27	Outside	0:06:47

(a) Action Details Window



Select Resident ID

Resident ID: A氏

Current Date: 2019/10/11

Start Time: 10:30:00

(b) Real-time Monitoring Window

Fig. 5.25. Graphical User Interface

5.7 Conclusion

This chapter explored various approaches for action recognition in the context of elderly monitoring. Three main approaches were investigated:

- **HOG features-based approach (SVM, HMM + ML)**: This traditional approach relies on hand-crafted features like HOG to represent the sequence of data. While these methods can achieve reasonable accuracy, especially the (SVM + HMM) method, they require significant domain knowledge for feature extraction and may struggle to capture complex temporal dynamics within activities.
- **Temporal features-based approach (SVM with motion information)**: This approach utilizes SVM to classify temporal features derived from body posture changes. These features capture motion information but cannot learn complex relationships between consecutive frames. This limitation can hinder the recognition of transition states.
- **Spatiotemporal features-based approach (MotionCRNN)**: This approach leverages DL architectures like CNN and RNN to automatically learn spatiotemporal features from the sequence of data. The proposed MotionCRNN model integrates the motion information with a CNN for spatial feature extraction and an RNN for capturing temporal dependencies between consecutive frames.

To conclude, the MotionCRNN model offers a robust and generalizable solution for action recognition in elderly monitoring applications. By combining DL with well-designed refinements (sequential-based majority voting and transition state reasoning), MotionCRNN effectively recognizes not only primitive actions but also critical transition states, providing valuable insights for caregivers. The system’s focus on privacy and real-time processing further strengthens its potential for real-world deployment.

Finally, a user-friendly GUI was designed to provide a platform for offline interaction between caregivers and the system, offering insights into the health trends and details of the activities of the elderly.

Chapter 6

Overall Conclusion and Future Research

This chapter concludes the exploration of action recognition for elderly monitoring. The research objectives outlined in Chapter 1 will be recalled and the key findings of the proposed system will be summarized. This chapter will discuss the overall effectiveness of the system in real-world scenarios, highlighting its strengths and potential limitations. Finally, the potential contributions of this research to the field of elderly care technology will be explored and promising directions for future research will be outlined.

6.1 Research Summary and Realization

This research aimed to develop a comprehensive activity monitoring and behavior analysis system to support the well-being of elderly individuals and reduce the burden on caregivers. It achieved this goal by leveraging cutting-edge technologies such as Deep Learning (DL), advanced Computer Vision (CV), Hidden Markov Model (HMM), and sequential analysis. The system prioritizes user privacy by utilizing stereo depth cameras, ensuring real-time monitoring without capturing personal details. To enhance its effectiveness, the research focused on the following key contributions:

(1) **Depth cameras for elderly monitoring**: This research explored the application of stereo depth cameras for real-time action recognition in indoor environments, prioritizing privacy preservation.

(2) **Transition state recognition**: We investigated various approaches using spatiotemporal features to achieve robust recognition of critical transition states between daily activities, offering valuable insights into residents' behavior patterns.

(3) **Hybrid HMM combinations**: The research investigated the effectiveness of combining HMMs with Machine Learning (ML) models for real-time action classification. The optimal hybrid (SVM + HMM) achieved an average accuracy of 84.14% for recognizing actions from three elderly datasets of the care center.

(4) **Convolutional Recurrent Neural Network (CRNN) integrations**: Leveraging CRNN in conjunction with motion information derived from body posture changes, the system achieves robust recognition of transition states. The proposed MotionCRNN model achieved an average accuracy of 99.42% for recognizing seven actions, including transition states. Moreover, it yielded a remarkable F1-score of 83.39% for transition state recognition, demonstrating its effectiveness in capturing these crucial moments. Compared to the (SVM + HMM) approach, the CRNN model achieved significantly higher accuracy.

(5) **Real-world validation**: Extensive testing with real-world data collected from elderly facilities (care center and hospital) validates the system's reliability and generalizability, demonstrating its effectiveness in real-world settings.

By fulfilling the objectives and delivering key contributions, this research provides a significant advancement in the field of elderly care technology. The proposed system offers real-time activity monitoring, facilitates early detection of potential health concerns, and empowers caregivers with valuable insights into resident behavior.

Overall, this system can aid elderly individuals to age safely, facilitating smarter living with the help of Artificial Intelligence (AI). Additionally, it can be deployed in smart care centers for remote monitoring and access to health details through a user-friendly Graphical User Interface (GUI), promoting independent living, and assisting caregivers. Furthermore, the effective recognition of specific actions and transition states can provide valuable insights into the well-being of the elderly, aiding in the early detection of potential health issues related to mobility and balance. It is important to recognize that modern technology can benefit all generations. By educating and assisting the elderly in using smart devices and tools, they can be empowered to experience independent living and smarter aging, especially as the elderly population continues to grow.

6.2 Limitations

While this study demonstrates the effectiveness of the proposed system for elderly activity monitoring, there are still some limitations. The first one is the trade-off between image quality and processing speed. The system utilizes depth images with a resolution of 320×180 pixels to achieve a balance between image detail, storage efficiency, and real-time processing. Higher resolutions offer more detail but could impact processing speed. Future research could explore techniques for optimizing image compression or utilizing more powerful hardware to enable higher resolutions without sacrificing real-time performance.

The second one is the potential for improved person detection and segmentation. In this study, person detection and segmentation using You Only Look Once (YOLOv5) and Segment Anything Model (SAM) models form the foundation of the system. While achieving promising results, the accuracy of these processes can be further enhanced. Exploring newer YOLO versions or alternative segmentation algorithms could lead to improved detection and segmentation accuracy, potentially leading to better action recognition performance.

The third one is related to the camera-to-person distance and generalizability. The current system converts depth images to colorized images for compatibility with RGB-based object detectors. However, achieving optimal performance depends on maintaining a specific camera-to-person distance consistent with the training data. Significant deviations from this distance can lead to detection errors. Future research can investigate leveraging 3D processing techniques that directly utilize depth information for person detection. This approach eliminates dependence on colorization and enables distance-based detection, potentially improving generalization and robustness in various environments.

Finally, the current system is optimized for single-resident scenarios. In multi-person settings, person tracking is crucial to handle occlusion and identify individual activities accurately. Future research should explore incorporating person tracking algorithms to enable robust monitoring in environments with multiple residents.

6.3 Future Research

Current research has established a robust framework for recognizing the daily activities and transition states of elderly residents. This makes the way for exciting future directions as shown in **Fig. 6.1**.

(1) Integration with Cloud Computing and Big Data Analysis

(a) **Resident profiling**: Integrate the system with cloud computing to automatically generate resident profiles based on recognized activities.

(b) **Activity analysis**: Leverage big data computing to analyze daily activity dairies and sleep patterns extracted from hourly or daily data.

(c) **Predictive analytics**: Employ ML and AI to predict health trends and wellness for individuals and groups based on the generated resident profiles.

(d) **Alerting system**: Develop an alerting system to flag potential health concerns like falls, extended sleep durations, or frequent bathroom visits.

(2) Real-world deployment and scalability

(a) **Camera streaming integration**: Test the system's functionality with camera streaming in real-world settings like hospitals, care centers, and smart homes.

(b) **Edge computing deployment**: Investigate modifying the model for deployment on edge devices with limited computational resources such as Mini PCs or Raspberry Pi computers. This would enable on-device processing and reduce reliance on cloud infrastructure.

These future directions could significantly enhance the system's capabilities, transforming it from a monitoring tool to a comprehensive platform for proactive care and health management. By integrating big data analysis and predictive models, the system can empower caregivers and healthcare professionals to detect potential health issues and take action early, ultimately improving the quality of life for elderly populations.

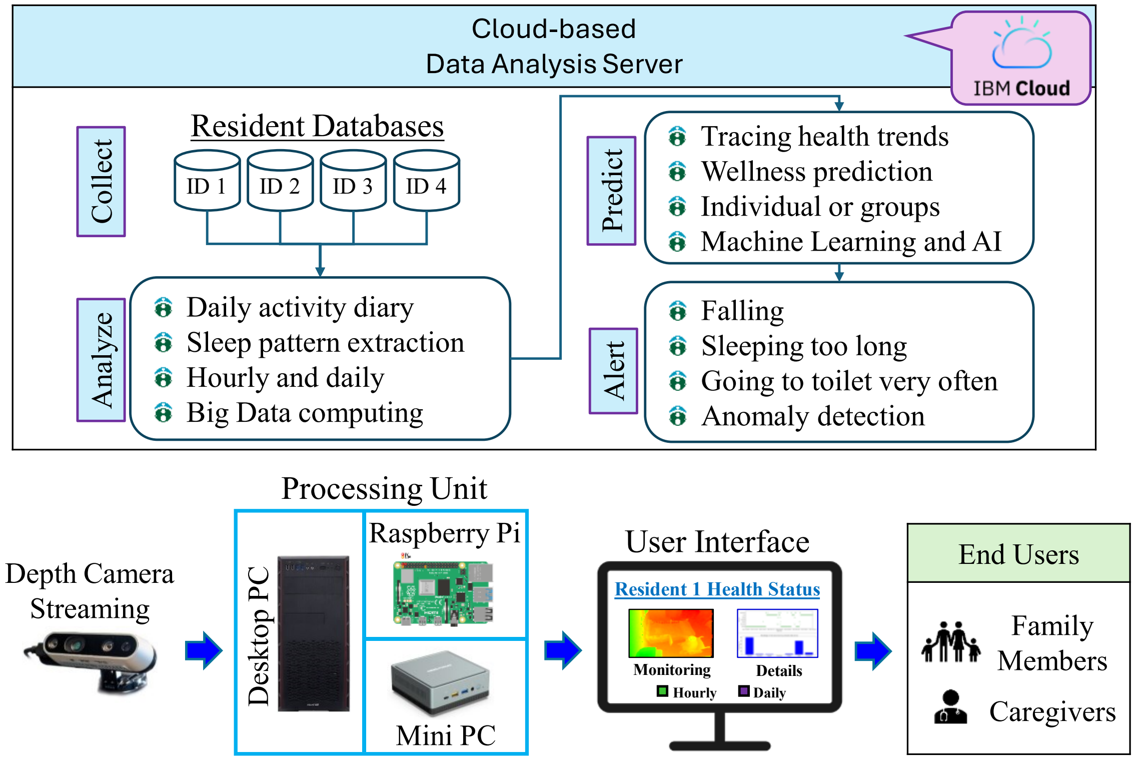


Fig. 6.1. Future Research

References

[1] "Demography - elderly population - OECD data," theOECD. Accessed: Feb. 20, 2024. [Online]. Available: http://data.oecd.org/pop/elderly-population.htm

[2] World Population Prospects—Population Division—United Nations. Accessed: Mar. 31, 2022. [Online]. Available: https://population.un.org/wpp/

[3] E. Freiberger, C. C. Sieber, and R. Kob, "Mobility in older community-dwelling persons: a narrative review," *Front. Physiol*, vol. 11, article no. 881, 2020. doi: https://doi.org/10.3389/fphys.2020.00881

[4] L. W. Keeler and M. J. Bernstein, "The future of aging in smart environments: four scenarios of the United States in 2050," *Futures*, vol. 133, article no. 102830, 2021. doi: https://doi.org/10.1016/j.futures.2021.102830

[5] I. Y. Song et al., "The landscape of smart aging: topics, applications, and agenda," *Data Knowl. Eng.*, vol. 115, pp. 68–79, 2018. doi: https://doi.org/10.1016/j.datak.2018.02.003

[6] M. P. De Freitas et al., "Artificial intelligence of things applied to assistive technology: a systematic literature review," *Sensors*, vol. 22, no. 21, article no. 8531, 2022. doi: https://doi.org/10.3390/s22218531

[7] M. E. N. Gomes et al., "Multi-human fall detection and localization in videos," *Comput. Vis. Image Und.*, vol. 220, article no. 103442, 2022. doi: https://doi.org/10.1016/j.cviu.2022.103442

[8] E. Teixeira et al., "Wearable devices for physical activity and healthcare monitoring in elderly people: a critical review," *Geriatrics*, vol. 6, no. 2, article no. 38, 2021. doi: https://doi.org/10.3390/geriatrics6020038

[9] A. Zhavoronkov et al., "Artificial intelligence for aging and longevity research: recent advances and perspectives," *Ageing Res. Rev*. vol. 49, pp. 49-66, 2019. doi: https://doi.org/10.1016/j.arr.2018.11.003

[10] Y. Yazdi and S. Acharya, "A new model for graduate education and innovation in medical technology," *Ann. Biomed. Eng*, vol. 41, pp. 1822-1833, 2013. doi: https://doi.org/10.1007/s10439-013-0869-4

[11] Assistive Technology. Accessed: Mar. 31, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/assistive-technology

[12] G. M. Weiss, K. Yoneda, and T. Hayajneh, “Smartphone and smartwatch-based biometrics using activities of daily living,” *IEEE Access*, vol. 7, pp. 133190–133202, 2019. doi: https://doi.org/10.1109/ACCESS.2019.2940729

[13] M. Uddin, W. Khaksar, and J. Torresen, “Ambient sensors for elderly care and independent living: a survey,” *Sensors*, vol. 18, article no. 2027, 2018. doi: https://doi.org/10.3390/s18072027

[14] M. Buzzelli, A. Albé, and G. Ciocca, “A vision-based system for monitoring elderly people at home,” *Appl. Sci.*, vol. 10, no. 1, 2020. doi: https://doi.org/10.3390/app10010374

[15] P. Jayashree et al., “Smart assistive technologies for aging society: requirements, response and reality,” *in Proc. 8th Int. Conf. Adv. Comput. (ICoAC)*, IEEE, pp. 111-116, 2017. doi: https://doi.org/10.1109/icoac.2017.7951755

[16] C. M. M. Mansoor, S. K. Chettri, and H. M. M. Naleer, “A remote health monitoring system for the elderly based on emerging technologies,” *in Proc. Int. Conf. Emerg. Global Trends in Eng. and Technol.*, Singapore, pp. 513-524, 2022. doi: https://doi.org/10.1007/978-981-99-4362-3_47

[17] A. H. Sapci, and H. A. Sapci, “Innovative assisted living tools, remote monitoring technologies, artificial intelligence-driven solutions, and robotic systems for aging societies: systematic review,” *JMIR Aging*, vol. 2, no. 2, article no. 15429, 2019. doi: https://doi.org/10.2196/15429

[18] X. Zhou et al., “Design of intelligent wearable device based on embedded system,” *in Proc. 9th Int. Forum Elect. Eng. Automat. (IFEEA)*, IEEE, pp. 202-205, 2022. doi: https://doi.org/10.1109/ifeea57288.2022.10037788

[19] X. Chen, “Smart technologies and aging society,” *in Smart Cities and Smart Commun.: Empowering Citizens through Intell. Technol.*, Springer Nature Singapore, pp. 131-146, 2022. doi: https://doi.org/10.1007/978-981-19-1146-0_7

[20] T. Thomas, C. Cashen, and S. Russ, “Leveraging smart grid technology for home health care,” *in Proc. Int. Conf. Consum. Electron.*, pp. 274-275, 2013. doi: https://doi.org/10.1109/icce.2013.6486892

[21] S. Iqbal, “Artificial intelligence tools and applications for elderly healthcare-review,” *in Proc. 9th Int. Conf. Comput. Artif. Intell.*, pp. 394-397, 2023. doi: https://doi.org/10.1145/3594315.3594347

[22] C. H. Lee et al., "Artificial intelligence-enabled digital transformation in elderly healthcare field: scoping review," *Adv. Eng. Inform.*, vol. 55, article no.101874, 2023. doi: https://doi.org/10.1016/j.aei.2023.101874

[23] S. Salomé, and E. Monfort, "The digital revolution and ageism: the ethical challenges of artificial intelligence for older people," *NPG Neurologie-Psychiatrie-Gériatrie*, 2023. doi: https://doi.org/10.1016/j.npg.2023.09.004

[24] M. Koc, "Artificial intelligence in geriatrics," *Turkish J. Geriatrics*, vol. 26, no. 4, 2023. doi: https://doi.org/10.29400/tjgeri.2023.362

[25] M. T. Harris, K. A. Blocker, and W. A. Rogers, "Older adults and smart technology: facilitators and barriers to use," *Front. Comput. Sci.*, vol. 4, article no. 835927, 2022. doi: https://doi.org/10.3389/fcomp.2022.835927

[26] G. Rubeis, "The disruptive power of artificial intelligence. ethical aspects of gerontechnology in elderly care," *Arch. Gerontology and Geriatrics*, vol. 91, article no. 104186, 2020. doi: https://doi.org/10.1016/j.archger.2020.104186

[27] T. Shiwani et al., "New horizons in artificial intelligence in the healthcare of older people," *Age Ageing*, vol. 52, no. 12, article no. afad219, pp. 1-11, 2023. doi: https://doi.org/10.1093/ageing/afad219

[28] Y. Yamout et al., "Beyond smart homes: an in-depth analysis of smart aging care system security," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1-35, 2024. doi: https://doi.org/10.1145/3610225

[29] K. M. Kokorelias et al., "Coadaptation between smart technologies and older adults over time: protocol for a scoping review," *JMIR Res. Protocols*, vol. 12, no. 1, 2023. doi: https://doi.org/10.2196/51129

[30] K. Deepa et al., "Elderly and visually impaired indoor activity monitoring based on wi-fi and deep hybrid convolutional neural network," *Sci. Rep.*, vol. 13, no. 1, article no. 22470, 2023, doi: https://doi.org/10.1038/s41598-023-48860-5

[31] M. S. Momin et al., "In-home older adults' activity pattern monitoring using depth sensors: a review," *Sensors*, vol. 22, no. 23, article no. 9067, 2022, doi: https://doi.org/10.3390/s22239067

[32] A. Kadambi, A. Bhandari, and R. Raskar, "3D depth cameras in vision: benefits and limitations of the hardware: with an emphasis on the first-and second-generation kinect models," *Comput. Vis. Mach. Learn. RGB-D Sensors*, Springer, Cham, pp. 3–26, 2014. doi: https://doi.org/10.1007/978-3-319-08651-4_1

[33] J. Park et al., “Development of an unobtrusive sleep monitoring system using a depth sensor,” *J. Sleep Disorders: Treatment and Care*, vol. 9, no. 3, article no. 1000231, pp. 1-7, 2020. doi: https://doi.org/10.37532/jsdtc.2020.9(3).231

[34] A. Jalal, S. Kamal, and D. Kim, “A depth video-based human detection and activity recognition using multi-features and embedded hidden markov models for health care monitoring systems,” *Int. J. Interactive Multimedia Artif. Intell.*, vol. 4, no. 4, p. 54, 2017. doi: https://doi.org/10.9781/ijimai.2017.447

[35] R. Jansi and R. Amutha, “Detection of fall for the elderly in an indoor environment using a tri-axial accelerometer and Kinect depth data,” *Multidim. Syst. Sign. Process*, vol. 31, no. 4, pp. 1207–1225, 2020. doi: https://doi.org/10.1007/s11045-020-00705-4

[36] C. J. Debono, M. Sacco, and J. Ellul, “Monitoring indoor living spaces using depth information,” *in Proc. 10th Int. Conf. Consum. Electron. (ICCE-Berlin)*, 2020, pp. 1–5. doi: https://doi.org/10.1109/ICCE-Berlin50680.2020.9352158

[37] J. Lee, and B. Ahn, “Real-time human action recognition with a low-cost RGB camera and mobile robot platform,” *Sensors*, vol. 20, article. 2886, 2020. doi: https://doi.org/10.3390/s20102886

[38] W. Wilkowska et al., “Insights into the older adults’ world: concepts of aging, care, and using assistive technology in late adulthood,” *Front. Public Health*, vol. 9, article no. 653931, 2021. https://doi.org/10.3389/fpubh.2021.653931

[39] A. S. Rajput, B. Raman, J. Imran, “Privacy-preserving human action recognition as a remote cloud service using RGB-D sensors and deep CNN,” *Expert Syst. Appl.*, vol. 152, 2020. doi: https://doi.org/10.1016/j.eswa.2020.113349

[40] Thi Thi Zin et al., “Real-time action recognition system for elderly people using stereo depth camera,” *Sensors*, vol. 21, article no. 5895, 2021. doi: https://doi.org/10.3390/s21175895

[41] Thi Thi Zin et at., “Elderly monitoring and action recognition system using stereo depth camera,” *in Proc. 9th Glob. Conf. Consum. Electron.*, Kobe, Japan, pp. 13–16, 2020. doi: https://doi.org/10.1109/GCCE50665.2020.9291785

[42] Y. Hbali et al., “Skeleton-based human activity recognition for elderly monitoring systems,” *IET Comput. Vis*, vol. 12, pp. 16 – 26, 2018. doi: https://doi.org/10.1049/iet-cvi.2017.0062

[43] H. Rahmani et al., “HOPC: histogram of oriented principal components of 3D pointclouds for action recognition,” *in Proc. Comput. Vis.*, Zurich, Switzerland, pp. 6–12, 2014. doi: https://doi.org/10.1007/978-3-319-10605-2_48

[44] J. H. Li et al., “Segmentation and recognition of basic and transitional activities for continuous physical human activity,” *IEEE Access*, vol. 7, pp. 42565-42576, 2019. doi: https://doi.org/10.1109/access.2019.2905575

[45] S. Aminikhanghahi, and D. J. Cook, “Using change point detection to automate daily activity segmentation,” *in Proc. Int. Conf. Pervasive Comput. Commun. Workshops*, IEEE, pp. 262-267, 2017. doi: https://doi.org/10.1109/percomw.2017.7917569

[46] D. Thakur and S. Biswas, “Online change point detection in application with transition-aware activity recognition,” *Trans. Human-Machine Syst.*, vol. 52, no. 6, pp. 1176-1185, 2022, doi: https://doi.org/10.1109/THMS.2022.3185533

[47] S. Irfan et al., “A novel hybrid deep learning model for human activity recognition based on transitional activities,” *Sensors*, vol. 21, no. 24, article no. 8227, 2021. doi: https://doi.org/10.3390/s21248227

[48] S. Aminikhanghahi, and D. J. Cook, “Enhancing activity recognition using cpd-based activity segmentation,” *Pervasive and Mobile Comput.*, vol. 53, pp. 75-89, 2019. doi: https://doi.org/10.1016/j.pmcj.2019.01.004

[49] L. Song et al., “TACNet: transition-aware context network for spatio-temporal action detection,” *in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11987–11995. doi: https://doi.org/10.1109/CVPR.2019.01226

[50] J. Kang et al., “Transition activity recognition using fuzzy logic and overlapped sliding window-based convolutional neural networks,” *J. Supercomputing*, vol. 76, no. 10, pp. 8003-8020, 2020. doi: https://doi.org/10.1007/s11227-018-2470-y

[51] C. Gu, C. Zhang, and S. Kuriyama, “Orientation-aware leg movement learning for action-driven human motion prediction.” *arXiv*, 2024. doi: https://doi.org/10.48550/arXiv.2310.14907

[52] J. Shi, D. Zuo, and Z. Zhang, “Transition activity recognition system based on standard deviation trend analysis,” *Sensors*, vol. 20, no. 11, 2020, doi: https://doi.org/10.3390/s20113117

[53] J. L. Reyes-Ortiz et al., “Transition-aware human activity recognition using smartphones,” *Neurocomputing*, vol. 171, pp. 754-767, 2016, doi: https://doi.org/10.1016/j.neucom.2015.07.085

[54] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *in Proc. the IEEE*, vol. 77, no. 2, pp. 257-286, 1989, doi: https://doi.org/10.1109/5.18626

[55] C. A. Ronao and S. B. Cho, “Human activity recognition using smartphone sensors with two-stage continuous hidden markov models,” *in Proc. 10th Int. Conf. Natural Comput. (ICNC)*, Xiamen, China, pp. 681-686, 2014. doi: https://doi.org/10.1109/ICNC.2014.6975918

[56] M. H. Kabir et al., “Two-layer hidden markov model for human activity recognition in home environments,” *Int. J. Distrib. Sens. Netw*, vol. 12, article no. 4560365, 2016. doi: https://doi.org/10.1155/2016/4560365

[57] P. Asghari, E. Soleimani, and E. Nazerfard, “Online human activity recognition employing hierarchical hidden markov models,” *J. Ambient Intell. Humaniz. Comput*, vol. 11, pp. 1141-1152, 2020. doi: https://doi.org/10.1007/s12652-019-01380-5

[58] Swe Nwe Nwe Htun, Thi Thi Zin, and Pyke Tin, “Image processing technique and hidden markov model for an elderly care monitoring system,” *J. Imaging*, vol. 6, article no. 49, 2020. doi: https://doi.org/10.3390/jimaging6060049

[59] A. Dubois and F. Charpillet, “Automatic fall detection system with a RGB-D camera using a hidden markov model,” *In Inclusive Society: Health and Wellbeing in the Community, and Care at Home*, Springer: Berlin/Heidelberg, Germany, pp. 259-266, 2013. doi: https://doi.org/10.1007/978-3-642-39470-6_33

[60] M. I. Khedher, M. A. El-Yacoubi, B. Dorizzi, “Human action recognition using continuous HMMs and HOG/HOF silhouette representation,” *in Proc. 1st Int. Conf. Pattern Recogn. Appl. Methods*, Vilamoura, Algarve, Portugal, pp. 503-508, 2012. doi: https://doi.org/10.5220/0003695905030508

[61] M. Z. Uddin et al., “Human activity recognition using body joint-angle features and hidden markov model,” *ETRI J.*, vol. 33, pp. 569-579, 2011. doi: https://doi.org/10.4218/etrij.11.0110.0314

[62] M. Mokari, H. Mohammadzade, and B. Ghojogh, “Recognizing involuntary actions from 3D skeleton data using body states,” *Sci. Iran*, vol. 27, pp. 1424-1436, 2018. doi: https://doi.org/10.24200/SCI.2018.20446

[63] B. Ghojogh, H. Mohammadzade, and M. Mokari, “Fisherposes for human action recognition using kinect sensor data,” *IEEE Sens. J.*, vol. 18, pp. 1612-1627, 2018. doi: https://doi.org/10.1109/JSEN.2017.2784425

[64] N. Manouchehri and N. Bouguila, “Human activity recognition with an hmm-based generative model,” *Sensors*, vol. 23, no. 3, 2023, doi: https://doi.org/10.3390/s23031390

[65] L. Wang et al., “A fusion of a deep neural network and a hidden Markov model to recognize the multiclass abnormal behavior of elderly people,” *Knowl. Syst.*, vol. 252, 2022, doi: https://doi.org/10.1016/j.knosys.2022.109351

[66] C. Zhao, J. G. Han, and X. Xu, “CNN and RNN based neural networks for action recognition,” *J. Phys.: Conf. Ser.*, vol. 1087, no. 6, article no. 062013, 2018, doi: https://doi.org/10.1088/1742-6596/1087/6/062013

[67] H. Zhao and X. Jin, “Human action recognition based on improved fusion attention cnn and rnn,” *in Proc. 5th Int. Conf. Comput. Intell. Appl. (ICCIA)*, pp. 108-112, 2020. doi: https://doi.org/10.1109/ICCIA49625.2020.00028

[68] S. Chopra, L. Zhang, and M. Jiang, “Human action recognition using multi-stream fusion and hybrid deep neural networks,” *in Proc. IEEE Int. Conf. Syst., Man, and Cybern. (SMC)*, pp. 4852–4858, 2023. doi: https://doi.org/10.1109/SMC53992.2023.10393912

[69] A. Ullah et al., “Action recognition in video sequences using deep bi-directional lstm with cnn features,” *IEEE Access*, vol. 6, pp. 1155-1166, 2018. doi: https://doi.org/10.1109/ACCESS.2017.2778011

[70] J. Wensel, H. Ullah, and A. Munir, “Vit-Ret: vision and recurrent transformer neural networks for human activity recognition in videos,” *IEEE Access*, 2023. doi: https://doi.org/10.1109/access.2023.3293813

[71] A. S. Rajput, B. Raman, and J. Imran, “Privacy-preserving human action recognition as a remote cloud service using RGB-D sensors and deep CNN,” *Expert Syst. Appl.*, vol. 152, article no. 113349, 2020. doi: https://doi.org/10.1016/j.eswa.2020.113349

[72] M. Dallel et al., “A sliding window based approach with majority voting for online human action recognition using spatial temporal graph convolutional neural networks,” *in Proc. 7th Int. Conf. Mach. Learn. Technol. (ICMLT), New York, USA: Assoc. Comput. Mach.*, pp. 155-163, 2022. doi: https://doi.org/10.1145/3529399.3529425

[73] T. S. Apon, A. Islam, and MD. G. Rabiul Alam, "Action recognition using transfer learning and majority voting for csgo," *in Proc. 13th Int. Conf. Inf. Commun. Technol. Syst. (ICTS)*, pp. 235-240, 2021. doi: https://doi.org/10.1109/ICTS52701.2021.9608407

[74] K. Safdar, S. Akbar, and A. Shoukat, "A majority voting based ensemble approach of deep learning classifiers for automated melanoma detection," *in Proc. Int. Conf. Innov. Comput. (ICIC)*, pp. 1-6, 2021. doi: https://doi.org/10.1109/ICIC53490.2021.9692915

[75] H. Yoo et al., "Precondition and effect reasoning for action recognition," *Comput. Vis. Image Und.*, vol. 232, article no. 103691, 2023. doi: https://doi.org/10.1016/j.cviu.2023.103691

[76] T. Zhuo et al., "Explainable video action reasoning via prior knowledge and state transitions," *in Proc. 27th ACM Int. Conf. Multimedia, in MM '19. New York, NY, USA: Assoc. Comput. Mach.*, pp. 521-529, 2019. doi: https://doi.org/10.1145/3343031.3351040

[77] S. Tetsuri, and G. J. Anders, "Depth image compression by colorization for Intel® RealSenseTM depth cameras," *Developer Documentation*. Accessed: Mar. 28, 2022. [Online]. Available: https://dev.intelrealsense.com/docs/depth-image-compression-by-colorization-for-intel-realsense-depth-cameras?_ga=2.62121196.1983099587.1648443850-119351473.1648443850

[78] J. Glenn, "Ultralytics YOLOv5,", 2020. Accessed: Mar. 31, 2022. [Online]. Available: https://github.com/ultralytics/yolov5

[79] T. T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839-2846, 2015. doi: https://doi.org/10.1016/j.patcog.2015.03.009

[80] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," *in Proc. Int. Conf. Syst., Sig. Image Processing*, pp. 237–242, 2020. doi: https://doi.org/10.1109/IWSSIP48289.2020.9145130

[81] D. Kim et al., "Action recognition with depth maps using HOG descriptors of multi-view motion appearance and history," *in Proc. 8th Int. Conf. Mobile Ubiquitous Comput., Syst., Services Technol., UBICOMM*, Rome, Italy, pp. 126-130, 2014.

[82] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. R. Stat. Soc. Ser. B*, vol. 39, pp. 1-22, 1977. doi: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

[83] A. Kirillov, et. al, “Segment anything,” *arXiv preprint*, 2023. doi: https://doi.org/10.48550/arXiv.2304.02643

[84] J. Glenn, “YOLOv5 SOTA Realtime Instance Segmentation,” Accessed: Jul. 19, 2023. [Online]. Available: https://github.com/ultralytics/yolov5/releases/v7.0

[85] M. Tan and Q. V. Le, “EfficientNet: rethinking model scaling for convolutional neural networks.” *arXiv*, 2020. doi: https://doi.org/10.48550/arXiv.1905.11946

[86] J. Chung et al., “Empirical evaluation of gated recurrent neural networks on sequence modeling.” *arXiv*, 2014. doi: https://doi.org/10.48550/arXiv.1412.3555

[87] M. Tan and Q. V. Le, “EfficientNetV2: smaller models and faster training,” *arXiv*, 2021. doi: https://doi.org/10.48550/arXiv.2104.00298

List of Publications

Major Publications

Journal Papers

[1] Ye Htet, Thi Thi Zin, Pyke Tin, H. Tamura, K. Kondo, and E. Chosa, “HMM-based Action Recognition System for Elderly Healthcare by Colorizing Depth Map,” *International Journal of Environmental Research and Public Health*, vol. 19, issue. 19, article no. 12055, 2022. doi: https://doi.org/10.3390/ijerph191912055.

[2] Ye Htet, Thi Thi Zin, Pyke Tin, H. Tamura, K. Kondo, S. Watanabe, E. Chosa, “Smarter Aging: Developing A Foundational Elderly Activity Monitoring System with AI and GUI Interface,” *IEEE Access*, vol. 12, 2024. doi: https://doi.org/10.1109/ACCESS.2024.3405954.

Conference Proceedings

[3] Ye Htet, Thi Thi Zin, H. Tamura, K. Kondo, and E. Chosa, “Action Recognition System for Senior Citizens Using Depth Image Colorization,” *in Proceedings of the IEEE 4th Global Conference on Life Sciences and Technologies (LifeTech 2022)*, Osaka, Japan. 07-09 March 2022, pp. 494-495, doi: https://doi.org/10.1109/LifeTech53646.2022.9754900

[4] Ye Htet, Thi Thi Zin, H. Tamura, K. Kondo, and E. Chosa, “Temporal-dependent Features based Inter-Action Transition State Recognition for Eldercare System,” *in Proceedings of the IEEE 13th International Conference on Consumer Electronics – Berlin (ICCE-Berlin 2023)*, Berlin, Germany, 03-05 September 2023, pp. 106-111, doi: https://doi.org/10.1109/ICCE-Berlin58801.2023.10375682

Other Publications

Journal Papers

[1] Thi Thi Zin, Ye Htet, San Chain Tun, and Pyke Tin, “Artificial Intelligence Fusion in Digital Transformation Techniques for Lameness Detection in Dairy Cattle,” *International Journal of Biomedical Soft Computing and Human Sciences*, vol. 28, no. 1, pp. 1-8, 2023, doi: https://doi.org/10.24466/ijbschs.28.1_1.

[2] Thi Thi Zin, Ye Htet, Tunn Cho Lwin, and Pyke Tin, “A Markov-Dependent stochastic approach to modeling lactation curves in dairy cows,” *Smart Agricultural Technology,* vol. 6, article no. 100335, 2023, doi: https://doi.org/10.1016/j.atech.2023.100335.

発明

[3] 推定装置、推定方法及びプログラム、特願 2022-149569、出願日：2022 年 9 月 20 日、発明者：ティティズイン、パイティン、イエテ、出願人：国立大学法人宮崎大学