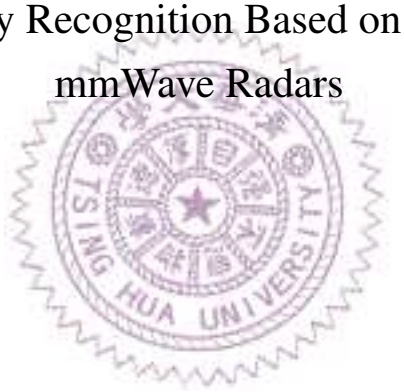
國立清華大學電機資訊學院資訊工程研究所
碩士論文
Department of Computer Science
College of Electrical Engineering and Computer Science
National Tsing Hua University
Master Thesis

使用毫米波雷達進行隱私保護之人體進食行為辨識
Food Intake Activity Recognition Based on Privacy-Preserving mmWave Radars



110062518
吳逸竑
Yi-Hung Wu

指導教授：徐正炘 博士
Advisor: Cheng-Hsin Hsu, Ph.D.

中華民國 112 年 9 月
September, 2023

中文摘要

本論文旨在探討使用毫米波雷達技術進行人類進食活動識別，並提出了一個包含RGB攝影機、深度攝影機和毫米波雷達數據的公開資料集，資料集含有24名參與者執行12種不同的桌邊活動，以不同隱私敏感性程度的儀器收集。 我們提出了四種算法。首個算法FIA結合了數個預處理技術（如體素化、邊界框構建和三線性插值）與CNN+Bi-LSTM神經網絡分類器，在全球設置下實現了91.49%的準確率，超越了當時使用體素化方法的SOTA。為了解決記憶體和儲存空間效率問題，我們引入了同樣為end-to-end的DPR算法，直接使用毫米波點雲的特徵，將GPU內存使用減少了79.38%。DPR還節省了約90%的磁碟空間。就準確性而言，在global設定下，DPR實現了95.59%的準確率，留一法設定下準確率達到了所有演算法中最高的72.46%。 我們還引入了一個預測骨骼特徵並將其用作分類器輸入的流程。我們提出的SPE和SPE+模型在骨骼估計方面表現出色，採用了ResNet架構，同樣使用了毫米波點雲的點坐標、訊號強度和速度作為輸入特徵。SPE和SPE+模型超越了現有的MARS和mmPose-NLP模型，成為誤差距離最小的模型。 在使用骨骼數據作為輸入的分類器方面，我們直接修改了廣泛使用的ST-GCN和2s-AGCN模型。這些模型在global設定下超越了DPR，實現了98.68%的準確率，但在留一法設置中僅達到了57.87%的準確率。然而，當使用理想的骨骼（mediapipe pose）作為輸入時，分類器達到了82.42%的準確率，突顯了GCN模型的潛力。 本研究提出了創新的方法、算法和多樣化的數據集，實現了準確性、內存效率和隱私考慮方面的重大進展。

Abstract

This thesis explores the use of mmWave radar technology for recognizing human food intake activities, and introduces a dataset comprising data from an RGB camera, a depth camera, and mmWave radar, with 24 participants performing 12 food intake activities with heterogeneous privacy sensitivity sensors. Four algorithms are presented. FIA, the initial algorithm, combines several preprocessing techniques (voxelization, bounding box, and trilinear interpolation) with a CNN+Bi-LSTM neural network classifier, achieving 91.49% accuracy in the global setup, surpassing voxelization-based methods. To address memory and classification issues, DPR, another end-to-end algorithm, directly uses mmWave point cloud features, reducing GPU memory usage by 79.38%. DPR also conserves 90% of disk space and achieves 95.59% accuracy globally, with the best accuracy of 72.46% in the leave-one-out setup. A pipeline for predicting skeleton features (SPE and SPE+) is introduced. These models outperform existing models like MARS and mmPose-NLP, boasting the smallest error distances. Modified ST-GCN and 2s-AGCN models achieve 98.68% accuracy in the global setup but 57.87% in the leave-one-out setup. However, utilizing the ideal skeleton (mediapipe pose) results in 82.42% accuracy, highlighting the GCN model's potential. This research presents innovative approaches, algorithms, and a diverse dataset, with advancements in accuracy, memory efficiency, and privacy considerations.

Contents

List of Figures

List of Tables

Chapter 1

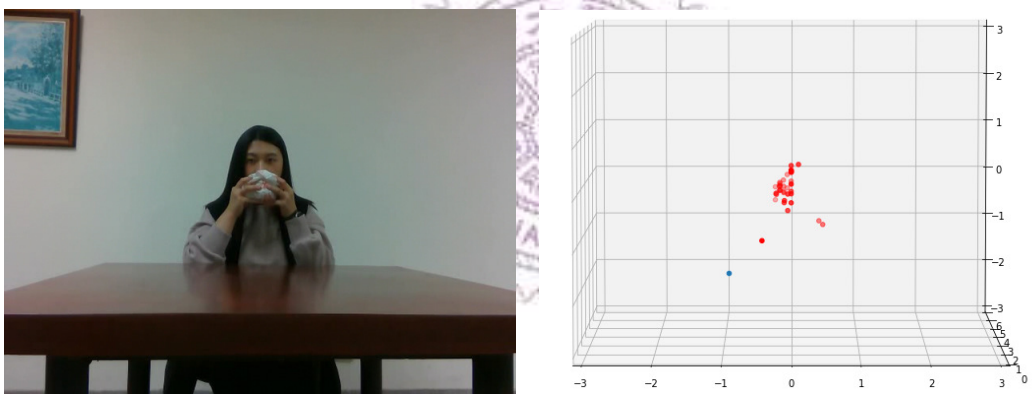
Introduction

Today's mainstream clinical methods for diet monitoring ask subjects to manually report or record food intake activities, which has been proven to be inaccurate. For example, Harnack et al. [23] discovered a large number of underreported intakes of food portions significantly beyond the recommended serving sizes. Therefore, an automatic food intake activity recognition system that can be deployed in smart environments, such as smarthomes, to pervasively detect when, for how long, and what food/drink a subject is eating/drinking, would be a good contribution towards accurate food-intake reporting. Such an automatic system has quite a few application scenarios. For example, in diet control, such a system can automatically monitor how many times during the day, for how long each time, and at what times a person is eating or drinking. In smarthome data monetization applications, such as D!fintech's PriMonDi [12], such information can be used to develop a model to predict the home residents' habits and offer deals that will benefit both food providers and the smarthome's residents. In telecare, especially for dementia patients who may forget to eat or drink, the system can remind them or inform a nurse that the patient has not received enough food or drink.

While much research has been done on recognizing what food a person is intaking [38], very few works have investigated the automatic recognition of when and for how long food intake takes place. Such recognition must be done completely autonomously and without requiring the subject to do anything in addition to just eating or drinking, in order to avoid human error or lethargy. As such, food logs which require the subjects themselves to record the food intake or take a picture of the food are not appropriate. In addition, it is highly preferred that the system be non-intrusive so that subjects do not need to wear special sensors or actuators. Finally, the system should preserve the privacy of the subjects, which rules out imaging systems in which the identity of the subject can be seen in the image, such as most camera-based systems.

In this paper, we propose a non-invasive and privacy-preserving food intake activity

recognition system that can recognize when and for how long a subject is eating or drinking. Our system can work either standalone, when only frequency or timing of food intake is important, or complementary to systems that recognize what food/drink is being taken. We use a mmWave point cloud obtained from an in-situ sensor that can protect the subject's privacy while recognizing activities, shown in Fig. 1.1(b) where a subject's identity is well hidden. In contrast, with an RGB camera, Fig. 1.1(a) clearly reveals the identity of the subject, which is undesired. To this end, we obtain the point cloud data generated by a mmWave radar and use Artificial Intelligence (AI), specifically a neural network, to recognize food intake activities and distinguish them from other activities such as swiping a smartphone, making a phone call, hand clapping, hand waving, and cleaning one's face with a napkin. At the time of writing, there is approximately an order of magnitude cost difference between a mmWave radar and an RGB camera (300 USD vs. 30 USD) [57]. While mmWave radars are more expensive. Then odder various advantages, such as privacy protection, and can be installed in private areas. Additionally, mmWave radars also work in low-lighting environments.



(a) (b)

Figure 1.1: Sample data from: (a) an RGB camera, and (b) a mmWave radar with fine-grained activities.

All of the solutions are shown in Fig. 1.2 The solution is split into two main streams, (i) end-to-end solutions directly using mmWave radar point cloud as the input of the model, which classifies the model classifies the activities. The *Food Intake Activity classifier* (FIA) algorithm incorporates voxelization, bounding box techniques, and trilinear interpolation as preprocessing steps for mmWave point cloud data. It subsequently employs a deep neural network model consisting of CNN and Bi-LSTM layers, as the first end-to-end solution model. On the other hand, the *Dynamic Point Cloud Recognizer* (DPR) algorithm utilizes features such as X, Y, and Z coordinates, intensity, and velocity from individual radar points within the mmWave point cloud data. It employs a deep neural network model comprising Convolution Neural Networks (CNN) and Long Short-Term

Memory (LSTM) layers, acting as the second end-to-end solution model. (ii) skeleton solutions, not directly using mmWave radar as the input of the classifier model, generate human skeleton features first, then use estimated human skeleton joints as the input of the classifier. The *Skeleton Pose Estimator* (SPE) algorithm employs a similar approach to DPR, utilizing features such as X, Y, and Z coordinates, intensity, and velocity from individual radar points. It utilizes a CNN network model to predict the positions of skeleton joints. Furthermore, an extended version of SPE, the SPE+, combines data from multiple frames and uses LSTM layers to learn time information, providing the model with a temporal context and thereby obtaining more accurate data. The estimated skeleton joints become the input of two different Graph Convolutional Network (GCN) models, ST-GCN and 2s-AGCN, respectively. GCN algorithms utilize a graph matrix to enable the model to understand and learn from skeletal connectivity features. Classifiers trained on these features have the potential for better performance compared to raw data. The dataset from heterogeneous sensors can be used in several research directions, including but not limited to: (a) food intake activity recognition, (b) human skeleton estimation, and (c) privacy leakage assessment of heterogeneous sensor data. We have implemented the first two applications in the thesis, The third one considers the amount of privacy leakage caused by different sensor types, which is a possible future work.

In summary, our work consists of five main contributions:

- *Food Intake Activity (FIA)*, an end-to-end food intake activity recognizer based on voxelization, using the mmWave point cloud as input.
- *Dynamic Point Cloud Recognizer (DPR)*, also an end-to-end food intake activity recognizer uses mmWave point cloud as input, but abandons voxelization for lower GPU memory consumption.
- *Skeletal Pose Estimator (SPE)*, an enhanced skeletal pose estimation model capable of generating more precise skeletons specifically for food intake activity recognition, uses the mmWave point cloud as input.
- A *Graph Convolution Network (GCN)*-based food intake activity recognizer using skeletons as input, which outperforms both end-to-end solutions.
- A food intake activity dataset that is collected with mmWave radar and an RGBD camera.

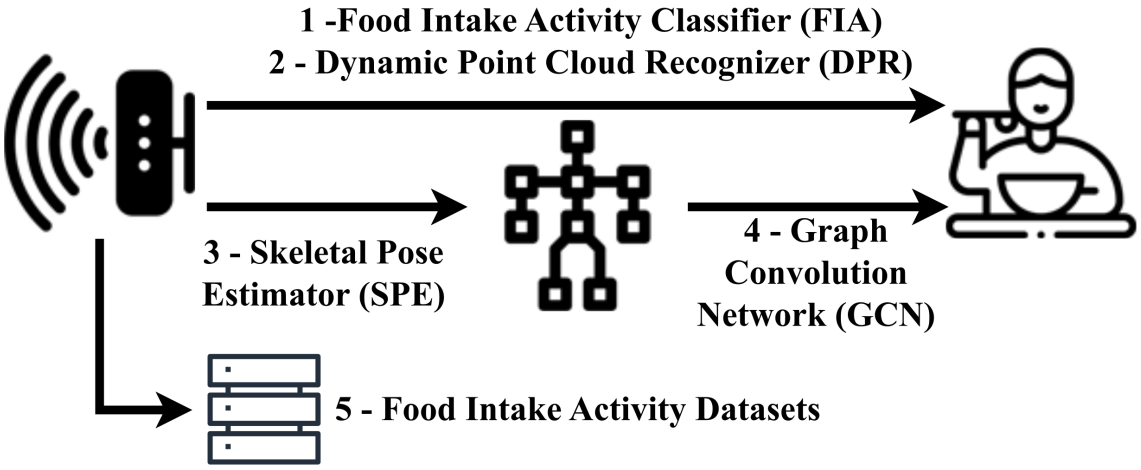


Figure 1.2: Five main contributions of the paper.

Chapter 2

Related Work

In this chapter, we introduce the related work relevant to our topic. Based on the four algorithms we have proposed and a public dataset, we divide the entire chapter into three major sections: food intake activity recognition, skeletal pose estimation, and food intake activity datasets.

2.1 Food Intake Activity Recognition

Keeping track of what food is being eaten by human subjects has been extensively studied in the literature. Interested readers may find recent surveys on food logging [38] and food recognition [39]. Differing from those studies, our work focuses on understanding when food intake happens and for how long.

Human activity recognition in general can be done via two approaches: by employing wearable or in-situ sensors. For the former approach, different bioelectric sensors such as Electromyography (EMG) or Electroencephalography (EEG) sensors can be attached to human subjects to detect their subtle muscle movements. In particular, human activity recognition systems using EMG [34, 40, 43] and EEG [45] sensors lead to relatively higher recognition accuracy than other wearable sensors. However, setting up these bioelectric sensors is cumbersome, and hence not suitable for day-to-day usage. Inertial sensors, such as accelerators and gyroscopes can also be used to detect subject movements. Inertial sensors are already included in modern smartphones, smartwatches, and smart wristbands, which are easier to carry around and have been adopted for human activity recognition [55, 1, 24, 64, 5, 36].

The most popular in-situ sensor for human activity recognition is probably the RGB camera. Many existing projects [17, 54, 6] have applied Convolution Neural Networks (CNN) to RGB images or videos for human activity recognition. More advanced neural networks haven also been applied to the RGB videos [66, 30] or the RGBD [59] ones.

However, deploying RGB cameras or other camera-based sensors in smarthomes invades the privacy of human subjects, because the subject identities could be recognized if the sensor data are sent to third parties, such as cloud service providers. To cope with such privacy concerns, researchers have proposed to employ Radio Frequency (RF) sensors for activity recognition. For instance, Wang et al. [61] proposed to use low-cost WiFi transceivers to recognize human eating activities. Their proposed solution places two WiFi transceivers with a human subject in between, which leads to installation burdens. Adding to that, WiFi signals are often affected by the noisy Industrial, Scientific, and Medical (ISM) bands. As better alternatives, mmWave radars have also been adopted for coarse-grained human activity recognition [69, 8, 53, 63, 20], where sparse point clouds are filtered and augmented in various ways for higher recognition accuracy. For example, Singh et al. [53] proposed to voxelize raw point cloud before sending the voxels into neural networks. Wang et al. [63] proposed a preprocessing approach to identify human movements in order to filter out noise from environments before voxelization. Different from prior voxelization-based studies, a recent work employed a Graph Neural Network (GNN) on dynamic edges generated from sparse point clouds [20].

Our work also recognizes human activities using in-situ mmWave radars, but we set out to recognize the more challenging fine-grained activities of food intake. Fine-grained food intake activity recognition has been considered; e.g., He et al. [26] surveyed the sensor types used for recognizing food intake activities, and Selamat and Ali [47] surveyed the approach of monitoring chewing activities. A majority of such studies were built on wearable sensors [44, 71, 14, 65, 28], like microphones, EMG/EGG sensors, and inertial sensors, which dictate attaching sensors and wires on human subjects. However, such a setup overhead may discourage human subjects from using the systems on a daily basis. In contrast, our work solely relies on in-situ sensors, which are less intrusive to people's life.

Two existing works [69, 8] also aim to recognize fine-grained activities, and thus are the closest ones to ours. Particularly, Xie et al. [69] fused the outputs from inertial sensors and 2D mmWave radars to recognize drinking, eating pasta, and eating soup from other activities. Although they reported $\sim$70% accuracy using a proprietary dataset, such accuracy is only achievable when the inertial sensor data from cumbersome wearable devices are available. In contrast, our work only adopts in-situ mmWave radars, yet achieves promising accuracy above 90%. Bhalla et al. [8] also adopted 2D mmWave radars to identify utensils such as forks, chopsticks, bare hands, etc. used in eating activities. Differing from their works, ours differentiates food intake from other activities. Moreover, since 2D mmWave radars capture less information compared to their 3D counterparts, we opt for using 3D mmWave radars in this work.

2.2 Skeletal Pose Estimation

So far, there is no non-intrusive skeleton detector available on the market that can directly capture human skeletons. The generation of skeletons primarily comes from two major sources. The first involves attaching *locators*, or *markers* to a subject's body, while the second relies on other data sources (e.g., RGB videos, depth images, mmWave radar point clouds). How to generate skeleton features from the available data is an active research area.

RGB-based approaches have been widely adopted for skeletal pose estimation. For example, Gkioxari et al. [19] proposed a deform-able part model to detect people and localize their joints. Pishchulin et al. [42] employed joint subset partitioning and labeling for multi-person pose estimation. Papandreou et al. [41] capitalized region-based CNNs to predict bounding boxes that potentially contain people, and then used ResNet to estimate joints of the person in each bounding box. Cao et al. [10] leveraged part affinity fields for realtime multi-person pose estimation. Despite good results, these RGB-based approaches raise privacy concerns and are mostly restricted to estimation in the 2D space.

mmWave-based approaches have emerged as a newer alternative for skeletal pose estimation. For instance, Sengupta et al. [49] presented *mm-Pose*, using two mmWave radars for skeletal pose estimation. In contrast, Wang et al. [62] used a single commodity mmWave radar for pose estimation. In these two works [49, 62], the preprocessing step involves projecting 3D point clouds onto two separate planes, followed by two CNNs, which incur higher overhead. Sengupta et al. [48] published *mmPose-NLP*, using NLP techniques to improve the precision of their earlier work [49]. However, their work is based on voxelization, which may not be memory efficient. Furthermore, it can only estimate where the voxel is, which is vulnerable to high inaccuracy when the resolution is low. Most recently, An et al. [3] proposed *MARS*, using a single mmWave radar for human skeleton estimation. Their point cloud preprocessing step involves mapping points to 5-channel feature maps before employing a single CNN for estimation. We adopted this recent work as a starting point and enhanced it to achieve higher accuracy specifically for food intake activity recognition.

2.3 Food Intake Activity Datasets

Existing datasets can be used in human activity recognition. We classify them into four classes: (i) coarse-grained activities with rich-media sensors, (ii) coarse-grained activities with less privacy-intrusive sensors, (iii) food-intake activities with wearable sensors, and (iv) food-intake activities with mmWave radar. For brevity, we focus on the publicly

available datasets at the time of writing.

2.3.1 Coarse-Grained Activities with Rich-Media Sensors

Earlier datasets of human activities are mostly from rich-media sensors, such as RGB cameras. For example, Zelnik-Manor and Irani [72] collected a video dataset of humans performing four coarse activities, including running and walking. Similarly, Gorelick et al. [21] considered 10 human activities in their dataset. Laptev et al. [46] generated a larger dataset with 25 people performing six activities, such as walking, jogging, and running. More recently, Fisher et al. [16] considered nine human activities, such as walking and meeting, in two different contexts: a research lab and a shopping mall. The video clips were collected using a wide-angle lens. They also collected a similar dataset [15] with two subsets from a train station and between multiple humans. Parts of their dataset are in the form of multiview (actually two-view) video clips. Different from our dataset, the aforementioned datasets [72, 46, 16, 15] only consider privacy-intrusive vision-based sensors, and they only consider coarse-grained activities, which are easier to recognize.

Less privacy-intrusive sensors have also been used to collect coarse-grained human activity recognition datasets. The widely-used ones are Inertial Measurement Unit (IMU) sensors, i.e., accelerometers and gyroscopes. For example, Anguita et al. [4] collected a dataset of 30 people performing six activities: standing, sitting, lying down, walking, walking downstairs, and walking upstairs, using IMUs in a waist-mounted smartphone. Zhang et al. [74] constructed a dataset of 14 subjects performing 12 activities: walking left, walking right, running forward, jumping, sitting, standing, and sleeping, using an embedded system with IMUs. Micucci et al. [37] released a dataset of 30 subjects performing nine activities and eight falls using IMUs in a smartphone. Bhat et al. [9] collected a dataset of 22 subjects performing seven activities: jumping, lying down, sitting, standing, walking, walking upstairs, and walking downstairs, using an ankle-mounted IMU and a wearable stretch sensor. Logacjov et al. [35] constructed a dataset of 22 subjects performing 12 activities, such as sitting, walking, standing, and cycling, using IMUs as well. Garcia-Gonzalez et al. [18] built a dataset of 19 subjects performing four activities, which are inactive, active, walking, and driving, using the accelerometers, gyroscopes, magnetometers, and GPS receivers in a smartphone. Sikder et al. [52] released a dataset of 90 subjects performing 18 activities, such as standing, sitting, jumping, and playing table tennis, using IMUs in a smartphone. Compared to our dataset, the above-mentioned IMU-based datasets require humans to wear or carry cumbersome sensors, and they only consider coarse-grained activities.

Wireless signals are another way to recognize coarse-grained human activities, e.g., Guo et al. [22] collected a dataset of 10 subjects performing: (i) 10 upper-body activ-

ities, such as horizontal arm waving, tossing paper, hand clapping, and high throwing, (ii) two lower-body activities, which are forward- and side-kicking, and (iii) four whole-body activities, which are squatting, sitting, bending, and walking using WiFi signals. Their dataset was also captured by in-situ sensors, which is similar to ours. However, it only contains coarse-grained activities, likely because recognizing fine-grained activities is too challenging with WiFi signals alone. Compared to WiFi signals, mmWave radar offers more information without additional subject privacy. Similar to our work, Huang et al. [29] also used a mmWave radar to construct a dataset of 17 subjects performing six activities: boxing, jumping, squatting, walking, circling, and lifting. However, their dataset only concerns coarse-grained activities and has not been made public.

2.3.2 Food-Intake Activities with Wearable Sensors

Although food intake activities are less common in human activity datasets, they are considered in a few earlier attempts using wearable sensors. For example, Amft et al. [2] constructed a dataset of four subjects performing four food intake activities, which are eating lasagna with a fork/knife, fetching and drinking a glass of liquid, eating soup with a spoon, and eating slices of bread with one hand. They employ several wearable sensors, including IMUs, electromyography sensors, and microphones. Similarly, Yatani et al. [71] used a custom-built wearable acoustic sensor to record the sound close to the subjects' throats when they performed one of the 12 activities, such as sitting, deep breathing, eating cookies, and drinking. Cheng et al. [11] collected a dataset of three subjects performing five activities: eating, sleeping, watching TV, working on a computer, and walking, using a capacitive sensor in a customized neckband. Farooq et al. [13] produced a dataset of 10 subjects performing five activities: sitting, talking, eating while sitting, eating while walking, and walking. They built a wearable embedded system with a piezo-electric strain sensor and an accelerometer. Unlike our dataset, the above datasets require subjects to wear custom-built sensors, making everyday usage more challenging.

2.3.3 Food-intake Activities with mmWave Radars

More recently, the less privacy-intrusive mmWave radar has been adopted to create food intake activity datasets. For example, Xie et al. [69] used mmWave radar to create a dataset of six subjects performing five activities: eating with a fork, eating with a fork/knife, eating with a spoon, eating with chopsticks, and eating with hands. Bhalla et al. [8] gathered a dataset of nine people engaged in 10 different activities: catching, clapping, dribbling, drinking, folding clothes, joggling, eating pasta, eating soup, brushing teeth, and walking. However, they used 2D mmWave radar, which only has two-axis

coordinates instead of three. Similarly, Wang et al. [60] also used a mmWave radar to produce a dataset of 48 subjects performing five activities: eating with a fork/knife, eating with chopsticks, eating with a spoon, eating with a hand, and drinking from a glass. They considered 48 meal sessions, composed of 3,121 eating gestures and 608 drinking gestures. However, to the best of our knowledge, these datasets have not been made public at the time of this writing. Moreover, they only consider a single sensor and may not be suitable for investigating the trade-off between recognition accuracy and privacy leakage. Last, in our earlier work [67], we employed both an RGB-D camera and a mmWave radar to construct a smaller dataset of six subjects. The current dataset paper is a significant extension of that workshop paper.

Chapter 3

Background

In this chapter, we introduce the background related to our topic. There are three major sections, starting with human activity recognition. In this section, we describe the importance of Human Activity Recognition (HAR), commonly used sensors, and various data processing techniques applied to HAR. The second section is dedicated to in-situ sensors, where we introduce the concept of in-situ sensors, their advantages compared to other types of sensors, and the differences between various in-situ sensors. Finally, we delve into food intake activity recognition, further breaking it down into food recognition and food intake activity recognition for detailed discussion.

3.1 Human Activity Recognition

Human Activity Recognition stands at the intersection of computer vision, machine learning, sensor technology, and data processing, aiming to understand and classify human activities or behaviors from sensor data. This field has attracted significant attention due to its broad applicability across domains such as healthcare, security, robotics, and human-computer interaction. The goal of HAR research is to develop algorithms and models capable of automatically recognizing and categorizing activities based on data obtained from various sensors.

Importance of HAR: HAR addresses fundamental challenges in understanding and interpreting human activities. It holds immense potential for applications like assisted living for the elderly, monitoring patient health, improving industrial safety, enhancing user experiences in augmented reality, and bolstering surveillance and security systems.

Sensors: HAR primarily relies on sensors, which have evolved rapidly. In the early days, researchers used accelerometers and gyroscopes, but more recently, data from cameras (RGB and depth), wearable devices (smartphones, smartwatches), and radar (such as mmWave radar) have become prominent data sources. These sensors capture diverse

information, enabling researchers to explore novel approaches.

Data preprocessing: HAR research commonly involves significant data preprocessing steps. This includes data fusion, noise reduction, feature extraction, and sometimes, using skeleton data rather than the raw data to represent human poses. In recent years, deep learning techniques, particularly convolutional and recurrent neural networks, have demonstrated remarkable effectiveness in handling raw sensor data.

Feature extraction: Getting features is crucial for a mature HAR system. Traditionally, features such as statistical moments, Fourier transforms, and wavelets were employed. However, deep learning has enabled the automatic extraction of relevant features directly from raw data, reducing the need for handcrafted features.

Challenges: HAR research faces numerous challenges, including variability in human activities, occlusions, diverse sensor modalities, and scalability to handle large datasets. Privacy concerns related to capturing human activity data in public spaces are also a significant consideration.

Benchmark Datasets and Competitions: The availability of benchmark datasets (e.g., UCI-HAR [31], Kinetics [32], and NTU RGB+D [50]) has accelerated progress in the field. Research competitions, such as the ActivityNet Challenge, encourage the development of innovative HAR models.

Applications: HAR is deployed in diverse real-world applications, including fall detection for the elderly, gesture-based control for consumer devices, recognizing activities in sports analytics, and identifying unusual behavior in surveillance systems.

Future Works: The future of HAR research is likely to involve further integration of multi-modal sensor data, enhancing the interpretability of deep learning models, addressing privacy concerns, and advancing towards real-time and edge-based implementations to support IoT and smart environments.

In summary, HAR research plays an important role in understanding human activities through sensor data, with applications spanning healthcare, security, entertainment, and beyond. Continuous advancements in sensor technology and machine learning techniques continue to drive innovation in this field, offering the potential to improve the quality of life and safety for individuals worldwide.

3.2 In-situ Sensors & Radars

In the realm of Human Activity Recognition (HAR) research, the utilization of in-situ sensors and radar technologies has significantly advanced the capabilities of activity detection and monitoring. These technologies offer unique advantages, addressing various challenges in HAR applications. Here, we explore the benefits of in-situ sensors and

radars separately, focusing on in-situ sensors first.

Advantages of in-situ sensors. The in-situ sensors have their advantages when employed in a smart environment:

- Non-Intrusiveness: In-situ sensors, such as infrared motion sensors, pressure sensors, and ultrasonic sensors, are non-intrusive by nature. They can detect human activities without requiring subjects to wear or carry any devices, preserving their comfort and privacy.

- Privacy-preserving: In-situ sensors are particularly advantageous in scenarios where privacy is a primary concern. Unlike cameras or wearable devices, they do not capture visual or personal information, making them suitable for healthcare settings and public spaces.

- Continuous monitoring: These sensors can provide continuous monitoring, making them suitable for applications like elderly care and healthcare monitoring. They can detect falls, irregular movements, or changes in vital signs without requiring active user participation.

- Low power consumption: Many in-situ sensors are designed for low power consumption, extending their battery life or reducing energy costs in smart home environments.

- Real-time responsiveness: In-situ sensors can offer real-time responsiveness, enabling timely alerts or interventions when abnormal activities are detected. This is crucial in healthcare applications, where rapid response can be life-saving.

The most widely used in-situ sensor is the RGB camera, but because of privacy concerns and some limitations, for example, it requires enough illumination and is easy to be affected by small obstacles such as rain, radar has become a hot in-situ sensor to replace RGB cameras for detection. The radars have their advantages when employed:

- Privacy: Radar technology, specifically millimeter-wave (mmWave) radar, offers exceptional privacy advantages. Unlike cameras, which capture detailed visual information, radar only measures motion and object presence, ensuring privacy compliance in sensitive areas.

- All-Weather Capability: Radars, including mmWave radar, are effective in all weather conditions, including darkness, rain, fog, or smoke. This makes them suitable for outdoor and indoor applications without environmental limitations.

- High Penetration: Radars can penetrate obstacles, including walls and other visual barriers, making them ideal for monitoring activities in obstructed environments. This capability is valuable in building automation and security applications.

- Versatile Sensing: Radar technology can sense not only human motion but also vital signs, gestures, and micro-movements. This versatility enables a wide range of HAR applications beyond simple activity recognition.

In summary, in-situ sensors and radar technologies have significantly contributed to the field of Human Activity Recognition. In-situ sensors offer non-intrusive, privacy-preserving, and continuous monitoring capabilities, while radar technologies excel in privacy preservation, all-weather sensing, and versatile motion detection. These technologies are fundamental in advancing HAR applications, ranging from healthcare and eldercare to building automation and security.

3.3 Food Intake Activity Recognition

In the realm of Human Activity Recognition (HAR) research, Food Intake Recognition represents a specialized subfield encompassing two major themes: Food Recognition and Food Intake Activity Recognition.

Food recognition: Food Recognition involves the identification and categorization of various food items or dishes based on visual or sensor data. This area has garnered significant attention due to its potential applications in health monitoring, dietary analysis, and food service automation. Researchers in this field strive to develop algorithms and models that can accurately recognize different types of foods from images or videos. Techniques employed range from traditional computer vision methods to deep learning approaches, leveraging datasets with labeled food images.

Food intake activity recognition: Food Intake Activity Recognition takes HAR a step further by not only recognizing general activities but specifically focusing on activities related to food consumption. This subfield aims to understand and classify actions like eating, drinking, or even identifying the quantity of food consumed. It plays a crucial role in health and nutrition monitoring, particularly for individuals with dietary restrictions or those seeking to manage their food intake for specific health goals. To accomplish Food Intake Activity Recognition, researchers employ a combination of sensory data sources, including wearable devices, depth cameras, RGB cameras, and even mmWave radar. These sensors capture fine-grained details of hand movements, utensil usage, and facial expressions during eating or drinking. Machine learning models are then trained on this multimodal data to accurately recognize specific food-related activities. Privacy con-

cerns are paramount in this field, as capturing individuals' eating habits can be invasive. Researchers have been exploring techniques to balance data privacy with the need for accurate recognition. Privacy-preserving technologies and the development of non-intrusive sensors have become crucial aspects of Food Intake Activity Recognition research.

In summary, Food Intake Recognition within the broader domain of HAR focuses on two critical aspects: Food Recognition, which involves identifying food items from visual data, and Food Intake Activity Recognition, which delves into recognizing activities related to food consumption. These research areas have the potential to revolutionize health monitoring, dietary analysis, and personalized nutrition planning.

Chapter 4

Problem Statement

In this chapter, we have introduced the characteristics of sensors commonly used in food intake activity recognition and highlighted the advantages of mmWave radar over these sensors in terms of privacy and convenience. Following that, we elaborate on the challenges of using mmWave radar for food intake activity recognition.

Food intake recognition is a highly relevant and popular topic in the field of human behavior analysis. It has the potential to be widely used in applications such as remote caring, dietary control, and personalized health monitoring. Accurate and efficient recognition of food intake actions can provide valuable insights into individuals' eating behaviors, facilitating interventions for promoting healthy habits and managing dietary requirements.

RGB cameras. Existing food intake recognition systems often rely on RGB cameras as the primary sensor for capturing and analyzing human actions. However, the use of RGB cameras raises significant privacy concerns. Many individuals are reluctant to have their images captured and stored due to privacy issues and the potential misuse of personal data. This poses a major challenge to the widespread adoption of food intake recognition systems that rely solely on RGB camera data.

Wearable devices. Another alternative sensor commonly used for activity recognition is wearable devices. These devices can capture various physiological signals and motion data, including hand gestures and arm movements. However, wearable devices have their own limitations, particularly in the context of food intake recognition. Users may forget to consistently wear the device or may find it uncomfortable, leading to incomplete data collection and compromised accuracy. Moreover, wearable devices may not capture the complete range of movements associated with food intakes, such as fine-grained hand movements and facial expressions.

mmWave radars. To address the privacy concerns and limitations of current sensors, mmWave radar technology presents a promising alternative for food intake recognition.

Unlike RGB cameras, mmWave radar generates sparse point clouds that provide depth and motion information without revealing the identity of the individuals being monitored. This privacy-preserving feature makes mmWave radar an attractive option for capturing human actions in sensitive environments, such as homes, healthcare facilities, or public dining areas. However, there are significant challenges associated with utilizing mmWave radar for food intake recognition.

Challenge 1: The sparse nature of the point clouds generated by mmWave radar. The maximum number of points per frame is typically limited to 64, which may not provide sufficient detail for accurately capturing and analyzing complex food intake actions. Sparse point clouds can result in limited coverage of the scene and may miss crucial hand movements, facial expressions, or interactions with food and utensils.

Challenge 2: The sensitivity of mmWave radar to moving targets. While mmWave radar excels at detecting and tracking larger-scale movements, it may struggle to discern and differentiate small-scale actions, such as those involved in food intake. Subtle movements, such as grasping a utensil, bringing food to the mouth, or chewing, require high precision and sensitivity, which mmWave radar may lack compared to other sensors.

Problem statement. Therefore, the main problem to be addressed in this study is *how to overcome the limitations of sparse point clouds and the sensitivity to small movements associated with mmWave radar in order to develop an effective and privacy-preserving food intake recognition system.* This entails exploring novel algorithms and techniques to enhance the resolution and completeness of point clouds, as well as to accurately recognize and classify food intake actions. By addressing these challenges, we can enhance the usability and privacy protection of food intake recognition systems, leading to improved remote caregiving, dietary control, and personalized health monitoring.

Solving these challenges requires interdisciplinary research, involving expertise in signal processing, computer vision, machine learning, and human-computer interaction. By leveraging the advantages of mmWave radar technology while mitigating its limitations, we can develop innovative approaches that enable accurate and privacy-aware food intake recognition. Ultimately, the successful development of such a system has significant implications for improving healthcare, promoting healthy eating habits, and enhancing the overall well-being of individuals.

Chapter 5

Proposed Solutions

In this chapter, we provide a detailed explanation of the algorithms we propose. Firstly, in the overview section, we briefly outline the contributions of these four algorithms within the entire pipeline. Subsequently, we delve into a description of each of the four algorithms.

5.1 Overview

Fig. 5.1 shows the pipeline of the food intake activity recognition. There are 4 algorithms proposed in this thesis. Our study encompasses four distinct algorithms aimed at advancing human activity recognition based on mmWave point clouds:

- FIA Algorithm: FIA utilizes voxelization, bounding box methods, and trilinear interpolation to address the issue of inconsistent point quantities in sparse point cloud data. This method efficiently overcomes the challenges posed by irregular point densities.

- DPR Algorithm: The DPR algorithm tackles sparse point cloud data limitations by capping each frame at a maximum of 64 points. Any deficit in points is resolved through zero-padding. This approach employs the original XYZ coordinates, intensity, and velocity parameters, efficiently mitigating the memory wastage encountered in voxelization solutions.

- The third algorithm, known as the Skeleton Pose Estimation (SPE), is designed to predict corresponding skeleton frames based on mmWave point cloud frames.

- The final algorithm leverages the Graph Convolutional Network (GCN) model. It utilizes the skeleton data as input to perform Human Activity Recognition (HAR).

These algorithms collectively represent diverse strategies to address the complexities of human activity recognition using mmWave radar data. From voxelization techniques to zero-padding solutions and advanced machine learning models, each algorithm contributes to a multifaceted approach in resolving challenges related to sparse point cloud data and effectively extracting meaningful insights for accurate human activity recognition.

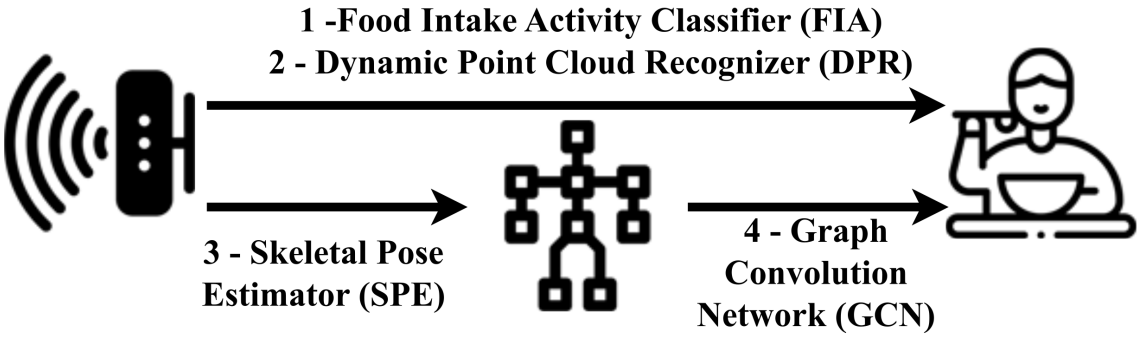
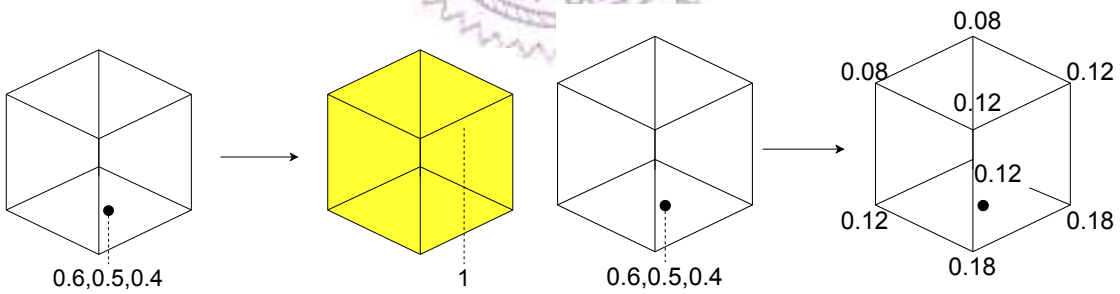


Figure 5.1: Proposed algorithm overview.

5.2 Food Intake Activity (FIA)



(a) (b)

Figure 5.2: Sample weights from (a) ordinary voxelization and (b) voxelization with trilinear interpolation.

In the field of mmWave radar data processing, a significant challenge arises from the varying number of points in each frame of the point cloud. This variability makes it difficult to directly utilize the point cloud as an input for neural networks. To address this issue, RADHAR [53] proposed a method called ”Voxelization” as a solution. Voxelization involves partitioning the 3D space into fixed-sized blocks and computing the number of points contained within each block. This approach provides neural networks with a consistent input size and simultaneously imbues the resulting 3D matrix with spatial sig-

nificance. To address the voxelization idea to fine-grained food intake activity recognition, we proposed the following steps.

Voxelization. For the voxelization, we divide the spatial space into wX, wY, wZ voxels along the three axes with a uniform dimension. Next, we count the number of points falling in each voxel, and form a 3D array to represent the voxelized point cloud frame.

Bounding box. Equally spacing voxels in the whole 3D space covered by the mmWave radar leads to wasted resources, as many voxels are allocated to the empty space. Prior work on coarse-grained human activity recognition [53] defines a *per-frame* bounding box by checking the maximum and minimum coordinates along the three axes. Let bX, bY, and bZ denote the size of the bounding box, which is divided into a cubical voxel with an edge length of r. For a current frame, we have $nX = bX/r$, $nY = bY/r$, and $nZ = bZ/r$. In our pilot tests, we found that a per-frame bounding box does not perform well in terms of recognizing fine-grained food intake activities. This may be attributed to the need for high-resolution voxels to recognize hand/face movements and the lack of sudden arm/leg movements. Hence, we proposed to have a fixed, rather than per-frame, bounding box. Notice that there exists a trade-off between the bounding box size and resolution, as the GPU memory is limited. On the one hand, we want to have high resolution (smaller r), but then we have to reduce the bounding box size (smaller bX, bY, bZ) and may miss some spatial information. We tried different parameters in evaluations to understand their implications. Finally, we set $(bX, bY, bZ) = (2, 3, 3)$ m and $r = 10$ cm as the default values.

Trilinear interpolation. One limitation of the ordinary voxelization approach is that each voxel keeps track of the numbers of points falling in it, which is an *integer*. An observation is that, no matter where a point in a voxel is located, the voxel value does not change, which may lead to unnecessary information loss. Inspired by the trilinear interpolation, we propose to distribute the weight of a point among all eight vertices of the voxel it falls in, based on the distance between the point to each voxel face. By doing so, we introduce continuous values on vertices instead of integer values on voxels, which preserves richer information. Fig. 5.2 shows the difference with/without trilinear interpolation.

To ease the math presentations, we let the bounding box corner closer to the mmWave radar, on the right of the subject, and on the floor as the new origin in the following derivations. First, we let $V(x, y, z)$ denote the (x, y, z)-th vertex weight, which is zeroed. Let P be all points of a frame. We iterate through $p = (x, y, z) \in P$. A voxel $(Px, Py, Pz) = (\lfloor \frac{x}{r} \rfloor, \lfloor \frac{x}{r} \rfloor, \lfloor \frac{z}{r} \rfloor)$ contains p. We compute the normalized distances to the three voxel faces as $\Delta x = (x \bmod r)/r$, $\Delta y = (y \bmod r)/r$, and $\Delta z = (z \bmod r)/r$.

Then, we increment the eight vertex weights by:

$$\begin{aligned} V(Px+\hat{x}, Py+\hat{y}, Pz+\hat{z}) \mathrel{+}= & [(1-\hat{x})\Delta x + \hat{x}(1-\Delta x)] \\ & \times [(1-\hat{y})\Delta y + \hat{y}(1-\Delta y)] \\ & \times [(1-\hat{z})\Delta z + \hat{z}(1-\Delta z)], \end{aligned} \tag{5.1}$$

where $\hat{x}, \hat{y}, \hat{z} \in \{0, 1\}$. After iterating through all points in the frame, we get the vertex weights V as a 3D array. Note that each vertex receives weights from points in all adjacent voxels. The number of vertex weights roughly equals the number of ordinary voxels; i.e., trilinear interpolation incurs negligible memory overhead.

However, the voxelization solution has a critical downside, as it costs enormous memory while almost all of the memory (around 98%) is 0, which is useless. Furthermore, the resolution is one key parameter that can heavily impact the performance; for example, a 10-cm resolution is a passable resolution for food intake activities and it already costs 4 megabytes of memory for a 2-second sample.

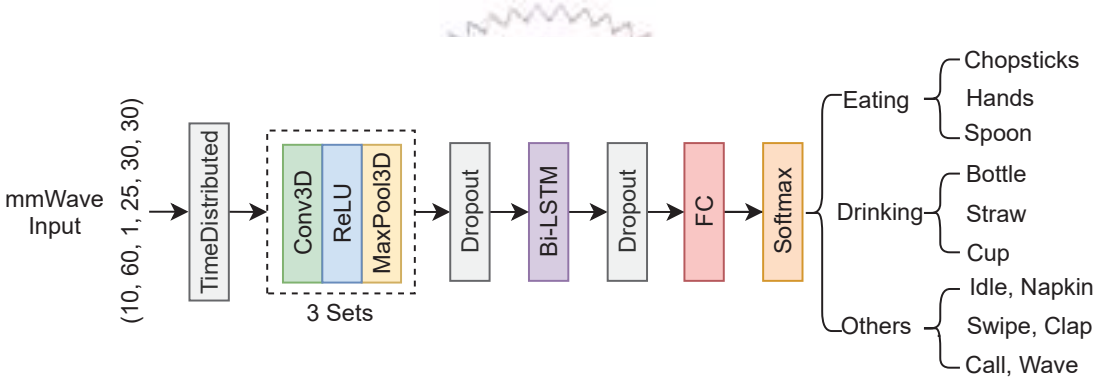


Figure 5.3: Our proposed neural network structure.

Fig. 5.3 gives the structure of our proposed neural network, which is mainly based on 3D CNN and Bi-Directional Long Short-Term Memory (Bi-LSTM) layers. Each mmWave input sample consists of 60 voxelized frames, which first goes through a time-distributed layer for lower complexity and time-step features. This layer is followed by three sets of convolution layers to identify motion features, and a Bi-LSTM layer to capture temporal features. These layers are separated by dropout layers to avoid overfitting. Compared to the neural networks in similar studies [8, 53, 69], our proposed network captures temporal features [8, 69] and has a simpler structure (fewer CNN layers) [53]. Next, we give the hyperparameters of our proposed neural network.

The mmWave input has six dimensions: batch size, frames per sample, number of channels, and voxel width, height, and depth. Each 3D CNN layer has 32 output channels, kernel size $3 \times 3 \times 3$, stride 1, and the same padding. Each max-pooling layer uses kernel size $2 \times 2 \times 2$ and stride 2. The dropout rates are 0.5 and 0.3, respectively. The Bi-LSTM

layer has 64 hidden units and the fully connected layer employs 128 neurons. Unless otherwise specified, we train the neural network with 50 epochs, a learning rate of 0.001, and a learning rate decay of 0.98.

5.3 Dynamic Point Cloud Recognizer (DPR)

In this section, we propose a *Dynamic Point Cloud Recognizer (DPR)*, adapting the model used in SPE to directly process dynamic point clouds for food intake activity recognition to reduce the GPU memory consumption compared to FIA.

Motivation FIA struggles to strike a balance between voxel side length and memory consumption due to the voxelization of sparse point clouds. Fig. 5.4 reveals the implication of different voxel lengths on: (i) the accuracy of food intake activity recognition and (ii) GPU memory consumption. It shows that finer voxels lead to higher accuracy at the expense of higher memory consumption. When the voxel length goes below a certain point (∼4.77 cm), the required GPU memory exceeds the hardware limitation (11 GiB, as indicated by the dashed line). Therefore, FIA's accuracy cannot be further improved below that voxel length on our server, motivating us to search for more compact representations, such as the 5-channel grids used by SPE. Based on SPE, we develop DPR, which significantly reduces memory consumption, enabling more efficient processing of point cloud data, while preserving the essential features required for accurate activity recognition.

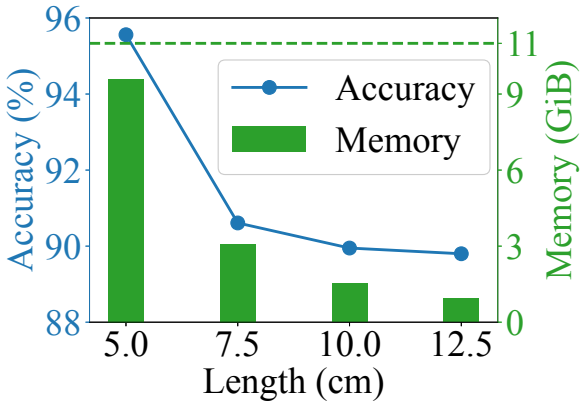


Figure 5.4: Implication of FIA's voxel length.

Proposed solution. DPR is built upon SPE's compact 8×8 grids, which capture spatial patterns. Unlike SPE, which directly feeds the output of the CNN into a fully connected layer, DPR's objective is to recognize activities from dynamic point clouds, which contain temporal information. Therefore, after passing point cloud frames through CNNs for spatial features, DPR leverages a *Recurrent Neural Network (RNN)*, more specifically

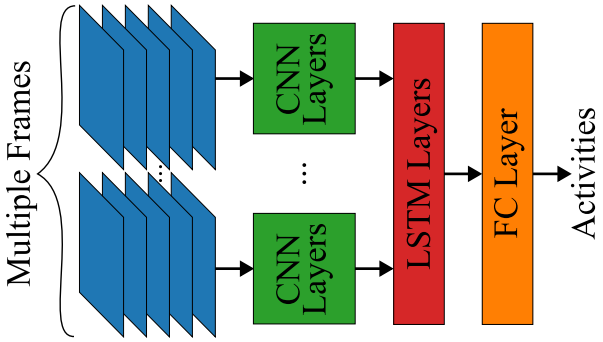


Figure 5.5: The structure of DPR algorithms.

Long Short-Term Memory (LSTM) [27], to capture temporal dependencies among frames. As illustrated in Fig. 5.5, DPR employs the combination of CNN and LSTM to handle the spatiotemporal information in dynamic point clouds.

DPR comes with a few system parameters. Let F be the number of consecutive frames sent into the model, L be the output length of each CNN (also the input length of the LSTM), N be the number of LSTM layers, H be the number of hidden LSTM states, D be the dropout rate to avoid overfitting, and let a Boolean variable B indicate whether *Bidirectional LSTM* is employed. We choose F=40 to cover a 4-sec duration, which is the time each repetition of activity was performed in the dataset. For the other parameters, we search for the optimal ones in the following section.

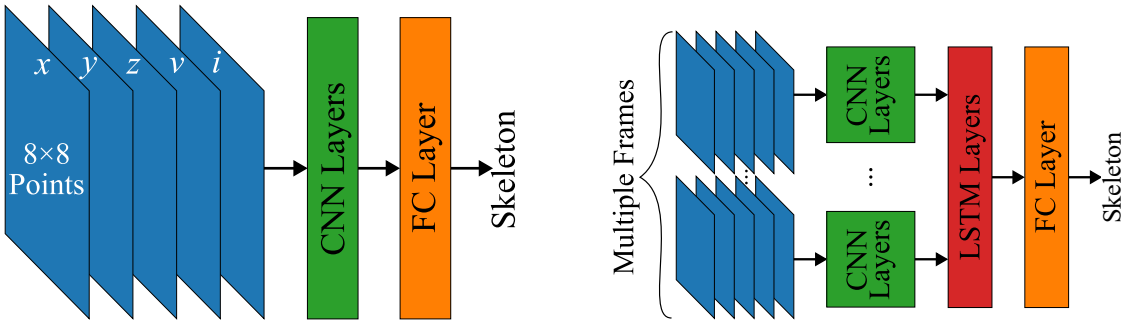
5.4 Skeletal Pose Estimator (SPE)

In this section, we develop an enhanced *Skeletal Pose Estimator (SPE)* capable of generating more precise skeletons for food intake activity recognition compared to *MARS* [3].

5.4.1 Motivation

MARS is originally designed for rehabilitation systems. When applied to estimate skeletons for food intake activities, it incurs significant errors in the estimated joints. More specifically, in our pilot tests[1], MARS suffered from an average error of $\sim$12 cm between the estimated and ground-truth joints, posing a challenge for accurate activity recognition. Such a significant error could be due to its relatively simple 2-layer CNN, which has limited its ability to capture intricate features. However, blindingly increasing the number of layers could lead to overfitting and the potential for vanishing gradients. Therefore, to enhance MARS, we have to carefully select the model structure and model depth to reduce the estimation error.

[1]We retrain all considered models using the same dataset throughout this thesis.



(a) (b)

Figure 5.6: The model structure of (a) MARS/SPE and (b) SPE+.

5.4.2 Proposed solution

Data preprocessing. Due to the distinct coordinate systems of mmWave point clouds and skeletons, it is necessary to harmonize these coordinate systems. We use the skeleton's coordinate system as the reference and adjust the x, y, and z coordinates of the mmWave point cloud data to ensure that all points utilize the same coordinate system. In this unified system, x represents the horizontal coordinate, y represents the depth coordinate, and z represents the vertical coordinate.

SPE/SPE+ implementation. Fig. 5.6(a) shows the MARS and SPE model structure. Each point cloud frame comes with 64 points arranged into an 8×8 grid with five channels[2], representing coordinates x, y, z, velocity v, and intensity i. To optimize the performance for the small feature map input in our case, we propose replacing the two-layer CNNs with more advanced and widely adopted network models, including: (i) *AlexNet* [33], a model structure that won the ImageNet competition, (ii) *GoogLeNet* [56], with *inceptions* for multi-scale features, and (iii) *ResNet* [25], with *residual connections* to resolve vanishing gradients. In the next section, we conduct experiments to compare the accuracy of the estimated skeletons from different network models. Additionally, inspired by DPR, we also tried to estimate skeletons by several frames' input. To handle the time-series data, an additional LSTM layer is set to get temporal features from the input. Fig. 5.6(b) shows the model structure of the SPE+.

5.5 Graph Convolution Network (GCN)

In this section, we use the human skeleton feature as the input of our classifier, we modified two GCN algorithms to fit our task, namely STGCN [70] and 2S-AGCN [51], and we will compare the performance of these two algorithms with different skeletons estimated

[2]The mmWave radar [58] gives at most 64 points per frame, which are zero-padded, and sorted based on x, then by y, and finally by z.

by different pose estimation models.

5.5.1 Motivation

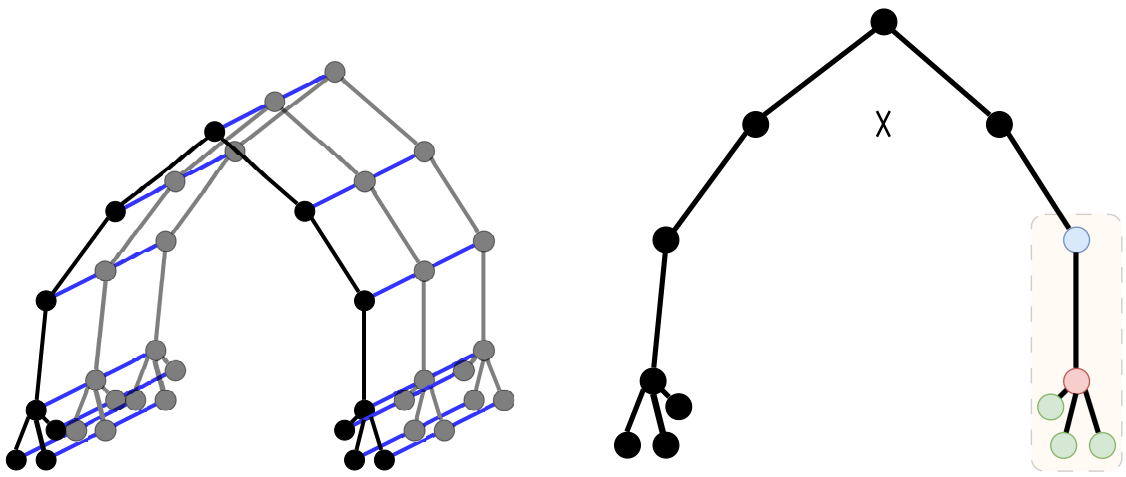
Human Activity Recognition (HAR) extensively employs human body skeleton features as a fundamental component. These features offer rapid insights into the details of behavior by swiftly comprehending the effects of skeletal movements on actions. In comparison to raw sparse point cloud data, the use of human body skeleton information is deemed to yield higher accuracy, enhancing the effectiveness of HAR applications.

In the current landscape of HAR using human skeleton features, Graph Convolutional Network (GCN) models have demonstrated the highest accuracy levels. Therefore, we have chosen to adopt the GCN model in our approach. GCN's ability to capture complex relationships within skeletal data, coupled with its superior accuracy, positions it as a suitable candidate for optimizing HAR performance.

5.5.2 Proposed solution

Our work involves the modification of two renowned Graph Convolutional Network (GCN) models: ST-GCN [70] and 2S-AGCN [51]. The first, ST-GCN, achieved groundbreaking performance improvement by integrating temporal information into the graph structure within the GCN model. This innovation was led by a pioneer who recognized the significance of incorporating time-based data. The second model, 2S-AGCN, is an evolution of the ST-GCN concept. This adaptation addresses the challenge of accurately recognizing dual-hand gestures, a limitation present in the original ST-GCN framework. The modifications introduced in 2S-AGCN were driven by the aspiration to refine hand gesture recognition, enhancing the model's ability to decipher complex hand movements. Since both models were invented for whole-body activity recognition, we modified the model to fit our task for the best accuracy.

Graph construction. The raw skeleton data in a single frame is consistently presented as a sequence of vectors. These vectors are 3D coordinates of human joints from a frame. Since there are multiple frames in an activity sequence, each potentially possesses different lengths across various samples. To capture the structured relationships among these joints in both the spatial and temporal dimensions, we utilize a spatio-temporal graph. The graph structure is introduced by the ST-GCN [70]. Figure. 5.7(a) shows an example spatio-temporal graph, which is a graph constructed with several frames of skeleton joints, linked with edges. There are two types of linking: spatial linking, which is represented as the black line, is the skeleton linking in a frame, while temporal linking, which is represented as the blue line, is the linking of the same joints in a skeleton but in different



(a) (b)

Figure 5.7: (a) The spatio-temporal graph and the (b) partition strategy for convolution.

frames.

Graph convolution. With the spatio-temporal graph structure, we employed the graph convolution methods to get the feature from the graph. After that, global average pooling layers and a softmax classifier can be applied to make predictions from the generated features.

The whole operation focuses on extracting the spatial feature; from the ST-GCN [70], the convolution formula is:

$$f_{out}(v_i) = \sum_{v_j \epsilon B_i} \frac{1}{Z_{ij}} f_{in}(v_j) \cdot w(l_i(v_j)) \tag{5.2}$$

where f means feature map, and v means the vertex of the graph. For each vertex v_i, it has its convolution area Bi, containing the vertexes(vj) with a distance of 1, and w is the weight function to implement the convolution which generates a weight vector. Each vertex has its own unique weight vector, and a special partitioning map li is introduced. The idea is shown in Fig. 5.7(b), where the $\times$ in the middle is the gravity center of the input skeleton, and it splits all the vertexes that are B_i into 3 subsets: (i) v_i itself, which is the red joint of the figure, (ii) the inner subset, which is the blue joint of the figure, for all $v_j \epsilon B_i$ the distance to the gravity center is shorter than v_i, and (iii) the outer subset, which is made up of the green joints of the figure. For all $v_j \epsilon B_i$, the distance to the gravity center is longer than v_i. Z_{ij} is a fraction to balance the contribution of each subset.

ST-GCN implementation. For the implementation part, the graph convolution formula from Eq. 3.2 transforms into:

$$f_{out} = \sum_{k}^{K_v} W_k (f_{in} A_k) \odot M_k \tag{5.3}$$

The shape of the feature map is $C \times T \times N$, where N is the number of vertexes, T is the frame length, and C is the number of channels. K_v represents the kernel size of the spatial dimension, and the value is set as 3 due to the partition strategy mentioned above. W_k represents the weight vector of convolution, just like the w in Eq. 3.2. A_k is the linking matrix including self-edge. M_k is an N $\times$ N attention map that indicates the importance of each vertex, and $\odot$ denotes the dot product.

2S-AGCN implementation. 2S-AGCN shares a similar idea with STGCN, but tries to solve the problem from the ST-GCN algorithms, the formula is:

$$f_{out} = \sum_{k}^{K_v} W_k f_{in}(A_k + B_k + C_k) \tag{5.4}$$

The ST-GCN uses only A_k to know if the vertexes are connected, and M_k to determine the feature relation strength between each node, while they multiply each other, which means the feature between the vertexes that is not connected in the skeleton will be lost, for example, the relation between the hands. In, 2S-AGCN, A_k is the linking matrix including self-edge, B_k is the attention map similar to M_k, and C_k is a data-dependent graph which learns a unique graph from the sample. It is a normalized embedded Gaussian function, which calculates the similarity of vertexes. The value in C_k is normalized between 0-1, representing the similarity. With the 2S-AGCN's convolution method, we can get more information from the limbs movement, which is more common for food intake activities.

2S-AGCN proposes learning features from not only joint coordinates but also from bone vectors. For the skeleton with N joints, it will be always $N - 1$ bones. The bone vectors come from its first joint $v1 = (x1, y1, z1)$ and its second joint $v2 = (x2, y2, z2)$, and the vector of the bone $V_{v1v2} = (x2 - x1, y2 - y1, z2 - z1)$. The last vector is pending with a $(0, 0, 0)$ vector.

Model structures.

Two algorithms share the same model structure. A TCN_GCN block is shown in Fig. 5.8. Spatial convolution layers and temporal convolution layers are followed by a batch normalization (BN) layer and a ReLU layer. There is a 0.5 dropout between spatial and temporal layers, and a residual connection is used in the block.

The whole model structure is shown in Fig. 5.9. It is the stack of TCN_GCN blocks. The output channels of the blocks are increasing; they are 64, 64, 64, 128, 128, 256, and 256, respectively. The output feature is followed with linear layers, a dropout layer with a value of 0.5 is placed to prevent overfitting, and the softmax layer does the final classification.

The 2S-AGCN's two streams network is shown in Fig. 5.10. Both the bone and joint data share the same graph construction and model structure, which means a bone can be

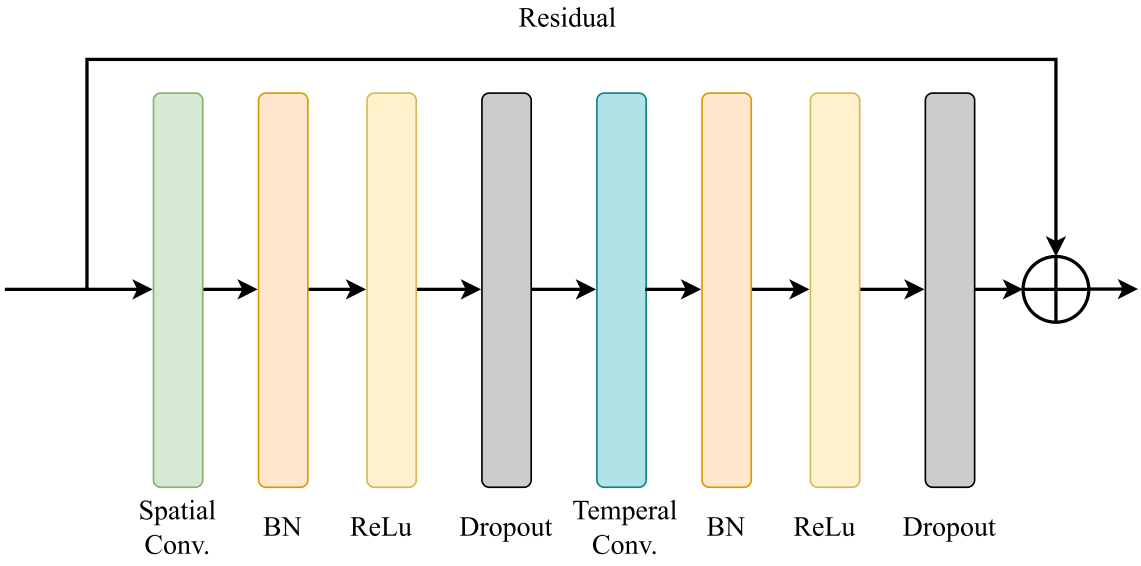


Figure 5.8: The structure of the TCN_GCN block.

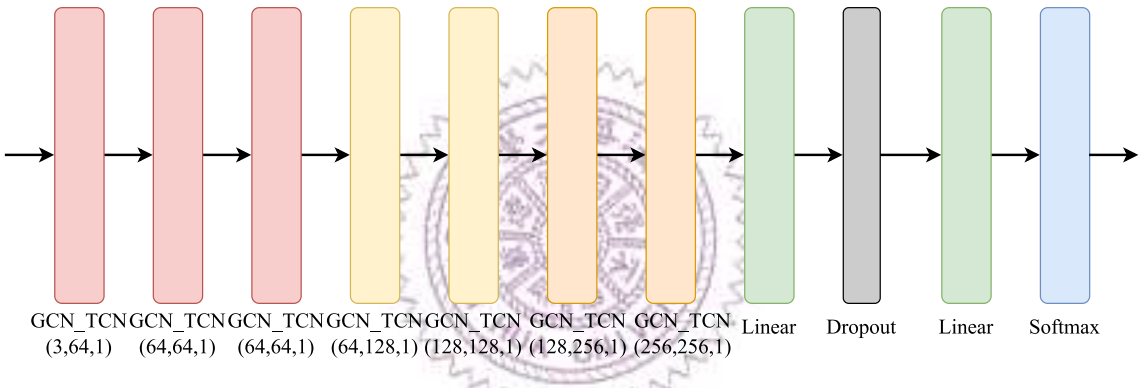


Figure 5.9: The structure of the GCN model.

viewed as a unique joint. Both stream features are stored and merged before being sent into the softmax with the same weight, giving the classifier the feature from both setups.

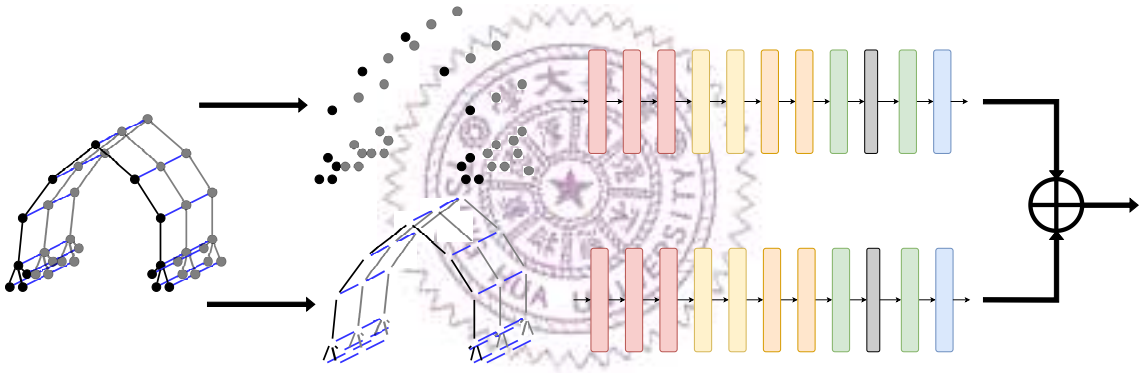


Figure 5.10: The pipeline of the 2-stream model.

Chapter 6

Dataset

In this chapter, we introduce the food intake activity dataset of 24 participants performing 12 fine-grained activities using sensors with heterogeneous levels of privacy sensitivity, including an RGB camera, a depth camera, and an mmWave radar.

6.1 Sensors

There are two types of radar systems, (i) pulse radar, and (ii) continuous wave (CW) radar. Pulse radar emits short pulses of radio waves and measures the time it takes for the pulse to return after hitting an object. They are designed for long-range object detection with low-range resolutions. CW radar, on the other hand, continuously emits a signal without interruptions, making it suitable for applications such as speed detection and fine-grained detection. Also, pulse radars emit much stronger electromagnetic radiation and are less suitable for indoor activity recognition.

Continuous-wave radars also have two varients: (i) *simple continuous-wave* radars, which only provide *velocity* data, and (ii) *Frequency-Modulated Continuous-Wave* (FMCW) radars, which offer both *range* and *velocity* data. Our study utilizes FMCW radar, which offers several advantages over other radar types. FMCW radar provides high range and velocity resolution, making it suitable for various applications. It is less susceptible to interference and noise compared to pulse radar. FMCW radar's continuous operation and ability to measure both range and velocity make it a versatile choice for sensing and detection tasks.

After surveying the off-the-shelf FMCW products, the Texas Instruments (TI) IWR1443BOOST 3D mmWave radar is the mmWave radar sensor for the dataset. As an FMCW radar, our mmWave radar allows us to calculate the distance to an object and to determine its velocity with good accuracy. Furthermore, compared to the 2D mmWave radars used in early studies [8, 69], 3D mmWave radars provide richer data since the

reflecting points come with *three* rather than *two* coordinates.

6.2 Dataset Collection

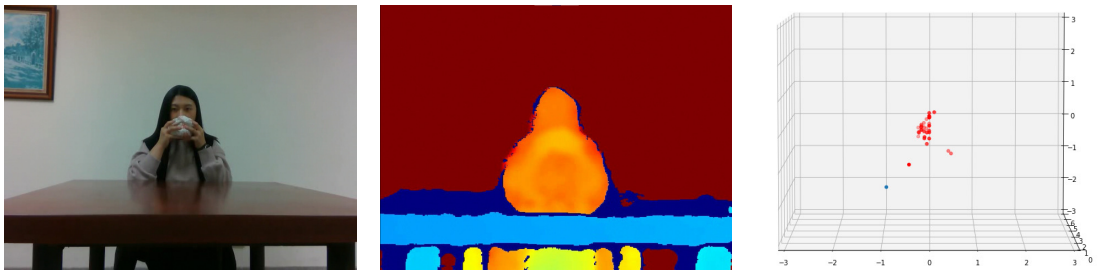
Table 6.1: Considered Activities

Act.	Description	Target Period (sec)	Act.	Description	Target Period (sec)
a01	Drinking tea with a cup	4	a07	Sitting still	Continuous
a02	Drinking tea with a bottle	4	a08	Picking up a call	4
a03	Drinking tea with a straw	4	a09	Wiping mouth with a tissue	4
a04	Eating a burger with hands	4	a10	Writing on a piece of paper	4
a05	Eating fruit with a fork	4	a11	Reading a book	4
a06	Eating noodles with chopsticks	4	a12	Scrolling a smartphone	Continuous

The dataset we are using is from [68]. This dataset provides a valuable resource for studying human behavior using mmWave radar and RGB-D video. The dataset encompasses a total of 12 activities, consisting of 6 related to food intake and 6 activities related to other table-side behaviors. The inclusion of mmWave radar data in this dataset introduces a novel approach to capturing human behavior while maintaining privacy. Unlike RGB and Depth video, which may contain identifiable features such as facial recognition or personal appearance, mmWave radar data focus on detecting and tracking the movement of objects without revealing personal details. This privacy-preserving feature ensures that individuals' identities are not compromised during data analysis and evaluation. The 6 food intake-related activities within the dataset provide insights into various aspects of eating behaviors. The difference between each activity is utensils. Each food intake-related activity includes the process of the subject picking up the food with utensils, bringing food to the mouth, chewing and swallowing, and putting the utensils back on the table. Using mmWave radar, researchers can monitor hand movements involved in food intake without compromising individuals' privacy. This enables detailed analysis of eating patterns and behaviors while safeguarding personal information. Moreover, the dataset also includes six other actions related to table-side activities. These actions include writing, reading, using electronic devices, or cleaning with tissue. Comparing them with food intake-related activities, researchers can explore the coordination between hand movements and these table-side behaviors, facilitating a deeper understanding of human actions in various contexts.

Fig. 6.1 presents the data of a subject eating a burger captured by three heterogeneous sensors: an RGB camera, a depth camera, and a mmWave radar from the dataset.

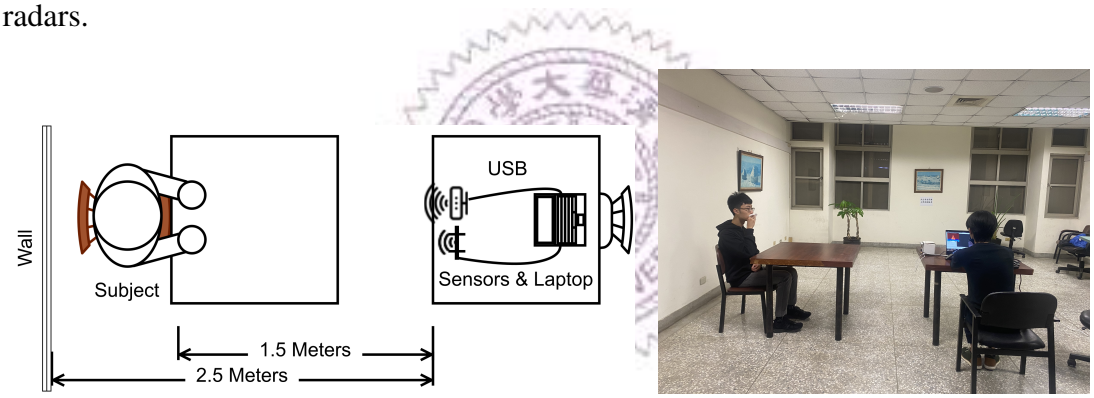
As illustrated in Fig. 6.2, human subjects performed various activities when sitting behind a table, mimicking the typical setup in a dining room. The table has a height of 75 cm. We placed the two above-mentioned sensors along with a laptop computer on another



(a) (b) (c)

Figure 6.1: Sample sensor data from: (a) an RGB camera, (b) a depth camera, and (c) a mmWave radar of a subject eating a burger.

table of the same height, which was 1.5 meters away from the subjects, and 2.5 meters away from the wall behind the subjects. The reason for having the wall behind the user is to avoid noise in depth images caused by open space. On the other hand, whether there is open space or not does not affect the mmWave, which is an advantage of using mmWave radars.



(a) (b)

Figure 6.2: The collection setup.

There are 24 subjects in the dataset: 12 men and 12 women, who were either university students or employees between 22 and 36 years old. We enumerate the subjects as s01, s02, ..., s24, where s01–s12 are male, and the rest are female. Subjects were asked to perform 12 different activities as summarized in Table 6.1. We ask each subject to repeat every activity for one minute, take a short (30 sec) break, and repeat the same activity for another minute.

Table 6.2 summarizes the software setup of the two sensors. We developed a set of scripts to automate the sensor data collection. During collection, we created two threads: one is responsible for dynamic 3D point cloud collection from the mmWave radar, and the other is responsible for gathering the RGB and depth video frames from the RGB-D camera. Our scripts are included in the dataset, allowing researchers and practitioners to initiate/stop data collection with a single keystroke.

Table 6.2: Sensors Used for Data Collection

Sensor	**mmWave Radar**	**RGB-D Camera**
Laptop OS	Ubuntu 20.04	
Model	TI IWR1443BOOST	Intel RealSense D435i
Driver	TI mmWave rospkg	Pyrealsense2
SDK	TI mmWave SDK	Librealsense
Data Type	Dynamic point clouds	RGB/Depth video clips
Frame Rate	10 fps	30 fps

To be able to drive the mmWave radar, we installed an open-source mmWave ROS Package [73]. With that, we wrote a script to collect point clouds through the mmWave SDK. Our mmWave radar is highly configurable. Table 6.3 gives the key configurations empirically derived through our pilot tests. Among these configurations, two key parameters are: (i) *peak grouping*, which instructs the radar to only report the strongest point among a group of nearby points, and (ii) *clutter removal*, which instructs the radar to skip the reflecting points that are stationary. To get denser point clouds, we set peak grouping to false when collecting the dataset. For clutter removal, it is not so clear whether we should enable or disable it. With clutter removal *on*, we may have a chance to gather more *moving* points as the mmWave module has a limitation of 64 processed points per frame. However, doing so may prevent the ML algorithms from "seeing" the stationary parts of the subjects or objects. Hence, we decided to collect the dataset twice, with and without clutter removal. Interested researchers can work with both through their ML algorithms to see which settings work better for them.

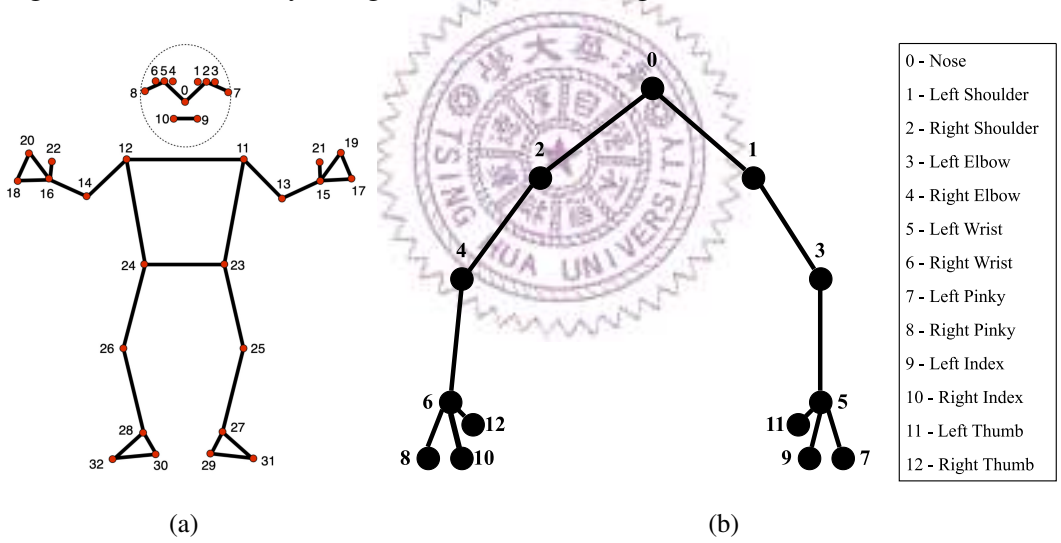
Table 6.3: Key mmWave Radar Configurations

Description	**Value**	**Description**	**Value**
Starting frequency	77 GHz	Range resolution	4.4 cm
Bandwidth	3.44 GHz	Max range	3.95 m
Frame rate	0.1 s	Velocity resolution	7 cm
No. chirps per frame	32	Max velocity	1 m
No. TX antennas	4	Peak grouping	*False*
No. RX antennas	3	Clutter removal	*On/Off*

6.3 Skeleton Generation

The original FIA dataset lacked essential human skeleton data, which is pivotal for effective human activity recognition. To overcome this limitation, a novel approach was adopted. RGB images from the dataset were employed as input data, and the Mediapipe

Pose Model [7] was utilized to synthesize human skeleton data. The Mediapipe Pose Model is designed for whole-body human pose estimation and provides a comprehensive set of 33 skeletal feature points shows in Fig. 6.3(a) However, we opted to select only 13 of these points due to specific reasons: (i)Upper-Body Focus: The dataset primarily captures the upper-body portion of individuals. Given this limitation, the lower-body skeleton data generated by the Mediapipe Pose Model were irrelevant and hence were excluded. (ii)Facial Landmarks Exclusion: The Mediapipe Pose Model generates human facial landmarks alongside skeletal points. However, as mmWave radar point clouds cannot detect facial features, these facial landmarks were removed from consideration. The 13 joints we chose are shown in Fig. 6.3(b) In light of these factors, we filtered the generated human skeleton data to suit the context of the FIA dataset. The synthesized data capture the upper-body skeletal structure while excluding irrelevant lower-body information and facial landmarks. This processed skeleton dataset would serve as ground truth for further experiments, offering accurate and relevant human skeletal data for subsequent stages of human activity recognition model training and evaluation.



(a) (b)

Figure 6.3: The landmarks from (a) Mediapipe, and (b) the FIA dataset.

Chapter 7

Evaluations with Global Models

In this chapter, we compare the performance of various models in a *global setup*, which is a commonly used training and testing division method. In the global setup, 80% of segments from all samples are designated as the training set, while 20% of segments are allocated to the testing set. In this global setup, data from each subject will appear in both the training and testing sets. In order not to make the data from a frame appear in both the training and the testing set, we employ the 80-20 split strategy by cutting each video sequence into 5 small sequences and then doing the 80-20 split. In that case, the model will not classify the activities just because it remembers the answer from the training test. We will begin by comparing the different parameters of the models. Subsequently, we will assess the performance differences among these models.

7.1 FIA Algorithm

In this section, we extensively evaluate the performance of our proposed food intake activity recognition system using the collected mmWave dataset.

7.1.1 Setup

We implemented our proposed Food Intake Activity (FIA) recognition pipeline using PyTorch 1.10. We ran our experiments on a Linux server having 2.1 GHz Xeon CPUs and NVIDIA 1080Ti GPUs with 10 GB memory. We considered three variants of our FIA algorithms: (i) *FIA-D*, which employs per-frame (dynamic) bounding boxes, (ii) *FIA-B*, which adopts a fixed bounding box, and (iii) *FIA-V*, which utilizes a fixed bounding box and weighted vertices. For comparison, we also implemented the state-of-the-art RadHAR [53], which also recognizes human activities using a 3D mmWave radar. Another work, m-activity [63], adopted the same dataset and network structure as RadHAR, but re-

placed the LSTM with GRU layers to trade the accuracy for shorter running time. Hence, we do not compare against (simpler) m-activity, for brevity.

Our FIA algorithms take the following parameters (where bold font indicates the default values): (i) *temporal aggregation frames*: $k \in \{\mathbf{1}, 3, 5, 7, 9\}$; (ii) *bounding box size*: $(bX, bY, bZ) \in \{(\mathbf{2}, \mathbf{3}, \mathbf{3}), (2, 3, 2), (2, 3, 1)\}$, which correspond to the **full-body**, half-body, and head-and-shoulder setups; and (iii) *resolution*: $r \in \{\mathbf{10}, 15, 20\}$. To quantify the performance, we adopt the following metrics: (i) accuracy, which is the fraction of correctly predicted activities, (ii) training time, (iii) inference time, and (iv) GPU memory usage. We use the dataset collected in advance to drive our experiments. Particularly, we arrange the dataset in two ways to answer two questions: (i) *when* a subject eats/drinks and (ii) *how* a subject eats/drinks. For the first question, we consider three general activities: *eating*, *drinking*, and *others*. For the second question, we consider all 12 detailed activities, which is more challenging. For the sake of presentation, we refer to these two arrangements as the *when* and *how* datasets in the rest of the paper.

7.1.2 Test Results

We start from the "when" dataset, followed by the "how" dataset.

Table 7.1: Accuracy Results from Different Algorithms in the Global Test with the When Dataset

Algorithm	When Dataset	How Dataset
RadHAR	76.43%	12.17%
FIA-D	90.79%	68.81%
FIA-B	93.56%	72.77%
FIA-V	96.73%	91.49%

Table 7.2: Confusion Matrix from the FIA-V Algorithm in the Global Test with the When Dataset

Accuracy	Drinking	Eating	Others
Drink	95.35%	2.13%	1.16%
Eating	2.71%	93.60%	0.46%
Others	1.94%	4.27%	98.38%

Our algorithms outperform the state-of-the-art. Table 7.1 presents the overall accuracy achieved by our algorithms as well as the start-of-the-art RadHAR [53], where both "when" (three activities) and "how" (12 activities) datasets are considered and default parameters are employed. We notice that RadHAR's accuracy is 76.43% while our

Table 7.3: The Overhead Comparison of Different Algorithms in Global Test

Algorithm	Preprocessing Time	Training Time	Inference Time	Memory
RadHAR	7.11 s / sample	482.41 s / epoch	0.063 s / sample	6580 MB
FIA-D	6.58 s / sample	167.06 s / epoch	0.055 s / sample	5192 MB
FIA-B	0.08 s / sample	165.51 s / epoch	0.058 s / sample	5780 MB
FIA-V	0.10 s / sample	174.64 s / epoch	0.058 s / sample	6705 MB

algorithms outperform RadHAR by at most 22.36%, reaching an accuracy of 96.73%. Among the three variants of our algorithms, FIA-V performs the best (96.73%), while FIA-D performs the worst (90.79%). This demonstrates the effectiveness of our proposed bounding box and weighted vertices on determining when a subject eats/drinks. For completeness, we give the accuracy achieved by FIA-V as a confusion matrix in Table 7.2. With the more challenging "how" dataset, our algorithms also reach as high as 91.49% accuracy, which is significantly higher than the testing accuracy of 12.17% achieved by RadHAR. This demonstrates that our FIA-V algorithm can indeed determine how a subject eats and drinks. For completeness, we also give the confusion matrix of FIA-V in Fig. 7.1. Table 7.3 shows the time and memory consumption from different algorithms. The preprocessing time for a sample of RadHAR is 7 seconds, which makes it impractical in real-time systems. On the contrary, thanks to the bounding box idea of FIA, it achieves a much shorter preprocessing time for each sample. FIA also has a shorter training time per epoch and inference time because of the lighter model structure. In summary, our FIA algorithms clearly outperform the state-of-the-art RadHAR. Moreover, because FIA-V performs the best among our proposed algorithms, we only report the performance from FIA-V in the following if not otherwise specified.

Further improvement test with different parameter values. Because our FIA algorithms have reached about 97% accuracy with the "when" dataset, which leave virtually no room for further improvement, we only vary the parameters of the FIA-V algorithm to see if we can further improve its accuracy with the "how" dataset. We plot the accuracy achieved by the FIA-V algorithm in Fig. 7.2. Fig. 7.2(a) reveals that the accuracy increases from 91.49% to 96.56% when the number of temporal aggregation frames is increased from 1 to 7. This can be attributed to the increased number of points in each aggregated frame. However, the accuracy drops when the number of temporal aggregation frames exceeds 7, probably because of the significant loss of temporal information. Fig. 7.2(b) reports the accuracy from different bounding box sizes. This figure shows that the half-body setup achieves the highest accuracy at 94.75%, and the head-and-shoulder setup achieves the lowest accuracy at 81.99%. This observation demonstrates that all the activities are performed around the upper body, and the half-body setup helps the classifier

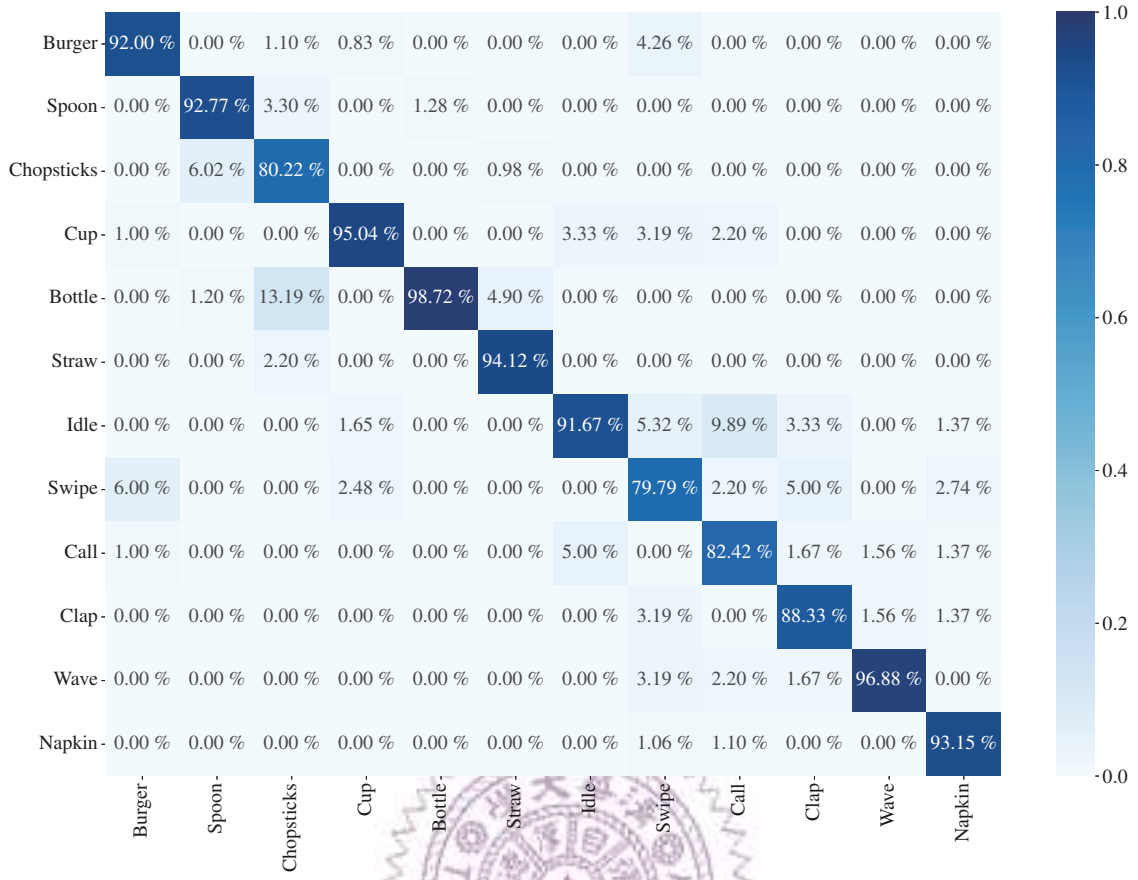


Figure 7.1: Confusion matrix from the FIA-V algorithm in the global test with the how dataset.

focus on the most important part. Fig. 7.2(c) shows the accuracy at different resolutions. We can see that even with all the proposed techniques, like weighted vertices, a smaller r still leads to higher accuracy: an increase of 5.95% is observed when reducing r from 20 to 10 cm. With the above comparisons, we have identified the best parameters of our FIA-V algorithm in the global test.

Testing with the public RadHAR dataset. To see if our proposed algorithms also work well with other public datasets, we compare FIA-D with RadHAR [53] using their dataset and their training/testing arrangement, which essentially is our global test. We chose the worst-performing FIA-D to be conservative. The RadHAR dataset includes 5 activities: (i) walking, (ii) jumping, (iii) boxing, (iv) jumping jacks, and (v) squats from two subjects. We trained/tested FIA-D and RadHAR following their experiment procedure. We found that both RadHAR and FIA-D achieved 91.50% accuracy. Furthermore, our training time is only 35% of that of RadHAR, thanks to our simpler neural network structure. In summary, our algorithms can also recognize coarse-grained activities accurately, with a shorter training time.

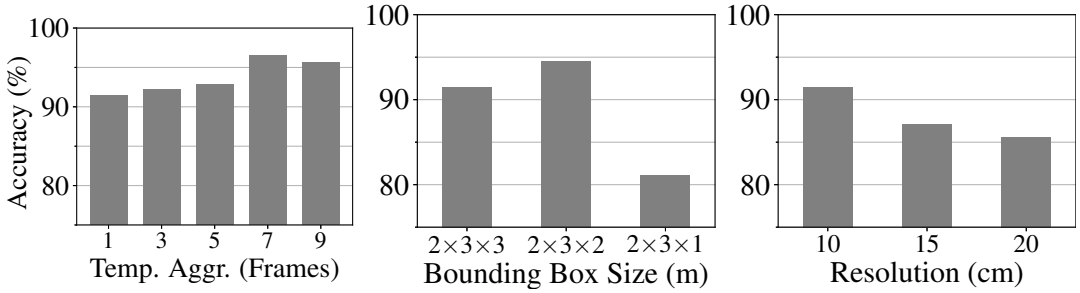


Figure 7.2: Accuracy results under different: (a) temporal aggregations, (b) bounding box sizes, and (c) resolutions. Sample results of the global test with the how dataset.

7.2 DPR Algorithm

7.2.1 Setup

We implemented DPR algorithm with PyTorch 1.10. We ran our experiments on a Linux server having 2.1 GHz Xeon CPUs and NVIDIA 1080Ti GPUs with 10 GB memory

We used the dataset collected in Chap.6 for the experiments. Since the *when* dataset is good enough in FIA, we only compared the performance difference between FIA and DPR in the *how* dataset to get a more significant difference.

DPR has several parameters. F is the number of consecutive frames and will be set as 40 with a stride of 10. We obtained 120 40-frame samples per activity per subject, which allowed us to get the feature from different states of the activity. This resulted in $24 \times 12 \times 120 = 34,560$ samples in total, for which we employed the 80-20 train-test split strategy with a batch size of 32. Let L be the output length of each CNN (also the input length of the LSTM), N be the number of LSTM layers, H be the number of hidden LSTM states, D be the dropout rate to avoid overfitting, and let a Boolean variable B indicate whether *Bidirectional LSTM* is employed. We chose F=40 to cover a 4-sec duration, which is the time each repetition of activity was performed in the dataset. For the other parameters, we search for the optimal ones in the following section.

7.2.2 Test Result

We evaluated DPR with various parameters: $L = \{\mathbf{39}, 256, 576\}$, $N = \{\mathbf{1}, 2, 3\}$, $H = \{\mathbf{64}, 128, 256\}$, $D = \{0.1, \mathbf{0.3}, 0.5\}$, and $B = \{$**true**, false$\}$, with the default parameters highlighted in **bold** font. Considering the huge number of combinations, we varied parameters individually while fixing others at their default values. Fig. 7.3 gives the accuracy under different parameter settings, from which we identify the optimal parameters: L^*=39, N^*=1, H^*=128, D^*=0.3, and B^*=true.

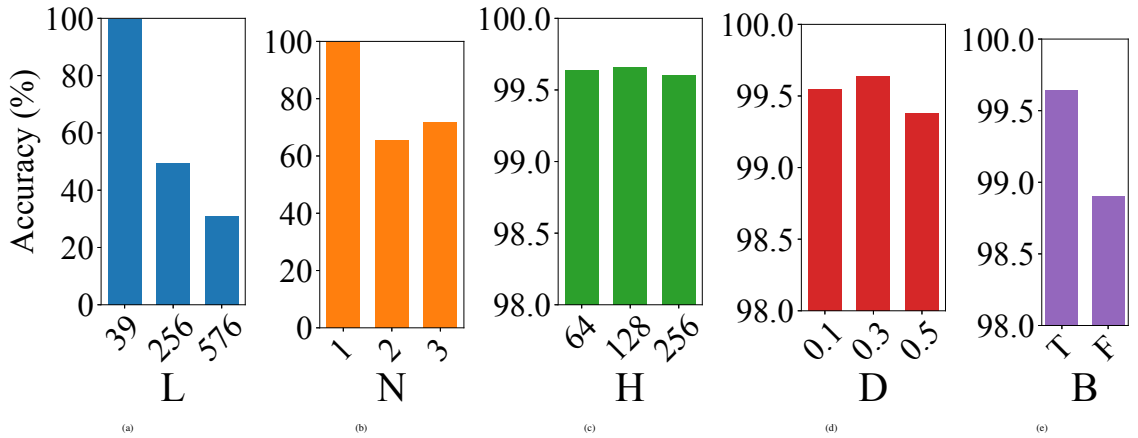


Figure 7.3: Accuracy comparison of different system parameters: (a) L, (b) N, (c) H, (d) D, and (e) B.

Table 7.4: The Overhead Comparison Between DPR and FIA

	Preprocessing Time (sample)	Training Time (epoch)	Inference Time (sample)	GPU Memory
FIA	100.15 ms	174.64 s	58.25 ms	6705 MiB
DPR	48.66 ms	53.75 s	7.7 ms	1373 MiB

DPR improves the accuracy and saves memory consumption compared to FIA. Fig. 7.4 shows the accuracy improvement for each activity, demonstrating that DPR outperforms FIA for all activities in the dataset. Specifically, using the optimal parameters, FIA achieves a classification accuracy of 95.56%, while DPR achieves 99.66%, indicating a 4.10% improvement Table 7.4 shows the resource consumption comparison between the two algorithms. FIA utilizes 6705 MiB of GPU memory, while DPR only uses 1373 MiB, resulting in a 79.38% reduction. The table also shows that, DPR has better time performance, only 30.78% and 13.21% of FIA in terms of training and inference times.

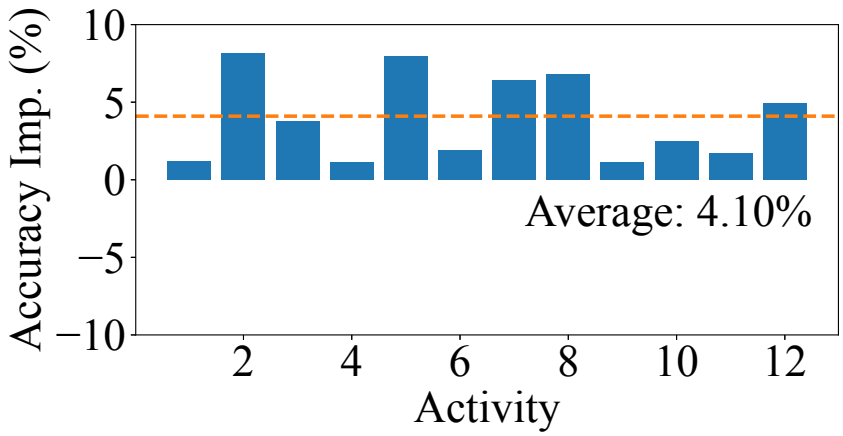


Figure 7.4: Accuracy improvement of DPR over FIA.

7.3 SPE Algorithm

In this section, we extensively evaluate the performance Skeletal Pose Estimator (SPE) capable of generating more precise skeletons for food intake activity recognition compared to MARS and mmPose-NLP(NLP).

7.3.1 Setup

We implemented the DPR algorithm with PyTorch 1.10. We ran our experiments on a Linux server having 2.1 GHz Xeon CPUs and NVIDIA 1080Ti GPUs with 10 GB memory We evaluated four model structures: MARS, AlexNet, GoogLeNet, and ResNet. For ResNet, we tested three variations with depths 18, 34, and 50, resulting in six models in total. Each model was trained for 100 epochs with a learning rate of 0.001 and a batch size of 512. These hardware specifications, library versions, and training settings are used throughout this thesis unless otherwise specified. Except for MARS, which is specifically designed for 8×8 inputs, other models require larger and different input sizes. For instance, AlexNet requires at least 63×63, and GoogLeNet requires at least 15×15. For a fair comparison, we scaled the input dimension to 64×64 for all models except MARS. Furthermore, these models are designed for RGB images, so we need to modify them in Torchvision 0.11.3 to accept five-channel inputs in their first CNN layers. The ground truth skeleton is from our proposed dataset in Chap.6.3, and randomly split with 80-20 train-test ratio, which leads to 270,848 training and 67,584 testing samples. We computed the *Euclidean distance* between each estimated and corresponding ground-truth joints in the 3D space as the *estimation error*. For the SPE+, we set a series of temporal aggregation values $3, 5, 7, 9, 11$ and examine which is the best setup for SPE+.

7.3.2 Test Result

The Recommended ResNet-34-based SPE model. Table 7.5 gives the resulting errors of 13 considered joints, along with 95% confidence intervals in parentheses. In this table, the network models are sorted in increasing complexity from left to right. The best-performing network models are highlighted in **bold** font. We make several observations: (i) ResNet outperforms the other three model structures, demonstrating the significance of residual connections in addressing vanishing gradients, (ii) ResNet-34 proves to be the most suitable model, as it exhibits the highest accuracy across all joints. It outperforms MARS [3] with an average reduction of 45.16% and a standard deviation of 2.79%. It also outperforms both its 18- and 50-layer counterparts, showing that more layers may not yield better results. Compared to MARS [3], our improved SPE can be utilized as

Table 7.5: Errors of the Skeletal Poses (cm)

	NLP	**MARS**	**AlexNet**	**GoogLeNet**	**ResNet-18**	**ResNet-34**	**ResNet-50**
Nose	9.85 (±0.18)	8.28 (±0.17)	6.91 (±0.06)	4.73 (±0.05)	4.41 (±0.05)	**4.22 (±0.05)**	4.75 (±0.08)
L. Shldr	7.93 (±0.18)	6.66 (±0.10)	5.65 (±0.05)	3.91 (±0.04)	3.74 (±0.04)	**3.58 (±0.04)**	3.93 (±0.05)
R. Shldr	7.84 (±0.18)	6.58 (±0.11)	5.54 (±0.05)	3.86 (±0.04)	3.67 (±0.04)	**3.52 (±0.04)**	3.91 (±0.06)
L. Elbow	10.31 (±0.18)	8.68 (±0.15)	7.13 (±0.05)	4.84 (±0.04)	4.59 (±0.04)	**4.39 (±0.04)**	4.83 (±0.05)
R. Elbow	9.97 (±0.18)	8.38 (±0.16)	7.05 (±0.05)	4.97 (±0.04)	4.76 (±0.04)	**4.57 (±0.04)**	4.94 (±0.04)
L. Wrist	13.68 (±0.18)	11.52 (±0.13)	9.60 (±0.06)	6.53 (±0.05)	6.36 (±0.05)	**6.06 (±0.05)**	6.55 (±0.06)
R. Wrist	14.02 (±0.18)	11.78 (±0.10)	10.19 (±0.06)	7.29 (±0.05)	7.15 (±0.05)	**6.85 (±0.05)**	7.33 (±0.06)
L. Pinky	14.91 (±0.18)	12.51 (±0.13)	10.44 (±0.07)	7.14 (±0.05)	6.96 (±0.05)	**6.62 (±0.05)**	7.13 (±0.06)
R. Pinky	15.79 (±0.18)	13.27 (±0.15)	11.47 (±0.07)	8.24 (±0.06)	8.08 (±0.06)	**7.74 (±0.06)**	8.28 (±0.06)
L. Index	14.91 (±0.18)	12.54 (±0.13)	10.50 (±0.07)	7.20 (±0.06)	7.00 (±0.05)	**6.67 (±0.05)**	7.21 (±0.06)
R. Index	15.68 (±0.18)	13.16 (±0.11)	11.41 (±0.07)	8.23 (±0.06)	8.06 (±0.06)	**7.73 (±0.06)**	8.26 (±0.06)
L. Thumb	14.23 (±0.18)	11.63 (±0.13)	9.70 (±0.06)	6.62 (±0.05)	6.46 (±0.05)	**6.15 (±0.05)**	6.63 (±0.05)
R. Thumb	13.84 (±0.18)	11.96 (±0.09)	10.36 (±0.06)	7.42 (±0.05)	7.29 (±0.05)	**6.99 (±0.05)**	7.47 (±0.06)
Average	12.26 (±1.45)	10.54 (±1.33)	8.92 (±1.16)	6.23 (±0.85)	6.04 (±0.86)	**5.78 (±0.83)**	6.25 (±0.85)

Table 7.6: Time and Memory Consumption Among Different Algorithms

	Preprocessing Time (sample)		Training Time (epoch)		Inference Time (sample)		GPU Memory
	Time	STD	Time	STD	Time	STD	MiB
mmPose-NLP	0.6 ms	0.10	277.71 s	2.61	145.8 ms	2.97	10840
MARS	1.17 ms	0.03	54.24 s	3.06	0.6 ms	0.03	1321
SPE	1.23 ms	0.04	201.67 s	3.27	6.3 ms	0.22	1675
SPE+	10.83 ms	0.94	488.62 s	3.53	6.9 ms	0.25	1867

a foundation of various activity recognition systems. In the remainder of this thesis, for brevity, we use the term SPE to refer to the ResNet-34-based SPE.

SPE has the best overall performance in global setup. Fig. 7.5(a) shows that in the global setup, SPE and SPE+ have the same performance, while Fig. 7.5(b) shows that the values of temporal aggregation make no significant performance improvement in the current circumstances. Table 7.6 shows the overhead comparison among SPE, SPE+, and two baseline algorithms: mmPose-NLP and MARS. mmPose-NLP has the shortest preprocessing time but has the longest inference time because of the complicated model and voxelization. Compared with SPE, SPE+ has a longer calculation time without improving the performance. Hence, we recommend SPE for high performance and low overhead.

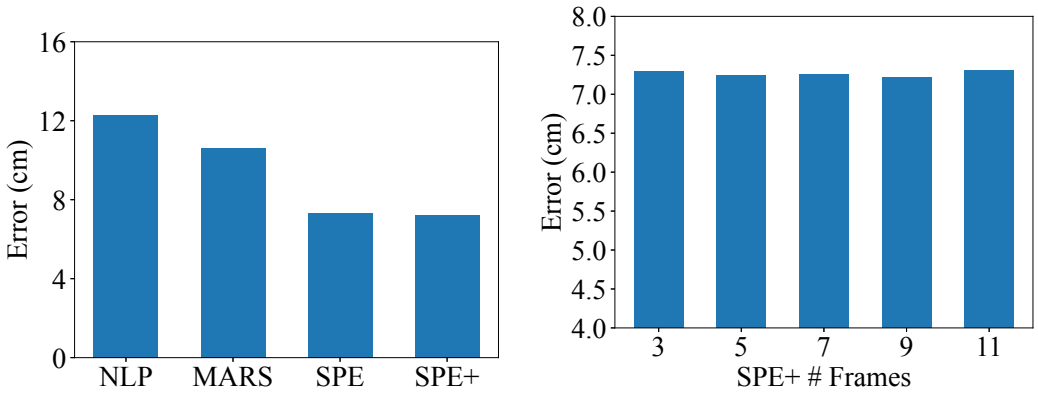


Figure 7.5: The accuracy result in distance among different (a) algorithms and (b) temporal aggregation values in SPE+.

7.4 Graph Convolution Network (GCN)

7.4.1 Setup

In this section, we compare the performance differences among various setups of GCN networks. Regarding the input skeleton data for GCN, we have four different inputs: (i) MARS, (ii) SPE, (iii) SPE+, generated from mmWave point cloud predictions, and the ideal input, which is the original mediapipe pose skeleton (MP).

As we delve into the performance comparison of different classifiers, it is noteworthy that FIA and DPR exhibit significant advantages over other algorithms within the global setup. Therefore, our comparisons will be limited to FIA and DPR, and the GCN-based models, ST-GCN and 2S-AGCN, to examine the outcomes.

To make sure it is fair to compare to the other end-to-end algorithms, we use the same sample generation setup as DPR's experiment, in which F is set as 40 with a stride of 10, 120 40-frame samples per activity per subject. and about $24 \times 12 \times 120 = 34,560$ samples in total.

7.4.2 Test Results

Table 7.7: The Overhead Comparison of GCN and DPR in Global Test

	Preprocessing Time (sample)		Training Time (epoch)		Inference Time (sample)		GPU Memory
	Time	STD	Time	STD	Time	STD	MiB
DPR	48.66 ms	1.39	53.75 s	0.64	7.7 ms	0.16	1373
GCN	317 ms	10.95	53.30 s	1.494	47 ms	1.03	1813

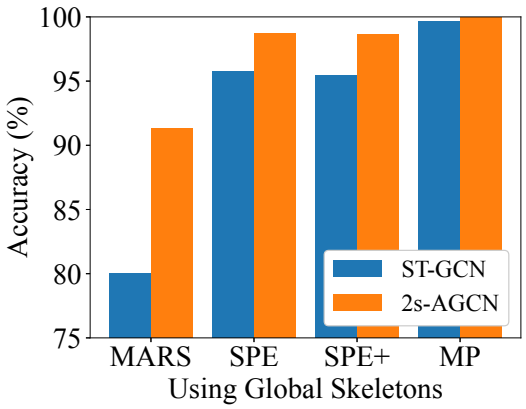


Figure 7.6: The accuracy of different algorithms.

Quality of the estimated skeletons affects the performance. Fig. 7.6 shows the performance difference among different algorithms, the estimated skeleton has a significant impact on the model's ability to classify activities. In our experiments, the accuracy for MARS was the lowest, at 80.02% for ST-GCN and 91.32% for 2s-AGCN. SPE achieved accuracies of 95.79% and 98.57% in the respective models, while SPE+ achieved 95.48% and 98.68%. The ideal mediapipe skeleton (error distance = 0) reached almost 100% accuracy. In summary, as indicated in Fig. 7.5(a), we observed that higher error distances result in lower accuracy. Since SPE and SPE+ have almost the same error distance, their skeleton data reach the same accuracy.

2s-AGCN outperformed ST-GCN with a significant difference. Fig. 7.6 also shows the performance improvement between 2s-AGCN and ST-GCN. Since food intake activity recognition highly relies on getting features from arms and hands, this is what 2s-AGCN's advantage is. 2s-AGCN algorithm also shows it has higher error tolerance while suffering from high error skeletons such as the MARS skeleton, as it can outperform ST-GCN by 11.30%.

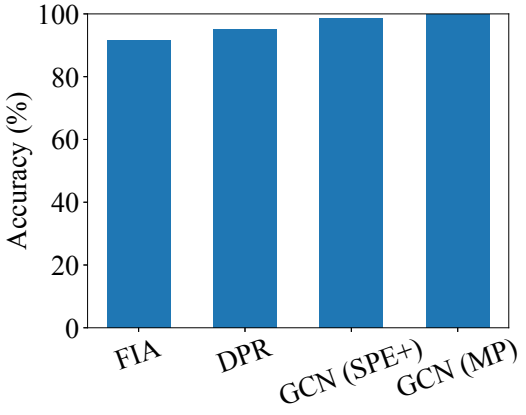


Figure 7.7: The accuracy of different algorithms.

GCN algorithms had better performance than the end-to-end algorithms. Fig. 7.7 reported that the 2s-AGCN algorithm with SPE+ skeleton reaches 98.68% accuracy. On the contrary, DPR reaches 95.59% and FIA achieves 91.95%. The result shows that GCN algorithms are outperforming end-to-end algorithms by 3.09% and 6.67%. In summary, under this circumstance, the estimated skeleton with a GCN model can help the classifier recognize activities better than directly using the raw data. Table 7.7 shows GCN has a longer preprocessing time, where we include the skeleton estimation time, GCN also has a long inference time because of the high complexity of GCN network. Nonetheless, for a 4-sec sample, both algorithms can recognize the subject's activities in real time.

Chapter 8

Evaluations with Leave-One-Out Models

In this chapter, we conduct a performance comparison of various models in the *leave-one-out setup*. This configuration is designed to simulate real-world classifier deployment scenarios. The data used in this setup will be sourced from a subject who has never encountered the model before, making it more challenging to do the classification. In this setup, all samples from one specific subject will constitute the testing set, while samples from all other subjects will form the training set. We will begin by comparing the different parameters of the models. Subsequently, we will assess the performance differences among these models.

8.1 FIA Algorithm

In this section, we extensively evaluate the performance of our proposed food intake activity recognition system using the collected mmWave dataset.

8.1.1 Setup

The hardware setup is completely the same as the global setup, and the four algorithms to be compared are three FIA algorithms: (i) *FIA-D*, (ii) *FIA-B*, (iii) *FIA-V*, and RadHAR [53].

Our FIA algorithms take the following parameters (where bold font indicates the default values): (i) *temporal aggregation frames*: $k \in \{\mathbf{1}, 3, 5, 7, 9\}$; (ii) *bounding box size*: $(bX, bY, bZ) \in \{(\mathbf{2}, \mathbf{3}, \mathbf{3}),\ (2, 3, 2),\ (2, 3, 1)\}$, which correspond to the **full-body**, half-body, and head-and-shoulder setups; and (iii) *resolution*: $r \in \{\mathbf{10}, 15, 20\}$. To quantify the performance, we adopt the following metrics: (i) accuracy, which is the fraction of

correctly predicted activities, (ii) training time, (iii) inference time, and (iv) GPU memory usage. We use the dataset collected in advance to drive our experiments. Particularly, we arrange the dataset in two ways to answer two questions: (i) *when* a subject eats/drinks and (ii) *how* a subject eats/drinks. For the first question, we consider three general activities: *eating*, *drinking*, and *others*. For the second question, we consider all 12 detailed activities, which is more challenging. For the sake of presentations, we refer to these two arrangements as the *when* and *how* datasets in the rest of the paper.

8.1.2 Test Results

We focus on the results from the "when" dataset first.

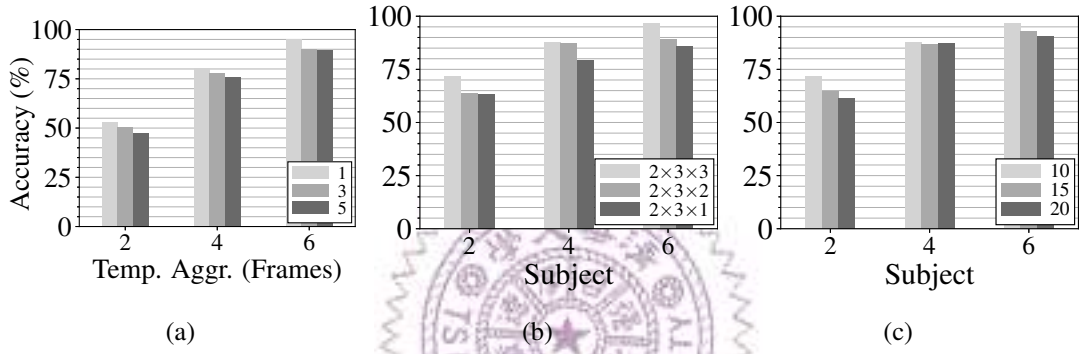


Figure 8.1: Accuracy results for different: (a) temporal aggregations, (b) bounding box sizes, and (c) resolutions. Sample results of the leave-one-out test with "when" dataset from subjects 2, 4, and 6 are shown.

Implications of various parameters. The best parameters of our FIA algorithms in the leave-one-out test could be different from those derived in the global test presented in Sec. 7.1. We report the sample results from subjects 2, 4, and 6 for brevity. Fig. 8.1(a) shows that when increasing the number of temporal aggregation frames from 1 to 5, the accuracy drops by about 5%. This indicates that the temporal aggregation does not improve the accuracy in the leave-one-out test, probably because the LSTM layer already takes care of the temporal features, and the eating/drinking speed of the users are different. Moreover, temporal aggregation reduces the temporal resolution of dynamic point clouds, which causes small performance drops. Fig. 8.1(b) reveals that the full-body setup achieves the highest accuracy between 71.75% and 96.55%, and the head-and-shoulder setup achieves the lowest accuracy between 63.12% and 85.67%. This observation demonstrates that even though all the activities are performed around the upper body, there are still useful features provided by the lower body. In addition, different heights of subjects might also play a role in accuracy. Fig. 8.1(c) depicts that a smaller r leads to higher accuracy: an increase between 1.33% to 10.25% is observed when re-

ducing r from 20 to 10 cm. With the above comparisons, we have identified the best parameters of our FIA-V algorithm in the leave-one-out test. In the following, we report the results with no temporal aggregation, full-body bounding box, and 10 cm resolution, unless specified otherwise.

Table 8.1: Sample Results from Different Algorithms in the Leave-One-Out Test with the When Dataset from Subject 6

Algorithm	Accuracy	Preprocessing Time	Training Time	Inference Time	Memory
RadHAR	52.13%	7.11 s	385.05 s	0.063 s	6580 MB
FIA-D	85.70%	6.58 s	133.65 s	0.055 s	5192 MB
FIA-B	94.66%	0.08 s	133.46 s	0.058 s	5780 MB
FIA-V	96.55%	0.10 s	138.60 s	0.058 s	6705 MB

Our algorithms outperform the state-of-the-art. We run a sample experiment with subject 6 as the testing subject and give the results in Table 8.1. We first observe that RadHAR failed to classify the three activities in the "when" dataset and has an accuracy of merely 52.13%, which is not much better than tossing a coin. In contrast, our algorithms outperform RadHAR by at most 44.22%, reaching an accuracy as high as 96.55% (FIA-V). This demonstrates the effectiveness of our proposed bounding box and weighted vertices methods for food intake activity recognition, particularly to answer the "when" question. Our algorithms also benefit from its simpler neural network structure, e.g., they consume about 1/3 of the training time of RadHAR, as indicated in Table 8.1. In summary, for the "when" dataset, our FIA algorithms also outperform the state-of-the-art by at most 44.22% and at least 33.57% in accuracy, as well as in shorter training and inference times. Thanks to the bounding box provided a fixed detection area and resolution, FIA-B & FIA-V algorithm can achieve real-time preprocessing for each sample. Since the gap between our algorithms and the state-of-the-art RadHAR [53] is clear with testing subject 6, we omit experiments with other subjects.

Achieved accuracy from different testing subjects. Next, we dive deeper and plot the confusion matrix with the median and best-performing results from the six experiments with different testing subjects in Fig. 8.2. We observe that our FIA-V algorithm can answer the "when" question. More specifically, the accuracy levels from the best-performing subject (along the diagonal) are: 95.12%, 91.67%, and 98.63%. We next report the accuracy and F1 scores achieved by different subjects using 6-fold experiments in Fig. 8.3. This figure reveals that the achieved accuracy varies between 60.32% and 96.55%. Subjects 4–6 achieve 85+% accuracy, which is nontrivial, as they are tested against a neural network trained only with *other* subjects. In addition, the F1 scores of different activities are consistent with the overall accuracy. We take a closer look at the less-than-perfect accuracy from subjects 1–2 and find the following probable causes: (i)

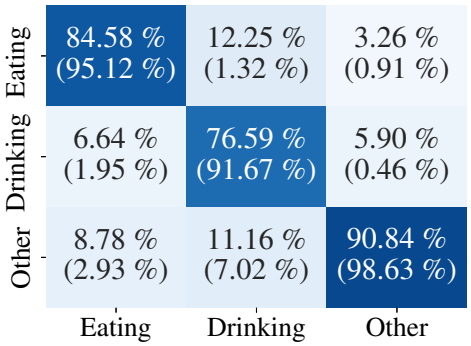


Figure 8.2: Confusion matrix of median and best-performing (within parentheses) subjects in the leave-one-out test with "when" dataset.

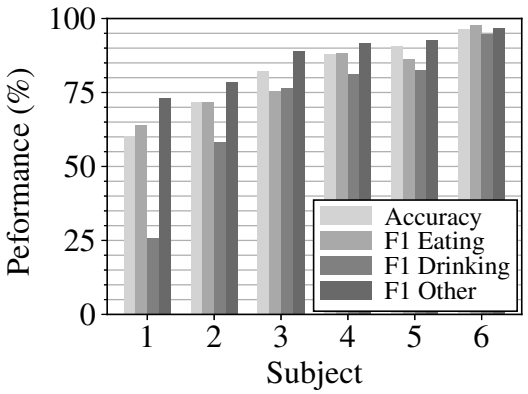


Figure 8.3: The performance results from individual testing subjects in leave-one-out test with "when" dataset.

Subject 1 is the only left-handed subject; (ii) Subject 2 performs activities much more slowly.

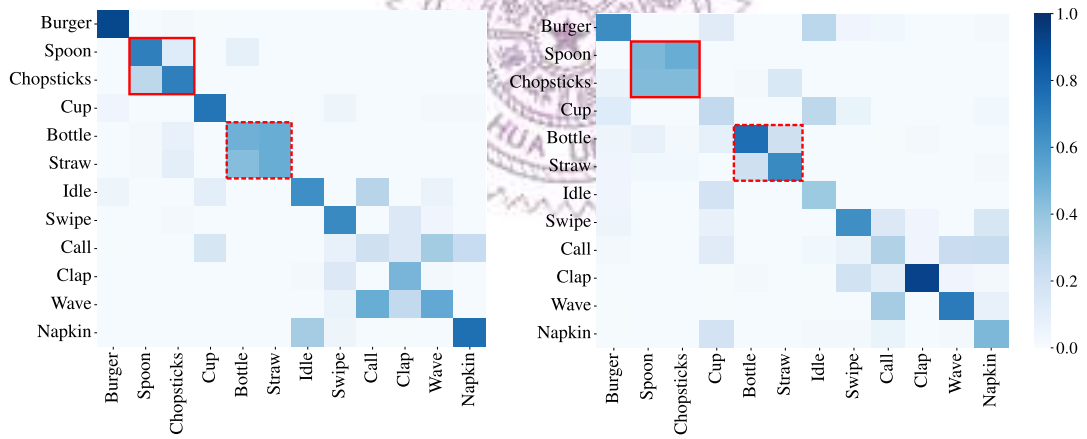


Figure 8.4: Confusion matrices of the leave-one-out test on the "how" dataset: (a) the best testing subject and (b) the median of six subjects.

The leave-one-out test of our FIA algorithms on the "how" dataset. For the "how" question (with 12 detailed activities) in the more challenging leave-one-out test (where testing subjects are excluded during training). We conduct 6-fold experiments, where one subject is reserved for testing. We give the confusion matrices of our 6-fold experiments from: (i) the best-performing subject and (ii) the median among 6 subjects in Fig. 8.4, where the overall accuracy is 62.89% and 53.33%, respectively.

8.2 DPR Algorithm

8.2.1 Setup

The hardware setup is completely the same as the global setup, and the parameters are also the same, including F, L, N, H, D, and B. We carried out a parameter search similar to the one in Sec. 7.2, using the same batch size of 32, and found the following optimal parameters: L^*=256, N^*=2, H^*=512, D^*=0.3, and B^*=true. Since there are 24 subjects in the dataset, we will run 24-fold cross-validation and discuss the findings from the results.

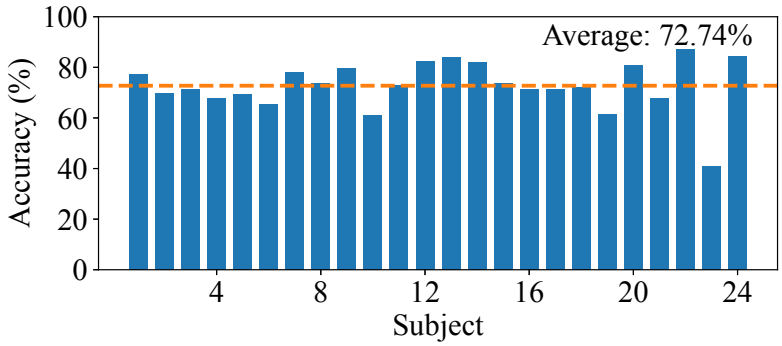


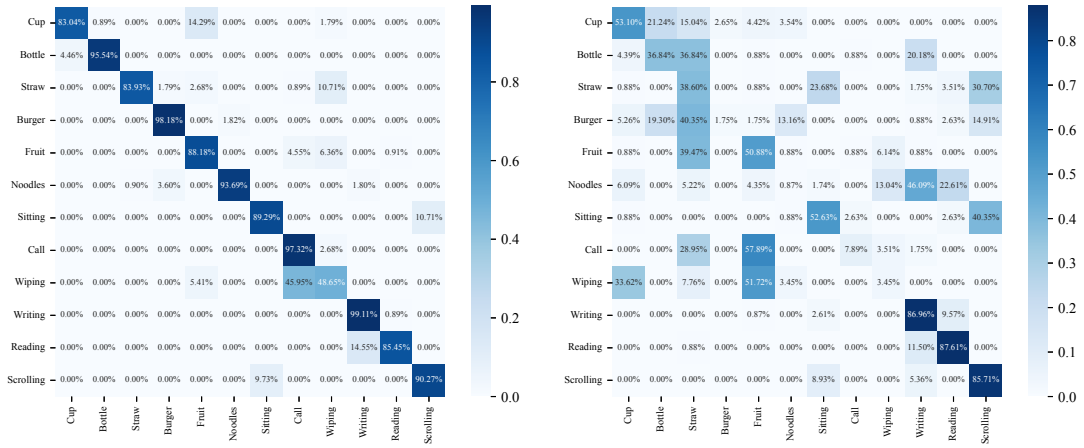
Figure 8.5: Accuracy of DPR in the leave-one-out test.

8.2.2 Test Results

Per-subject accuracy. Fig. 8.5 reports the results of the 24-fold cross-validation, showcasing the variability in accuracy depending on the subject being tested. The accuracy ranges from a high of 86.97% to a low of 40.77%, with left-handed subjects posing a particular challenge. On average, DPR achieves an accuracy of 72.74%, which a standard deviation of 9.64%, as indicated by the dashed line. This shows that the lack of prior exposure to unseen subjects during training could hinder the model's accuracy.

Misrecognized activities. We next dive into the results from two sample subjects, trying to identify the activities that are more likely to be mixed up. Fig. 8.6 gives the confusion matrices from the best and worst subjects in terms of accuracy (reported in Fig. 8.5). For the best subject (22), most of the activities are correctly recognized, with the exception of around half of the *wiping mouth* samples misrecognized as *picking up a call*. This is understandable, as these two activities are hard to differentiate without recognizing a cellphone or tissue using RGB images. For the worst subject (23), only a few activities, such as writing, reading, and scrolling, are correctly recognized. Further investigation into the RGB data revealed that the subject performed the activities in a

more relaxed or “lazy” manner compared to other subjects, which might have resulted in distinct patterns in the dynamic point clouds that deviated from the norm, making her an outlier in the dataset. This can be improved in the future by collecting a larger dataset that represents enough data from people with different styles of eating and drinking.



(a) (b)

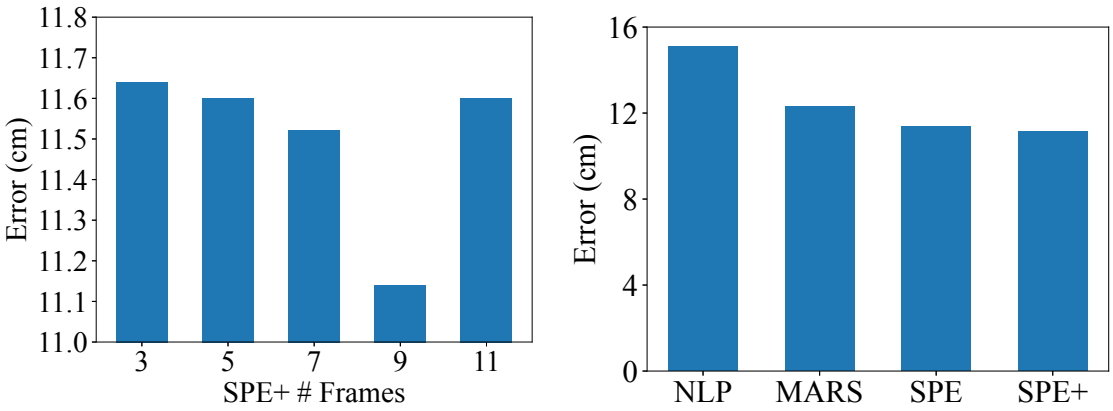
Figure 8.6: The confusion matrices for the (a) best and (b) worst subject of DPR.

8.3 SPE algorithm

In this section, we extensively evaluate the performance Skeletal Pose Estimator(SPE) capable of generating more precise skeletons for food intake activity recognition compared to MARS and mmPose-NLP(NLP).

8.3.1 Setup

The hardware setup is completely the same as the global setup. To examine the performance difference among models, we set a fixed parameter setup for SPE(SPE+), which is the ResNet-34-based SPE. The ground truth skeleton is from our proposed dataset in Chap.6.3. For the leave-one-out setup, $23/24$ of the dataset is in the training set, which is around 324,331 frames, while the testing set contains $1/24$ of the dataset, which is around 14,101 frames. We computed the *Euclidean distance* between each estimated and corresponding ground-truth joints in the 3D space as the *estimation error*. For the SPE+, we set a series of temporal aggregation values $3, 5, 7, 9, 11$ and examine which is the best setup for SPE+.



(a) (b)

Figure 8.7: The error distance from (a) SPE+ with different temporal aggregation values, and (b) different skeleton estimation algorithms.

8.3.2 Test Results

Implications of various parameters. Fig. 8.7(a) reveals that the error distance decreases from 11.64 cm to 11.14 cm when the number of temporal aggregation frames is increased from 3 to 9. However, the accuracy drops when the number of temporal aggregation frames exceeds 9, making setting temporal aggregation value 9 the best parameter for SPE+.

SPE+ outperformed the other algorithms in leave-one-out setup. Fig. 8.7(b) reported the error distances of different algorithms, SPE+ has the best performance of 11.14 cm on average, on the contrary mmpose-NLP, MARS, and SPE reaches 15.11, 12.29, and 11.38 cm on average, with a standard deviation of 2.45, 2.14, 1.52, and 1.51 cm, respectively. It shows that the temporal features do help the model to estimate a more accurate skeleton than directly estimating a skeleton by one single frame.

8.4 GCN algorithm

In this section, we extensively evaluate the performance Graph Convolution Network(GCN) classifiers from the estimation skeleton from different skeleton estimation models, and we compare GCN classifier's performance with end-to-end algorithms.

8.4.1 Setup

In this section, we compare the performance of GCN networks with the same algorithms in the global setup. Regarding the input skeleton data for GCN, we have four different inputs: (i) MARS, (ii) SPE, (iii) SPE+, and the original mediapipe pose skeleton (MP).

For the SPE+ skeleton, we generate the estimated skeleton using the SPE+ model with a value of 9 frames for temporal aggregation.

As the performance comparison of different classifiers, the FIA and DPR still have huge advantages over other algorithms. Therefore, our comparisons will only compare with FIA and DPR, and the GCN-based models, ST-GCN and 2S-AGCN, to examine the outcomes.

8.4.2 Test Results

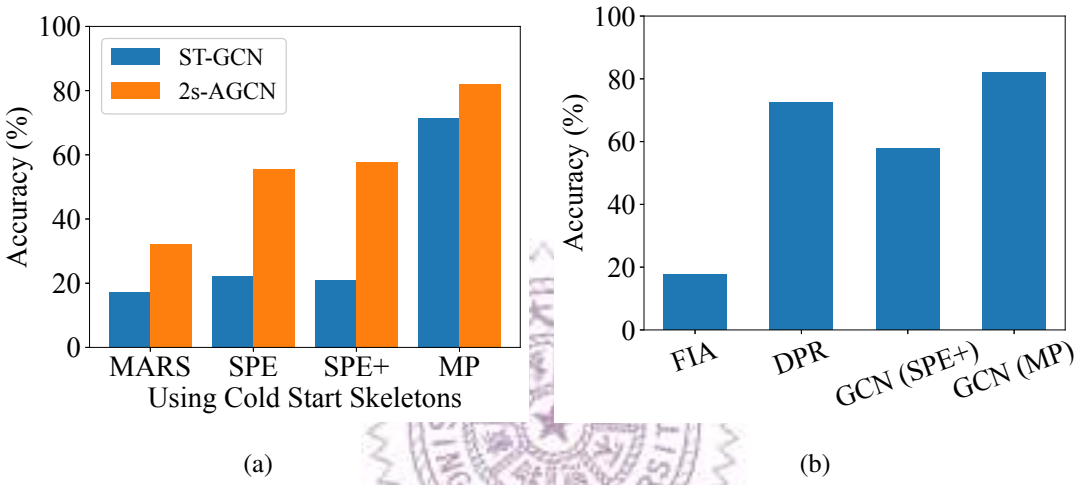


Figure 8.8: The accuracy comparison of (a) GCN with different estimated skeletons, and (b) accuracy with different algorithms.

Quality of the estimated skeletons affects the performance critically. Fig. 8.8(a) shows the performance difference among different algorithms, the estimated skeleton has a significant impact on the model's ability to classify activities. In our experiments, the accuracy for MARS was the lowest, with the high error distance skeleton with hard questions, MARS skeleton reaches only 17.10% and 32.13% accuracy in ST-GCN and 2s-AGCN, respectively. The SPE and SPE+ skeletons, because of their better quality skeleton, reach 22.15% and 20.88% for the ST-GCN model, and 55.65% and 57.87% for the 2s-AGCN model. The mediapipe pose skeleton's result indicates the ideal result for the GCN model; it reaches 71.51% for the ST-GCN model and 82.15% for the 2s-AGCN model, with a standard deviation of 12.75% and 10.95%.

2s-AGCN outperformed ST-GCN with a significant difference. Fig. 8.8(a) also shows the performance difference between 2s-AGCN and ST-GCN. 2s-AGCN has outperformed ST-GCN by 10.64% to 33.50%. It clearly shows that when dealing with more challenging questions with worse-quality skeleton input, 2s-AGCN's design gives it a higher error tolerance and tends to get the features from the input. Furthermore, food

intake activity recognition highly relies on getting features from arms and hands, which gives 2s-AGCN better performance.

GCN algorithms compared with end-to-end algorithms. Fig. 8.8(b) shows the performance of the leave-one-out setup. FIA algorithms only achieve 17.63% accuracy on average, which means they can not recognize a new subject's activities. The DPR algorithm has the highest accuracy for mmWave radar input, reaching 72.42% accuracy on average. While the GCN with SPE+ estimated skeleton, because the classifier is suffering from a low-quality skeleton with a huge error distance, it reaches 57.87% accuracy under current circumstances. As for the GCN with ideal skeletons, which is from the mediapipe pose, the GCN algorithm reaches 82.15% accuracy, which shows that the GCN algorithm has the potential to have better performance than end-to-end algorithms if the skeleton quality is improved.

Chapter 9

Conclusions & Future Works

In this chapter, we organize the results of all algorithms in both the global and leave-one-out scenarios. Additionally, in the future work section, we propose potential improvements to address the performance observed in the leave-one-out setup. We also explore opportunities for extending the use of mmWave radar point cloud data to other activity recognition scenarios.

9.1 Concluding Remarks

In this thesis, we endeavor to employ mmWave radar for recognizing human food intake activities. We present a total of a publicly available dataset and 4 algorithms. The dataset is the first food intake activity dataset with heterogeneous levels of privacy sensitivity, including an RGB camera, a depth camera, and a mmWave radar, and the dataset contains 24 participants performing 12 fine-grained activities. The first algorithm, FIA, serves as our initial attempt to address this problem. We employ preprocessing techniques such as voxelization, bounding box construction, and trilinear interpolation. These are combined with a CNN+Bi-LSTM neural network classifier. In the global setup, FIA achieves an accuracy of 91.49%, surpassing the state-of-the-art (SOTA) that employed voxelization methods at that time. To overcome the issues of excessive memory usage, imprecise classification of subtle movements, and poor performance in the leave-one-out setup observed with FIA, we introduce the end-to-end DPR algorithm. DPR directly utilizes the point coordinates, intensity, and velocity from the mmWave point cloud as input features. This results in a remarkable reduction of 79.38% in GPU memory usage compared to FIA, as FIA requires the pre-generation of large voxelized data. Additionally, DPR saves approximately 90% of disk space. As for accuracy, DPR demonstrates a notable improvement in the global setup, achieving an accuracy of 95.59%. DPR also achieved the current best accuracy of 72.46% in the leave-one-out setup.

While attempting to address the issues encountered with FIA, we introduced a pipeline that predicts skeleton features and uses them as classifier inputs. In the context of skeleton estimation, we proposed SPE and SPE+. These models utilize a ResNet architecture and similarly utilize the mmWave point cloud's point coordinates, intensity, and velocity as input features. SPE and SPE+ outperform the existing MARS and mmPose-NLP models and are currently the models with the smallest error distance. Regarding classifiers that use skeleton data as input, we directly modified the widely used ST-GCN and 2s-AGCN models. In the global setup, these models surpassed DPR, achieving an accuracy of 98.68%. Unfortunately, in the leave-one-out setup, due to the suboptimal quality of the currently estimated skeleton, the SPE+GCN pipeline only achieved an accuracy of 57.87%. However, when using the ideal skeleton, mediapipe pose (MP), as input, the classifier reached an accuracy of 82.42%, demonstrating the potential of the GCN model.

9.2 Future Work

In this section, we propose our future work to improve the classifier model and the further application of the classifier on different recognition tasks.

First, our dataset is the first to combine mmWave point cloud data with RGBD images for food intake activities recognition. Currently, the subjects in the dataset are limited to the same area and perform actions at a similar frequency. In the future, we plan to collect a larger-scale dataset, encompassing variations in subject distances, action frequencies, and angles. This approach will enable us to develop an even more generalized model suitable for real-world system applications.

In the leave-one-out setup, the current algorithms still have room for improvement. We have identified three main areas for future enhancements:

- **Adjustment of the overall pipeline:** Since current end-to-end algorithms only use mmWave radar point cloud data as input, incorporating data from other privacy-preserving sensors, such as depth images from our proposed dataset, for two-stream fusion model training could lead to better performance.

- **Refinement of the skeleton models:** In our experimental results, we observed the potential of skeleton classifiers. However, we also noted that the quality of skeleton data significantly impacts classifier performance. Therefore, continuous improvement of skeleton estimation models can positively influence overall accuracy.

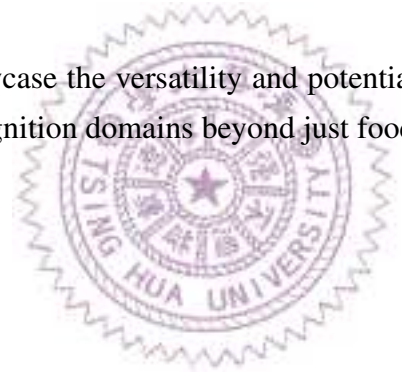
- **Transfer learning as personalization model:** We mentioned that the leave-one-out setup simulates a real-world scenario where a classifier is used by a subject it has never seen before. In practical situations, users can provide feedback to refine

the model based on their usage patterns, achieving personal optimization, by the feedback given by the subject. Hence, we plan to introduce a small amount of subject data through transfer learning to potentially improve performance.

The mmWave radar is not limited to food intake activity recognition, and thus, we can extend the use of this sensor and classifier to other indoor fine-grained activity recognition scenarios:

- **Driver Monitoring System (DMS):** Recognizing driver behavior is a popular topic, and given that most driving actions are related to hand movements, there is a certain degree of relevance between DMS and food intake activity recognition.
- **Gesture Recognition:** The mmWave radar has been proven effective in detecting small moving objects. Utilizing it for gesture recognition is also a trending area of interest. Leveraging our skeleton pipeline for gesture recognition is a feasible development as well.

These applications showcase the versatility and potential of mmWave radar technology in various activity recognition domains beyond just food intake.



Bibliography

[1] N. Ahmed, J. Rafiq, and M. Islam. Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors*, 20(1):317:1–317:19, 2020.

[2] O. Amft and G. Tröster. Recognition of dietary activity events using on-body sensors. *Artificial Intelligence in Medicine*, 42(2):121–136, 2008.

[3] S. An and U. Ogras. Mars: mmwave-based assistive rehabilitation system for smart healthcare. *ACM Transactions on Embedded Computing Systems*, 20(5s):1–22, 2021.

[4] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proc. of the International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 437–442, 2013.

[5] S. Balli, E. Saugbacs, and M. Peker. Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm. *Measurement and Control*, 52(1-2):37–45, 2019.

[6] F. Baradel, C. Wolf, and J. Mille. Human activity recognition with pose-driven attention to rgb. In *Proc. of British Machine Vision Conference (BMVC)*, pages 1–14, 2018.

[7] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann. Blazepose: On-device real-time body pose tracking. *arXiv preprint arXiv:2006.10204*, 2020.

[8] S. Bhalla, M. Goel, and R. Khurana. Imu2doppler: Cross-modal domain adaptation for doppler-based activity recognition using imu data. *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):145:1–145:20, 2021.

[9] G. Bhat, N. Tran, H. Shill, and U. Ogras. w-har: An activity recognition dataset and framework using low-power wearable devices. *Sensors*, 20(18):5356, 2020.

[10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[11] J. Cheng, B. Zhou, K. Kunze, C. C. Rheinländer, S. Wille, N. Wehn, J. Weppner, and P. Lukowicz. Activity recognition and nutrition monitoring in every day situations with a textile capacitive neckband. In *Proc. of the ACM Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 155–158, 2013. Demo Paper.

[12] Dfintech. Cisco visual networking index: Forecast and methodology, 2016-2021, 2022.

[13] M. Farooq and E. Sazonov. A novel wearable device for food intake and physical activity recognition. *Sensors*, 16(7):1067, 2016.

[14] M. Farooq and E. Sazonov. Accelerometer-based detection of food intake in free-living individuals. *IEEE Sensors Journal*, 18(9):3752–3758, 2018.

[15] R. Fisher, S. Blunsden, and E. Andrade. Behave: Computer-assisted prescreening of video streams for unusual activities, 2011.

[16] R. Fisher, J. Santos-Victor, and J. Crowley. Caviar: Context aware vision using image-based active recognition, 2011.

[17] A. Franco, A. Magnani, and D. Maio. A multimodal approach for human activity recognition based on skeleton and rgb data. *Pattern Recognition Letters*, 131:293–299, 2020.

[18] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco, and M. Luaces. A public domain dataset for real-life human activity recognition using smartphone sensors. *Sensors*, 20(8):2200, 2020.

[19] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3582–3589, 2014.

[20] P. Gong, C. Wang, and L. Zhang. Mmpoint-gnn: Graph neural network with dynamic edges for human activity recognition through a millimeter-wave radar. In *Proc. of International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2021.

[21] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[22] L. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, J. Yang, and S. Guo. Wiar: A public dataset for wifi-based activity recognition. *IEEE Access*, 7:154935–154945, 2019.

[23] L. Harnack, L. Steffen, D. Arnett, S. Gao, and R. Luepker. Accuracy of estimation of large food portions. *Journal of the American Dietetic Association*, 104(5):804–806, 2004.

[24] M. Hassan, M. Uddin, A. Mohamed, and A. Almogren. A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81:307–313, 2018.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] S. He, S. Li, A. Nag, S. Feng, T. Han, S. Mukhopadhyay, and W. Powel. A comprehensive review of the use of sensors for food intake detection. *Sensors and Actuators A: Physical*, 315:112318:1–112318:16, 2020.

[27] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] J. Hu, W. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5344–5352, 2015.

[29] Y. Huang, W. Li, Z. Dou, W. Zou, A. Zhang, and Z. Li. Activity recognition based on millimeter-wave radar by fusing point cloud and range–doppler information. *Signals*, 3(2):266–283, 2022.

[30] A. Iosifidis, E. Marami, A. Tefas, and I. Pitas. Eating and drinking activity recognition based on discriminant analysis of fuzzy distances and activity volumes. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2201–2204, 2012.

[31] A. Jain and V. Kanhangad. Human activity classification in smartphones using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, 18(3):1169–1177, 2017.

[32] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[34] H. Liu and T. Schultz. A wearable real-time human activity recognition system using biosensors integrated into a knee bandage. In *Proc. of International Conference on Biomedical Electronics and Devices*, pages 47–55, 2019.

[35] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork. Harth: A human activity recognition dataset for machine learning. *Sensors*, 21(23):7853, 2021.

[36] S. Mekruksavanich and A. Jitpattanakul. Smartwatch-based human activity recognition using hybrid lstm network. pages 1–4, 2020.

[37] D. Micucci, M. Mobilio, and P. Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.

[38] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain. A survey on food computing. *ACM Computing Surveys*, 52(5):1–36, 2019.

[39] W. Min, L. Liu, Z. Luo, and S. Jiang. Ingredient-guided cascaded multi-attention network for food recognition. In *Proc. ACM International Conference on Multimedia (MM)*, pages 1331–1339, 2019.

[40] A. Moin, A. Zhou, A. Rahimi, A. Menon, S. Benatti, G. Alexandrov, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan, et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics*, 4(1):54–63, 2021.

[41] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 4903–4911, 2017.

[42] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

[43] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao. Intelligent human-computer interaction based on surface emg gesture recognition. *IEEE Access*, 7:61378–61387, 2019.

[44] N. Rashid, M. Dautta, P. Tseng, and M. Faruque. Hear: Fog-enabled energy-aware online human eating activity recognition. *IEEE Internet of Things Journal*, 8(2):860–868, 2020.

[45] A. Salehzadeh, A. Calitz, and J. Greyling. Human activity recognition using deep electroencephalography learning. *Biomedical Signal Processing and Control*, 62:102094, 2020.

[46] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proc. of the International Conference on Pattern Recognition (ICPR).*, pages III:32–III:36, 2004.

[47] N. Selamat and S. Ali. Automatic food intake monitoring based on chewing activity: A survey. *IEEE Access*, 8:48846–48869, 2020.

[48] A. Sengupta and S. Cao. mmpose-nlp: A natural language processing approach to precise skeletal pose estimation using mmwave radars. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[49] A. Sengupta, F. Jin, R. Zhang, and S. Cao. mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, 20(17):10032–10044, 2020.

[50] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[51] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.

[52] N. Sikder and A.-A. Nahid. Ku-har: An open dataset for heterogeneous human activity recognition. *Pattern Recognition Letters*, 146:46–54, 2021.

[53] A. Singh, S. Sandha, L. Garcia, and M. Srivastava. Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proc. of the ACM Workshop on Millimeter-wave Networks and Sensing Systems (mmNets)*, pages 51–56, 2019.

[54] T. Singh and D. Vishwakarma. A deeply coupled convnet for human activity recognition using dynamic and rgb images. *Neural Computing and Applications*, 33(1):469–485, 2021.

[55] A. Stisen, H. Blunck, S. Bhattacharya, T. Prentow, M. Kjaergaard, A. Dey, T. Sonne, and M. Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proc. of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 127–140, 2015.

[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[57] Texas Instrument. Iwr1443 data sheet, product information and support — ti.com, 2023.

[58] Texas Instruments. Iwr1443boost evaluation module mmwave sensing solution - user's guide, 2020.

[59] K. Verma and B. Singh. Deep multi-model fusion for human activity recognition using evolutionary algorithms. *International Journal of Interactive Multimedia & Artificial Intelligence*, 7(2), 2021.

[60] C. Wang, T. S. Kumar, W. De Raedt, G. Camps, H. Hallez, and B. Vanrumste. Eat-radar: Continuous fine-grained eating gesture detection using fmcw radar and 3d temporal convolutional network. *arXiv preprint arXiv:2211.04253*, 2022.

[61] C. Wang, Z. Lin, Y. Xie, X. Guo, Y. Ren, and Y. Chen. Wieat: Fine-grained device-free eating monitoring leveraging wi-fi signals. pages 1–9, 2020.

[62] K. Wang, Q. Wang, F. Xue, and W. Chen. 3d-skeleton estimation based on commodity millimeter wave radar. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 1339–1343. IEEE, 2020.

[63] Y. Wang, H. Liu, K. Cui, A. Zhou, W. Li, and H. Ma. m-activity: Accurate and real-time human activity recognition via millimeter wave radar. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8298–8302, 2021.

[64] G. Weiss, K. Yoneda, and T. Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.

[65] A. Wellnitz, J. Wolff, C. Haubelt, and T. Kirste. Fluid intake recognition using inertial sensors. In *Proc. of international Workshop on Sensor-based Activity Recognition and Interaction (iWOAR)*, pages 1–7, 2019.

[66] Z. Wharton, A. Behera, Y. Liu, and N. Bessis. Coarse temporal attention network (cta-net) for driver's activity recognition. In *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1279–1289, 2021.

[67] Y.-H. Wu, Y. Chen, S. Shirmohammadi, and C.-H. Hsu. Ai-assisted food intake activity recognition using 3d mmwave radars. In *Proc. of the ACM International Workshop on Multimedia Assisted Dietary Management (MADiMa)*, pages 81–89, 2022.

[68] Y.-H. Wu, H.-C. Chiang, S. Shirmohammadi, and C.-H. Hsu. A dataset of food intake activities using sensors with heterogeneous privacy sensitivity levels. In *Proc. of the 14th Conference on ACM Multimedia Systems*, pages 416–422, 2023.

[69] Y. Xie, R. Jiang, X. Guo, Y. Wang, J. Cheng, and Y. Chen. mmeat: Millimeter wave-enabled environment-invariant eating behavior monitoring. *Smart Health*, 23:10023:1–10023:8, 2022.

[70] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proc. of the AAAI conference on artificial intelligence*, volume 32, 2018.

[71] K. Yatani and K. Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *Proc. of ACM Conference on Ubiquitous Computing (UbiComp)*, pages 341–350, 2012.

[72] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Proc. of Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II:123–II:130, 2001.

[73] L. Zhang. Github-radar-lab/ti_mmwave_rospkg, 2019.

[74] M. Zhang and A. A. Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proc. of ACM Conference on Ubiquitous Computing (UbiComp)*, pages 1036–1043, 2012.