Privacy Preserving Human Action Recognition with Joint Edge-Cloud Computing

by

Chi Chung Chan

BASc, Simon Fraser University, 2018

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science

in the
School of Engineering Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2023

Declaration of Committee

Name:	**Chi Chung Chan**
Degree:	**Master of Applied Science**
Thesis title:	**Privacy Preserving Human Action Recognition with Joint Edge-Cloud Computing**
Committee:	**Chair:** Shahram Payandeh Professor, Engineering Science
	Jie Liang Supervisor Professor, Engineering Science
	Zhenman Fang Committee Member Assistant Professor, Engineering Science
	Faisal Beg Examiner Professor, Engineering Science

Abstract

Recognition of human actions is crucial in several fields, including medical usage, human interaction applications, and video surveillance systems. In this thesis, we aim to develop a joint edge-cloud system for recognizing human actions that incorporate embedded devices, a cloud server, and a mobile app. A significant requirement for the application is privacy protection, which prohibits the direct transmission of video from the devices. However, the computational resources of embedded devices are limited and inflexible due to constraints in CPU/GPU, memory, and power supply. As a result, the embedded devices cannot execute overly complex algorithms. In this thesis, a joint edge-cloud computing approach is developed, where the embedded device runs the relatively simple human pose estimation algorithm to convert the original human action into skeleton animation, which is transmitted to the cloud. This not only protects the user's privacy, but also reduces the transmission cost. In addition, the cloud can then utilize the more powerful computational resource to run more advanced algorithms, so that more complicated human activities can be detected from the input skeleton animation.

We examine various options for implementing the system using Amazon Web Services (AWS) as the cloud, analyze their feasibility and costs, and choose the best solution. Additionally, we conduct various experiments. The results of this thesis can help developers and researchers decide whether to deploy their algorithms on the edge, or the cloud. The findings may also be beneficial for other areas of artificial intelligence applications.

Keywords: joint edge-cloud computing; artificial intelligence; cross-platform; human action detection; industrial integration; privacy preserved monitoring

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Jie Liang, for unwavering guidance, patience, and supervision throughout the entire duration of this project. He provided invaluable suggestions and ideas for improving my thesis development, which overcame the difficulties in many research challenges.

I would also like to extend my appreciation to Dr. Shahram Payandeh, the committee chair, Dr. Zhenman Fang, the committee member, and Dr. Faisal Beg, the examiner for the time and effort invested in thoroughly reviewing and evaluating my thesis.

Additionally, I am grateful to my coworkers, Andrew Au, Dr. Bruce Zheng, and Michael Ng, for their valuable input and ideas related to the task, drawing from their professional expertise. Additionally, they have generously shared their experiences from their own MASc and Ph.D. studies, providing me with invaluable insights and guidance for my study, which have significantly influenced my professional and academic growth.

I would like to express my heartfelt appreciation for the understanding and support extended to me by my colleagues throughout the completion of my thesis. Their support has been crucial in enabling me to successfully balance my work and study. The valuable work experience I gained has significantly enriched my knowledge of industrial application and contributed a large portion of my thesis.

I would also like to express my sincere gratitude to AWS (Amazon Web Services) for providing me with their service credits which allow me to utilize cloud computing resources and services throughout my academic pursuits.

I would like to express my deepest gratitude to all those researchers and scholars whose work has laid the foundation for this thesis. Without their invaluable contribution, this thesis would not have been possible.

Finally, I feel truly blessed to have had such loving and supportive family members and parents by my side throughout my academic journey. I would like to express a special acknowledgment to my late grandfather, whose influence has been instrumental in instilling in me the importance of senior care. I am especially grateful for the immense care and support provided by my grandparents during my academic studies in Canada.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Recognizing human actions is a critical task in various areas, such as medical monitoring, video surveillance, and human-robot interaction. Traditional surveillance methods rely on human monitoring, which is limited by a person's ability to monitor multiple cameras simultaneously and over long. To address this issue and enhance the efficiency and accuracy of surveillance systems, researchers have turned their attention to the development of automatic human activity recognition (HAR) methods. They aim to automate the process of identifying and categorizing human actions computationally and eliminating the need for constant human intervention. By leveraging techniques such as machine learning and computer vision, computers can analyze video data, detect specific actions, and provide valuable insights to improve monitoring, decision-making, and response mechanisms in various domains. These automation methods have the potential to revolutionize the way we monitor and understand human behavior in dynamic environments, enabling more effective medical monitoring, enhanced video surveillance, and seamless human-robot interaction.

HAR has long been a challenging task in the field of computer vision, particularly prior to the 21st century. It was in 1950 that researchers first began exploring the concept of artificial intelligence computationally. [59] Over the years, advancements in artificial intelligence have significantly improved the accuracy of HAR systems. By enabling computers to understand and interpret human actions, these systems have the potential to autonomously respond to specific events without the need for human reviewers. Moreover, they can analyze an individual's behavior and health patterns, providing valuable insights and aiding in decision-making processes. In recent years, deep learning methods found their primary success and were predominantly utilized in computer vision applications.

The task of HAR poses challenges to computer vision due to the complex graphical and temporal nature of the computations involved. Fortunately, recent advancements in deep learning have contributed to overcoming these challenges. Deep learning methods have

demonstrated higher accuracy and faster processing speeds compared to early approaches. State-of-the-art models have also shown robustness for commercial use and adaptability to embedded devices, making them practical for real-world applications.

In this thesis, we are interested in applications in senior care, including aging at home, independent living, assisted living, and long-term care facilities. By leveraging HAR technology, the workload of facility staff, such as nurses, security guards, and family caretakers, can be reduced significantly. Additionally, HAR systems can provide timely alerts for emergency events, enabling proactive responses.

As more HAR methods are developed and deployed, concerns regarding privacy have garnered increased attentions from the users and the research community. To protect the privacy of the people being monitored, it is desired to process the data locally using some embedded devices, and do not send the original images or videos out of the devices. However, running HAR algorithms on embedded devices faces limitations imposed by the hardware. These limitations restrict the flexibility in selecting and deploying sophisticated HAR models. Even when more complex and advanced models are available, their utilization may be hindered by the computational power required. Consequently, relying solely on embedded devices is not a sustainable long-term solution. In the realm of deep learning products, customers place significant emphasis on the accuracy and effectiveness of the results generated. If the performance is below expectation, it can lead to customer dissatisfaction and ultimately tarnish the company's reputation in the long run.

Our goal is to research novel algorithms and methods that can recognize more activities of daily living (ADL) with clinical relevance, such as eating, drinking, taking medication, cooking, and dressing. This allows us to better evaluate seniors' physical and cognitive status, determine their level of health, and recommend activities that may prevent, mitigate, or reverse some health issues. In the long term, we also plan to develop algorithms to detect some behavioral diseases earlier, such as Parkinson's disease, dementia, and depression. However, it is difficult to perform these tasks in the embedded devices.

One promising approach to address the limitations of embedded devices is to harness the power of cloud computing. By utilizing cloud resources, a long-term advanced solutions can be achieved, enhancing the performance of human action recognition. One of the major concept is how to develop a high-performance joint edge-cloud computing while respecting the privacy of the users and data safety. And, this is achieved by employing the skeleton-based HAR [22][64][51].

With the skeleton-based approach, the sensor will first apply pose estimation algorithms to accurately determine the locations of keypoints on each person, thus forming the basic skeleton of the individual. This skeletal data is then used as input to the skeleton-based

HAR model for further prediction. By implementing this method, the processing burden on embedded devices is significantly reduced, as they only need to transmit the extracted keypoint data to the cloud for analysis and recognition, rather than transmitting entire video streams.

This joint edge-cloud computing approach offers several advantages. Firstly, it enables real-time processing and inference on the edge, enhancing response times for critical applications. Additionally, it ensures data privacy by reducing the need to transmit raw video feeds, as only the keypoint data, which contains essential information for recognition, is transmitted. This way, the users' privacy is respected, and the risk of sensitive information exposure is minimized.

Moreover, the cloud aspect of this solution allows for the utilization of more powerful hardware and extensive computing resources. This, in turn, facilitates the implementation of complex and advanced machine learning algorithms for human action recognition. As cloud computing resources are scalable, the system can easily accommodate increasing demands as the number of connected embedded devices and users grows.

By striking a balance between edge and cloud computing, this high-performance joint approach paves the way for a more efficient and privacy-conscious human action recognition system, catering to the needs of diverse applications across various domains. As the technology continues to evolve, it holds great potential in overcoming the limitations of embedded devices and unlocking new possibilities for the future of human-centric smart systems.

1.2 Contribution

This thesis presents a high-performance privacy-reserved action recognition system, leveraging cloud computing assistance. The study comprehensively outlines the system's pipeline, optimizing human action recognition's performance and continuous ability for low-cost embedded video capture devices. By successfully overcoming several limitations, such as secured data transfer, data storage, and deep learning algorithm computation constraints, the proposed system yields promising results.

Moreover, the system's modular architecture enhances flexibility, allowing users to effortlessly select the most suitable and advanced deep learning model tailored to their specific requirements. In conjunction with the privacy-reserved action recognition system, the architecture efficiently combines low-end embedded devices and cloud computing, and ensures security, cost-effectiveness, and scalability, facilitating seamless integration.

Additionally, the thesis introduces three key components to complement the system. Firstly, the Data Recording Component (DRC) optimizes data usage, ensuring secure data

storage and transfer through the use of a skeleton recording sensor and a minimized data format for HAR. Furthermore, various data compression methods are explored, effectively reducing HAR data transfer by 2.5 times.

Secondly, the Action Recognition Component (ARC) is defined for cloud processing. It examines various HAR modalities, including signal-based and vision-based methods. Focusing on computer vision, RGB image-based and 2D skeleton-based methods are studied, with the latter offering greater privacy protection. The thesis successfully demonstrates the possibility and feasibility of conducting HAR in the cloud using the skeleton data provided by the DRC.

Thirdly, the thesis introduces the Application Programming Interface (API) of the system, enabling third parties to query analysis results from the ARC. The integration of Amazon's Alexa Together emergency service in the cloud pipeline showcases the system's flexibility for third-party integration, particularly in real-world commercial use cases such as fall detection. A part of the work in this thesis is published in [18].

1.3 Thesis Organization

The thesis is structured into 7 chapters.

In Chapter 1, the problems that this thesis aims to solve are introduced, followed by a brief summary of the contributions made in this work.

In Chapter 2 reviews machine learning in artificial intelligence, with a focus on understanding the required components for the high-level architecture.

In Chapter 3 reviews different cloud computing service models, highlighting the benefits of using cloud computing and designing the architecture based on the six important pillars of cloud computing.

In Chapter 4 defines the DRC of the system, powered by pose estimation methods.

In Chapter 5, the ARC of the system is defined, powered by skeleton-based action recognition, with a summary of the problems encountered during the experiments and the possible solutions.

In Chapter 6 presents an actual use case of the whole system that can be applied in commercial settings.

Finally, in Chapter 7, a summary of all the work conducted in this study is presented, with a discussion on the future improvements of the design.

Chapter 2

Study of Human Action Recognition

2.1 Introduction of Artificial Intelligence

Artificial Intelligence (AI) refers to the field of study and development of machines, computers, and robots that possess human-like abilities to reason and solve problems. Within the realm of AI, machines are capable of performing tasks that typically require human intelligence, including visual perception, natural language processing, decision-making, question answering, and more.

Machine Learning, a subfield of artificial intelligence in computer science, aims to replicate how humans learn from experience and information. It involves using algorithms and training data to enable computers to learn and make predictions. There are three fundamental paradigms of machine learning: supervised learning, unsupervised learning, and reinforcement learning. [21]

1. Supervised Learning: In this type, the training data is labeled, meaning it contains the desired categories or answers that the model needs to estimate. The labels can be discrete (categories) or continuous (values), and the model learns to classify or regress based on this labeled data.

2. Unsupervised Learning: In this type, the model is trained using unlabeled data, where the categories or groups are unknown during the training process. One example is clustering, where the model is given a set of unlabeled data and needs to identify patterns or groupings based on its own analysis of the data.

3. Reinforcement Learning: This type involves decision-making processes, often seen in AI playing games. The model learns through interactions with an environment, receiving feedback in the form of rewards or penalties based on its actions. It seeks to optimize its decision-making strategy to maximize the cumulative reward.

These three types of machine learning provide different approaches to solving problems and learning from data, allowing computers to perform tasks and make predictions based on patterns and information. In this thesis, we are more focus on the supervised Learning model, as that model is sufficient enough for the current system on the classification of action recognition. And, number of action recognition datasets are available, like NTU RGB+D[42], UCF101[52], ETRI-Activity3D[34] and HMDB51[40].

2.2 Introduction of Deep Learning for Classification

Deep learning, a subfield of machine learning, has become especially powerful in classification tasks due to the use of neural networks, particularly deep neural networks or deep learning models. Deep neural networks consist of multiple layers of interconnected artificial neurons, allowing them to automatically extract complex and hierarchical features from the input data, making them well-suited for a wide range of classification problems, like image and speech recognition, as well as natural language processing. The process of training deep neural networks involves fine-tuning their internal parameters—weights and biases—to minimize the difference between their predictions and actual data. [27]

To train a model, it typically involves three essential stages: training, validation, and testing. During the training phase, the model learns from the provided data, adjusting its internal parameters to minimize errors. Validation is a crucial step aimed at assessing the model's ability to generalize to unseen data and to prevent overfitting, which occurs when the model becomes too specialized to the training data. Finally, testing evaluates the model's accuracy and performance using entirely new and unseen data, providing a clear measure of its predictive capabilities in real-world scenarios. The accuracy serves as a key metric, measuring the proportion of correctly classified instances. [27]

For the training, backpropagation is an algorithm commonly used. It enables the iterative adjustment of a network's parameters by propagating errors backward through the layers, updating weights and biases to minimize the difference between predicted and actual outputs. Through this process, neural networks learn to make better predictions and are widely applied in various machine learning and deep learning tasks, playing a central role in the advancement of artificial intelligence.

2.2.1 Convolutional Neural Network

In the last decade, developments in deep learning have revolutionized many research fields and industries. In computer vision, a series of convolutional neural network (CNN)-based deep learning algorithms have been developed, which outperformed human in just a few years. Convolutional neural network is one type of neural network that is usually applied for image classification. For example, AlexNet, an architecture in the realm of Convolutional

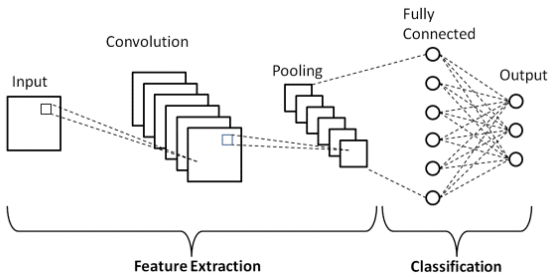


Figure 2.1: Convolutional Neural Network [30]

Neural Networks (CNNs), is renowned for its influential role in computer vision tasks. In the ImageNet Large Scale Visual Recognition Challenge in 2012, the model established a new benchmark for image classification tasks. It comprises the fundamental eight layers commonly utilized in CNNs: the Convolution layer, which applies filters to detect features; local response normalization for feature scaling; max pooling and fully connected layers with ReLU activation for feature extraction; fully connected layers without activation for classification; and the dropout layer to mitigate overfitting. In particular, max pooling and average pooling layers are instrumental in down-sampling the data, reducing spatial dimensions and capturing salient features, thereby making AlexNet a pivotal model in image classification and object recognition tasks. Commonly, CNN network contains the layers as shown in Figure 2.1.

Batch normalization or normalization layers are often used in CNN to normalize the inputs to a layer that has a mean close to zero and a standard deviation close to one. stabilizing training, faster and better when centered around zero and have similar variances, and faster convergence during training.

Next, we briefly introduce some representative image classification models using 2D CNN, including Inception[55], MobileNet[31], ShuffleNet[66], EfficientNet[57] and ResNet[29].

Inception

Inception network contains "inception modules," which employ multiple convolutional filters of varying sizes within the same layer to capture features at different scales. This approach enhances the model's ability to recognize complex patterns in images, making it highly effective for tasks like image classification and object detection. Inception has been influential in the development of modern convolutional neural networks.

ResNet

Residual Network (ResNet) [29] contains skip connections to bypass one or more layers in the networks and allows the network to learn residual information. These residual con-

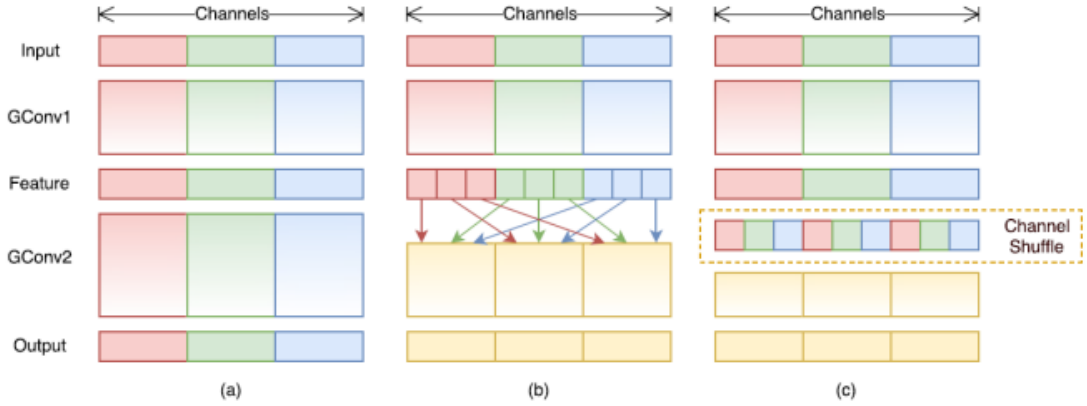


Figure 2.2: (a) Two Stacked Group Convolutions (b) Shuffle the channels, (c) Channel Shuffle Operation of (b) [66]

nections enhance training efficiency and permit the construction of deeper networks with minimal performance degradation compared to conventional CNN architectures. ResNet find widespread use in tasks such as image classification, object detection, and computer vision applications. In reference to [10], there is an adaptation of 2D ResNet into a 3D CNN for the purpose of HAR. This adaptation has resulted in a significant boost in accuracy, surpassing 90%, when evaluated on the Kinetics 400 dataset.

MobileNet

MobileNet [31] is designed for efficient and lightweight deep learning on mobile devices and embedded systems. It achieves this efficiency through depthwise separable convolutions, making it ideal for applications with limited computational resources. MobileNet is widely used for tasks like image classification, object detection, and semantic segmentation in mobile apps, robotics, and IoT devices.

ShuffleNet

ShuffleNet uses channel shuffle operations, as shown in Figure 2.2, to reduce computation, making it suitable for resource-constrained devices. ShuffleNet excels in applications like image classification and object detection, offering a balance between performance and computational efficiency.

EfficientNet

EfficientNet achieves a harmonious blend of model performance and computational efficiency by employing a compound scaling technique that uniformly adjusts the network's depth, width, and image resolution. This balanced scaling approach makes it particularly

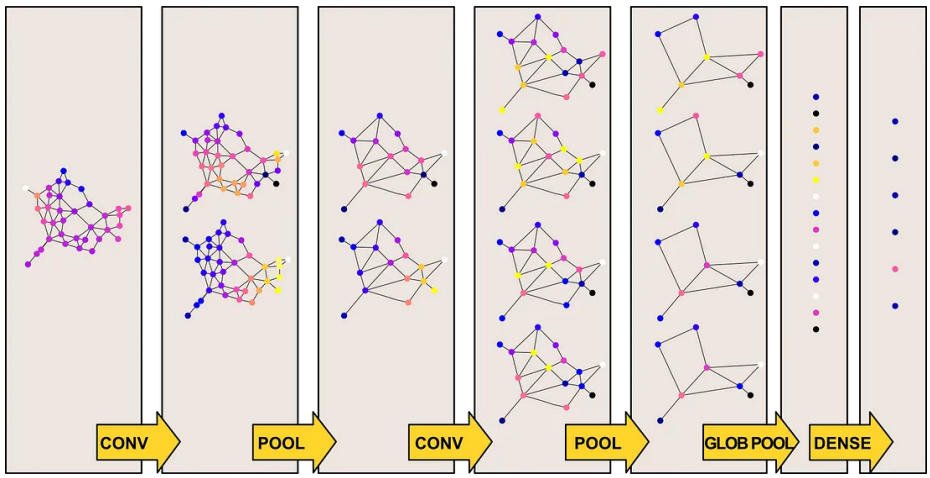


Figure 2.3: Graph Convolutional Network[16]

suitable for running efficiently on devices with limited resources, such as mobile phones and IoT devices.

2.2.2 Graph Convolutional Network

The graph convolutional network (GCN) was first introduced by Thomas Kipf and Max Welling in 2017 [38]. it is a type of neural network architecture designed for analyzing and processing data that can be represented as a graph or a network. Graphs are mathematical structures that consist of nodes and edges, where nodes represent entities and edges represent relationships or connections between these entities.

"For a specific graph-based neural network model $f(X, A)$, the formal expression of the GCNs layer is $H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$, where $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph G with added self-connections. I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function. $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the l^{th} layer; $H^{(0)} = X$." [38]

GCNs have found application in numerous domains, from the analysis of social networks, where individuals and their connections form the nodes and edges of a graph, to the field of biology, where they have been used for predicting protein-protein interactions based on complex molecular networks. In natural language processing, GCNs are instrumental in tasks such as semantic parsing and knowledge graph reasoning, allowing the modeling of intricate relationships between words, entities, or concepts. Their adaptability and effectiveness in diverse areas make GCNs an invaluable tool in the realm of artificial intelligence.

Furthermore, in the realm of computer vision and human action recognition, GCNs have gained prominence. GCNs, when applied to the analysis of human skeleton key points and

their corresponding edges, offer a compelling solution for recognizing complex human actions. By considering the spatial relationships and interactions between key points, GCNs can enhance the accuracy and robustness of human action recognition systems. We'll delve deeper into the application of GCNs in this context, highlighting the potential for more precise and effective recognition of human activities. This chapter thus provides an introduction to the power and versatility of GCNs in modeling and learning from data with intricate connectivity patterns while emphasizing their significance in human action recognition.

2.3 Introduction of Human Action Recognition

Human action recognition is a field of artificial intelligence that focuses on understanding and interpreting human movements and gestures. It involves the development of algorithms and systems capable of automatically identifying and categorizing actions performed by individuals from different data modalities, The goal is to enable machines to comprehend and interpret human behavior. It is usually categorized as supervised learning in classification, as it involves training models to classify and recognize various human actions based on the provided labeled datasets.

2.3.1 Data Modalities

With the aid of computer resources, we are not limited to using only vision, but can also use other data modalities, such as ECG data [45], depth [9], infrared sequence [44], audio [67], radio signal [41], etc, to predict an action. Multiple modalities can be used for human action recognition, such as RGB, skeleton, depth, infrared, point cloud, event stream, audio, acceleration, radar, and WiFi signal. [54] And, skeleton data is the better modality that can also protect the user's privacy by detecting and capturing only the skeleton data and streaming to the server, as opposite to the RGB method which would need to stream the video for cloud-based deep learning processing. Also, unlike other modalities, it does not expose the person or background image, environment information or audio sound. At the same time, it also allows the mobile application to visualize the playback of the skeleton action by using only the skeleton data and frame timestamps for the users reviewing the event. Comparing with the traditional video record data, it only needs to store the action prediction data, the person prediction data, the event prediction data, and skeleton data instead of the actual video itself. It saved the cost of the network bandwidth, storage, and processing speed. Also, it keeps our client data secured and private as there is no actual video being sent to or stored in the server. For example, 95 seconds of the recorded stick figure data, only has around 38.2 KB of data. That means it will take less than 2-megabyte data for a 1-hour record data. It is significantly less than a recorded actual video, which could take more than 700 megabytes of data.

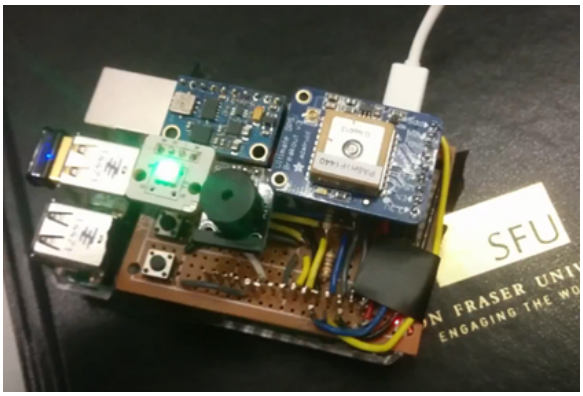


Figure 2.4: a fall action detector utilizing accelerometer and gyroscope signals

Physical Signal-Based Methods

In addition to skeleton-based recognition, another approach that can be incorporated into the system to respect user privacy is the physical signal-based method. Commercial products in this domain mainly revolve around wearable devices or radio signal recognition devices [63]. For instance, we have successfully developed a fall detector during undergraduate study, as depicted in Figure 2.4, by utilizing accelerometers and gyroscopes to calculate the angle and g-force associated with a person's fall. Furthermore, recent research [26] has introduced a reliable deep neural network model for recognizing typical patient behaviors encompassing six distinct human actions. It is worth noting that a limitation of using accelerometers and gyroscopes for human action detection is the requirement for users to wear the devices. Based on feedback from healthcare facilities, some patients may be hesitant to wear additional devices, which can pose challenges in certain scenarios. Nevertheless, a study has indicated that relying solely on data from low-dimensional sensors lacks the necessary depth for effective action recognition. It is recommended to incorporate cameras or a combination of cameras and low-dimensional sensors. [58]

The use of wearable devices, although effective in capturing physical signals for action recognition, may not be universally applicable due to user preferences and acceptance. It becomes crucial to strike a balance between privacy preservation and user comfort. Finding alternative methods that do not rely on wearables but still capture the necessary physical signals is an area of ongoing research. By exploring such solutions, the system can cater to a broader range of individuals while ensuring their privacy is respected.

Vision-Based Based Methods

Another commonly used approach for human action recognition is the vision-based method, which utilizes deep learning techniques. This method leverages visual data, typically obtained from cameras or video sources, to analyze and interpret human actions. [12]

[11] [18] However, when comparing vision-based devices for human monitoring, there are important considerations regarding mobility and monitoring coverage. Once the user leaves the area covered by the stationary device, their actions cannot be monitored.

To address the limitations of stationary devices, physical signal based are sometime considered as it can capture physical signals without being restricted to a fixed position. These devices offer mobility and can track the user's actions regardless of their location.

2.4 Deep Learning Models in Computer Vision

The advent of deep learning has revolutionized the field of action recognition, enabling researchers to achieve impressive accuracy levels. This section provides a comprehensive overview of the modalities used in action recognition, focusing specifically on RGB images and skeleton key points input.

RGB-based deep learning models have demonstrated exceptional performance in action recognition tasks. Convolutional Neural Networks (CNNs), including 3D CNNs, 2D CNNs, and spatiotemporal networks such as C3D and I3D, are commonly employed in this context. These models learn to extract relevant features from video frames and capture temporal patterns to identify actions. However, using RGB images has certain limitations. These images may contain extraneous or repetitive information, and their recognition performance can be influenced by factors like lighting conditions and occlusion.

In contrast, skeleton-based methods rely solely on the key points of a human skeleton for action recognition. By focusing on skeletal information, these methods are less susceptible to irrelevant or redundant data and can effectively capture the temporal dynamics of actions. Prominent skeleton-based approaches include Graph Convolutional Networks (GCNs), Recurrent Neural Networks (RNNs), and Graph Attention Networks (GATs). Recent studies have demonstrated that skeleton-based methods can achieve comparable or even superior performance to RGB-based methods, particularly in scenarios involving occlusion or low-quality RGB images. Moreover, skeleton-based methods offer the potential for enhanced privacy preservation, as they eliminate the need to transmit actual video data. However, a critical challenge in employing skeleton-based methods lies in the reliable and accurate extraction of skeletons from RGB images, which can be demanding in certain situations.

Deep learning has significantly advanced the accuracy of action recognition, with both RGB-based and skeleton-based modalities demonstrating their unique strengths and considerations. While RGB-based methods excel in capturing visual details, skeleton-based approaches offer improved resistance to irrelevant information and potential privacy advantages. As the field continues to evolve, further research and development are needed to

refine the extraction of skeletons from RGB images and enhance the overall performance and applicability of action recognition systems.

2.4.1 RGB Image-Based Models

After the completion of AlexNet in the ImageNet Large Scale Visual Recognition Challenge in 2012, convolutional neural networks (CNNs) gained widespread recognition as a highly effective technique for image classification. With a top-5 error rate of 15.3%, significantly lower than the second-place error rate of 26.2%, CNNs became well-known for their performance. [39] According to [15], the advantages of image classification can also be applied to video classification by transforming the network from a 2DConvNet to a 3DConvNet. However, while a very deep 2D CNN may achieve high accuracy in classification, it may not be suitable for embedded systems. This is due to the large model size and memory usage requirements. Therefore, in the context of embedded systems, it is preferable to have a model with fewer parameters that still maintains high accuracy. This approach reduces memory requirements, power consumption, and training time.

Given the memory constraint, it is more appropriate to use a model with fewer parameters in an embedded device. Table 1 illustrates several models, namely ShuffleNet-v2, MobileNetv2, Inception-v2, and EfficientNet-B1, which offer high accuracy while having a lower number of parameters. Additionally, EfficientNet-B7 stands out as a high-performance network when compared to other models with a similar number of parameters. Consequently, my focus will primarily be on reviewing these four model types and summarizing their main concepts.

After AlexNet completed on ImageNet Large Scale Visual Recognition Challenge in 2012, the convolutional neural network (CNN) becomes more famous and known as a high-performance technique for image classification. In imaged based method, 2DConvNet and 3DConvNet. classification can also apply to video classification after converting the network from 2DConvNet to 3DConvNet. In the research, some higher accuracy models trend to have deeper 2D CNN model, which means a larger model in terms of file size and memory usage. Hence, they are not ideal for a low constraint embedded system. ShuffleNet-v2 [66], MobileNetv2 [49], Inception-v2 [56], and EfficientNet-B1 [57] come with high accuracy in image classification and a lower number of parameters.

Video classification applications are commonly utilized in security, home automation, and robot interaction, primarily operating on small, low-power embedded devices with limited computing capabilities. Therefore, there is a strong demand for optimized, accurate, and diverse algorithms. Over the past decade, deep learning methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have achieved significant success in human action recognition. Several architectures have been developed, including

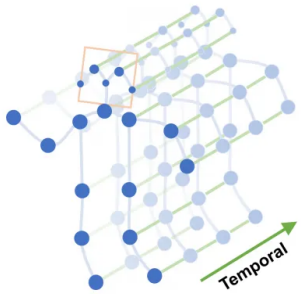


Figure 2.5: The spatial-temporal graph of a skeleton sequence.Blue dots denote the body joints The inter-frame edges connect the same joints between consecutive frames. [64]

2DConvNet + LSTM, 3DConvNet, Two Stream, 3D-Fused Two-Stream, and Two-Stream 3D-ConvNet, as depicted in Figure 8 [1]. Deep learning has demonstrated notable success in image classification, achieving over 75% accuracy on UCF101 with small models. However, video classification poses additional challenges due to the temporal domain, which increases the complexity of training and results analysis. Simply examining a single frame or image is insufficient to predict the entire action. For instance, a photograph showing a person holding a door allows one to infer that the person is moving the door. However, more information is required to determine if the person intends to open or close the door, as future knowledge of the action is necessary.

One approach to reducing the computational burden on low-constraint devices is to offload the action recognition process to a more powerful device or cloud services. For instance, a centralized machine learning device with enhanced processing capabilities could be employed to handle action recognition tasks by connecting to all other sensors. However, this solution would necessitate additional physical space and incur extra expenses for the user. Furthermore, it introduces maintenance challenges, as it adds another hardware component to be managed by the company.

In contrast, using cloud services for processing, especially with image-based methods, raises privacy concerns because it requires transmitting the actual images or videos from the edge devices to the cloud. This makes it an unsuitable choice for a commercial product. Consequently, the skeleton key points-based method emerges as a more viable and acceptable solution for this research.

2.4.2 2D Skeleton key points Based Reconigtion Models

Skeleton-based recognition is a privacy-preserving approach that ensures the confidentiality of data when consumed by third parties. In order to maintain user privacy, the system refrains from sending or displaying actual videos to data consumers. Instead, only

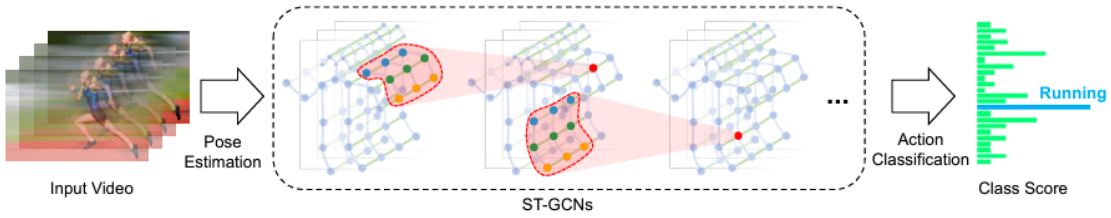


Figure 2.6: Process flow of ST-GCN [64]

skeleton data, which represents the essential information about human body movements, is transmitted to the cloud.

In this thesis, we conducted a comparative analysis of two deep-learning skeleton-based HAR approaches: ST-GCN[64] and PoseC3D[22]. ST-GCN is renowned as the pioneering graph convolution network model specifically designed for skeleton-based action recognition. Over time, the concept of graph convolution networks has been further developed, leading to the emergence of other notable GCN-based methods such as HD-GCN[65] and 2s-AGCN [51]. In contrast, the PoseC3D model, which gained prominence in 2021, is based on a ResNet architecture instead of GCN. It has been recognized as a state-of-the-art model for the NTU RGB+D dataset, demonstrating its effectiveness in skeleton-based action recognition tasks. However, in this research, our primary focus was on examining the fundamental GCN model and the ResNet model to present a comprehensive architectural analysis, as described in Section 3.3. By comparing these two approaches, valuable insights can be gained into the strengths and weaknesses of each model in the context of skeleton-based HAR.

ST-GCN [64]

Spatio-Temporal Graph Convolutional Network (ST-GCN), is a specialized deep learning architecture designed for processing spatio-temporal data, with a particular focus on human action recognition in videos. It is a powerful tool for extracting meaningful features from video sequences, making it valuable in applications such as video surveillance, gesture recognition, and sports analysis. ST-GCN builds upon the concept of GCNs and extends it to accommodate the temporal dimension. As illustrated in 2.5, the blue edge connections between nodes symbolize the interconnections between nodes across successive frames of skeleton data.

The input format for ST-GCN is represented as (1, 3, 300, 18, 2), where each dimension corresponds to (batch, channel, frame, joint, person). In this context, channel=0 represents the x-coordinate of the joint, which is normalized within the range [-0.5, 0.5]. Channel=1 corresponds to the y-coordinate, and channel=2 represents the confidence value. The "joint" refers to the index of the joint in the skeleton, and "person" indicates the person's index.

The output format is the confidence value for (batch, class, output_frame, joint, person). To detect an action, you add up the confidence values for each class and select the class with the highest value. The "output_frame" is determined by dividing the number of input frames by 4. The "voting_label" is computed by summing the values along multiple axes (axis=3, axis=2, axis=1), and then finding the index of the maximum value along axis=0.

As shown in Fig. 2.6, the input video would be first processed by the pose estimation module turned into skeleton node sequences and from a spatial-temporal graph, then the multiple layers of ST-GCNs will be applied and generate feature maps on the graph. Then, it would be classified by Softmax classifier to different action classes.

PoseC3D [22]

Although GCN approaches represent a significant advancement in the field of skeleton-based action recognition, they are having challenges related to robustness, interoperability, and scalability. In response to these challenges, the PoseC3D [22] approach was developed. PoseC3D offers a fresh perspective by employing a 3D heatmap volume as the fundamental representation of human skeletons, in contrast to the graph sequences used by GCN-based methods. This shift in representation yields several notable advantages. In the original paper, the accuracy of PoseC3D shows a promising result, as it achieved 87% accuracy on the UCF101 dataset and 69.3% on HMDB51 dataset.

First, PoseC3D excels in learning spatio-temporal features, making it highly effective in capturing the nuances of human actions. It also demonstrates increased robustness against pose estimation errors, a common challenge in skeleton-based action recognition. Moreover, PoseC3D exhibits strong generalization capabilities, particularly in cross-dataset settings, which is crucial for real-world applicability.

Secondly, PoseC3D is able to handle multiple-person scenarios without incurring additional computational costs, a limitation of some other methods. Additionally, the hierarchical features extracted by PoseC3D can be seamlessly integrated with other modalities at early fusion stages, offering a versatile and customizable approach to boosting performance.

2.5 Deep learning approach in recording device

2.5.1 Pose Estimation

The field of pose estimation plays a crucial role in utilizing the pose and orientation of individuals. It involves the integration of sensors, radio technologies, and computer vision techniques. In this work, the focus is on the computer vision method, particularly in the context of a camera device and its application in skeleton-based action recognition, as discussed in section 5.3.2. The pose estimation method employed here aims to estimate the

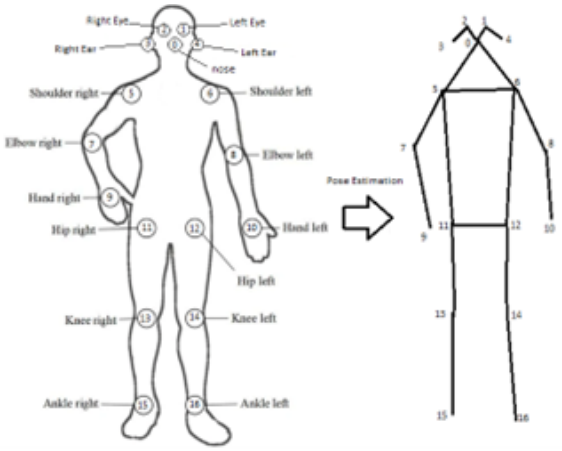


Figure 2.7: Human pose to skeleton stick Mapping in this study [7]

skeleton key points of various body parts, including the eyes, ears, nose, elbows, hands, hips, knees, and ankles, as illustrated in Figure 2.7. Once the pose estimation is accomplished, the application only requires connecting these skeleton key points to generate the visual representation of a skeleton stick figure.

Pose estimation using images typically adopts either a top-down approach, a bottom-up approach, or a combination of both strategies. The top-down approach involves initially detecting individuals in the input image using a person detector and then performing pose estimation within the bounding box of each detected person. However, when multiple people are present in the image, the number of person detectors needed to track each person's pose increases, leading to potential runtime complexity and performance challenges.

On the other hand, the bottom-up approach directly predicts all the keypoints of all individuals in the image and subsequently combines these related keypoints to form complete body poses. This approach offers the potential to reduce runtime complexity when dealing with an increasing number of people in the image. It is worth noting that single-person pose estimation serves as the foundation for multi-person pose estimation. Various methods and frameworks are employed for multi-person pose estimation using a top-down strategy. Notable approaches include OpenPose[14], MoveNet[13], and PoseNet[37]. Each offers distinct features and advantages within the domain of multi-person pose estimation.

When comparing the three methods, OpenPose stands out for its flexibility in detecting multiple persons, although its accuracy is comparatively lower. On the other hand, MoveNet excels in providing higher accuracy for estimating a person's skeleton. PoseNet, while suitable for single-person pose prediction, exhibits slightly lower accuracy but requires less computational power. [36]

In [14], it stated the challenges when dealing with pose estimation. First, each image may contain unknown number of people at any position and scale. Second, interactions between people induces a complex spatial interface Third, runtime complexity tends to grow with the number of people in the image, making real-time performance a challenge.

2.5.2 Object Detection

Object detection is a fundamental computer vision task that involves identifying and locating objects within an image or a video. It is a crucial step in HAR. The goal of object detection is to not only recognize the presence of objects but also provide their precise spatial coordinates in the image. This capability enables machines to understand the environments of the scene. Object detection has been achieved by deep learning with remarkable accuracy and efficiency in identifying across diverse scenarios. As the field of computer vision continues to progress, object detection plays a pivotal role in empowering machines to perceive and comprehend the world around us.

The recording component of the system utilizes the top-down approach, which requires the deployment of an object detector to identify humans in the scene. In this particular system, a Faster R-CNN model has been employed for object detection [46], in combination with the HRNet pose estimation for HAR [62]. The Faster R-CNN model is known for its ability to perform real-time object detection by utilizing region proposal networks.

Chapter 3

Cloud Architecture

3.1 Introduction of Cloud and Services

The concept of cloud computing involves a network of servers that work together to run software, process data, and store files remotely over the internet instead of on local devices. This centralized approach allows for real-time collaboration and communication between multiple users. A basic cloud network typically includes a software application server, a database, a load balancer, and network-attached storage, though other types of service servers may also be incorporated. There are five fundamental properties that define cloud computing: broad network access, on-demand self-service, resource pooling or shared services, rapid elasticity, and measured services. [48] As the number of servers in a cloud grows, it can become challenging to manage software and hardware, physical space, and installations. To reduce these costs, developers often prefer to operate their cloud infrastructure using third-party cloud service providers. Cloud service providers manage configuration, maintainability, and server hosting for the users, and allow users to activate additional services almost instantly. Several well-known cloud service providers are available, including Amazon AWS, Microsoft Azure, and Google Cloud. Amazon AWS has the largest market share, followed by Microsoft Azure. Among these providers, Microsoft Azure has the highest number of service regions, with 60 regions, while Google Cloud and AWS have 34 and 26 regions, respectively.

Several cloud service models have been introduced, including SaaS, PaaS, IaaS, FaaS, and MLaaS, all of which are utilized in this study. These service models can also be combined with other services.

Infrastructure as a Service (IaaS) allows users to specify the hardware specifications and software to be used on the system. [48] In Platform as a Service (PaaS), users can specify the software they want to use without having to configure the hardware specifications as the cloud provider takes care of it. [48]

Software as a Service (SaaS) allows users to only provide the data input and desired outcome for the service as the cloud provider takes care of the software and hardware processing. [48] Function as a Service (FaaS) enables users to implement their software logic without dealing with the infrastructure, hosting servers, or maintaining version frameworks, as it is done by the cloud provider. The service scales based on demand and is suitable for short-running, stateless computation, and event-driven applications. Aside from the public cloud provider service, if someone desires to own their FaaS service, they can investigate an open project, called Fn Project. [2] Machine Learning as a Service (MLaaS) is similar to PaaS but provides a flexible and scalable machine learning platform that processes data, trains models, validates and makes inferences, and can deploy the serverless service like FaaS [47] Both FaaS and MLaaS do not require users to maintain servers, allowing for fast on-demand deployment that reduces development time and saves the company the cost of unnecessary hardware resources.

3.2 Cloud Services

Based on the large development community and support discussed in 3.1, this work is mainly designed using AWS. Besides that, AWS also provided service credits for the research. This section concludes with a summary and noteworthy part of each service which are used for the overview of the private cloud and the design of the skeleton-based HAR system architecture in the next section. For more detail of AWS Web Service can be found in [32]. There are alternative services from other service providers, Azure, Google, and Alibaba. Or other GPU cloud-based providers like CoreWeave, Lambda, Vultr, etc.

With assist of cloud computing, we would need to handle the sensitive data like the recording file over the internet. Therefore, the proper cloud service, configuration and security measures are required to review on the implementation. Especially, some countries have strict regulations when dealing with privacy and health related data, such as Health Insurance Portability and Accountability Act (HIPAA) in United States. HIPPA is a federal law that requires the creation of a national standards to protect sensitive patient health information from being disclosed without the patient's consent or knowledge. Therefore, a privacy HAR architecture should ensure the data is safely transferred and stored in the cloud with correct access control right.

In this study, various cloud services were explored and evaluated for their suitability in the proposed system architecture. The following sections provide a summary of the different cloud services considered and the alternatives available:

3.2.1 Cloud IoT Core

To ensure secure communication between sensors and the cloud, AWS IoT Core was chosen as the primary service. It offers the AWS IoT Credential Provider, which generates X.509 client certificates for each sensor. These certificates, managed through AWS Identity and Access Management (IAM), enable encryption and decryption of message data, ensuring mutual authentication. Alternative services like Azure IoT Hub, Google IoT Core, and Alibaba IoT Platform also offer similar functionalities.

3.2.2 Cloud Identity and Access Management

AWS IAM was utilized to manage identities and access control for users and devices. It enables authorization for the use of AWS Video Stream and AWS S3 services by the sensors. Alternative services include Azure Active Directory, Google Cloud IAM, and Alibaba Resource Access Management.

3.2.3 Cloud Storage

In our implementation, we have opted for AWS S3 as the cloud storage solution for our data. AWS S3 offers several advantages, including fast and reliable access to data, high durability, and availability. Moreover, S3 excels in ensuring the security, confidentiality, and integrity of stored data, making it a suitable choice for our needs.

	SSE-S3	SSE-KMS	DSSE-KMS	SSE-C
Manage Key Polices	N/A	N/A	N/A	Available
AWS CloudTrail logs	N/A	Available	Available	Available
Key Rotation	By S3	By S3	By S3	By Customer
Data Shareability	Available	N/A	N/A	Available

Table 3.1: Comparsion between 4 server-side encryption choices on S3 [4]

In Amazon S3, we have four optional choices for data encryption: server-side encryption with Amazon S3 managed keys (SSE-S3), server-side encryption with AWS Key Management Service (SSE-KMS), dual-layer server-side encryption with AWS KMS keys (DSSE-KMS), and server-side encryption with customer-provided keys (SSE-C). By default, SSE-S3 is enabled for encryption. In Table 3.1, you can see that managed key policies, AWS CloudTrail logs, key rotation, and data shareability are the primary differences among these four options.

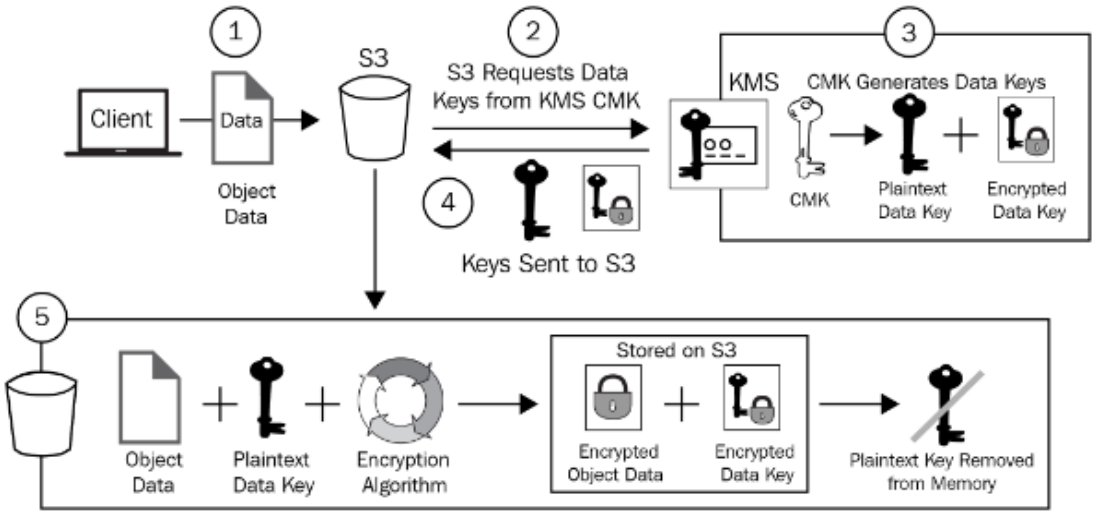


Figure 3.1: Five-step encryption process when using SSE-KMS[50]

The key distinction between SSE-KMS and DSSE-KMS is that DSSE-KMS offers an additional layer of encryption, enhancing security to meet compliance requirements. In our system, we have chosen to use SSE-KMS because we require AWS CloudTrail logs to monitor encryption access. Additionally, our data contains only skeletal information which is less sensitive data, so we do not need the extra encryption layer or managed key policies.

We also enable the bucket key as part of the encryption process to enhance its efficiency. The bucket key, generated by the AWS KMS Key and having a short lifespan, is stored within the S3 bucket. This approach ensures quicker key access, thereby expediting encryption requests and reducing the need for frequent key requests from AWS KMS. By doing so, it minimizes AWS KMS service usage by up to 99% [17]. The bucket key is then employed to generate the plaintext data key, as shown in Step 5 in Figure 3.1, for the transformation of plaintext objects into ciphertext objects. Once the encryption process is complete, the plaintext data key is promptly disposed of, and the bucket key is added as metadata to the ciphertext object.

In order to ensure that the bucket complies with regulatory requirements such as HIPAA, EU Data Protection, and FERAMP, AWS S3 offers a crucial feature known as AWS Macie. This tool enables periodic scanning of objects within the bucket to identify the presence of sensitive data, unencrypted information, or publicly accessible data. When any of these data types are detected, AWS Macie promptly notifies us, helping to maintain regulatory compliance and data security.

Integrity plays a crucial role in the realm of data storage. It is essential to guarantee that our data remains intact and uncorrupted during both storage and retrieval processes.

Amazon S3 offers support for a checksum feature, which allows applications to verify the data's integrity. This checksum is a unique string generated by a checksum algorithm, functioning much like a fingerprint for the file. Applications can use this checksum to compare objects that have been downloaded or uploaded, ensuring their identity and integrity by matching the checksum at both ends of the process.

In addition to the default MD5 checksum, Amazon S3 offers compatibility with four additional checksum algorithms: SHA-256, SHA-1, CRC32, and CRC32C. These algorithms vary in terms of speed and security. Speed generally increases from left to right in the list, with CRC32C being the fastest, but also less secure compared to the others. Depending on the specific requirements of the application, our system will opt for CRC32C when fast processing is the priority. However, if a higher level of security is mandated, SHA-256 will be the chosen checksum algorithm.

It's worth noting that alternative cloud storage services like Azure Blob Storage, Google Cloud Storage, and Alibaba Cloud Object Storage Service can also be considered for similar requirements. These services offer comparable features and can be evaluated based on factors such as performance, pricing, and specific integration needs.

By leveraging AWS S3's secure storage capabilities, including encryption and compliance configurations, we can confidently store and protect sensitive data in our privacy-reserved system.

3.2.4 Cloud Function

In the traditional method, the functions or applications are typically hosted on a server. However, this approach often imposes a significant workload for server maintenance and development. Ensuring the stability and scalability of the server to support the increasing number of requests can become a major challenge, especially for smaller teams with limited resources.

To address this issue, the concept of function-as-a-service (FaaS) was introduced. FaaS provides a serverless computing model where developers can focus solely on writing and deploying individual functions or microservices without the need to manage the underlying infrastructure. With FaaS, developers no longer have to worry about server maintenance, provisioning, or scalability. The cloud provider takes care of automatically scaling the infrastructure based on demand, relieving the burden from the development team.

By adopting a FaaS approach, teams can streamline their development processes, reduce operational overhead, and enhance scalability. This allows them to allocate more time and resources to developing the core functionality of their applications rather than managing infrastructure, ultimately improving efficiency and productivity.

In our implementation, we opted to use AWS Lambda as our preferred FaaS solution. We found that AWS Lambda offered a reliable and powerful backend service that fulfilled our requirements for stability, scalability, and high availability. Additionally, the educational resources and support provided by AWS were factors that influenced our decision.

For instance, Azure Functions from Microsoft Azure, Google Cloud Functions from Google Cloud, and Alibaba Function Compute from Alibaba Cloud are other viable options that can be considered. These services offer similar functionality to AWS Lambda, allowing developers to deploy and run individual functions or microservices without the need to manage the underlying infrastructure.

3.2.5 Cloud Machine Learning

Sagemaker, a Machine Learning-as-a-Service (MLaaS) solution, was used for hosting real-time and asynchronous inference for deep learning and machine learning models. It provides serverless inference capabilities. [47] Azure Machine Learning Studio, Google Datalab, and Alibaba Machine Learning Platform for AI are alternative services to consider.

3.2.6 Cloud API Gateway

In our research, we utilized AWS API Gateway, a powerful tool that enables us to host RESTful APIs for seamless two-way communication between various applications. This gateway acts as a primary entry point through which applications can interact with cloud-based data and features. It's a fully managed service offered by AWS, designed to streamline the development process and facilitate the scaling of APIs.

Some of the key features of the API Gateway include traffic management, Cross-Origin Resource Sharing (CORS) support, access control, authorization mechanisms, throttling, comprehensive monitoring, and version management. Similar services in other cloud platforms include Azure API Management, Google Cloud Endpoints, and Alibaba Cloud API Gateway.

By considering these various cloud services and their alternatives, the architecture of the proposed system was designed to leverage the strengths and features of each service, ensuring a robust and scalable infrastructure for the intended application.

3.3 Cloud and System Architecture

A well-architected system is composed of six pillars that define its design: reliability, operational excellence, security, performance efficiency, cost optimization, and sustainability. [23] Each of these pillars serves a specific purpose in ensuring the success and effectiveness of the architecture. A system operation on cloud computing should also encompass

fundamental scalability, availability, and maintainability. These achievements collectively contribute to the overall success and effectiveness of cloud computing. They enable systems to operate reliably, scale seamlessly, remain available to users, and be easily maintained to ensure optimal performance.

Reliability focuses on ensuring that the system operates continuously without unexpected behavior, even under varying workloads. It aims to provide a reliable and dependable system that users can rely on for consistent performance.

Operational excellence aims to achieve continuous improvement by aligning the system's operations with the overall business goals. By understanding the operations and optimizing workflows, operational excellence ensures that the system operates efficiently and effectively.

Security is crucial in protecting online assets, data, and systems. It involves implementing appropriate security measures to safeguard the system against unauthorized access, data breaches, and other potential threats.

Performance efficiency aims to provide the necessary flexibility in computing resources to support changes efficiently. It ensures that the system can adapt and scale according to technological advancements and evolving requirements, enabling optimal performance.

Cost optimization focuses on optimizing the cost of system operations to achieve the most cost-effective solution. It involves identifying cost-saving opportunities, optimizing resource allocation, and making informed decisions to minimize expenses without compromising performance.

Sustainability aims to achieve maximum outcomes by provisioning the minimum necessary resources. It emphasizes the efficient utilization of resources, reducing waste, and promoting environmentally conscious practices.

Scalability is the ability of the system to handle increased demand by expanding its resources, such as servers or storage, to accommodate the growing workload. This ensures that the system can effectively scale up or down based on the needs of the users or applications. Moreover, with an increasing number of sensors deployed in the field, the system must be capable of managing substantial data volumes and processing all the data seamlessly, without encountering any limitations.

Availability measures the amount of time the system remains operational and accessible to its users. It indicates the system's ability to provide uninterrupted service and minimize downtime. However, the geographical distance between the servers and the users can introduce network latency, which refers to the delay in data transfer due to the longer travel distance. This latency can impact the availability and responsiveness of the system, especially when users are located far away from the servers.

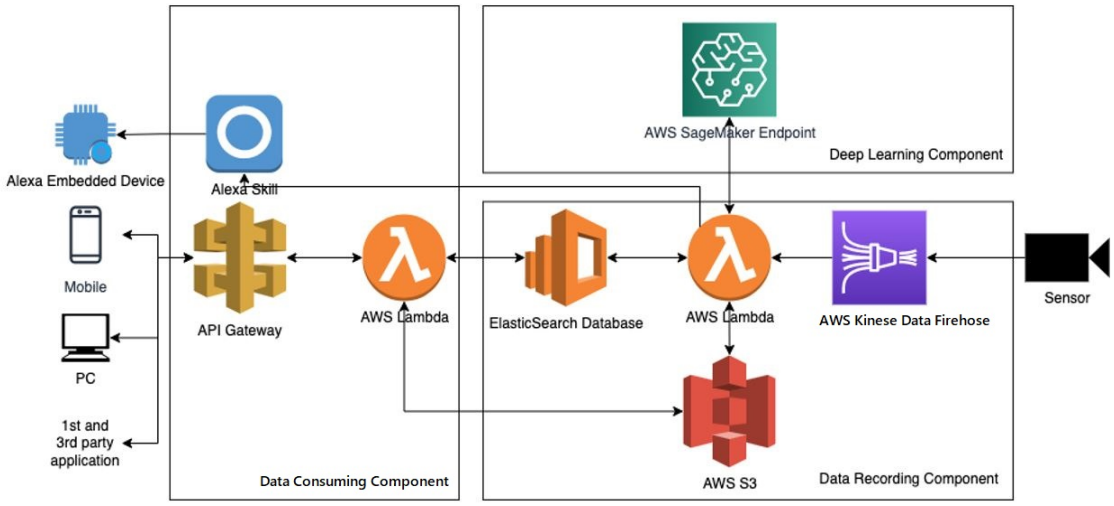


Figure 3.2: Overview of the proposed skeleton-based action recognition pipeline with cloud computing

Maintainability focuses on the ease of managing and repairing the system. It involves promptly identifying and addressing failed components, ensuring quick recovery and minimizing service disruptions. By implementing effective maintenance practices and having mechanisms in place to detect and resolve issues, the system can maintain its performance and reliability over time.

3.4 High Level of the Proposed Architecture

Our system's end-to-end architecture is depicted in Fig. 3.2 showcasing the various components that make up the entire pipeline. The majority of these components operate as event-driven applications, facilitating seamless and efficient data flow throughout the system. It's important to note that this architecture is not limited to the exclusive use of AWS. Instead, alternative services highlighted in each sub-section 3.2.1 to 3.2.6 or similar offerings from different providers can be employed as suitable replacements. The recording component serves as the initial and crucial element within the system. Its primary function is to manage the intake of data, which is gathered from sensors or cameras. To ensure secure transmission, the collected data is sent to the cloud through the utilization of an AWS Kinesis Data Firehose. Each device involved in the system is granted authorization through a client credential grant by the AWS IoT credentials provider, while the AWS Access Management (IAM) role effectively governs the resource usage scope. These measures ensure proper authentication and access control, safeguarding the system's integrity and protecting sensitive information.

The system can be divided into three distinct components, each playing a crucial role in its overall functionality: the deep learning component, the API component, and the data recording component.

The data recording component assumes the responsibility of securely consuming data from the camera source and storing it in an encrypted format. This ensures the confidentiality and integrity of the recorded data. By employing robust encryption techniques, the system safeguards against unauthorized access and potential data breaches.

The deep learning component takes on the task of performing Human Activity Recognition (HAR) using advanced deep learning algorithms. It leverages the recorded data to analyze and predict human activities accurately. Once the HAR process is complete, the deep learning component communicates with the data recording component, providing it with the predicted results. These results are then securely stored in the database, accompanied by a reference index that links them to the corresponding recorded data.

The API component plays a crucial role in the system by handling authentication and authorization processes for applications that consume the recording data. It ensures that only authorized and authenticated applications can access and retrieve the recorded data. By enforcing strict access control measures, the API component maintains the privacy and security of the recording data, allowing only authorized parties to interact with it. Together, these three components work in harmony to create a robust and efficient system that securely records, analyzes, and provides access to human activity data for various applications and purposes.

3.5 Comparing to Edge Computing

When relying solely on edge computing, the computational power of a device is constrained by its specifications at the time of production. There are two main challenges associated with this approach. Firstly, incorporating more advanced hardware during the initial development stage would increase the cost of the devices. Additionally, as newer and more capable hardware is released, older-generation devices may not have sufficient capabilities to run improved models designed for the new hardware. This would necessitate firmware updates to be pushed to the devices upon the release of new models, introducing complexities such as increased time and potential errors during the update process. Customers may find it burdensome to handle numerous firmware deployments, especially for devices installed in fixed positions that are difficult to reach. In case any issues arise during the update, it could result in time-consuming support for both the customers and the service providers. Thus, reducing the frequency of firmware deployments is generally preferred as it minimizes the need for extensive testing and customer support. However, this is not a sustainable long-term solution.

In contrast, cloud processing can address the aforementioned challenges. Firstly, it eliminates the need for high-end hardware configuration in the devices themselves. Instead, the devices can focus on performing simple tasks, such as pose estimation and small-scale fall detection models. This approach significantly reduces hardware costs and enhances adaptability for customers. By leveraging the power of the cloud, more advanced and large-scale models can be employed to process the results from the cameras, further reducing false alerts and improving overall accuracy.

However, using cloud processing requires ensuring secure communication between the devices and the cloud, which will be discussed in Chapter 4.

Chapter 4

Data Recording Component

In this chapter, we explore the Data Recording component in the cloud, which plays a crucial role in capturing, processing, and storing data from the sensor. The component can be divided into three essential parts: Data Capture, Data Transit, and Data Storage.

4.1 Data Capture

4.1.1 Data Format

Data transfer costs can become a considerable issue when utilizing cloud services, as larger data sizes lead to longer transfer times and higher expenses. To mitigate this concern, it is essential for the embedded devices to transmit only the essential data in a minimized data format. As a result, the devices exclusively send the keypoints data obtained in binary after applying the pose estimation algorithm discussed in Chapter 2, along with the time duration for each frame. This selective data transmission approach helps optimize the transfer process and reduces the overall data volume sent to the cloud.

To ensure flexibility and accommodate potential changes in the input data requirements for the HAR components, we also added a versioning header to the binary data. This approach allows for a versatile data structure without compromising backward compatibility when format changes are introduced.

Compared to conventional video recording data, our system efficiently transmits only the estimated skeleton keypoints data, omitting the need to send the actual video. This decision results in substantial savings in terms of network bandwidth, storage, and processing speed. Importantly, it also enhances user privacy, as no actual video content is sent to the server, ensuring confidentiality.

4.1.2 Data Compression

In cloud services, the efficient transfer of data is a significant concern due to the potential impact of larger data sizes on various aspects, such as transfer time, server storage costs, and overall expenses. To further address these challenges, we implement data compression techniques on the chosen data format before transmitting it to the server. In a comprehensive study conducted in 2017,[28] researchers extensively evaluated seven compression algorithms and provided insightful recommendations. They suggested that the appropriate algorithm choice should be based on specific requirements, including compression ratio, compression speed, and decompression speed. Among the recommended algorithms were Deflate[60], Bzip2 [5], LZMA [1], and PPMd[3]. Furthermore, another noteworthy lossless data compression algorithm called zstd[6] emerged in 2016 through the collaborative efforts of Yann Collet and Chip Turner at Facebook. This algorithm has gained considerable attention and is being actively tested and explored for its potential in optimizing data compression and transfer in various applications.

In our experiment, we conducted tests using four Python libraries specifically designed for compression: zlib, bz2, lzma, and zstd. The zlib library is a standard built-in library in Python that supports compression levels ranging from 1 to 9. Higher compression levels provide a slower but more efficient compression ratio [15]. Similarly, the bz2 library, also a standard Python built-in library, allows for compression levels from 1 to 9, with higher levels offering improved compression ratios at the cost of slower processing ratio [16]. The lzma library is another built-in library in Python specifically designed for compression purposes [17]. In addition to these built-in libraries, we utilized the zstd library, developed by Yann Collet and Chip Turner at Facebook, which exhibits a balance between compression ratio and speed. It offers multiple compression levels ranging from -100 to 22, with higher levels yielding higher compression ratios but slower processing. Our experiments were conducted using Python version 3.9.13 and a Ryzen Threadripper 2920x CPU. The results varied depending on the specific library employed. We measured the compression and decompression times before saving the file to disk, allowing us to compare the performance of the different libraries in our tests. By comparing the original data, it becomes evident that utilizing efficient data compression techniques can yield substantial savings in data transfer and storage costs, particularly in heavy usage systems.

We gathered recording data to assess various compression algorithms, showcasing the file sizes in Table 4.1. After evaluating the outcomes showing Table 4.2, we selected the LZMA algorithm for our compression needs. Since the skeleton record data is written into the system only once, the extended write time has minimal impact on the overall system. Moreover, as the data needs to be stored for over a year, we prioritize a higher compression ratio to optimize storage efficiency. In addition, we recognize that the user interface response

is crucial for a positive user experience. Therefore, a faster decompression time is essential, as it enables quicker responses to user requests, contributing to seamless and satisfactory user interaction.

Sample Name	Number of Files	Average Size (KB)	Total Size (GB)
Single Recording	1	36.372	0.0000364
Multiple Recording	112832	48.068	5.424

Table 4.1: Recording File Size

Compression Algorithm	Compressed Size (GB)	Compress Ratio	Average Compression Time (ms)	Average Decompression Time (ms)
LZMA	1.955	2.774	12934.519	987.723
PPMd	2.028	2.675	6096.790	1002.160
Bzip2 (level 9)	2.103	2.579	3092.195	1255.949
Deflate (level 9)	2.586	2.097	2713.763	165.302

Table 4.2: Compressed Recording File Size, Ratio, and Compression and Decompression Time

4.2 Data Transit

4.2.1 Data Streaming

In order to handle real-time, streaming data processing, we have opted to utilize AWS Kinesis Data Firehose, which is a cost-effective and fully managed service to process and analyze real-time streaming data at any scale. The data can be video, audio, application logs, website clickstreams, and IoT telemetry data, which is very suitable for our applications.

As shown in Figure 4.1, upon the arrival of recorded data at the Kinesis service, it undergoes processing through a lambda transformation function. This function's role is to convert the binary data from the sensor into a usable format before storing it in the database. Importantly, this process skips the initial database storage step, allowing the streaming data to accumulate within the Firehose buffer. Subsequently, the data stored in the buffer is processed at a designated interval, eliminating the requirement for immediate HTTP responses. This approach not only promotes scalability but also enhances data efficiency, enabling the function to efficiently consume data from the sensor.

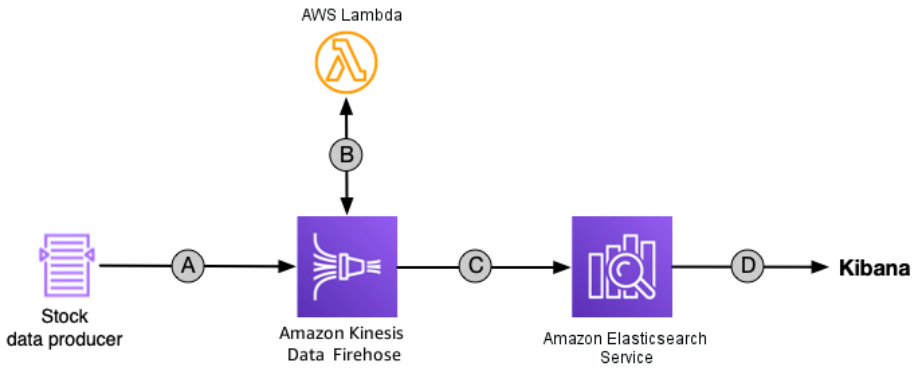


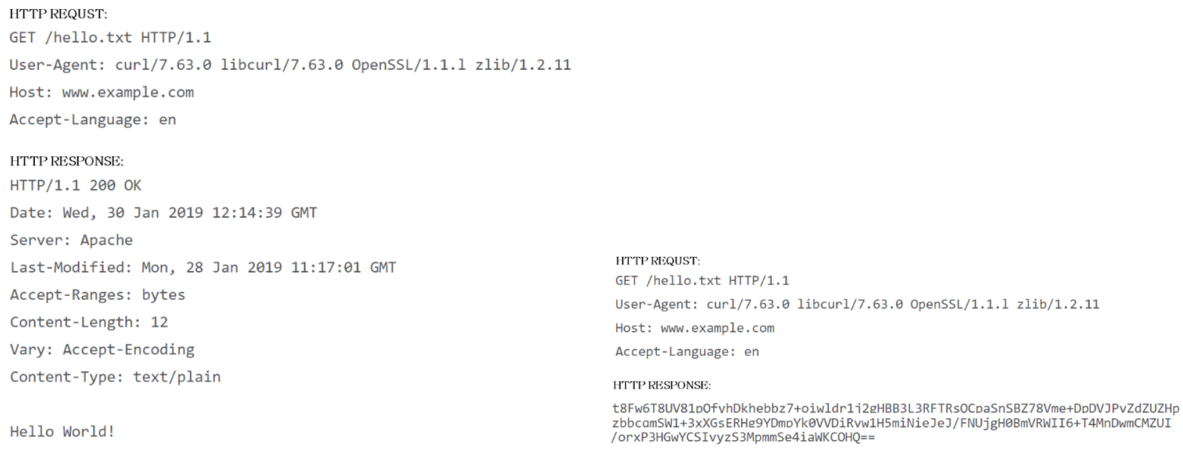
Figure 4.1: Data streaming process flow with Firehose [43]

AWS lambda is a powerful serverless, event-driven compute service that lets users run code for virtually any type of application or backend service without provisioning or managing servers. It has many advantages. For example, it uses the pay-only-for-what-is-used model, which charges zero base costs to host users' code, and only bills users based on when that code is executed, thereby minimizing the operating cost. Another important feature is that it scales automatically to accommodate sudden spike in usage, and users do not need to do anything.

4.2.2 Data Security in Transit

Ensuring data security is vital to safeguard against data breaches and malware infections during the transfer and storage of data from edge devices to the cloud. As highlighted in a cloud security report [33], critical security threats include insecure interfaces or APIs, unauthorized access, exfiltration of sensitive data, hijacking of accounts, cyberattacks, service traffic interception, and external sharing of data. To preserve user data privacy, it is imperative to establish appropriate configurations for data storage and transfer. This section will delve into the various data protection methods that have been deployed to minimize the risks and potential harm associated with data leakage.

Data breaches commonly occur due to information leaks during the transfer of data. Failing to encrypt data during transmission significantly increases the vulnerability to hacking and unauthorized access. To mitigate this risk, it is crucial to employ secure protocols such as Hypertext Transfer Protocol Secure (HTTPS) instead of the standard Hypertext Transfer Protocol (HTTP). HTTPS utilizes Transport Layer Security (TLS) to encrypt data while it is being transmitted over the internet. This encryption ensures that intercepted data remains unreadable to potential attackers. Figure 4.2 provides a visual comparison between the responses of HTTP and HTTPS. In addition to encryption, HTTPS also offers protection against various attacks, including on-path attacks, DNS hijacking, BGP hijacking, and domain spoofing [19]. Moreover, HTTPS enables the server to authenticate its identity to



a) b)

Figure 4.2: HTTP Request and Response: a) encrypted b) not encrypted

the client, establishing trust between the parties involved. By utilizing the secure HTTPS protocol, both parties can communicate securely and ensure that the transmitted data remains confidential and accessible only to authorized entities. In our solution, we chose to use TLS 1.3.

4.3 Data Storage

4.3.1 Data Index

We have implemented a comprehensive component to effectively manage and process the recording data in our system. To ensure efficient referencing and search capabilities, we have indexed the recording files based on timestamps, enabling easy retrieval and search within our system. However, our needs go beyond merely storing the recording data itself. To provide a more structured approach, we have separated the prediction data from the recording metadata and stored it in a database, associating it with the appropriate index. This approach allows for better organization and retrieval of data, facilitating future research and analysis.

In order to handle real-time, streaming data processing, we have opted to utilize AWS Kinesis Data Firehose, which is a cost-effective and fully managed service to process and analyze real-time streaming data at any scale. The data can be video, audio, application logs, website clickstreams, and IoT telemetry data, which is very suitable for our applications.

When the recording data arrives at the Kinesis service, it undergoes processing by a lambda transfer function, which decompresses and deserializes the binary data format, and

indexes the data into the database then stores the data. AWS lambda is a powerful serverless, event-driven compute service that lets users run code for virtually any type of application or backend service without provisioning or managing servers. It has many advantages. For example, it uses the pay-only-for-what-is-used model, which charges zero base costs to host users' code, and only bills users based on when that code is executed, thereby minimizing the operating cost. Another important feature is that it scales automatically to accommodate sudden spike in usage, and users do not need to do anything.

Subsequently, the original binary data is stored in AWS S3 (Simple Storage Service) storage, which can be accessed and consumed by the other components later on. This setup ensures stable and parallel processing of the recordings, enhancing the overall efficiency of our system, because AWS S3 is the most fundamental and global Infrastructure as a Service (IaaS) solution provided by AWS, which facilitates highly-scalable, secured and low-latency data storage from the cloud.

Given the escalating volume of recorded metadata over time, it was imperative for us to select a database engine capable of handling substantial data quantities and enabling rapid data analysis, fitting into the realm of "Big Data" sciences. To serve this purpose, we have opted for Elasticsearch as our database engine. Elasticsearch is a full-text search engine based NoSQL database. It provides a scalable search solution, supports multi-tenancy, and offers real-time searching capabilities. It gathers unstructured data from various sources, organizes it into searchable indexes, and stores it using user-specified mapping, which can also be automatically derived from the data. Thanks to its distributed architecture, we can efficiently and swiftly search, analyze, and visualize vast amounts of data, as it employs a distributed search and analytics engine. This choice equips us to process and analyze extensive data more efficiently and rapidly compared to traditional options like MySQL or MongoDB.

4.3.2 Data Security in Storage

Ensuring the security of the stored data is of utmost importance. In this regard, we have adopted a multi-faceted approach. Firstly, we have implemented minimum access controls for the resources, dictating who or which service components can access the stored files and the level of security applied. Additionally, we have employed encryption techniques to safeguard user privacy. By encrypting the data, unauthorized access to the storage files is rendered ineffective without proper decryption. To accomplish this, we have leveraged AWS Identity and Access Management (IAM) to establish minimum access rights for AWS S3 storage. As mentioned in Section 3.2.3, we have enabled SSE-KMS. SSE-KMS utilizes industry-standard AES-256 encryption, which has been recognized for its robust security compared to other algorithms such as DES or 3DES. AES is not only highly secure but

also efficient, with widespread support in both hardware and software environments, as highlighted in recent research [61] [53].

Encryption involves the transformation of information or data to safeguard it from unauthorized access. In today's digital age, it is crucial to protect the information stored in our computers or transmitted over the internet from potential attacks. Various cryptographic techniques can be employed for this purpose. The choice of cryptographic method typically depends on factors like response time, bandwidth, confidentiality, and integrity required by a specific application.

Chapter 5

Deep Learning Component

5.1 Implementation

In Chapter 2 of our research, we thoroughly explored various models for human action recognition and carefully evaluated their respective strengths and weaknesses. After a comprehensive analysis, we made the decision to implement our system using the PoseC3D and ST-GCN models. The primary objective behind this selection was to examine the differences between a ResNet-based model and a GCN-based model.

By implementing both models in our system, we aimed to gain valuable insights into their performance, accuracy, and computational efficiency. Additionally, this approach allowed us to compare how the ResNet-based and GCN-based architectures influence the overall action recognition results. The comprehensive evaluation of these models helped us understand their respective advantages and limitations, enabling us to make well-informed decisions based on their suitability for different use cases and system requirements.

Choosing PoseC3D and ST-GCN for our implementation ensured that our research is aligned with established benchmarks in the field of human action recognition. This allowed us to build upon existing knowledge and contribute valuable findings to the research community while also enabling us to address the specific objectives and goals of our study.

5.2 Experiments

In our experiment, we utilized data from both the NTU RGB+D and ETRI-Activity3D Datasets. A snapshot of the samples are as shown in Figure 5.3 and further details are outlined in Table 5.1. The NTU RGB+D dataset comprises 60 distinct action classes, whereas the ETRI-Activity3D Dataset encompasses 55 action classes. Notably, these datasets exhibit a higher correlation among their action classes and were recorded in varied environments—lab-based and home-based. Hence, these datasets have the potential to offer a more comprehensive evaluation.

In the experiment, we use a pre-trained model on PoseC3D with a backbone SlowOnly-ResNet50 network architecture. Both pre-trained models of PoseC3D and ST-GCN are trained with NTU RGB+D dataset and used frame sample strategy with uniform sampling to sampling frames. PoseC3D pre-trained model is sampling with 48 frames while the ST-GCN pre-trained model is sampling with 100 frames.

We conduct experiments on the inference speed and time within cloud side using samples data from the dataset. And, we also used AltumView Sentinare V2 sensors to conduct the pose estimation for the skeletal data as the input source of the system to perform the prediction on a real scenario environment and observe the problems described in this Chapter. Sentinare V2 sensor uses the Rockchip V1126 platform which is based on a quad-core ARM Cortex A7 32-bit core processor with 1200MHz base clock speed and 2.0 trillion operations per second (TOPS) neural processing unit(NPU).

5.2.1 Inference Speed

The inference time plays a crucial role in the overall cost and computational efficiency of the HAR system. Especially when the HAR system is deployed across thousands, tens of thousands, or even hundreds of thousands of sensors, the inference time becomes a significant factor in determining the system's monetary and computational costs.

When the HAR system is utilized by a large number of sensors, the inference time will directly multiply with the sheer volume of recording data received by the system. As a result, a longer inference time can lead to substantially higher computational costs and fee, as more computing resources are required to process the incoming data. Moreover, the inference time also greatly influences the user experience. A lengthier inference time means that the action prediction will take more time to be available to the end-users, resulting in a delay in receiving the desired information or response. This delay can negatively impact the user experience, as users might expect quick and real-time action recognition results.

To compare the inference speed of different GPUs and CPUs, we selected three videos from the NTU-RGB+D and ETRI-Activity 3D datasets, shown in Table 5.3. These videos were processed with three models, all yielding the same correct prediction results on both PoseC3D and ST-GCN model.

In serverless computing, a "cold start" refers to the initial invocation of a serverless function that hasn't been used for a certain time. When a serverless function is invoked for the first time or after a period of inactivity, the serverless platform needs to prepare the runtime environment for that function. This includes allocating memory, initializing dependencies, and booting up the runtime environment in which the function will execute. To ensure accurate measurement of inference time during our experiment, we took steps to

mitigate the impact of cold starts. Specifically, we removed the first inference result from each instance type, as the initial invocation could be affected by the cold start process, leading to increased latency for the first inference.

First, we conducted 101 inferences on three different samples using different setups: AWS Lambda endpoint with CPU, a local machine with and without GPU, and an AWS Sagemaker GPU endpoint. By comparing the inference times across different GPUs and CPUs, we obtained clear insights into the variations in computational performance. Removing this first result, which could be affected by cold start delays,we calculated the average inference time based on the remaining 100 results for each setup. After removing the first result, the inference time which used with CPU options, AWS lambda serverless and AMD Ryzen 2920x, and GPU options, RTX3080ti and Nvidia T4, are presented in Figure 5.1.

From the result, it is evident that the PoseC3D model incurs significantly higher computational time compared to the ST-GCN model. This observation is reasonable since the PoseC3D model is ResNet-based, which involves more parameters and computations compared to the ST-GCN model as discussed in Chapter 2. The substantial difference in inference time between the two models becomes a critical consideration when selecting the PoseC3D model for a system with time constraints. While the PoseC3D model may offer higher accuracy or other advantages, its longer inference time can be a limiting factor in certain real-time applications or systems where latency is a concern.

When deciding which model to use, it's essential to carefully weigh the trade-offs between computational cost, inference time, and desired system performance. The results provided valuable insights into the computational efficiency of each model, enabling informed decisions on model selection based on specific project requirements and constraints.

Secondly, we conducted experiments using different GPUs, RTX3080 Ti, Nvidia A10G and Nvidia T4, with the PoseC3D method to assess the potential for achieving faster inference times using serverless options on AWS Sagemaker. We also conducted the inference test on other CPUs options, AMD Ryzen 2920x, AWS lambda serverless, Intel Xeon 2nd gen and Intel Xeon 1st gen option. The Intel Xeon 2nd generation comes with the base clock speed of 3.1GHz processors, Intel Xeon 1st generation comes with a base clock speed of 3.4GHz processors, and AMD Ryzen 2920x comes with the base clock speed of 3.5GHz processors, while the base clock speed of serverless options us unclear since it is using the available power of virtual CPUs across the serverless services. The results are presented in Figure 5.2.

From the result, we can see that the RTX3080 Ti and Nvidia A10G have a similar inference time, while Nivida would take approximately double the time. And the best CPU option in the test, AMD Ryzen 2920x, took about 7 times of the RTX3080 ti. And, the

serverless options and Xeon processor takes more than 11 times for the inferences. Therefore, we can see the GPU options are better choices if there is an inference speed requirement of the system.

The experiment demonstrated that the inference speed could be enhanced as cloud hardware improves over time. With the continual advancements in cloud infrastructure, including more powerful and efficient GPUs, the PoseC3D model has the potential to become a more cost-effective and viable option for time-constrained systems. Faster and more efficient hardware on the cloud platform would lead to significant improvements in the overall inference time, making the PoseC3D model a more desirable choice for real-time applications and systems with stringent latency requirements.

By staying abreast of hardware advancements on the cloud and adapting accordingly, we can leverage the most optimal computational options available. This adaptability ensures that the PoseC3D model can keep up with the demands of time-sensitive systems, providing superior performance and responsiveness for users. As cloud hardware continues to evolve, the potential for PoseC3D to become an affordable and efficient choice for time-constrained systems becomes increasingly promising.

These two experiments shed light on the performance differences of the various models on different hardware configurations. The inference speed is a crucial factor to consider, especially in large-scale deployments, as it directly impacts the computational cost and user experience. By understanding the inference times for different models on distinct GPUs and CPUs, we can optimize the system's performance and choose the most suitable hardware configurations to achieve the desired speed and efficiency.

Considering both the computational cost and user experience, a shortened inference time is preferred when cost efficiency and system responsiveness are significant concerns. Reducing the inference time not only helps in managing computational resources more efficiently but also enhances the overall user experience by providing faster and more real-time action predictions. By optimizing the inference time, the HAR system can achieve better performance, scalability, and cost-effectiveness, making it a more reliable and practical solution for large-scale deployments.

5.2.2 Sampling Methods of Recording Data

During our experiment, the length of the recording data proved to be dynamic due to the sensors' nature of capturing data only when a person is detected in the scene. As a

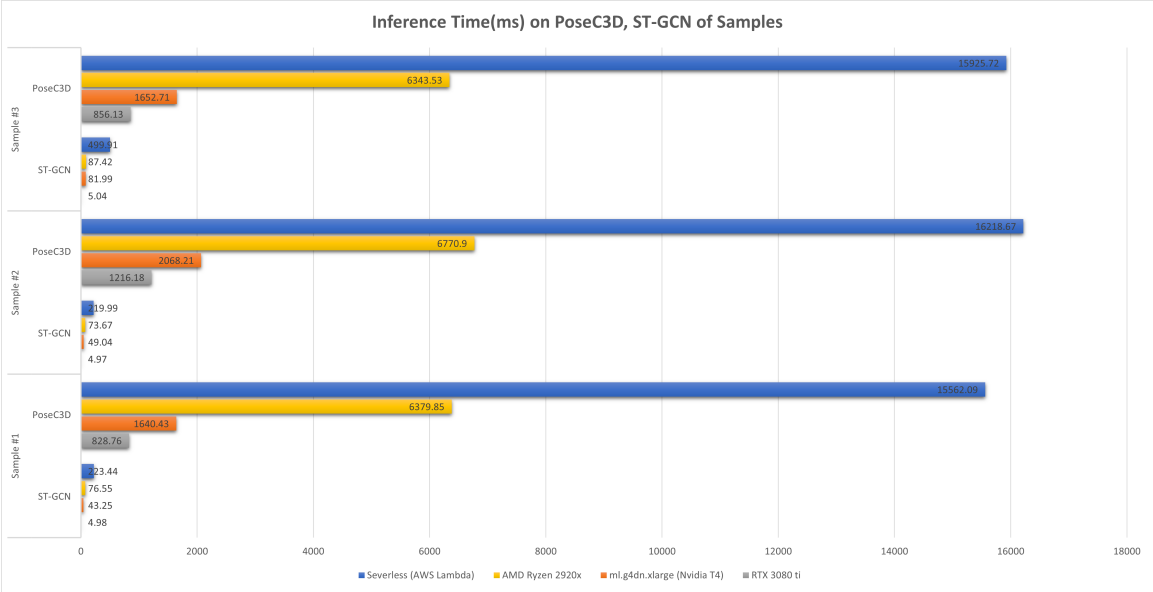


Figure 5.1: Chart of Inference Time of the Samples

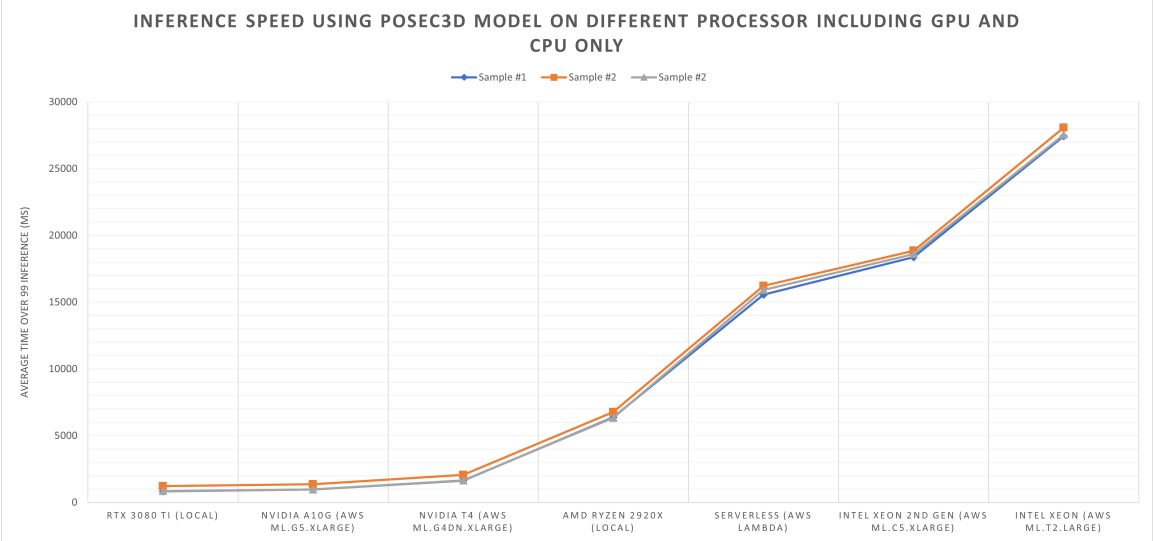
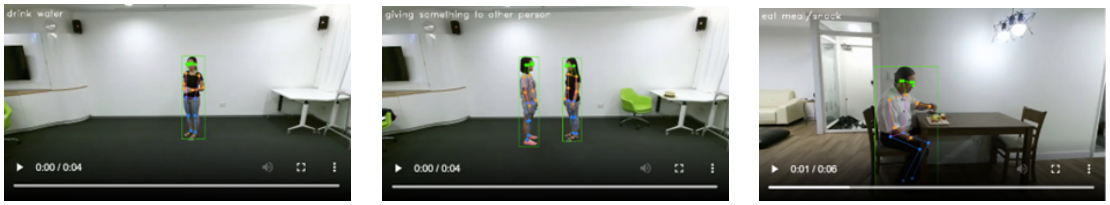


Figure 5.2: Chart of Inference Time of the Samples



a) b) c)

Figure 5.3: Sample: 1) Drink water [42] 2) Give something to another person [42] 3) Eat meal [34]

Action Type	Sample Time (s)	# of Keypoints per Person	Number Of Person
Drink Water [42]	4	103	1
Give sth. to another [42]	4	115	2
Eat meal [34]	8	161	1

Table 5.1: Detail of the Samples for the Inference Test

result, the duration of a person's presence in the scene can vary significantly, leading to recording data with varying lengths, either short or long. Additionally, since the sensors are positioned differently, they capture the scene from distinct angles. To account for these variations and capture a more comprehensive view of the actions, we set up three cameras at different angles to obtain samples from multiple perspectives.

In contrast, the training dataset, NTU-RGB dataset, contains actions performed by individuals in predefined positions within the scene. To ensure compatibility with existing deep learning networks and pre-trained models, we needed to preprocess the recording data before inputting it into our HAR system. And, we chose to use the average time of the action duration in the dataset, 3 seconds, for the sampling reference. To achieve this, we conducted three different sampling methods to preprocess the recording data:

Random Sampling: This method involves selecting samples randomly within a specified range of time. However, this approach may not accurately represent the true action duration in the recording data.

Recursive Sampling: In this method, we sample the skeleton data at fixed intervals, such as every 3 seconds after each second passes. This approach provides consistent sampling intervals but may not fully capture the variability in action durations present in the recording data.

Static Sampling: Static sampling entails taking samples at fixed periods and then taking another sample after a specified time interval has elapsed. For instance, the skeleton data is sampled every 3 seconds, and the next sample is taken 3 seconds after the first sample. This method provides a more structured approach to sampling, aligning with the typical action duration in the NTU-RGB dataset.

By applying these sampling methods, we aimed to preprocess the recording data effectively, making it compatible with existing deep learning networks and pre-trained models used in the HAR system. This preprocessing step is crucial in ensuring accurate and meaningful action recognition results from the varied and dynamic recording data captured by the multiple sensors.

Sensor 1 Sensor 2 Sensor 3

Figure 5.4: 3 Cameras positions: Sensor 1) Left, Sensor 2) Middle, Sensor 3) Right

5.2.3 Experiment Setup

During the experiment, we strategically positioned three Sentinare 2 sensors at three different angles, as depicted in Figure 5.4. The objective was to create a setup that closely resembled real-life settings rather than a controlled laboratory environment. By doing so, we aimed to gain a more comprehensive understanding of the actual scenarios and evaluate the accuracy of our action prediction system in real-world conditions.

The environment where the experiment took place was designed to replicate everyday living spaces. It included a sofa, a table with various objects, and open windows, which allowed natural sunlight to illuminate the area. This setup was chosen to introduce real-life elements and challenges that individuals might encounter in their daily routines.

By conducting the experiment in such a setting, we could better assess the performance of the action recognition system in handling various factors like object interactions, diverse lighting conditions, and potential occlusions caused by furniture or other objects. This approach enabled us to obtain valuable insights into the system's capabilities and its ability to accurately predict actions in more authentic scenarios, ultimately enhancing the system's practicality and reliability for real-world use.

5.2.4 Experiment Result

In the initial experiment, several challenges were identified that affected the accuracy of Pose Estimation and led to incorrect predictions.

Problem 1: Brightness Condition

The impact of varying brightness conditions emerged as a significant challenge affecting the accuracy of Pose Estimation and subsequently influencing the prediction results of the video recognition system. In real-world scenarios, video recordings often encounter diverse lighting conditions, ranging from well-lit environments to dimly lit or poorly illuminated settings. These fluctuations in brightness can lead to substantial variations in the appearance of human subjects and the surrounding background, posing a considerable obstacle for accurate pose estimation.

Under brightly lit conditions, the high contrast between light and shadow might cause certain body parts to be overexposed or underexposed, resulting in misinterpretations during the Pose Estimation process. Similarly, in low-light situations, the lack of sufficient illumination may obscure critical details of the human body, making it challenging for the Pose Estimation algorithms to precisely capture the key points and skeletal structure.

Moreover, rapid changes in lighting conditions, such as sudden flashes or flickering lights, can further degrade the quality of the video frames. This can cause temporary inconsistencies in the appearance of the subjects, leading to inaccurate estimations of their poses. Consequently, when the Pose Estimation process generates imprecise skeletal representations, the subsequent action recognition models may yield incorrect predictions, impacting the overall performance and reliability of the system.

Addressing the challenges posed by varying brightness conditions requires robust preprocessing techniques and adaptive algorithms capable of dynamically adjusting to different lighting environments. Advanced image enhancement methods, such as contrast adjustment and histogram equalization, can be employed to standardize brightness levels and improve the overall visibility of key body points. Additionally, incorporating machine learning models that can learn from different lighting scenarios and adapt their internal representations accordingly could lead to enhanced pose estimation accuracy and more reliable action recognition results.

To mitigate the adverse effects of brightness conditions, continuous research and development in the field of computer vision and deep learning are essential. By refining existing algorithms and developing innovative approaches, it becomes possible to build a video recognition system that demonstrates remarkable resilience to varying lighting conditions and consistently delivers accurate and dependable action recognition outcomes across diverse real-world settings.

To address the issue of the brightness condition affecting pose estimation results, it may be necessary to make hardware changes to the sensor. One approach is to apply a polarizing filter or replace the lens with an auto-iris lens. By using a polarizing filter, the amount of light reaching the camera sensor can be controlled, albeit without automatic adjustment. On the other hand, an auto-iris lens provides more advanced functionality but comes at a higher cost. Consideration should be given to the specific requirements and budget constraints when deciding between these two option

Problem 2: Object Blocking

The challenge of object blocking emerged as a critical factor affecting the accuracy of Pose Estimation and, consequently, influencing the prediction results of the video recognition system. The occurrence of object blocking was evident in various scenarios. During the

indoor experiment, objects such as furniture, walls, or other obstacles could obstruct specific body parts, resulting in overlapping body segments within the camera's view. These occlusions posed challenges for the accurate detection and tracking of key points and skeletal structure by Pose Estimation algorithms.

It is important to note that the experiment also highlighted the potential for occlusions in outdoor environments. In such cases, natural elements like foliage or structures might partially obscure individuals, further complicating the accurate estimation of key body points by the Pose Estimation algorithms.

Overall, these occlusions significantly affected the quality of the pose estimation process, ultimately impacting the accuracy and reliability of the action recognition results. Addressing these challenges is crucial to enhance the performance and effectiveness of the entire video recognition system.

Problem 3: Random Low Frame Rate

Furthermore, the inital results from the PoseC3D and ST-GCN models did not meet the expected accuracy, and incorrect action labels were assigned. For instance, in one example, the falling action was misclassified as "cross toe touch" by the PoseC3D model and as "touchback" and "cheer up" by the ST-GCN model. The main reason is that the sensor produces an uneven distribution of frame over the time of action due to limited computational power on the device, between 3 to 7 frames per seconds (FPS). The failure to correctly detect the falling action was attributed to the random and low frame rate of the recorded file, as shown in Figure 5.5. And, it leads to not enough temporal information of the data to predict a correct action. To validate this hypothesis, an additional test using the ETRI-Activity dataset was conducted to examine whether the models could accurately predict actions under such conditions. The results of this test led to proposed improvement solutions, which will be discussed in later this section.

To tackle this challenge, there are three possible methods can be taken. First, increasing the number of frames extracted from the sensors, however, this method is not suitable for us as the hardware of the devices is fixed and nonupgradable. Secondly, we can retrain the model by reducing the number of sampling frames from the training dataset, however, it would take more time to retrain the model. It would require more time for the research and it may need to redesign the network architecture of the model to work with low frame rate videos, which is beyond the scope of this work and can be a future research topic. Therefore, we took the third method, frame interpolation method. We conducted experiments with a frame interpolation method, aiming to increase the frame rate and enhance the accuracy of detection, especially in light of the problem of random low frame rates. Increasing the num-

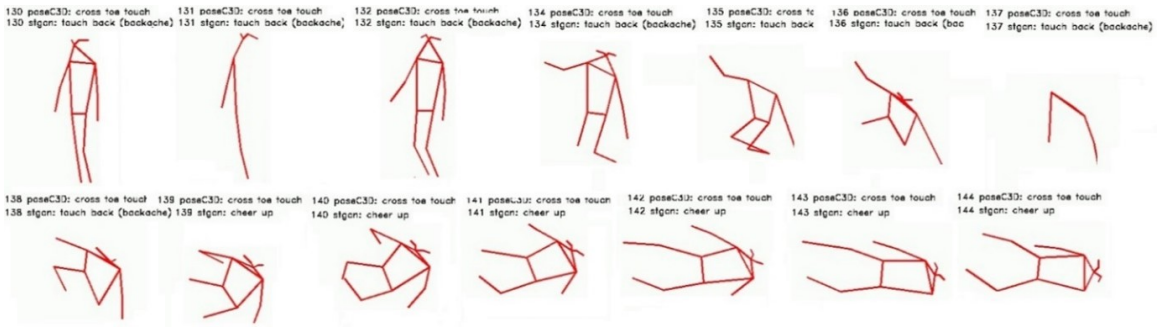


Figure 5.5: Prediction Result on PoseC3D and ST-GCN without implementing the frame interpolation method

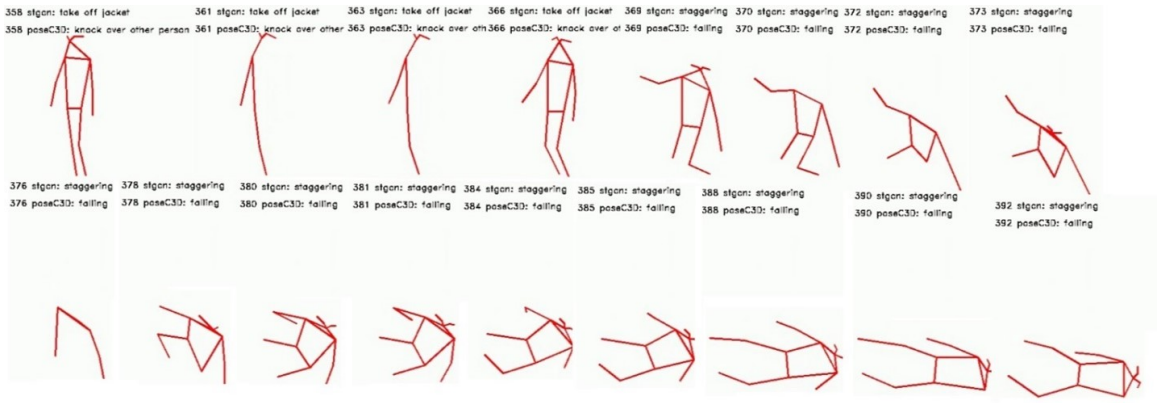


Figure 5.6: Prediction Result on PoseC3D and ST-GCN with implementing the frame interpolation method #2

ber of frames proves to be crucial for achieving improved detection performance. However, accomplishing this without requiring hardware changes becomes equally important.

Frame Interpolation Method #1: In this approach, we simply duplicated the same frame over the delta time of the skeleton frame. For instance, if frame 0 has a delta time of 50 milliseconds, we would duplicate frame 0 at intervals of every 1 millisecond. To execute this method successfully, we need to be aware of the delta time in milliseconds between the previous frame and the current frame, along with the desired frame per second for the in-filling result. With this information, we can calculate the number of in-filling frames required between frames.

$$N = \frac{\Delta T_{ij}}{1/FPS * 1000 \text{ ms}}$$

$$K = 0, 1, 2, \ldots, N$$

$$(x, y)_k = (x, y)_i \quad \text{——————————} \quad \text{Frame Interpolation Method \#1}$$

$$(x, y)_k = \frac{(x, y)_j - (x, y)_i}{N} * K + (x, y)_i \quad \text{——————} \quad \text{Frame Interpolation Method \#2}$$

Frame Interpolation Method #2: In this method involves increasing the frame rate by referencing the x, y location of the previous frame. Similar to Method 1, we require knowledge of the delta time in milliseconds between the previous frame and the current frame, as well as the desired frame per second for the in-filling result. With this information, we can calculate the number of in-filling frames needed between the frames. The formula is then used to determine the estimated x, y location of the in-filling frame index between the frames.

To better visualize the effectiveness of these methods, Figure 5.6 provides a visual representation of the expanded films, comparing them with the original clip displayed in Figure 5.5. As depicted in Figure 5.5 and Figure 5.6, the overall frames become smoother, and we can observe a remarkable improvement in the accuracy of the PoseC3D and ST-GCN results. The predictions have become more correct and reasonable after implementing these frame interpolation methods. These advancements are essential for enhancing the performance of the video recognition system and further consolidating the accuracy of action recognition results.

Problem 4: Ambiguous Skeleton Pattern

During the examination of the experiment results, a notable issue arose concerning certain action scenarios that exhibited similar skeleton motion patterns, even presenting challenges for human observers. Figure 5.7 visually illustrates some of these misclassified actions, including "Pick Up," "Use Phone," "Phone Call," and "Sit Down." In the case of ST-GCN, "Pick Up" is mispredicted as "Staggering," "Use Phone" as "Touch Back," "Phone Call" as "Reading," and "Sit Down" as "Touch Back." On the other hand, PoseC3D misclassifies "Pick Up" as "Vomiting," "Use Phone" as "Check Time," "Phone Call" as "Drink Water," and "Sit Down" as "Squat Down."

These mis-classifications indicate the challenge posed by similar skeleton patterns, emphasizing the need for improved action recognition methods to overcome these ambiguities and achieve more accurate and reliable results. Addressing these issues is crucial to enhance the performance and practicality of the video recognition system in various real-world applications.

To address the challenge of ambiguous skeleton patterns and improve action recognition accuracy, object detection can play a crucial role. By considering the interaction between objects and skeleton keypoints, actions with similar patterns can be differentiated based on the objects involved. For instance, holding a phone or a book at the hand keypoint would indicate different actions. While recent research has introduced a skeleton-based method that combines object localization and human action recognition to tackle this issue, such an approach may not be suitable for low computational power embedded devices.

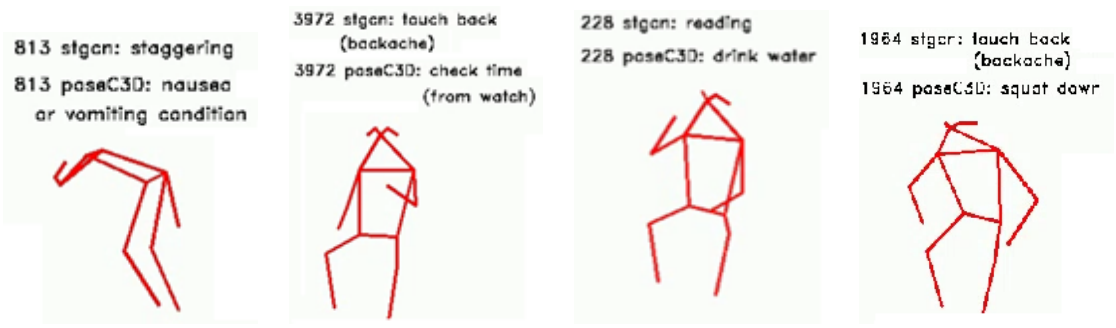
In Figure 6.2, we visually illustrate the significance of object detection keypoints combined with the skeleton keypoints. This integration can potentially alter the estimated action result based on the objects held by the observed individuals. This valuable context aids in improving the precision of action recognition, making it more contextually aware and adaptable to various scenarios. Implementing object detection in conjunction with skeleton-based action recognition is an advanced approach that can significantly contribute to the system's accuracy and reliability, especially in real-world settings where actions are often associated with specific objects or interactions.

For example, when the skeleton is holding a phone, it becomes considerably easier to identify the action as "Use Phone." Similarly, recognizing an individual as "Reading" is facilitated when they are holding a book, and the action of "Watch Time" can be inferred when the person is wearing a watch. This contextual awareness allows the system to make more informed and precise action predictions, resulting in a more sophisticated and insightful understanding of human activities in different contexts.

It is also crucial to take into account the removal of undesired labels that have a low chance of occurring in the specific use case. For instance, when focusing on everyday activities of home users, a more practical approach is to generalize frequently occurring actions rather than relying on complex object detection. By generalizing actions with similar patterns and excluding undesired action types from the action labels, we can greatly enhance the accuracy of labeling desired daily activities.

This approach streamlines the recognition process and reduces the risk of misclassification for actions that are less relevant or infrequently performed in the given context. By emphasizing the accurate identification and classification of commonly performed actions, the video recognition system becomes more reliable and effective for everyday users.

By striking a balance between specificity and generalization in action labeling, we can create a user-friendly and efficient system that seamlessly integrates into daily life. This enhancement significantly improves the practicality and usability of the video recognition system, making it a valuable tool for real-world applications in home environments.



a) b) c) d)

Figure 5.7: Misclassified Actions: a) Pick Up b) Use Phone c) Phone Call d) Sit Down

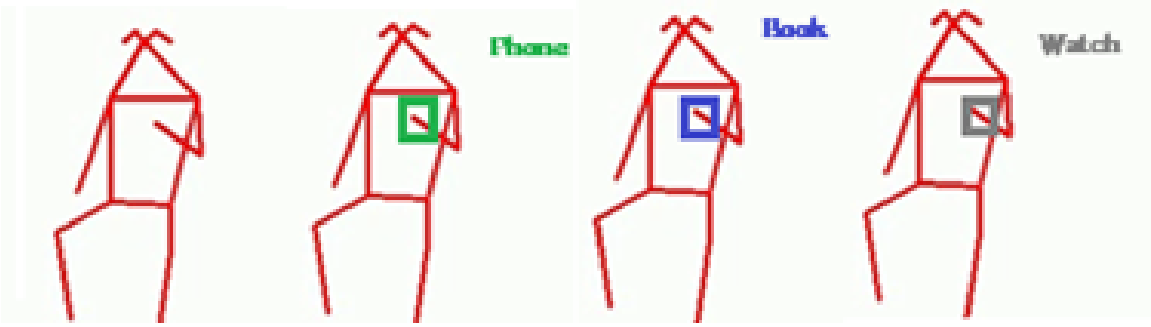


Figure 5.8: Modified Image for Illustration of Object Interacted HAR Method

Chapter 6

Interface Component

To ensure the user data privacy, we would also integrate the secured application programming interface (API) to allow the 1st and 3rd party application for accessing the data from the system. The API allows the developers to retrieve the data from the cloud network. And we used the industry open standard to implement the API and ensure the data access is secured, organized, and identifiable. By using the open standard, third-party developers can integrate their application quickly and avoid any inconsistent structure across different applications.

6.1 RESTful API

Representational state transfer (RESTful) APIs enable interoperability between different software systems and platforms. They provide a common language and set of protocols, allowing applications developed by different vendors to work together effectively. This promotes compatibility and facilitates integration, making it easier to connect diverse systems and components.

By using RESTful APIs, we can leverage existing functionality and services without having to reinvent the wheel. APIs provide pre-defined methods and functions that abstract complex operations, making it easier and faster to implement specific features or access certain resources. This accelerates development time and improves overall efficiency. It also presents a simplified and consistent interface that is easier to understand and work with. This reduces the learning curve, simplifies development efforts, and enhances developer productivity.

6.2 OAuth 2.0 Integration

To allow 3rd party applications to have the access right for our RESTful API, we used open authorization 2.0 (OAuth 2.0), which is an open standard that is commonly used to allow data access permission by the service users. The standard specification is

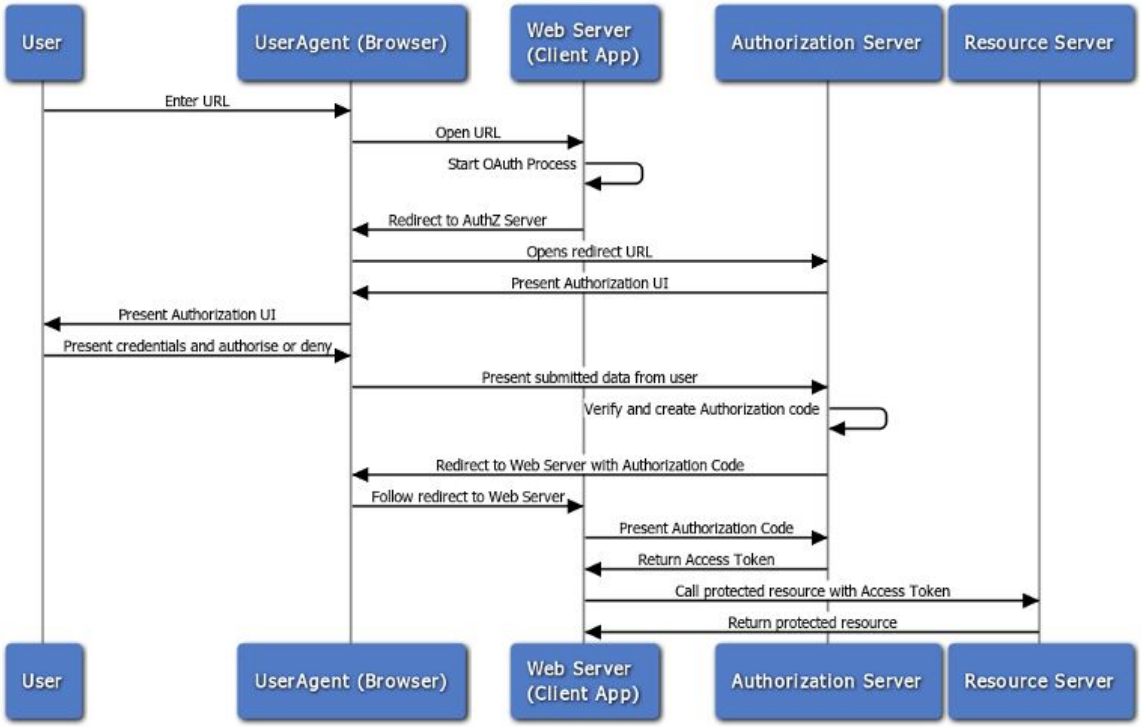


Figure 6.1: OAuth 2.0 Authorization Code Flow [20]

organized by Internet Engineering Task Force (IETF), which is a standards organization for the internet and organizes the technical standards for the internet protocol [24] [25]. As a high privacy system, it should avoid the data being accessed by any 3rd party client without the user's consent. In OAuth 2.0, there are 7 possible grant types, authorization code, client credentials, refresh token, proof key for code exchange, device token, implicit flow, and password grant. The implicit flow and password grants are legacy grant types, which are considered as insecure. In the studies, only the first three types are implemented in the current system. And, it has been successfully used by mobile, smart assistant, and third-party integration.

Authorization Code Flow

The Authorization Code Flow is utilized when 3rd party applications need access to user data within our system. The application first submits an authorization code request for the user scope to our server and proceeds with a user consent flow. In the user consent flow, our server redirects the application's UI to our login system, where users can view the data areas the application requests access to through our API interface. Once the user consents to the application's request, the application is issued an authorization code. This code can then be exchanged for a pair of access and refresh tokens, each with an expiration time. The access token allows the application to access our API interface on behalf of the

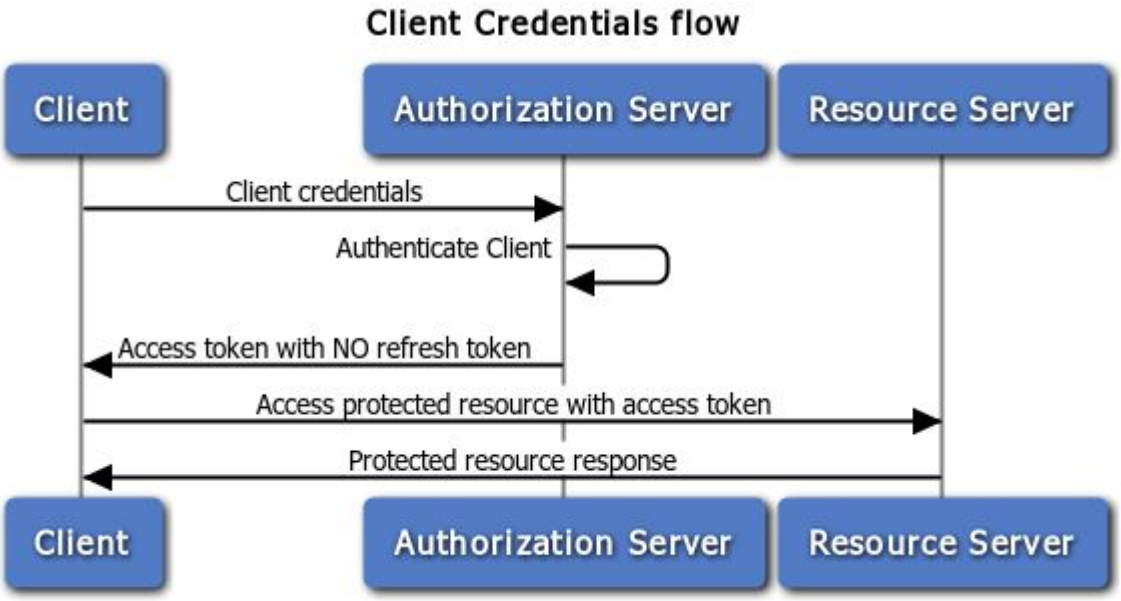


Figure 6.2: OAuth 2.0 Client Credentials Flow [20]

user, while the refresh token is used to obtain a new access token without requiring the user consent flow again. The complete flow is illustrated in Figure 6.1

Client Credentials

The Client Credentials grant type in OAuth 2.0 allows 3rd party applications to access resources in our system without user involvement. When the application wants to access our API, it sends its client credentials, which includes client ID and client secret, to the authorization server, which issues an access token granting access to specified resources. Unlike other grant types tied to user-specific scopes, Client Credentials provides access to resources not linked to any particular user. This flow is useful for backend services or machine-to-machine communication, where user authentication is unnecessary. However, it should be restricted to trusted applications to ensure secure and authorized resource access.

Refresh Token Flow

Refresh Token – As both the access token and refresh token have expiration time periods, the application can use the refresh token to obtain a new pair of access and refresh tokens from the authorization server. This allows the application to continue accessing the protected resource without requiring the user consent flow again.

By implementing OAuth 2.0, our system ensures secure and controlled access for 3rd party applications, enhancing user privacy and data protection while enabling seamless integration with various platforms.

6.3 Notification Components

The notification component plays a vital role in the system by handling system notifications and emails, particularly in providing analyzed results from the deep learning components. The analyzed results obtained from human action recognition are valuable information for medical monitoring purposes. Initially, we had implemented our notification system on an EC2 instance. However, this approach proved insufficient when a large number of alerts started flooding into the system. The EC2 instance became significantly slower due to the overwhelming influx of alerts. To ensure stable system operation, we made the decision to implement the notification component using AWS Lambda. By utilizing AWS Lambda, the system benefits from automatic scaling, allowing it to handle a high volume of requests without performance degradation.

6.4 Integration with Alexa Together

In addition to the user interface applications, the system boasts the capability of seamless integration with smart assistant technologies. The US market offers various popular smart assistants, such as Google Assistant, Amazon Alexa, and Apple Siri, among others. As a proof of concept, the system has successfully integrated with fall action detection, enabling it to monitor falls among seniors using the Amazon Alexa Together service [35]. Through this integration, the system provides 24/7 urgent response, allowing seniors to connect with emergency services swiftly when needed. This comprehensive approach ensures prompt assistance in case of an accident, significantly enhancing the safety and support for seniors.

To achieve integration with Alexa Together, the first step was to utilize the Alexa Smart Home skill in the Alexa Skills Kit (ASK) with account linking. Two options were available for the account linking process: the Alexa-app only flow and the app-to-app flow. The Alexa-app only flow served as a simple solution, particularly for those starting out, as it eliminated the need to develop a separate mobile application. Users could enable the Alexa Skill in the Alexa app and link their Alexa account through the authorization website of the system. This process utilized the Authorization code flow, as described in Section 6.2, to establish a connection with our system. Users then logged in to our system and provided consent for any necessary permissions for the Alexa app.

However, for systems with an existing dedicated mobile application, the Alexa-app only flow might not deliver the desired seamless user experience since it involved users linking with an authorization website. In such cases, the app-to-app flow proved to be a more suitable option. This approach allowed users to directly link their Alexa account and smoothly transition to the designated mobile application without any redirection to an authorization website. Implementing this flow also required support for the deep linking feature within the mobile application, which ensured a seamless transition for users.

Upon successful account linking integration, the system gained access to users' Alexa accounts, enabling it to send push notifications to Alexa Together. Specifically, when the system detected a fall action, it forwarded the corresponding notification to Alexa Together. Subsequently, Alexa devices initiated automatic communication with the users to assess if any assistance was needed. If the user required help or failed to respond, Alexa Together escalated the event for verification by human personnel. This streamlined process minimized the occurrence of false alerts while reducing the duration of human monitoring. Consequently, it significantly improved the accuracy of alerts and optimized the overall monitoring system's efficiency.

By integrating with the Amazon Alexa Together service, the system offered an essential safety and support feature for seniors, easing the workloads for both users and their caregivers.

Based on the proof of concept implementation, it is evident that the system can further expand the use case of the HAR and provide more interactive features for users. This ongoing enhancement will enable the system to adapt to diverse scenarios and user needs, making it an even more versatile and user-friendly solution for various applications.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

In the quest for developing advanced systems for Human Action Recognition (HAR), this thesis embarked on a journey to create a robust, privacy-aware, and seamlessly integrated solution. Through the course of this research, we have explored various aspects of HAR systems, addressing core components ranging from data collection through to secure interfaces and integration with smart assistant technologies. This journey has been driven by a commitment to enhance the state-of-the-art in HAR and make meaningful contributions to the field. As we draw this thesis to a close, we reflect on key findings, challenges, and contributions, encapsulating the essence of this research endeavor.

In Chapter 2 of the introduction of human action recognition, the use of Graph Convolutional Networks (GCNs) is discussed, highlighting their ability to improve accuracy and robustness by considering spatial relationships between skeleton key points. Privacy-preserving approaches, including skeleton data transmission, are emphasized, and various data modalities for recognition are explored. The text also addresses the trade-offs between privacy, efficiency, and accuracy in the context of deep learning models for computer vision, including RGB-based and skeleton-based approaches, as well as the use of object detection and pose estimation in recording devices. The need to balance user preferences, privacy, and effective processing in various scenarios is a central theme throughout.

In Chapter 3 of the thesis, the proposed system architecture leverages cloud computing services to create a secure, efficient, and scalable infrastructure for human activity recognition. It incorporates components for data recording, deep learning-based activity recognition, and API access control. Cloud services like AWS Kinesis Data Firehose, AWS IoT Core, AWS IAM, and AWS S3 are utilized to ensure data security, authentication, and access control. The architecture aligns with principles of reliability, operational excellence, security, performance efficiency, cost optimization, and sustainability. By adopting cloud-based solutions, the system enhances scalability, availability, and maintainability while reducing the

burden of server maintenance and offering cost-effective, real-time data processing. Comparatively, this cloud-based approach provides more flexibility and adaptability than relying solely on edge computing.

In chapter 4, the Data Recording component in the cloud is explored, which serves a pivotal role in capturing, processing, and storing data from sensors. This component encompasses three integral parts: Data Capture, Data Transit, and Data Storage. Data Capture focuses on efficient data transfer to minimize transfer costs and enhance data security. This is achieved by transmitting only essential data in a compact format, including keypoints data and frame duration. The data is also versioned to maintain flexibility without compromising backward compatibility. Data Compression techniques are employed to further optimize data transfer by implementing the LZMA algorithm, balancing compression ratio and speed. In the Data Transit section, AWS Kinesis Data Firehose is employed for real-time data processing, while data security in transit is ensured through secure protocols like HTTPS with TLS 1.3. The Data Storage component indexes recording data for efficient retrieval and employs AWS S3 for stable and parallel processing. Elasticsearch is chosen as the database engine, capable of handling large data volumes, providing scalability, multi-tenancy support, and real-time searching. Data security in storage is maintained through minimum access controls, encryption, and server-side encryption using AWS Key Management Service (SSE-KMS) for AES-256 encryption, ensuring data confidentiality, integrity, and availability.

Chapter 5 demonstrated the experimental setup and results for a human action recognition (HAR) system involving three sensors positioned at different angles, aiming to replicate real-life settings rather than controlled laboratory environments. They examine several challenges that impact the accuracy of the pose estimation and subsequent action predictions. The first challenge arises from varying brightness conditions, where diverse lighting affects pose estimation, leading to incorrect predictions. To address this, the authors suggest image enhancement and machine learning approaches to adapt to different lighting scenarios. The second challenge involves object blocking, where furniture and obstacles obstruct body parts and pose difficulties for keypoint detection. To mitigate this, the authors recommend sophisticated pre-processing techniques and adaptive algorithms. A third challenge is posed by random low frame rates in the recorded files, leading to incorrect action labels. The authors propose frame interpolation methods to increase the frame rate and improve detection accuracy. The final challenge relates to ambiguous skeleton patterns, as some actions exhibit similar motion patterns. The authors advocate combining object detection with skeleton data to provide context for improved action recognition, making the system more adaptable and accurate in real-world scenarios. They also recommend generalizing frequently occurring actions while excluding less relevant or infrequent ones to enhance system relia-

bility. These insights aim to enhance the practicality and reliability of the video recognition system for various real-world applications.

Chapter 6 shows that the system incorporates a comprehensive interface component that focuses on user data privacy and accessibility. This component integrates a secured application programming interface (API) to facilitate data access for both 1st and 3rd party applications while ensuring data security. It employs RESTful APIs to enable interoperability between different software systems and platforms, promoting compatibility and efficiency. OAuth 2.0 integration is used to grant access rights to 3rd party applications securely. The notification component handles system notifications and emails, enhancing scalability through AWS Lambda. The system also showcases successful integration with Amazon Alexa, offering fall action detection and urgent response capabilities to enhance the safety and support for seniors. The integration options provide a seamless user experience, allowing users to enable Alexa skills and establish connections with the system, ultimately improving the efficiency of the monitoring system. This comprehensive approach emphasizes user privacy, integration, and safety, making it a versatile and user-friendly solution with potential for further expansion and adaptation to diverse scenarios and user needs.

The research conducted in this thesis encompassed several fundamental components, each of which has yielded valuable insights and contributions:

Data Collection and Processing: The thesis commenced with the critical process of data collection. It leveraged multiple sensors strategically positioned in real-life environments, replicating everyday living spaces. This approach enabled a comprehensive understanding of real-world scenarios and provided insights into the system's performance in such settings.

Privacy-Aware Interface: The development of a privacy-aware interface component was crucial in safeguarding user data. This component integrated a secured API using industry open standards, ensuring organized and identifiable data access. The adoption of Representational State Transfer (RESTful) APIs promoted interoperability, compatibility, and efficient integration, reducing the learning curve for developers and enhancing their productivity.

OAuth 2.0 Integration: Security and controlled access were at the forefront of our concerns. The adoption of OAuth 2.0 standard specification allowed for secure data access permission while ensuring that 3rd party applications could not access user data without their consent. The integration of various grant types, such as authorization code, client credentials, and refresh token, bolstered data security and user privacy.

Processing Components: The processing component was identified as a critical element for system performance. The transition from EC2 to AWS Lambda provided automatic scaling, ensuring stability and performance even in the face of a high volume of alerts.

Integration with Smart Assistants: The thesis successfully showcased integration with Amazon Alexa, expanding the system's use case. This integration provided an essential safety and support feature for seniors, enhancing the accuracy of alerts and the overall efficiency of the monitoring system.

7.2 Future Work

In our forthcoming research endeavors, we will focus on several pivotal areas to further refine and advance our Human Action Recognition (HAR) system. Firstly, we plan to implement quantization techniques on the models, a strategy known to have the potential to significantly optimize their performance. By reducing the precision of numerical values within the models, we can attain improved efficiency without compromising the accuracy, ensuring a more resource-efficient system.

Secondly, our attention will be directed towards the exploration and integration of more advanced GCN-based HAR models into our system. Our experimentation has revealed that GCN-based models offer notable advantages in terms of speed and cost efficiency compared to the ResNet-based models. Therefore, by embracing and enhancing these GCN-based approaches, we can further elevate the accuracy and overall performance of our HAR system.

Furthermore, we remain dedicated to addressing problems discussed in Sec. 5.2.4, a challenge that has surfaced in our current system. We recognize the significance of finding an effective solution to this problem, and we are committed to researching and implementing approaches that can mitigate its impact. By proactively addressing this issue, we aim to fortify the overall robustness and reliability of our HAR system.

Our future work revolves around three key pillars: optimizing the models through quantization, exploring advanced GCN-based HAR models, and finding effective solutions to the challenges. These research endeavors are poised to continually enhance and advance our HAR system, ensuring its effectiveness and applicability in real-world scenarios.

To further enhance action detection, we will investigate the potential benefits of incorporating object detection assistance, as recently explored in related research [8]. This integration may offer valuable context and improve action recognition results in scenarios where objects are involved. Also, it is possible to explore the area in use of the multiple cameras-based HAR, like the research in [35], by using more spatial information to improve the prediction result.

Additionally, we will explore the application of depth sensor technology for data collection and model training, aligning with the latest advancements in data acquisition and enhancing the overall capabilities of our HAR system.

After we finalize the HAR model that fits our need, we can also optimize job allocation on the cloud to minimize costs for both service providers and customers.

Bibliography

[1] Compression using the lzma algorithm. `https://docs.python.org/3/library/lzma.html` [Accessed: 25 Oct 2022].

[2] Fn project. `https://fnproject.io/` [Accessed: 08 Oct 2022].

[3] Ppmd compression library. `https://ppmd-cffi.readthedocs.io/` [Accessed: 19 June 2023].

[4] Protecting data with server-side encryption - amazon simple storage service. https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html [Accessed: 22 Oct 2022].

[5] Support for bzip2 compression. `https://docs.python.org/3/library/bz2.html` [Accessed: 25 Oct 2022].

[6] Zstandard - real-time data compression algorithm. `http://facebook.github.io/zstd/` [Accessed: 26 Oct 2022].

[7] Wisam Hamed Abd, Ahmed T. Sadiq, and Khalid Ali Hussein. Human fall down recognition using coordinates key points skeleton. In *2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, pages 232–237, 2022.

[8] Dustin Aganian, Mona Köhler, Sebastian Baake, Markus Eisenbach, and Horst-Michael Gross. How object information improves skeleton-based human action recognition in assembly tasks. *ArXiv e-prints*, 2023. arXiv:2306.05844 [cs.CV].

[9] Heba Hamdy Ali, Hossam M. Moftah, and Aliaa A.A. Youssif. Depth-based human activity recognition: A comparative perspective study on feature extraction. *Future Computing and Informatics Journal*, 3(1):51–67, 2018.

[10] N. Archana and K. Hareesh. Real-time human activity recognition using resnet and 3d convolutional neural networks. In *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, pages 173–177, Piscataway, 2021. IEEE.

[11] Andrew Tsun Hong Au, Dong Zhang, Chi Chung Chan, and Jiannan Zheng. Method and system for privacy-preserving health monitoring, US patent application 17/408,490, Aug. 23, 2021.

[12] Andrew Tsun Hong Au, Dong Zhang, Chi Chung Chan, and Jiannan Zheng. Deep-learning-based fall detection based on human keypoints, US patent application 17/534,448, allowed, Oct. 24, 2023.

[13] Rishabh Bajpai and Deepak Joshi. Movenet: A deep neural network for joint profile prediction across variable walking speeds and slopes. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.

[14] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 43(01):172–186, jan 2021.

[15] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2018. arXiv:1705.07750 [cs.CV].

[16] Francesco Casalegno. Graph convolutional networks — deep learning on graphs, 2023. https://towardsdatascience.com/graph-convolutional-networks-deep-99d7fee5706f [Accessed: 22 Oct 2022].

[17] Will Cavin. Reducing aws key management service costs by up to 99bucket keys, 2023. https://aws.amazon.com/blogs/storage/reducing-aws-key-management-service-costs-by-up-to-99-with-s3-bucket-keys/ [Accessed: 22 Aug 2023].

[18] Chi Chung Chan, Dong Zhang, Yu Gao, Andrew Tsun-Hong Au, Zachary DeVries, and Jie Liang. Privacy-preserving human action recognition, storage, and retrieval via joint edge and cloud computing, US patent application 17/522,901, Nov. 9, 2021.

[19] Cloudflare. Why is http not secure? | http vs. https. `https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/` [Accessed: 19 June 2023].

[20] Oracle Corporation. Api gateway oauth 2.0 authentication flows. `https://docs.oracle.com/cd/E50612_01/doc.11122/oauth_guide/content/oauth_flows.html` [Accessed: 01 Nov 2022].

[21] Coursera. 3 types of machine learning you should know, 2023. https://www.coursera.org/articles/types-of-machine-learning [Accessed: 13 Oct 2022].

[22] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition, 2022. arXiv:2104.13586 [cs.CV].

[23] Seth Eliot and Lara Valverde. The 6 pillars of the aws well-architected framework, 2018. `https://aws.amazon.com/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework/` [Accessed: 26 May 2022].

[24] Internet Engineering Task Force. Rfc 6749: The oauth 2.0 authorization framework, 2012. `https://datatracker.ietf.org/doc/html/rfc6749` [Accessed: 15 Oct 2022].

[25] Internet Engineering Task Force. Rfc 7636: Proof key for code exchange, 2015. `https://www.rfc-editor.org/rfc/rfc7636` [Accessed: 16 Oct 2022].

[26] Esther Fridriksdottir and Alberto Bonomi. Accelerometer-based human activity recognition for patient monitoring using a deep neural network. *Sensors*, 20:6424, 11 2020.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[28] Apoorv Gupta, Aman Bansal, and Vidhi Khanduja. Modern lossless compression techniques: Review, comparison and analysis. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–8, 2017.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. arXiv:1512.03385 [cs.CV].

[30] Phung Van Hiep and Rhee Eun Joo. A deep learning approach for classification of cloud image patches on small datasets. *Journal of Information and Communication Convergence Engineering*, 16(3):173–178, 2018.

[31] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. arXiv:1704.04861 [cs.CV].

[32] Amazon Web Services Inc. Overview of amazon web services: Aws whitepaper, 2022. `https://www.citedrive.com/overleaf` [Accessed: 06 Oct 2022].

[33] Fortinet Inc. 2022 clould security report, 2022. `https://www.fortinet.com/content/dam/fortinet/assets/analyst-reports/report-2022-cloud-security.pdf` [Accessed: 30 Sept 2022].

[34] Jinhyeok Jang, Dohyung Kim, Cheonshu Park, Minsu Jang, Jaeyeon Lee, and Jaehong Kim. Etri-activity3d: A large-scale rgb-d dataset for robots to recognize daily activities of the elderly, 2020. arXiv:2003.01920 [cs.RO].

[35] Youjin Jang, Inbae Jeong, Moein Younesi Heravi, Sajib Sarkar, Hyunkyu Shin, and Yong Han Ahn. Multi-camera-based human activity recognition for human-robot collaboration in construction. *Sensors*, 23, 2023.

[36] BeomJun Jo and Seong-Jong Kim. Comparative analysis of openpose, posenet, and movenet models for pose estimation in mobile devices. *Traitement du Signal*, 2022.

[37] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization, 2016. arXiv:1505.07427 [cs.CV].

[38] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. arXiv:1609.02907 [cs.LG].

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[40] Serre Lab. Hmdb: a large human motion database, 2011. https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/ [Accessed: 21 Oct 2022].

[41] Gunsik Lim, Beomseok Oh, Donghyun Kim, and Kar-Ann Toh. Human activity recognition via score level fusion of wi-fi csi signals. *Sensors*, 23(16), 2023.

[42] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. NTU RGBd 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, oct 2020.

[43] Tarik Makota. Ingest streaming data into amazon elasticsearch service within the privacy of your vpc with amazon kinesis data firehose, 2020. https://aws.amazon.com/blogs/big-data/ingest-streaming-data-into-amazon-elasticsearch-service-within-the-privacy-of-your-vpc-with-amazon-kinesis-data-firehose/ [Accessed: 25 Oct 2022].

[44] Samah A.F. Manssor, Zhengyun Ren, Rong Huang, and Shaoyuan Sun. Human activity recognition in thermal infrared imaging based on deep recurrent neural networks. In *2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–7, 2021.

[45] Sakorn Mekruksavanich, Ponnipa Jantawong, and Anuchit Jitpattanakul. Recognition of human activity from ecg and imu signals using deep learning networks. In *2022 IEEE Region 10 Symposium (TENSYMP)*, pages 1–5, 2022.

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. arXiv:1506.01497 [cs.CV].

[47] Mauro Ribeiro, Katarina Grolinger, and Miriam A.M. Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 896–902, 2015.

[48] Nayan B. Ruparelia. *Cloud Computing*. The MIT Press, 04 2016.

[49] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. arXiv:1801.0438 [cs.CV].

[50] Steve Scott. *AWS Certified Security Specialty Exam Guide: Build Your Cloud Security Knowledge and Expertise as an AWS Certified Security Specialist (SCS-C01)*. Packt Publishing Ltd, 2020.

[51] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition, 2019. arXiv:1805.07694 [cs.CV].

[52] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. arXiv:1212.0402 [cs.CV].

[53] Ahmad-Loay Sousi, Dalia Yehya, and Mohamad Joudi. Aes encryption: Study evaluation, 11 2020. `https://www.researchgate.net/publication/346446212_AES_Encryption_Study_Evaluation` [Accessed: 21 Oct 2022].

[54] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022.

[55] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. arXiv:1409.4842 [cs.CV].

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. arXiv:1512.00567 [cs.CV].

[57] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. arXiv:1905.11946 [cs.LG].

[58] Catherine Tong, Shyam A. Tailor, and Nicholas D. Lane. Are accelerometers for activity recognition a dead-end?, 2020. arXiv:2001.08111 [cs.CV].

[59] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950.

[60] W3C. Deflate compressed data format specification version 1.3. `https://www.w3.org/Graphics/PNG/RFC-1951` [Accessed: 19 June 2023].

[61] Mohammed Nazeh Abdul Wahid, Abdul Wahid Ali, Babak Esparham, and Mohamed Marwan. A comparison of cryptographic algorithms: Des, 3des, aes, rsa and blowfish for guessing attacks prevention. *Journal of Computer Science Applications and Information Technology, Symbiosis*, 2018.

[62] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition, 2020. arXiv:1908.07919 [cs.CV].

[63] Shuangquan Wang and Gang Zhou. A review on radio based activity recognition. *Digital Communications and Networks*, 9, 03 2015.

[64] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition, 2018. arXiv:1801.07455 [cs.CV].

[65] Zhi Yang, Kang Li, Haitao Gan, Zhongwei Huang, and Ming Shi. Hd-gcn:a hybrid diffusion graph convolutional network, 2023. arXiv:2303.17966 [cs.LG].

[66] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices, 2017. arXiv:1707.01083 [cs.CV].

[67] Yunhua Zhang, Hazel Doughty, Ling Shao, and Cees G. M. Snoek. Audio-adaptive activity recognition across video domains, 2022. arXiv:2203.14240 [cs.CV].