# 國立臺灣師範大學

# 資訊工程學系

# 碩士論文

指導教授：方瓊瑤　博士

以深度學習技術為基礎之線上人體動作辨識應用於室內移動型智慧機器人

Online Human Action Recognition Using Deep Learning for Indoor Smart Mobile Robots

研究生：謝日棠　撰

中華民國一百零九年七月

# 誌謝

當我的論文寫到這裡時，也代表著鳳凰花已經陸續的盛開了。首先，我要感謝一直在背後支持我的父母與家人。他們讓我可以在無經濟壓力的情況下攻讀碩士班的學業以及在我低潮時給予我鼓勵。

然而，要完成一本碩士班的論文所需要的不單單是時間，更加重要的是教授與學生之間的合作與討論。在這裡我要特別感謝方瓊瑤教授在我研究生生涯的期間給予我的指導與栽培，也感謝方瓊瑤教授提供了我許多到國外交流與學習的機會。在撰寫論文的期間，方瓊瑤教授總是用心且不厭其煩的指導我在論文上的該注意細節與寫作技巧，使論文可以更加完善。然而方瓊瑤教授不僅僅是在學業上給予我指導，在日常生活上處理事情的方法與態度，方瓊瑤教授也會給予我指導，使我獲益良多。再來，我要感謝陳世旺教授在實驗室開會時，時常都會給予我許多在實驗或論文上的建議。且在我學業或是論文上遇到困難時，都會非常熱情且認真的幫助我解決問題。然後也要感謝黃仲誼博士和羅安鈞博士抽空審查我的論文並蒞臨本論文口試且給予許多建議，使本論文改進完整。

另外我也要感謝實驗室的夥伴們后玲、簡佑如、林旭政、蔡妤涓、曾永權、徐秉琛、胡雅雯、江孟霖、柯皓中在研究期間給予我的幫助與陪伴，感謝你們協助拍攝資料庫影片，分擔實驗室的事物，以及陪伴我在實驗室的時光。最後我也要感謝廖俐智、Riki Otaki 和 Fredrik Nilsson 在學業上給我的幫助。謹以此論文獻給所有給予我幫助與鼓勵的人。

謝日棠謹致

國立臺灣師範大學資訊工程學系

中華民國 109 年 7 月

# 摘要

本研究提出一種以深度學習技術為基礎應用於室內移動型智慧機器人之線上人體動作辨識系統。此系統利用輸入的視覺資訊且在攝影機朝向目標人物移動的狀況下進行線上人體動作辨識，主要目的在提供智慧型人機互動除了聲控與螢幕觸控外更多的介面選擇。

本系統採用三種視覺輸入資訊，分別為彩色影像資訊、短期動態資訊以及人體骨架資訊。且在進行人體偵測時涵蓋五個階段，分別為人體偵測階段、人體追蹤階段、特徵擷取階段、動作辨識階段以及結果整合階段。本系統首先使用一種二維姿態估測方法用來偵測影像中的人物位置，之後利用 Deep SORT 追蹤方式進行人物追蹤。之後，在已追蹤到的人物身上擷取人體動作特徵以便後續的動作辨識。本系統擷取的人體動作特徵有三種，分別為空間特徵、短期動態特徵以及骨架特徵。在動作辨識階段，本系統將三種人體動作特徵分別輸入三種訓練好的神經網路(LSTM networks)進行人體動作分類。最後，將上述三個不同神經網路的輸出結果整合後作為系統的分類結果輸出以期達到最佳成效。

另外，本研究建立一個移動式攝影機下的人體動作資料庫(CVIU Moving Camera Human Action dataset)。此資料庫共計3646個人體動作影片，其中包含三個不同攝影角度的11種單人動作和5種雙人互動動作。單人動作包括站著喝水、坐著喝水、站著吃食物、坐著吃食物、滑手機、坐下、起立、使用筆記型電腦、直走、橫走和閱讀。雙人互動動作包括踢腿、擁抱、搬東西、走向對方和走離對方。此資料庫的影片也使用來訓練與評估本系統。實驗結果顯示，空間特徵之分類器的辨識率達96.64%，短期動態特徵之分類器的辨識率達81.87%，而骨架特徵之分類器的辨識率則為68.10%。最後，三種特徵之整合辨識率可達96.84%。

關鍵字: 線上人體動作辨識、室內移動行智慧機器人、移動式攝影機、深度學習、長短期記憶、雙向長短期記憶、強化時序長短期記憶、空間特徵、時序特徵、結構特徵。

# Abstract

This research proposes a vision-based online human action recognition system. This system uses deep learning methods to recognise human action under moving camera circumstances. The proposed system consists of five stages: human detection, human tracking, feature extraction, action classification and fusion. The system uses three kinds of input information: colour intensity, short-term dynamic information and skeletal joints.

In the human detection stage, a two-dimensional (2D) pose estimator method is used to detect a human. In the human tracking stage, a deep SORT tracking method is used to track the human. In the feature extraction stage, three kinds of features, spatial, temporal and structural, are extracted to analyse human actions. In the action classification stage, three kinds of features of human actions are respectively classified by three kinds of long short-term memory (LSTM) classifiers. In the fusion stage, a fusion method is used to leverage the three output results from the LSTM classifiers.

This study constructs a computer vision and image understanding (CVIU) Moving Camera Human Action dataset (CVIU dataset), containing 3,646 human action sequences, including 11 types of single human actions and 5 types of interactive human actions. Single human actions include drink in sit and stand positions, eat in sit and stand positions, play with a phone, sit down, stand up, use a laptop, walk straight, walk horizontal, and read. Interactive human actions include kick, hug, carry object, walk toward each other, and walk away from each other. This dataset was used to train and evaluate the proposed system. Experimental results showed that the recognition rates of spatial features, temporal features and structural features were 96.64%, 81.87% and 68.10%, respectively. Finally, the fusion result of human action recognition for indoor smart mobile robots in this study was 96.84%.

Keywords: Online human action recognition, indoor smart mobile robot, deep learning, long short-term memory, bi-directional long short-term memory, temporal enhancement long short-term memory, spatial feature, temporal feature, structural feature.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1 Introduction

## 1.1 Research Motivation

Indoor smart mobile robots have rapidly been adopted for human society and are widely used in public or private indoor spaces for guidance, entertainment, home service, security and so on. For example, a guidance robot such as Troika [1], shown in Figure 1.1 (a), moves around in the airport and provides directions and guidance for tourists. Entertainment robots such as Aibo [2], which is a dog-shaped entertainment robot as shown in Figure 1.1 (b), can be used to play with children or pets in the house. Home service robots such as Zenbo [3], shown in Figure 1.1 (c), are used to provide company to family members. Multifunctional smart robots such as Pepper [4], shown in Figure 1.1 (d), can be used as receptionists at offices and banks, home companions at home, and educational robots at schools, universities, and colleges.

These kinds of robots have a level of interaction and self-determination abilities, which are due to the “intelligence” of the robots. This intelligence is created through artificial intelligence techniques. Robots with intelligence are called smart robots.


(a) Troika [5] (b) Aibo [6] (c) Zenbo [7] (d) Pepper [8]

Figure 1.1 Indoor smart mobile robots (a) Troika (b) Aibo (c) Zenbo (d) Pepper

The aforementioned indoor smart mobile robots, such as Troika [1], Aibo [2], Zenbo [3], and Pepper [4], have already been released and used in houses, airports, stores, and other indoor spaces. These robots are respectively produced by Lucky-Goldstar (LG), a South Korean multinational electronics company; Sony, a Japanese

multinational conglomerate corporation; Asus, a Taiwan-based multinational computer and phone hardware and electronics company; and SoftBank Robotics, a holding company in the SoftBank Group. All of these robots mainly interact via voice commands. Zenbo can also interact via a touch screen.

In summary, indoor smart mobile robots are mainly interactive through the application of voice recognition systems and touch screen systems. Indeed, verbal commands and screen touching commands are direct and smart human-robot interactive techniques. However, voice recognition systems typically have limitations with respect to different languages, various accents and even speaking tone. A touch screen system limits the possible distance between the user and the robot. That is, a user must be close enough to touch the screen or to see the content of icons shown on the screen.

Vision-based recognition systems provide an alternative type of direct and smart human-robot interaction. The users interact with the robot through a vision-based human action recognition system. With this system, users are only required to perform a daily life action in front of the robot, and the robot is expected to see and recognise the action and then perform the corresponding reflection. For example, if a robot sees the user sits on a chair, then the robot can move to the user and provide the user some water and food. With this approach, users who speak different languages can smoothly interact with the robot. Further, because of the vision-based setting, the robot is capable of interacting with a human remotely. Thus, the barriers and limitations associated with voice recognition and touch screen systems can be solved by using a vision-based online human action recognition system. Such systems can therefore diversify human-robot interaction approaches for future robot products.

Moreover, many global market companies have a positive outlook on robot markets and have forecasted increases in the coming years in smart robots, indoor robots, mobile robots, service robots, and other robots. Therefore, robot markets, no doubt, will become a bull market of the world.

The smart robot market is a promising prospect according to research from Maximize Market Research, as shown in Figure 1.2 [9], where the number below the bar indicates the years. The number above the bar indicates the market value to the corresponding years, and the unit is billion USD. The research from Maximize Market Research has reported and forecasted the value of the smart robot market from 2017 to 2026. In 2017, the smart robot market was valued at USD 4.54 billion and the market is

expected to grow to USD 29.46 billion by 2026 at a Compound Annual Growth Rate (CAGR) of 23.1% over the forecast period from 2017 to 2026.

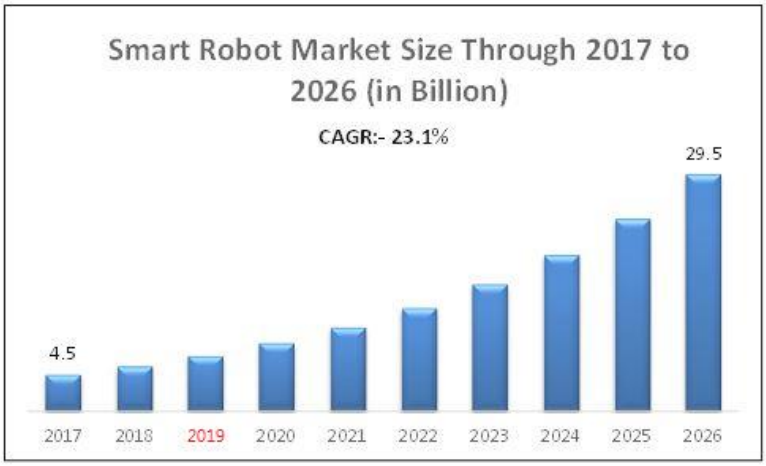



Figure 1.2 Smart robot market from 2017 to 2026 as reported by Maximize Market Research [9]

This research also reported and forecasted the value of the global indoor robot market from 2018 to 2026, as shown in Figure 1.3 [10], where the number below the bar indicates the year and colours indicate particular regions. The global indoor robot market is predicted to have a CAGR of 28.9% over the forecast period from 2018 to 2026.

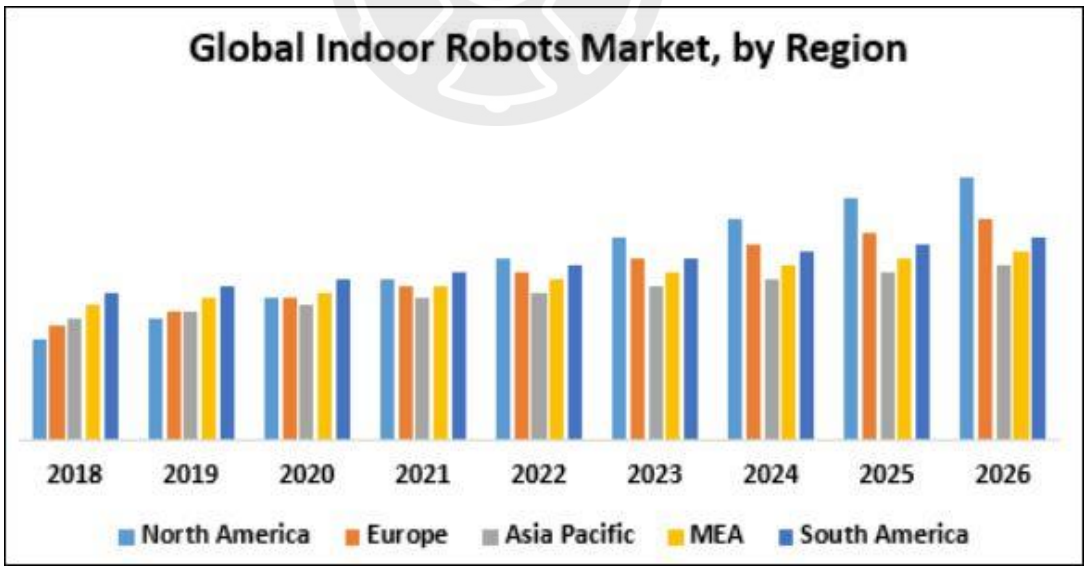
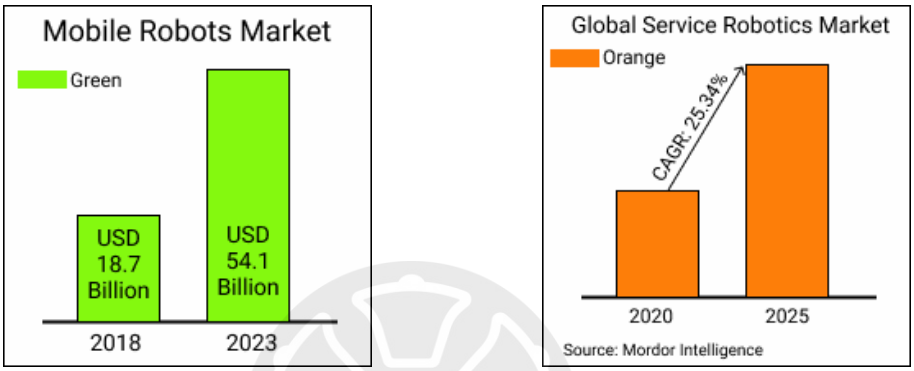



Figure 1.3 Global indoor robot market from 2018 to 2026 as reported by Maximize Market Research [10]

Markets And Markets reported and forecasted the value of global mobile robot market from 2018 to 2023, as shown in Figure 1.4 (a) [11], where the number below the green bar indicates years. The number in the green bar indicates the market value for the

corresponding years and the unit is billion USD. In 2018, the mobile robot market was valued at USD 18.7 billion and the market is expected to grow to USD 54.1 billion by 2023 at a CAGR of 23.71% over the forecast period from 2018 to 2023.

Mordor Intelligence [12] reported and forecasted the value of the global service robotics market from 2020 to 2025, as shown in Figure 1.4 (b), where the number below the orange bar indicates the years and the arrow indicates the CAGR during 2020 to 2025. The value of global service robotics market in 2019 was USD 14.39 billion and it is expected to grow to USD 63.80 billion with a CAGR of 25.34% over the forecast period.




(a) (b)

Figure 1.4 Global robotic market (a) global mobile robotics market from 2018 to 2023 as reported by Markets And Markets [11] (b) global service robotics market from 2020 to 2025 as reported by Mordor Intelligence [12]

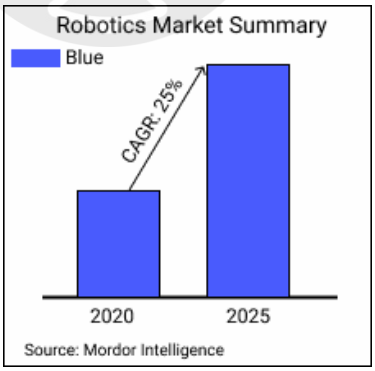



Figure 1.5 Robotics market summary from 2020 to 2025 reported by Mordor Intelligence [13]

The overall robotic market is shown in Figure 1.5 [13], where the number below the blue bar indicates the years. The arrow indicates the CAGR during 2020 to 2025. Mordor Intelligence [13] reported the value of the robotic market was USD 39.72 billion in 2019 and predicted it to have a CAGR of 25% over the forecast period from 2020 to 2025.

Furthermore, Mordor Intelligence also shows the overall robotics market growth rate during 2019 to 2024 by region, as shown in Figure 1.6 [13], where different colours indicate different growth rates. Specifically, green regions indicate high growth rates, yellow regions indicate medium growth rates and red regions indicate low growth rates. The colours cover over half the world. Undoubtably, robotics markets have a huge economic impact globally.

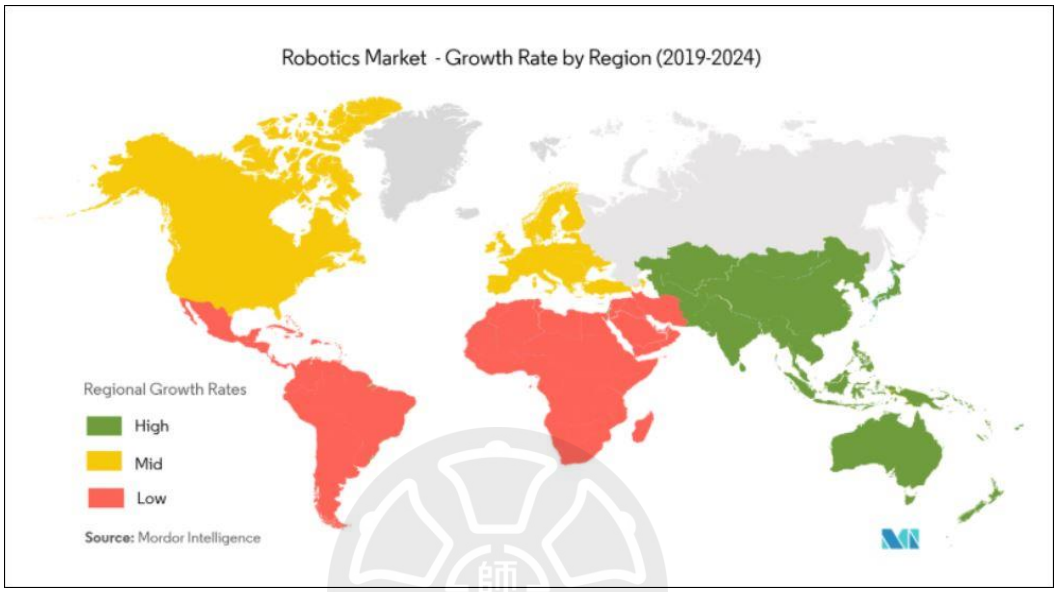



Figure 1.6 Robotics market growth rates by regions [13]

Indoor smart mobile robots seem to have a tremendous economic outlook and a high chance of bringing considerable economic benefit to many countries. With such high growth rates in the indoor smart mobile robot markets, it is clear such robots will be widely used in the foreseeable future. Therefore, a diversity of hardware and software products is necessary to satisfy different kinds of customer requirement. Here, hardware refers to the physical parts of the robots, such as the central processing unit, robot appearance, and monitor. The software refers to the abstract part of the robots, such as control systems, input recognition systems, output systems, and inference systems. Improvements in both hardware and software will increase the economic values of the robots. This research focuses on improving the input recognition system, which is part of the software.

Different types of input recognition systems process different kinds of input information and can result in different types of human-robot interactions. For example, a voice recognition system lets robots interact with humans via voice commands, a touch screen system lets robots interact with humans via screen touching, and a human action

recognition system lets robots interact with humans via human action commands.

This research develops a vision-based online human action recognition system for indoor smart mobile robots. The system is expected to let a robot recognise human actions while the robot is moving towards the user as well as recognise human actions online.

## 1.2 Background and Difficulty

Human action recognition has been a challenging computer vision problem in video analysis for decades. Methods of human action recognition can be divided into online and offline approaches. Offline methods classify human actions after obtaining the entire sequence. By contrast, online methods can classify actions from only a partial sequence. Both types of method classify the action of the current frame based on the information from previous frames. Only online methods have the characteristic of early action classification.

Online human action recognition can be done using a traditional machine learning approach or a deep learning approach. A machine learning example is Hoai and De la Torre's [Hoa12] proposed maximum-margin framework based on a structured output support vector machine (SVM) to achieve online action prediction. A deep learning example is De Geest and Tuyelaars' [De18] proposed two-stream feedback neural network built based on a recurrent neural network (RNN) with long-short term memory networks (LSTM) [Hoc97]. Both approaches are popular for solving the online human action recognition problem. However, this research utilises a deep learning method to build the online human action recognition system and tries to explore the characteristics of recurrent neural networks.

Most online action recognition systems [Hoa12] [De18] are designed to process video obtained from stationary camera videos. To design this kind of system, there are two main problems to solve:

(1) the transformation of real-world three-dimensional (3D) human action into two-dimensional (2D) video might cause object occlusion,

(2) the non-rigid body of the target person might cause difficulty in human tracking and recognition.

In addition, it is hard to simulate the vision of a mobile robot from stationary

camera videos. Mobile robots are capable of moving so a moving camera is required to simulate vision. In the rest of the article, human action videos recorded by stationary camera are called S-Videos and those recorded by moving camera are called M-Videos.

Developing an online action recognition system with M-Videos is more difficult than with S-Videos. For example, while the camera is moving, (1) the background is changing in every frame; (2) distances between target persons and the camera are changing; therefore, the size of the target person in the video is also changing; (3) illumination of each frame may not remain consistent; and (4) a moving camera may experience camera vibration.

Trends in smart mobile indoor robots are emerging, and robots will service elders in their house, help humans in the airport, and be used in other indoor spaces. Further, as mentioned above, interaction through human action commands is a direct and smart human-robot interactive approach. In response to these future trends, this research proposes a vision-based online human action recognition system using a deep learning method to recognise human actions under moving camera circumstances for indoor smart mobile robots.

## 1.3 Research Contribution

This research has three main contributions: the collection of an M-Video dataset of human actions, a human action recognition system to recognise human actions under moving camera circumstances, and a proposed method that simultaneously utilises multiple types of feature information to recognise human actions.

(1) *M-Video dataset*

Many human action datasets have been established and provided by different groups and universities for human action recognition experiments. However, many of these datasets, including NTU RGB + D 120 Dataset (NTU) [14], Berkeley Multimodal Human Action Dataset [15], KTH-Dataset [16], SBU Kinect Interaction Dataset [17], and PKU Multi-Modality Dataset (PKU-MMD) [18], are S-Videos. M-Videos datasets have rarely been established.

Therefore, this research collects an M-Video dataset called computer vision and image understanding (CVIU) Moving Camera Human Action dataset. The human actions are recorded while the camera is moving towards the target persons. Chapter 4

provides details about this dataset.

(2) *Human action recognition system under moving camera circumstances*

Most research in this field focuses on stationary camera circumstances. However, there has been little development of human action recognition systems under moving camera circumstances, which the current research aims to do. The proposed system applies human detection and human tracking to the target persons, and then extracts three types of features from the target persons to provide respective LSTM classifiers to analyse the human actions. Finally, a fusion method is used to integrate these three output results to determine the final classification for the human action. Experimental results show that the proposed model is robust and stable.

(3) *Utilise three kinds of feature information simultaneously*

The developed system recognises human actions using three kinds of feature information simultaneously: features obtained from red-green-blue (RGB) colour images, features obtained from optical flow and features generated from human skeletal joints. Each type of feature has a tailored LSTM model and an output result. We then use fusion methods to integrate these three output results to improve the human action recognition rate. Experimental results show that these three kinds of features can cover each other's deficiencies.

## 1.4 Thesis Framework

This thesis comprises 5 chapters. Chapter 1 introduces the research motivation, research background and difficulty. Chapter 2 discusses related works. Chapter 3 illustrates and details the system flowchart. Chapter 4 presents experimental results to show the improvement of the proposed system. Finally, Chapter 5 concludes this research and presents future works.

# Chapter 2 Related Work

This chapter discusses some relevant research on human action recognition. The first part introduces various types of features, including spatial and temporal features. The second part introduces human action classifiers.

## 2.1 Features of Human Action Recognition

Extracting suitable features to represent different actions is key to achieving human action recognition. Spatial and temporal features are those associated with space and time, respectively. Generally speaking, spatial features can be extracted in one frame whereas temporal features can be extracted from at least two successive frames.

(1) *Spatial Features*

Skeletal joints are one type of spatial feature. Many datasets have proposed skeletal joint information for researchers, such as NTU Dataset [14], PKU-MMD Dataset [18], and SBU Kinect Interaction Dataset [17]. NTU [14] and PKU-MMD Dataset [18] provide 25 3D-location joints for each person. SBU Kinect Interaction Dataset [17] provide 15 3D-location joints for each person. Many researchers [Han18] [Jun18] [Sha16] [Son18] [Tu18] [Li17] [Liu18] have used skeletal joints as features to classify human actions, although some have used skeletal joints provided by the established datasets and some have extracted their own data.

Skeletal joints can be preprocessed to increase their quality as features. Jun and Choe [Jun18] presented data-augmentation methods, such as tilting, flipping, and scale variation on the skeletal joints, to enlarge their training dataset. Tu *et al.* [Tu18] proposed an LSTM auto-encoder model (LSTM-AE) to eliminate noise and preserve the whole action representation of the skeletal joints. Song *et al.* [Son18] proposed a spatial and temporal attention model to detect and recognise human actions. They also preprocessed the skeletal joints to maintain consistency for joint position and different perspectives. They smoothed each skeletal joint position to decrease the impact of noise before human action recognition and implemented an attention-based model to enhance the important skeletal joints.

Skeletal joints can be also used to extract higher-level features. Li *et al.* [Li17]

calculated the Euclidean distance between each pair of skeletal joints, and the area of the triangle region among three neighbouring skeletal joints as higher-level features of human action classification.

Liu *et al.* [Liu18] proposed a tree-structure based traversal method to represent the structure of skeletal joints. This kind of representation links neighbouring skeletal joints to enhance their interdependency.

Soomro *et al.* [Soo19] proposed an online action localization and prediction system. They extracted individual skeletal joints using a Convolutional Pose Machines (CPM) [Wei16] method. Moreover, they proposed a high level structural information method to reduce the influence of noise by smoothing the locations of obtained skeletal joints, and to minimise the displacements of joint locations by scaling the height of the skeletal joints.

Colour/intensity information is another type of spatial feature extracted by various methods. Some researchers [Ull18] [De18] [Ouy19] [Hua19] [You19] [Goe18] [Du18] [Liu19] have used neural network methods, and others [Ni11] [Liu10] have used traditional image processing methods.

Ni and Xu [Ni11] proposed a statistical model based on sparse representation of space-time features to recognise human actions. This model uses the Harris3D detector to find the point of interest in space-time, and then applies a Histogram of Gradients (HOG) descriptor to extract the spatial features. Liu *et al.* [Liu10] proposed an action recognition framework based on multiple features. The proposed method uses Cuboids [Dol05] and 2D Scale-Invariant Feature Transform (SIFT) to extract local spatial features. Moreover, a frame differencing method is implemented to focus on the region of interest and 2D Gabor filters are applied to extract global spatial features.

Ullah *et al.* [Ull18] proposed a human action recognition model using a bi-directional LSTM model (BiLSTM) [Sch97]. The proposed model extracts spatial features from the last fully connected layer of a pre-trained convolutional neural network (CNN) model, AlexNet [Kri12]. Ouyang *et al.* [Ouy19] proposed a network consisting of a 3D CNN model [Tra15] and an LSTM model to recognise human actions. In this architecture, they split an action sequence into 25 clips and randomly select 16 frames in each clip. These selected frames are resized into $112 \times 112$ pixels to be input into the 3D CNN. The output of the last fully connected layer of the 3D CNN is defined as the spatial features.

Huang *et al.* [Hua19], You and Jiang [You19], Liu *et al.* [Liu19], and Goel *et al.* [Goe18] developed online systems. Huang *et al.* [Hua19] proposed an online action detection and prediction model based on a convolutional recurrent neural network (RNN). In this model, spatial features are extracted from the output of the last convolutional layer of a pre-trained Visual Geometry Group (VGG)-16 model [Sim14]. The feature dimensions are then reduced by employing a $1 \times 1$ convolutional layer.

You and Jiang [You19] proposed a deep neural network model, Action4DNet, to recognise human actions. This model uses 3D CNNs to extract lower-level spatial features of each person. These extracted features are passed through an attention model [Bah14] and a global max-pooling layer [Lin13] to extract higher-level spatial features.

Goel *et al.* [Goe18] proposed an online human activity detection algorithm using support vector machine (SVM). To extract spatial features, they proposed a Person-Centred CNN (PC-CNN) method. PC-CNN first uses a Single Shot Multibox Object Detector (SSD) [Liu16] to detect persons. Next, the regions of detected persons are cropped and resized into $224 \times 224$ pixels. Finally, the resized regions are sent into a ResNet-152 [He16] network to extract spatial features from the last flatten layer.

(2) *Temporal Features*

LSTM networks are powerful for learning long-term dependencies and modelling sequential data. Moreover, LSTM networks can solve the problem of vanishing gradients associated with the fundamental network structure, RNN, in the training stage. Therefore, many researchers [Li17] [Liu17] [Tu18] [Son18] [Jun18] [Han18] [Liu18] [Liu19] [Ull19] [De18] [You19] [Ull18] [Du18] [Ouy19] have adopted LSTM models to extract temporal features.

Song *et al.* [Son18] proposed a spatial and temporal attention model to exploit the importance of each frame. In this model, they added a temporal attention model, which can define the importance level of each frame, to improve the LSTM model. Liu *et al.* [Liu19] first passed skeletal joints through convolution operations to extract richer temporal statistics and then input these into the LSTM model to extract temporal features.

De Geest and Tuyelaars [De18] proposed a two-stream LSTM feedback network to detect and classify actions. This network used a two-stream LSTM model to extract temporal features. One LSTM stream is used to interpret the input frames, and the other is used to capture the temporal dependencies. Ullah *et al.* [Ull18] proposed an action

recognition model based on a bidirectional LSTM (BiLSTM) network. They regularly sampled one-sixth of the frames in a sequence and input these into the BiLSTM model to extract the temporal dependencies. The goal of frame sampling is to reduce the computational complexity of the proposed system.

Ouyang *et al.* [Ouy19] proposed a human action recognition network using both a 3D CNN model and an LSTM model. In this architecture, they passed the input sequences through the 3D CNN to enhance the temporal feature representation. They then sent the enhanced temporal features to the LSTM model to extract the final temporal features.

Optical flow is a type of temporal feature that is widely used to observe short-term dynamics. Jagadeesh and Patil [Jag16] addressed a vision-based human action detection and recognition method using optical flow. This method calculates optical flow between frames and then converts the calculated optical flow data to binary images. Finally, they applied the HOG descriptor to the optical flow to extract temporal features. Ullah *et al.* [Ull19] proposed an activity recognition network based on multilayer LSTM models. In this network, they used a pre-trained optical flow detection neural network, FlowNet2 [Ilg17], to obtain optical flow. They extracted the feature maps from the final convolutional layers of FlowNet2 [Ilg17] and used a global average pooling to obtain temporal features.

In summary, the above research adopted two kinds of spatial features: human skeletal joints and colour information. Importantly, skeletal joints can be used to roughly describe the structure of human poses whereas colour information contains more details of human poses.

As mentioned above, colour information can be extracted by neural network methods and traditional image processing methods. Using traditional image processing methods, researchers should decide feature extraction methods themselves. However, the results of selected methods are expected and may be unsuitable to classify human actions. By contrast, with neural network methods, researchers have a higher probability of finding unexpected and suitable spatial features since the neural networks can learn automatically. Therefore, this research adopts neural network methods to extract spatial features.

Moreover, this research adopts two kinds of temporal extraction methods for human action sequences: optical flow methods and the LSTM network. Optical flow

methods can capture short-term dynamics and LSTM networks can capture long-term dynamics. By knowing the temporal dynamics of the sequences, the system can discover the discrimination of each human action in the temporal domain.

## 2.2 Models of Human Action Recognition

In recent years, deep learning methods have been widely studied and developed for human action recognition. Many researchers [Li17] [Liu17] [Tu18] [Son18] [Jun18] [Han18] [Liu18] [Liu19] [Ull19] [De18] [You19] [Hua19] [Ull18] [Cio18] [Du18] [Ouy19] [Cha19] have used deep learning methods to develop their human action recognition models. Some of these studies use the offline approach [Cha19] [Wan16] [Li17] [Ijj14].

Wang *et al.* [Wan16] proposed a spatio-temporal features representation method, Joint Trajectory Maps (JTM), to use with the 2D CNN model, AlexNet [Kri12], to recognise human actions. JTM features are generated by three Cartesian planes of human action trajectories and are sent into an AlexNet [Kri12] model to recognise human actions. However, 2D CNN models could not learn temporal information, so the information of human action temporal dynamics may be lost.

Chang *et al.* [Cha19] proposed a 3D VGG-13 model to recognise human actions. The authors replaced the 2D CNNs in the original VGG-13 with 3D CNNs to construct the 3D VGG-13 network. 3D CNNs are used to learn the spatial and temporal features, but they focus on learning local spatial and temporal features of sequences. Such local information may be easily affected by noise and there might be a risk relating to lost global information of whole sequences.

On the other hand, some researchers [Liu17] [You19] [Liu19] [De18] have developed their human action recognition system using online approaches. De Geest and Tuyelaars [De18] developed a two-stream feedback network to detect and classify human actions. The two-stream feedback network consists of an upper stream LSTM network, a lower stream LSTM network, and a fully connected layer. The upper stream LSTM is used to interpret the input information. The lower stream is used to capture temporal information. Moreover, the fully connected layer is used to project the features into the action classes. In this study, the intensities of RGB colour model are the input information and are first analysed by a CNN model. The results are then sent into the

two-stream feedback network to detect and classify human actions. However, this study ignored other kinds of features, such as skeletal joints or short-term dynamic features. We think each kind of feature can be uniquely analysed for human actions, which would combine to increase the accuracy of human action recognition.

Liu *et al.* [Liu19] proposed a Multi-Modality Multi-Task RNN to classify and forecast human actions. The human action forecast aimed to find the start and end points of an action. This network is a two-stream system. The first stream processes the skeletal joint information, and the second processes the colour intensity information. These two types of information are first processed by using convolutional layers individually. Then, the features extracted from the convolutional layers are sent into a deep LSTM network with two subtask networks for action classification and forecast, respectively. The deep LSTM network alternately stacks three LSTM layers and three fully connected layers, with a fully connected layer with softmax at the end. The two subtask networks mainly consist of fully connected layers. However, this proposed network ignores the uniqueness of each type of input information. Different kinds of input information have corresponding suitable networks, such as various stacks of LSTM layers and various orders of fully connected layers and LSTM layers. We believe that a tailored network for various types of input information can get more meaningful results.

In summary, this research adopts three types of features to analyse various aspects of human actions: colour intensity, short-term dynamic information, and skeletal joints. Our proposed system is based on LSTM networks. Compare with 3D CNNs, which have weaknesses related to analysing global information, LSTM networks is superior for learning global temporal features. LSTM networks treat each frame of the input sequence as one input vector and analyse the relationship of all input vectors directly. This means that the temporal dependencies of sequences can be enhanced. Additionally, this research tries to implement corresponding tailored LSTM networks for different characteristics of features.

# Chapter 3 Online Human Action Recognition System

This chapter discusses the online human action recognition system flowchart proposed by this study. We briefly introduce the purpose of this research and then illustrate and detail the system flowchart.

## 3.1 Research Purpose

This research aims to provide diverse human-robot interaction options for indoor smart mobile robots and to overcome the limitations of voice recognition systems and touch screen systems. We aim to solve the camera moving and online recognition problems. This research develops a system using neural networks due to the recent development and robustness of deep learning techniques. That is, this research proposes an online human action recognition system using deep learning techniques. By analysing the human actions through the proposed system, the actions can be successfully recognised and indoor smart mobile robots can give the corresponding reflection to users.

## 3.2 System Flowchart

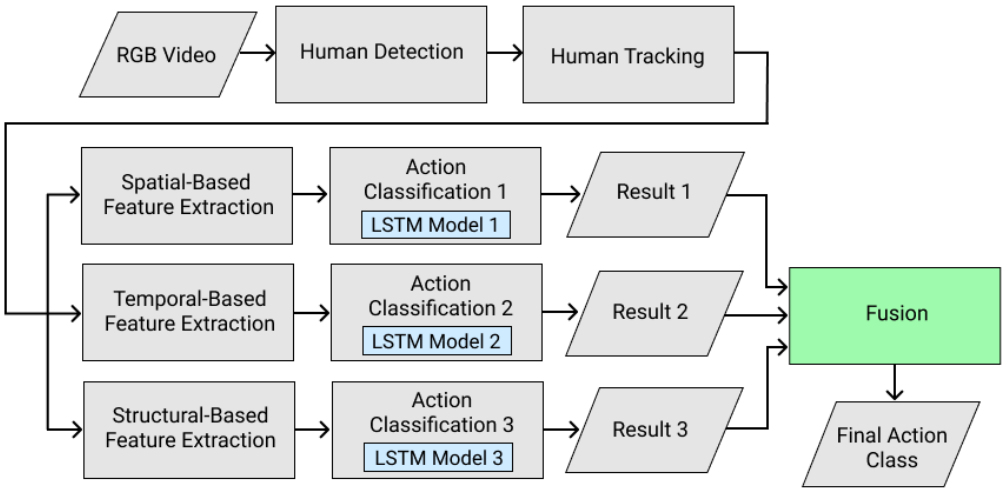



Figure 3.1 Flowchart of online human action recognition system

The system flowchart is shown in Figure 3.1. This flowchart has five stages: human detection, human tracking, feature extraction, action classification, and fusion. Note that feature extraction involves three types of features, spatial, temporal features and structural, and they each have their own classifier.

After the RGB videos are input into the system, the persons existing in the video are detected and tracked. Here, the detected persons are called target persons. Next, three kinds of features are extracted from the regions of target persons in each frame. These features are then input into their corresponding action classifiers. Finally, the outputs of the three action classifiers are fused together to determine the final human action.

### 3.2.1 Human Detection

The system adopts OpenPose, a real-time multi-person 2D human pose estimator proposed by Cao *et al.* [Cao19], to detect humans because it has high speed and accuracy. Figure 3.2 compares OpenPose with other human pose estimators proposed in the literature, including Alpha-Pose [Fan17], Mask R-CNN [He17], PersonLab [Pap18], and METU [Koc18]. In Figure 3.2, the horizontal axis indicates the frames per second (FPS) of a video where each frame contains three target persons. The vertical axis indicates the accuracy (mean average precision) of the results of the human pose estimators. The OpenPose estimator [Cao19] use in this research is highlighted in red triangles.

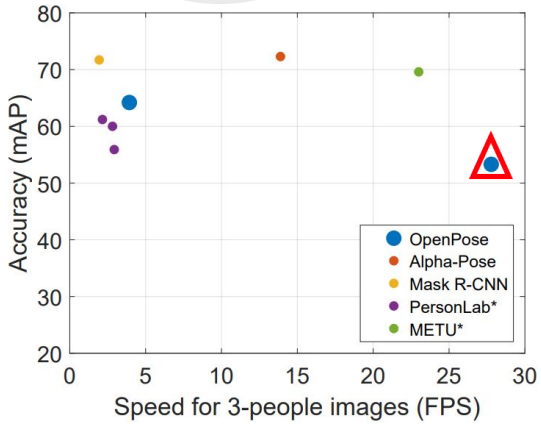
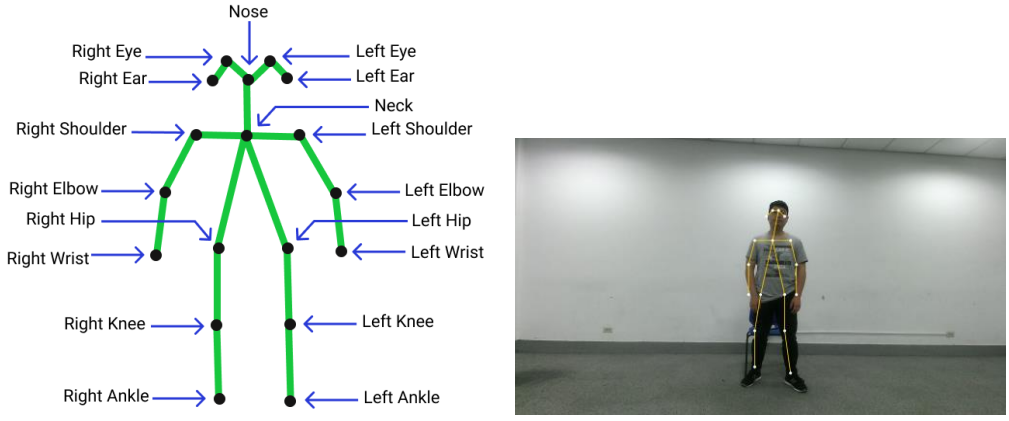



Figure 3.2 Comparison of human pose estimators [Cao19]

From Figure 3.2, the OpenPose estimator [Cao19] has the highest FPS, which is the most important property of online systems. Although the OpenPose estimator [Cao19] sometimes fails to detect all the skeletal joints, this shortcoming does not affect the human detection results. Moreover, our system will fill these missing skeletal joints

to improve the OpenPose estimator [Cao19] in the human tracking stage.




(a) (b)

Figure 3.3 Human skeletal joints (a) location of joints (b) result of joints extraction

(a) (b)

Figure 3.4 Results of human detection (a) completed joint extraction (b) incomplete joint extraction

The OpenPose estimator [Cao19] can extract 18 human skeletal joints for each person. These skeletal joints are two hips, two knees, two ankles, two shoulders, two elbows, two wrists, two ears, two eyes, a nose, and a neck, as shown in Figure 3.3 (a). An example of the results of joint extraction is shown in Figure 3.3 (b). By using the skeletal joints information, the system can enclose and detect the human successfully. The human detection results are shown in Figures 3.4 (a) and (b), which respectively show examples of complete and incomplete extraction. One can observe that the proposed system can detect the human, whether or not skeletal joints are fully extracted.

## 3.2.2 Human Tracking

After the human detection stage, this system uses a Deep Simple Online and Realtime Tracking (Deep SORT) method proposed by Wojke *et al.* [Woj17] to track each person in the input video. Some examples of human tracking results are shown in Figure 3.5, where the symbols shown above the bounding boxes, e.g., P-1, P-2, indicate the

person index of the target person. The green and blue bounding boxes show the results of human detection. One can observe that Deep SORT [Woj17] correctly tracks the humans.

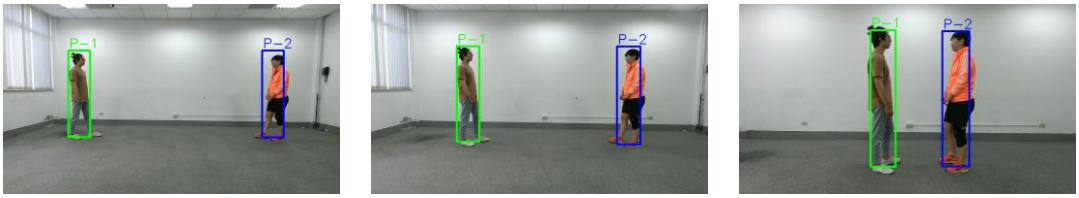



Figure 3.5 Results of human tracking

As mentioned above, the OpenPose estimator [Cao19] can extract 18 human skeletal joints for each person. However, the skeletal joints on the head are removed in this study because they are not as important for human action detection, and they are easily detected incorrectly. Therefore, the 18 skeletal joints are reduced to 13 joints in this stage, as shown in Figure 3.6 (a). The detection result is shown in Figure 3.6 (b). After the skeletal joint reduction, some missing skeletal joints are filled in at the human tracking stage.

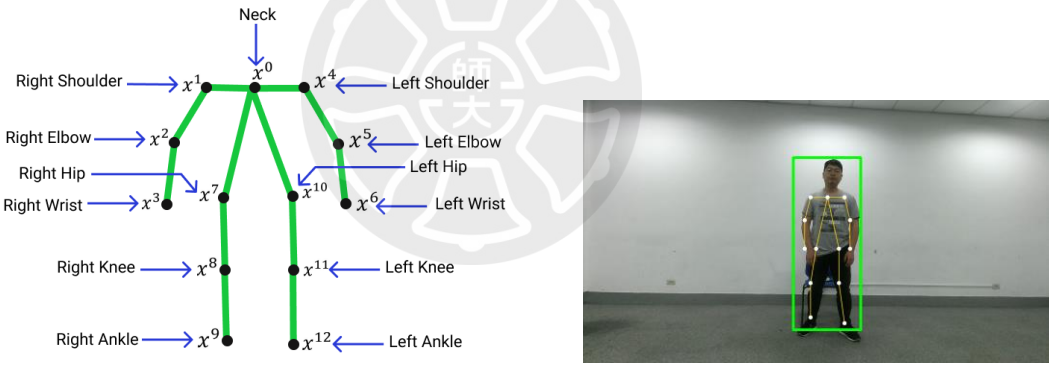



Figure 3.6 Reduced skeletal joints (a) skeletal joints without head joints (b) detection result of skeletal joints without head joints

Two approaches are used to fill the missing joints. Assume a missing joint $x_i$ has not been detected at frame $i$. Then, we have the following cases.

Case (1): The neck joint, $x_i^0$, is found at frame $i$. A missing joint $x_i$ can be predicted by the relative difference between the neck joint and its corresponding joint at frame $i-1$, $(x_{i-1} - x_{i-1}^0)$, as shown in Equation (1).

$$x_i = x_i^0 + (x_{i-1} - x_{i-1}^0) \times S \quad (1)$$

where $i$ indicates frame number, $x_i$ indicates a missing joint at frame $i$, and $x_{i-1}$ indicates the corresponding detected joints of the missing joint $x_i$ at frame $i-1$. Note that $x_{i-1}$ is detected and not a missing joint. Symbols $x_i^0$ and $x_{i-1}^0$ indicate the neck

joints at frames $i$ and $i-1$, respectively.

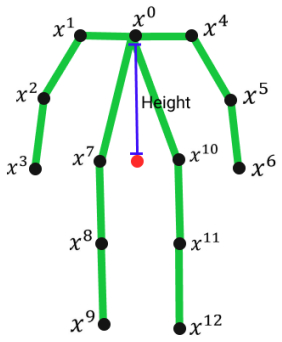
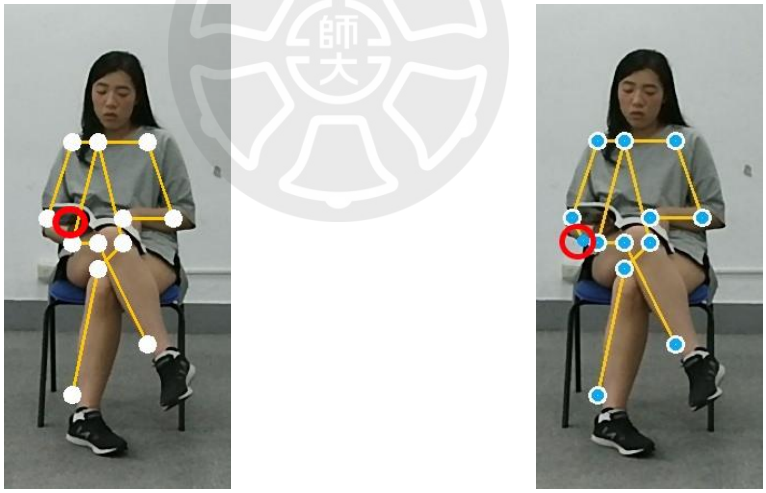



Figure 3.7 Schematic diagram of body height

In Equation (1), $S$ is defined as $S = \frac{H_i}{H_{i-1}}$, where $H_i$ and $H_{i-1}$ are the body heights in the frames $i$ and $i-1$, respectively. The body height is the Euclidean distance between the neck joint and the centre between the hip joints, illustrated by the red point in Figure 3.7. Thus, $S$ can maintain the consistency of the human height between frame $i$ and $i-1$. The camera moving problem can be fixed partially by considering the scale change of the same person in two successive frames.



(a) (b)

Figure 3.8 An example of joint filling (a) a missing joint (b) result of joint filling

Figure 3.8 shows an example of joint filling. In Figure 3.8 (a), the white points are the extracted skeletal joints. However, the right wrist joint has not been successfully extracted, as highlighted by a red circle. Figure 3.8 (b) shows the result of joint filling. In Figure 3.8 (b), the white circles indicate the original extracted skeletal joints, and the blue points indicate the filled joints. Clearly, the missing right wrist joint has been filled, as highlighted by a red circle.

Case (2): The neck joint, $x_i^0$, is not found at frame $i$. A missing joint $x_i$ can be predicted by the relative difference between its corresponding missing joints at frames $i-1$ and $i-2$, $(x_{i-1} - x_{i-2})$, as follows:

$$x_i = x_{i-1} + (x_{i-1} - x_{i-2}) \quad (2)$$

where $i$ indicates frame number, $x_i$ indicates the missing joints at frame $i$, and $x_{i-1}$, and $x_{i-2}$ indicate the corresponding detected joints of the missing joints $x_i$ at frames $i-1$ and $i-2$, respectively. Note that $x_{i-1}$ and $x_{i-2}$ are not missing. The difference between frame $i-1$ and $i-2$, $(x_{i-1} - x_{i-2})$, is used to determine the moving direction of joints to predict the missing joints at frame $i$.

Similarly to Figure 3.8, Figure 3.9 illustrates an example of joint filling. In this example, only three joints are detected successfully, and the others, including the neck joint, have not been extracted, as shown in Figure 3.9 (a). Figure 3.9 (b) shows the result of joint filling. In this situation, the system may obtain some joint information from the current frame; therefore, the degree of similarity between the filled joints and the real joints is lower. However, the joint filling step is still helpful for the following human action recognition stage.

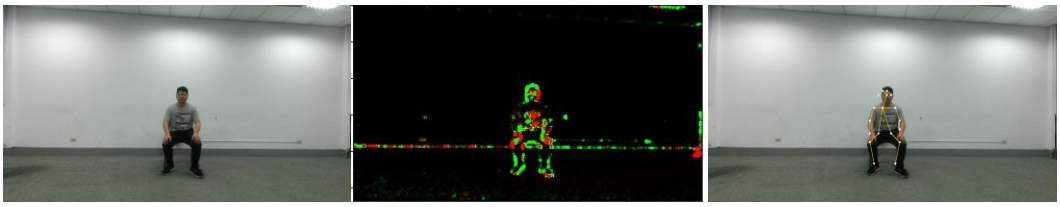
(a) (b)

Figure 3.9 An example of filling the missing joints (a) the missing joints (b) result of joint filling

## 3.2.3 Feature Extraction

As mentioned above, three types of features, spatial, temporal and structural, are used to distinguish human actions and provide information for the action classification stage. The spatial features are extracted from RGB colour images, as shown in Figure 3.8 (a). Temporal features are extracted from optical flow, as shown in Figure 3.8 (b).

Structural features are extracted from human skeletal joints, as shown in Figure 3.8 (c).



(a) (b) (c)

Figure 3.10 Input information (a) RGB colour images (b) optical flow (c) human skeletal joints

Both spatial and temporal features are extracted by the pre-trained CNN model, InceptionV3, which was proposed by Szegedy *et al*. [Sze16] in 2016. Table 3.1 (a) outlines the InceptionV3 architecture, including the input size and patch size of every layer. Specifically, the system extracts human action features from the output of the final pool layer, which has dimensions of $1 \times 1 \times 2048$. Table 3.1 (b) shows the evaluation results comparing InceptionV3 with other networks such as PReLU [He15], BN-Inception [Iof15], VGGNet [Sim14], and GoogLeNet [Sze15] proposed by Szegedy *et al*. One can observe that InceptionV3 has the lowest error rate for both Top-1 error and Top-5 error. Further, InceptionV3 is pre-trained on the ImageNet dataset.

Note that the system crops and resizes the input frames before extracting spatial and temporal features. In the cropping step, the system broadens the bounding box by 100 pixels in both left and right and 150 pixels in both top and bottom to increase the spatial information.

Table 3.1 InceptionV3 (a) outline of InceptionV3 architecture (b) evaluation results comparing InceptionV3 with other models [Sze16]

(a)

| type | patch size/stride or remarks | input size |
|---|---|---|
| conv | 3×3/2 | 299×299×3 |
| conv | 3×3/1 | 149×149×32 |
| conv padded | 3×3/1 | 147×147×32 |
| pool | 3×3/2 | 147×147×64 |
| conv | 3×3/1 | 73×73×64 |
| conv | 3×3/2 | 71×71×80 |
| conv | 3×3/1 | 35×35×192 |
| 3×Inception | Inception modules | 35×35×288 |
| 5×Inception | Inception modules | 17×17×768 |
| 2×Inception | Inception modules | 8×8×1280 |
| pool | 8 × 8 | 8 × 8 × 2048 |
| linear | logits | 1 × 1 × 2048 |
| softmax | classifier | 1 × 1 × 1000 |

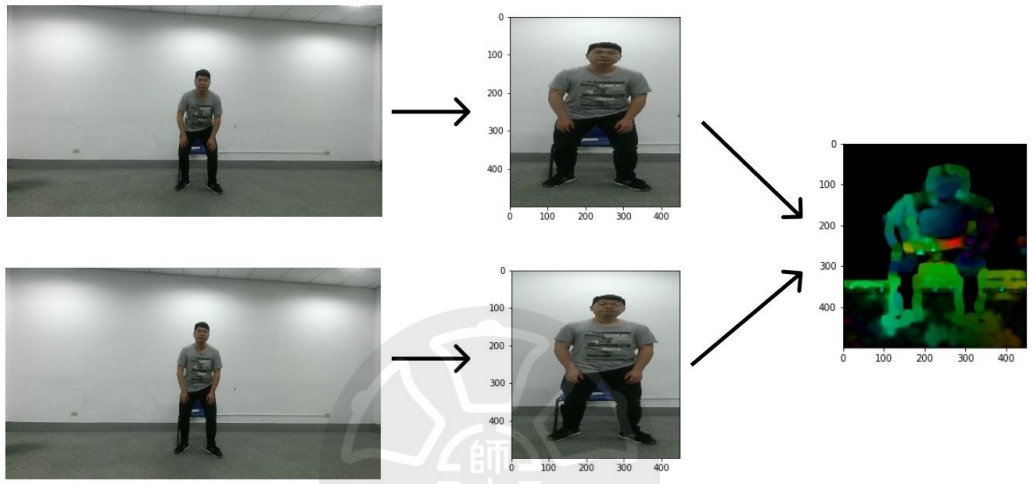
(b)

| Network | Models Evaluated | Crops Evaluated | Top-1 Error | Top-5 Error |
|---|---|---|---|---|
| VGGNet | 2 | - | 23.7% | 6.8% |
| GoogLeNet | 7 | 144 | - | 6.67% |
| PReLU | - | - | - | 4.94% |
| BN-Inception | 6 | 144 | 20.1% | 4.9% |
| Inception-v3 | 4 | 144 | **17.2%** | **3.58%*** |

Once the system crops the target persons, the cropped human region is resized into 500×450 pixels and sent into InceptionV3 [Sze16] for spatial feature extraction. Note that the cropped human region contains one person if only one person appears in a frame,

but it contains two persons if two persons appear in that frame. The cropped and resized human regions are called CR regions hereinafter. The system calculates the Farneback optical flow using two successive CR regions, and sends it into another InceptionV3 [Sze16] to extract temporal features.

Cropping and resizing the human region can partially fix the camera moving problem because cropping can force the system to focus on the target persons, and resizing can make the human regions consistent in all frames. Moreover, resizing the cropped human regions lets them fit the input shape of InceptionV3 [Sze16].



(a) (b) (c)

Figure 3.11 An example of optical flow calculation (a) two successive input frames (b) their corresponding CR regions (c) optical flow

Figure 3.11 shows an example of the process to obtain optical flow. Figures 3.11 (a) and (b) show two successive input frames and their corresponding CR regions, respectively. Figure 3.11 (c) shows the optical flow obtained by those successive CR regions. The arrows between Figures 3.11 (a), (b) and (c) indicate the processing direction. In summary, each frame can extract a $1 \times 1 \times 2048$ dimension spatial feature vector and two successive frames can extract a temporal feature vector of the same size. Moreover, each input sequence with $N$ frames can construct a feature map whose size is $N \times 2048$.

Structural features are obtained by calculating the relationship between each pair of skeletal joints. As mentioned above, each person has 13 skeletal joints that can be extracted. Thus, single human actions and interactive human actions by two persons respectively contain 13 and 26 skeletal joints in each frame. However, the system preserves sufficient memory space to record 26 skeletal joints in each frame, whether

the frame has one or two persons appearing. The system applies zero-padding to frames containing under 26 skeletal joints for the purpose of preparing information for structural feature extraction.

Next, the system calculates two kinds of distances on pairwise skeletal joints and concatenates them to be the structural features. One is the Manhattan distance (1-norm) and the other is the Euclidean distance (2-norm). In each frame, the system can calculate $2 \times C_2^{26} (= 650)$ 1-norm features and $1 \times C_2^{26} (= 325)$ 2-norm features for pairwise skeletal joints. Especially, 1-norm features calculate the location difference of pairwise skeletal joints on x-axis and y-axis respectively. Concatenating these features, the system can obtain $3 \times C_2^{26} (= 650 + 325 = 975)$ features. Moreover, each input sequence with $N$ frames can construct a feature map whose size is $N \times 975$.

Figure 3.12 shows two examples of the visualization results of spatial, temporal and structural feature maps. The human action in these examples, as shown in Figure 3.12 (a), is "walk toward to each other". The two sequences each contain 20 ($N = 20$) frames. Figures 3.12 (b), (c) and (d) show their corresponding spatial (green), temporal (purple), and structural (blue) feature maps, respectively. The horizontal axis indicates the dimension of feature vectors and the vertical axis indicates frame numbers. In particular, the structural feature maps have a second horizontal axis on the bottom, which shows 1-norm features (blue) from 0 to 650 and 2-norm features (red) from 650 to 975. The shade of colours in these feature maps indicate the magnitude of the extracted features. The corresponding ruler is shown on the right side of the feature maps, indicating that smaller values have a lighter colour. In spatial and temporal feature maps (see Figures 3.12 (b) and (c)), if the values are greater than one, they are represented in red.

Figure 3.13 shows another two examples of the visualization results of spatial, temporal and structural feature maps, this time for the drink in stand position, as shown in Figure 3.13 (a). Similarly to above, Figures 3.13 (b), (c) and (d) show the corresponding spatial (green), temporal (purple) and structural (blue) feature maps, respectively.

From these feature maps, one can observe that similar human actions have similar values of features and vice versa. This kind of characteristic can lead the classifiers to more easily obtain successful classification results.

The structural feature maps contain information about the relationship between

skeletal joints for both single and interactive actions. For example, in the feature maps of the action “walk toward each other” shown in Figure 3.12 (d), the values of the features are slowly decreasing from time step 0 to 20. This kind of variation means that the skeletal joints are getting closer, which matches the action. By contrast, in the feature maps of the action “drink in stand position” shown in Figure 3.13 (d), the values of the features barely change from time step 0 to 20. This kind of variation means that the skeletal joints only minorly change, which matches the action.

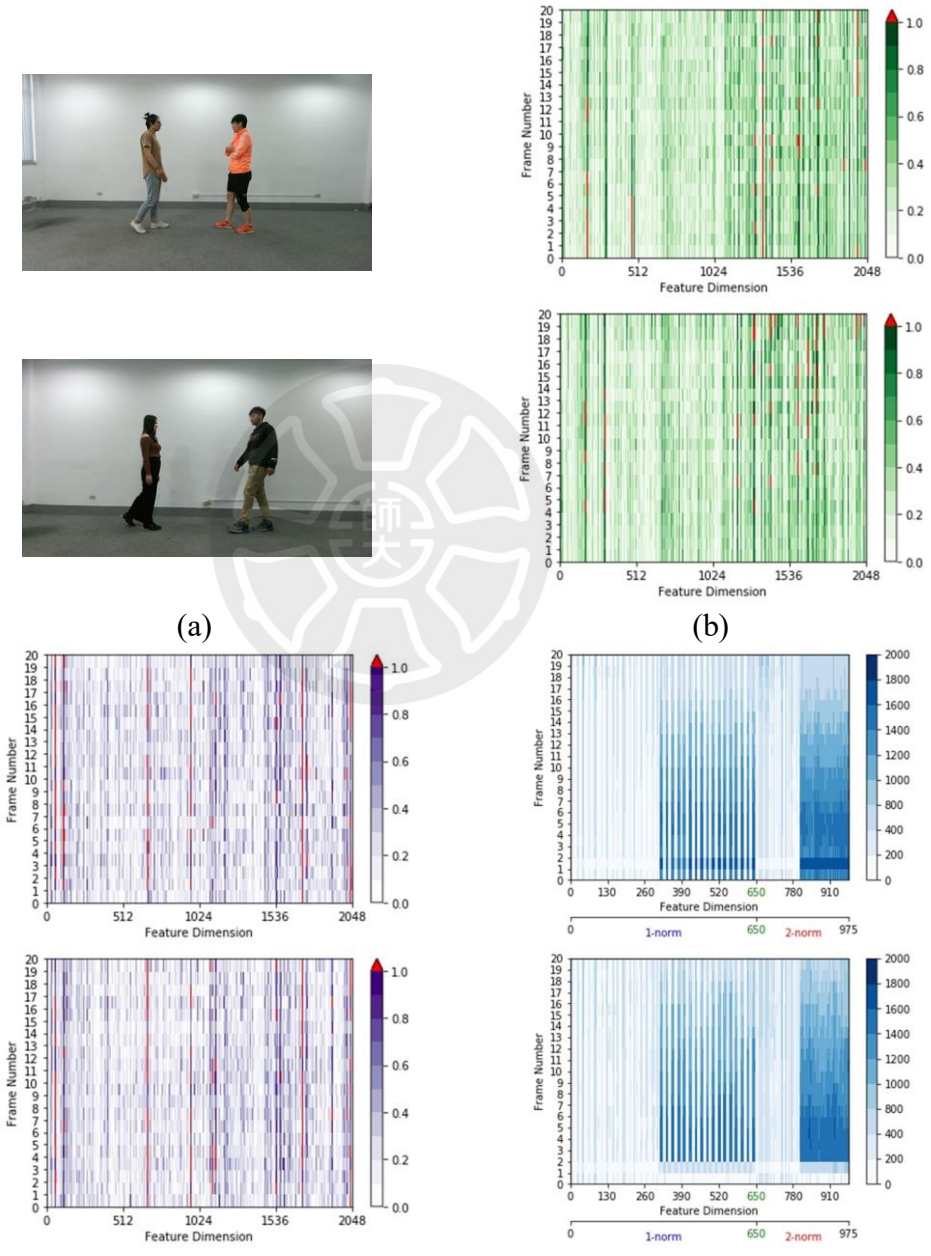



Figure 3.12 Two examples of feature map visualization (a) human actions (walk toward each other) (b) corresponding spatial feature maps (c) temporal feature maps (d) structural feature maps

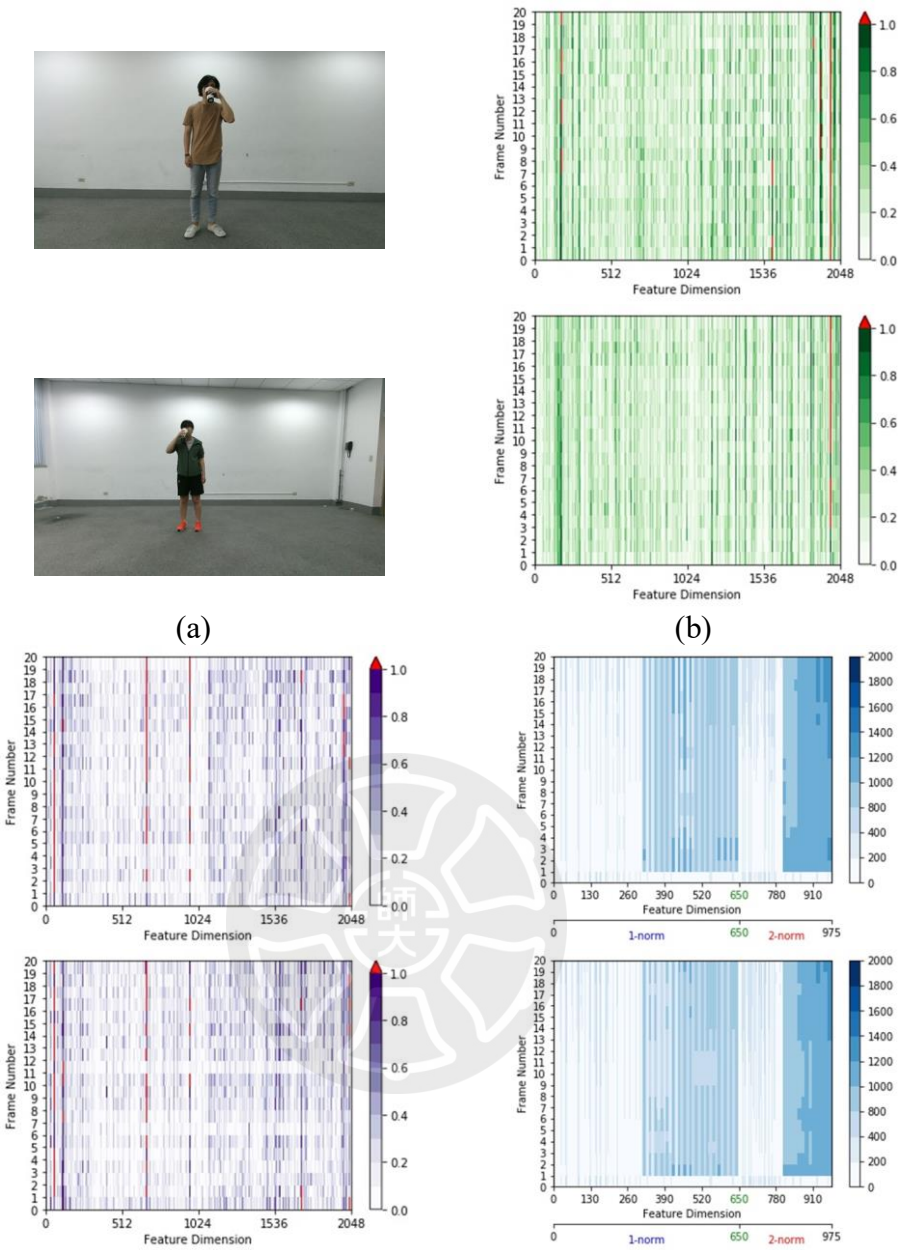



Figure 3.13 Examples of feature map visualisation (a) human actions (drink in stand position) (b) corresponding spatial feature maps (c) temporal feature maps (d) structural feature maps

## 3.2.4 Action Classification

This research adopts LSTM networks to classify human action. Each type of feature can be well classified by an appropriate and targeted network. Therefore, twelve kinds of LSTM networks are implemented to find appropriate ones for the three types of features. A new proposed temporal enhancement LSTM (TE-LSTM) is among the

implemented twelve networks. This subsection gives a brief overview of the LSTM networks and describes the LSTM models.

(1) *Overview of the LSTM Network*

The LSTM network improves on the RNN. As mentioned above, the RNN suffers from vanishing gradients in the training stage, and the LSTM network solves this problem.

The LSTM network consists of at least one LSTM layer, and each LSTM layer contains many LSTM cells. An LSTM cell includes a forget gate, an input gate, and an output gate, as shown in Figure 3.14. These three gates regulate, store, and add or remove the information at each cell. The input gate decides what new information should be added to cell state. The forget gate decides which cell states should be retained or removed. The output gate decides the final output vector based on the processed cell state. An LSTM cell is shown in Figure 3.14.

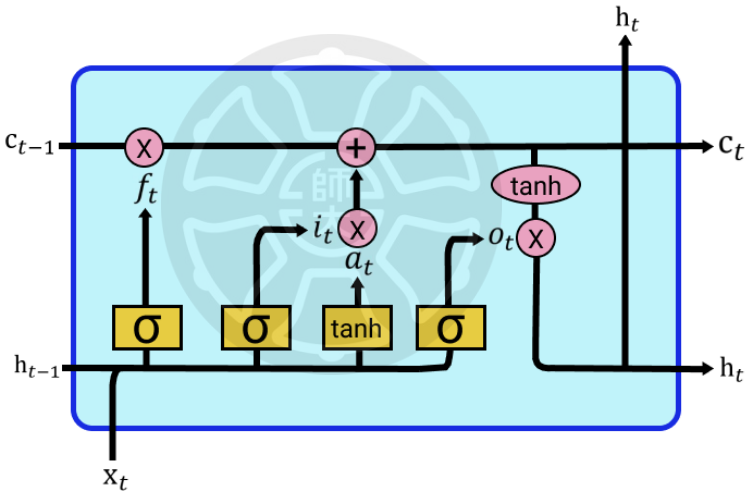



Figure 3.14 An LSTM cell

Given an input vector, $x_t$, at time step $t$ and a hidden state vector, $h_{t-1}$, at $t-1$, the output values of forget gate ($f_t$), input gate ($i_t$), output gate ($o_t$), and memory cell candidate ($a_t$), as shown in Figure 3.14, can be obtained by the following equations.

$$f_t = \sigma_s(w_f x_t + u_f h_{t-1} + b_f). \quad (5)$$

$$i_t = \sigma_s(w_i x_t + u_i h_{t-1} + b_i). \quad (6)$$

$$o_t = \sigma_s(w_o x_t + u_o h_{t-1} + b_o). \quad (7)$$

$$a_t = \sigma_h(w_c x_t + u_c h_{t-1} + b_c). \quad (8)$$

where $w_f$, $w_i$, $w_o$, $w_c$, $u_f$, $u_i$, $u_o$, and $u_c$, indicate parameter matrices. The symbols $b_f$, $b_i$, $b_o$, and $b_c$, are bias vectors, and $\sigma_s$ and $\sigma_h$ indicate the sigmoid

function and the tangent function, respectively.

The cell state vector, $c_{t-1}$, at $t-1$ can be regarded as the memory of the previous time step and can be used to make the connection with the cell state vector, $c_t$, at $t$. Then, $c_t$ can be calculated by

$$c_t = f_t * c_{t-1} + i_t * a_t. \quad (9)$$

Finally, the hidden state vector, $h_t$, at time step $t$ is defined as

$$h_t = o_t * \sigma_h(c_t). \quad (10)$$

where the operator $*$ denotes the element-wise product.

(2) *LSTM Networks*

Twelve kinds of LSTM networks, including LSTM networks with various layers, BiLSTM networks with various layers and the proposed TE-LSTM networks, are implemented to find suitable ones for the three types of features. Table 3.2 shows the structures of three LSTM networks with one, two and three layers, respectively. Different types of features can be input to train LSTM models. In Table 3.2, $1L_{Sp}$, $1L_{Te}$, and $1L_{St}$ indicate the structures of the LSTM networks with one-layer LSTM that are trained by spatial features, temporal features, and structural features, respectively. Similarly, $2/3L_{Sp}$, $2/3L_{Te}$, and $2/3L_{St}$ indicate the structures of the LSTM networks with two/three-layer LSTM trained by spatial features, temporal features, and structural features, respectively.

$\text{LSTM}_i$, $i = 1,2,3,4$, as shown in Table 3.2, indicate the hidden state units of the $i$th LSTM layers, and $\text{FC}_j$, $j = 1,2$, indicates the neuron numbers of full-connected layers. The neuron number of $\text{FC}_2$, 16, is equal to the number of human action classes.

Table 3.2 Structures of LSTM networks

| Structure | 1-Layer LSTM | | | 2-Layer LSTM | | | 3-Layer LSTM | | |
|---|---|---|---|---|---|---|---|---|---|
| | $1L_{Sp}$ | $1L_{Te}$ | $1L_{St}$ | $2L_{Sp}$ | $2L_{Te}$ | $2L_{St}$ | $3L_{Sp}$ | $3L_{Te}$ | $3L_{St}$ |
| $\text{LSTM}_1$ | 1024 | 512 | 1024 | 1024 | 512 | 1024 | 1024 | 512 | 1024 |
| $\text{LSTM}_2$ | | | | 512 | 256 | 512 | 1024 | 512 | 1024 |
| $\text{LSTM}_3$ | | | | | | | 512 | 256 | 512 |
| $\text{FC}_1$ | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| $\text{FC}_2$ | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |

Table 3.3 shows the structures of four BiLSTM networks with one, two, three, and four layers. As in Table 3.2, $1/2/3/4B_{Sp}$, $1/2/3/4B_{Te}$ and $1/2/3/4B_{St}$ indicate the structures of the BiLSTM networks with one/two/three/four-layer BiLSTM which are trained by spatial features, temporal features, and structural features, respectively. Further,

$\mathrm{BiLSTM}_i$, $i = 1,2,3,4$, indicates the hidden state units of the BiLSTM layers.

Table 3.3 Structures of BiLSTM networks

| Structure | 1-Layer BiLSTM | | | 2-Layer BiLSTM | | | 3-Layer BiLSTM | | | 4-Layer BiLSTM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $1B_{Sp}$ | $1B_{Te}$ | $1B_{St}$ | $2B_{Sp}$ | $2B_{Te}$ | $2B_{St}$ | $3B_{Sp}$ | $3B_{Te}$ | $3B_{St}$ | $4B_{Sp}$ | $4B_{Te}$ | $4B_{St}$ |
| $\mathrm{BiLSTM}_1$ | 2048 | 2048 | 1024 | 2048 | 2048 | 1024 | 2048 | 2048 | 1024 | 2048 | 2048 | 1024 |
| $\mathrm{BiLSTM}_2$ | | | | 1024 | 1024 | 512 | 2048 | 2048 | 1024 | 2048 | 2048 | 1024 |
| $\mathrm{BiLSTM}_3$ | | | | | | | 1024 | 1024 | 512 | 1024 | 1024 | 512 |
| $\mathrm{BiLSTM}_4$ | | | | | | | | | | 512 | 512 | 256 |
| $\mathrm{FC}_1$ | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| $\mathrm{FC}_2$ | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |

Additionally, one BiLSTM networks consists of two LSTM layers to process input sequences in two directions. One processes the input sequence from the first frame to the last frame (forward) along the time axis, and the other processes it from the last frame to the first frame (backward). In BiLSTM networks, the relationships of temporal variations can be analysed in both forward and backward directions. In summary, BiLSTM networks may capture better temporal dependencies than LSTM networks in some cases.

This study implemented five types of temporal enhancement (TE)-LSTM networks: TE-LSTM1, TE-LSTM2, TE-LSTM3, TE-LSTM4 and TE-LSTM5. These all have the same structure, a general TE-LSTM structure, but some of the layers use different LSTM models.

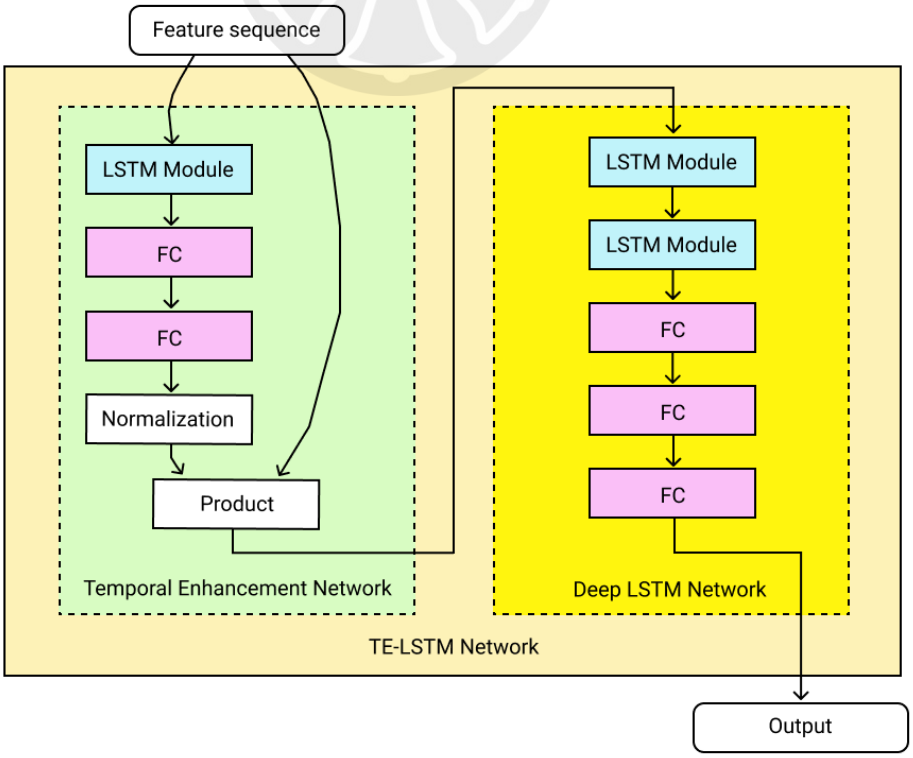



Figure 3.15 Network of a general TE-LSTM

A general TE-LSTM network comprises a TE network and a deep LSTM network, as shown in Figure 3.15. The TE network consists of an LSTM module and two fully connected layers. The deep LSTM network consists of two LSTM modules and three fully connected layers. Furthermore, the TE network can analyse the sequences to find their important parts and then pay more attention to those parts. By going through the TE network, the temporal information of sequences can be enhanced. The deep LSTM network then analyses the enhanced temporal sequences and classifies human actions. Noted that the LSTM modules can be either LSTM network or BiLSTM network.

In the TE network, the feature sequences are first passed through the LSTM module and two fully connected layers to analyse the input sequences. Next, the analysed sequences are normalised using the softmax normalisation method. Finally, the normalisation outputs are multiplied by the feature sequences using element-wise product operation. In the deep LSTM network, the product result is passed through two LSTM modules and three fully connected layers sequentially to classify the human actions.

Table 3.4 Structure of TE-LSTM networks (a) structure of TE-LSTM1 (b) structure of TE-LSTM2

(a)

| Structure | TE-LSTM1 | | |
|---|---|---|---|
| | $T1_{Sp}$ | $T1_{Te}$ | $T1_{St}$ |
| $LSTM_1^1$ | 2048 | 2048 | 975 |
| $FC_1^1$ | 2048 | 2048 | 975 |
| $FC_2^1$ | 2048 | 2048 | 975 |
| *FVN* | ✓ | ✓ | ✓ |
| ⊙ | ✓ | ✓ | ✓ |
| $LSTM_2^1$ | 512 | 512 | 512 |
| $LSTM_3^1$ | 256 | 256 | 256 |
| $FC_3^1$ | 128 | 128 | 128 |
| $FC_4^1$ | 128 | 128 | 128 |
| $FC_5^1$ | 16 | 16 | 16 |

(b)

| Structure | TE-LSTM2 | | |
|---|---|---|---|
| | $T2_{Sp}$ | $T2_{Te}$ | $T2_{St}$ |
| $LSTM_1^2$ | 2048 | 2048 | 975 |
| $FC_1^2$ | 2048 | 2048 | 975 |
| $FC_2^2$ | 2048 | 2048 | 975 |
| *FVN* | × | × | × |
| ⊙ | × | × | × |
| $LSTM_2^2$ | 512 | 512 | 512 |
| $LSTM_3^2$ | 256 | 256 | 256 |
| $FC_3^2$ | 128 | 128 | 128 |
| $FC_4^2$ | 128 | 128 | 128 |
| $FC_5^2$ | 16 | 16 | 16 |

Tables 3.4, 3.5 and 3.6 show the structures of these five TE-LSTM networks. Similarly to Table 3.2, $T1/2/3/4/5_{Sp}$, $T1/2/3/4/5_{Te}$, and $T1/2/3/4/5_{St}$ indicate the structures of the TE-LSTM1/2/3/4/5 networks trained by spatial features, temporal features and structural features, respectively.

In Tables 3.4, 3.5 and 3.6, $LSTM_i^n$, $i = 1,2,3$, $n = 1,2,3,4$, and $BiLSTM_i^n$, $i = 1,2,3$, $n = 3,4,5$ indicate the hidden state units of the *i*th LSTM/BiLSTM layers of the

$n$th TE-LSTM networks, respectively. Here, $FC_j^n$, $j = 1,2,3,4,5$, $n = 1,2,3,4,5$, indicates the neuron numbers of fully-connected layers. The neuron number of $FC_5^n$, 16, is equal to the number of human action classes. Moreover, *FVN* and ⊙ respectively indicate whether the features vector normalization, softmax normalization, and the element-wise product has been applied. A tick indicates the technique has been applied, and a cross indicates otherwise.

Table 3.5 Structure of TE-LSTM networks (a) structure of TE-LSTM3 (b) structure of TE-LSTM4

(a)

| Structure | TE-LSTM3 | | |
|---|---|---|---|
| | $T3_{Sp}$ | $T3_{Te}$ | $T3_{St}$ |
| $BiLSTM_1^3$ | 4096 | 4096 | 1950 |
| $FC_1^3$ | 2048 | 2048 | 975 |
| $FC_2^3$ | 2048 | 2048 | 975 |
| *FVN* | ✓ | ✓ | ✓ |
| ⊙ | ✓ | ✓ | ✓ |
| $LSTM_1^3$ | 512 | 512 | 512 |
| $LSTM_2^3$ | 256 | 256 | 256 |
| $FC_3^3$ | 128 | 128 | 128 |
| $FC_4^3$ | 128 | 128 | 128 |
| $FC_5^3$ | 16 | 16 | 16 |

(b)

| Structure | TE-LSTM4 | | |
|---|---|---|---|
| | $T4_{Sp}$ | $T4_{Te}$ | $T4_{St}$ |
| $LSTM_1^4$ | 2048 | 2048 | 975 |
| $FC_1^4$ | 2048 | 2048 | 975 |
| $FC_2^4$ | 2048 | 2048 | 975 |
| *FVN* | ✓ | ✓ | ✓ |
| ⊙ | ✓ | ✓ | ✓ |
| $BiLSTM_1^4$ | 1024 | 1024 | 1024 |
| $BiLSTM_2^4$ | 512 | 512 | 512 |
| $FC_3^4$ | 128 | 128 | 128 |
| $FC_4^4$ | 128 | 128 | 128 |
| $FC_5^4$ | 16 | 16 | 16 |

Table 3.6 Structure of TE-LSTM5

| Structure | TE-LSTM5 | | |
|---|---|---|---|
| | $T5_{Sp}$ | $T5_{Te}$ | $T5_{St}$ |
| $BiLSTM_1^5$ | 4096 | 4096 | 1950 |
| $FC_1^5$ | 2048 | 2048 | 975 |
| $FC_2^5$ | 2048 | 2048 | 975 |
| *FVN* | ✓ | ✓ | ✓ |
| ⊙ | ✓ | ✓ | ✓ |
| $BiLSTM_2^5$ | 1024 | 1024 | 1024 |
| $BiLSTM_3^5$ | 512 | 512 | 512 |
| $FC_3^5$ | 128 | 128 | 128 |
| $FC_4^5$ | 128 | 128 | 128 |
| $FC_5^5$ | 16 | 16 | 16 |

### 3.2.5 Fusion

Let the outputs of the three LSTM classifiers trained by spatial features, temporal features, and structural features at time $t$ be $v_t^{Sp}$, $v_t^{Te}$, and $v_t^{St}$ respectively. A fusion

method should be used to integrate these three outputs to determine the fusion action class $c_t^{fu}$. Noted that each of $v_t^{Sp}$, $v_t^{Te}$, and $v_t^{St}$ contains 16 probability values corresponding to the 16 action classes and the probability values are between 0 to 1. Here, $o_t^{Sp}$, $o_t^{Te}$, and $o_t^{St}$ indicate the highest probability values of $v_t^{Sp}$, $v_t^{Te}$, and $v_t^{St}$ at time $t$, and their corresponding action classes are $c_t^{Sp}$, $c_t^{Te}$, and $c_t^{St}$, respectively.

In this study, two kinds of fusion method are implemented and compared with each other to find the characteristics of fusion. Both methods consider the output action class from the previous time, $c_{t-1}^{fu}$, to classify the human action at the current time.

In the first fusion method, the output classes of the three types of features, $c_t^{Sp}$, $c_t^{Te}$, and $c_t^{St}$, have their corresponding highest probability values $o_t^{Sp}$, $o_t^{Te}$, and $o_t^{St}$, respectively. The fusion action class $c_t^{fu}$ is assigned by the class with the maximum probability values among $o_t^{Sp}$, $o_t^{Te}$, and $o_t^{St}$, if $c_t^{Sp}$, $c_t^{Te}$, and $c_t^{St}$ are all different from $c_{t-1}^{fu}$. For example, consider the case where $c_t^{Sp}$, $c_t^{Te}$, and $c_t^{St}$ are different from $c_{t-1}^{fu}$, and $max[o_t^{Sp}, o_t^{Te}, o_t^{St}] = o_t^{Sp}$. Then, $c_t^f = c_t^{Sp}$. Otherwise, $c_t^{fu}$ is assigned to $c_{t-1}^{fu}$.

In the second fusion method, the output class can be determined by the following equation.

$$c_t^{fu} = \begin{cases} c_t^{Sp} & \text{if } c_{t-1}^{fu} \neq c_t^{Sp} \text{ and } c_{t-1}^{fu} \neq c_t^{Te} \text{ and } c_{t-1}^{fu} \neq c_t^{St} \\ c_{t-1}^{fu} & \text{otherwise} \end{cases} \tag{11}$$

Experimental results show that the LSTM classifier trained by spatial features has higher recognition rates compared with those classifiers trained by temporal features and structural features, respectively. This suggests that the action class $c_t^{Sp}$ is sometimes more trustworthy than $c_t^{Te}$ and $c_t^{St}$. Therefore, in the second fusion method, the fusion action class $c_t^{fu}$ is assigned to $c_t^{Sp}$ if $c_t^{Sp}$, $c_t^{Te}$, and $c_t^{St}$ are all different from $c_{t-1}^{fu}$. Otherwise, $c_t^{fu}$ is assigned to $c_{t-1}^{fu}$.

# Chapter 4 Experimental Results

This chapter describes the research environment and equipment and provides details about the CVIU Moving Camera Human Action Dataset. The action classification results for the three types of features individually and the fusion results of action classification are presented. We also show the online human action recognition results of a multi-action sequence.

## 4.1 Research Environment and Equipment Setup

The focal point of this research is to recognise human actions under moving camera circumstances. This research simulates the moving camera circumstances by moving a four-wheel movable cart with a Kinect v2 sensor on it in a clean background classroom. The cart is 76.5 cm high and 45 cm wide. The CVIU Moving Camera Human Action dataset was built with this environment and equipment. This research was implemented in Python 3.7 using Keras 2.3, TensorFlow 1.15 and OpenCV4.1 run on NVIDIA GeForce GTX 1080 Ti on Ubuntu 16.04.

## 4.2 CVIU Moving Camera Human Action Dataset

We established an M-Video dataset called the CVIU Moving Camera Human Action dataset (CVIU dataset). The CVIU dataset contains 3,646 human action sequences (252,048 frames), including 11 types of single and 5 types of interactive human actions. The types of single human actions include drink in sit and stand positions, eat in sit and stand positions, play with a phone, sit down, stand up, use a laptop, walk straight, walk horizontal, and read. The types of interactive human actions include kick, hug, carry object, walk toward each other, and walk away from each other.

This dataset was recorded from three perspectives and each human action sequence was recorded while the camera was slowly moving towards the target persons. The first recording perspective, D1, had the Kinect v2 sensor facing the target person. The second recording perspective, D2, had the Kinect v2 sensor on the right side of the target person with a $45^{\circ}$ angle. The third recording perspective, D3, has the Kinect v2

sensor on the left side of the target person with a $45^{\circ}$ angle. Figure 4.1 provides a schematic diagram of the recording dataset including the three recording perspectives, and the recording environment and equipment. Figure 4.2 shows three recording perspectives for a human action sequence, “carry object”.

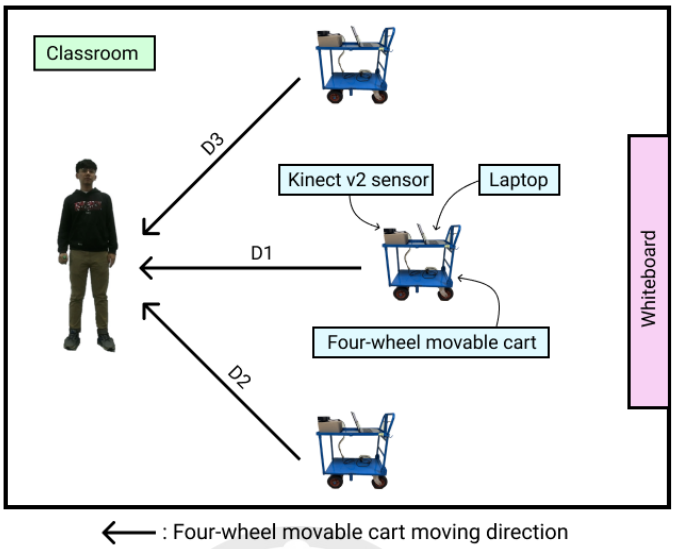



Figure 4.1 Schematic diagram of recording dataset

(a) D1

(b) D2

(c) D3

Figure 4.2 Three recording perspectives for “carry object” (a) D1, (b) D2, (c) D3

## 4.3 Action Classification Results of Three Types of Features

This research adopts the CVIU dataset to train and test the networks. In the training stage, action sequences are subsampled and extracted three types of features. And, these three types of features are individually fed into the LSTM networks for training. In the

testing stage, full action sequences are classified by the system in an online manner. Thus, the system outputs an action classification result by feeding 20 continuous frames. However, for the next 20 frames, the system inputs the last 10 frames from the previous 20 frames and the next continuous 10 frames. That is, the system outputs the first action classification result after feeding 20 frames continuously. Then, the system outputs a result every 10 frames. This kind of method can speed up the system without affecting the recognition of human action in the sequences. Note that the training and testing data are distinct. This subsection describes frame sampling number selection, preprocessing, and the action classification results of the CVIU dataset.

(1) *Decision of Frame Sampling Number*

This research uses the CVIU dataset for frame sampling number decision, which implements a one-layer LSTM network, to find a suitable frame sampling number. We chose six classes of human action from the CVIU dataset, three single and three interactive classes. The single human actions are use a laptop, drink in stand position, and eat in stand position. The interactive human actions are walk toward to each other, kick, and carry object. We used 240 sequences for training (40 for each action), 60 sequences for validating (10 for each actions), and 180 sequences for testing (30 for each action).

Table 4.1 Decision results of frame sampling number

| F1-Measure / Network / Action | Frame Sampling 10 | | | Frame Sampling 15 | | | Frame Sampling 20 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $1L_{Sp}$ | $1L_{Te}$ | $1L_{St}$ | $1L_{Sp}$ | $1L_{Te}$ | $1L_{St}$ | $1L_{Sp}$ | $1L_{Te}$ | $1L_{St}$ |
| Use a laptop | 1.000 | 0.789 | 0.842 | 0.984 | 0.732 | 0.692 | 0.984 | 0.841 | 0.596 |
| Drink in stand position | 0.815 | 0.261 | 0.238 | 0.929 | 0.392 | 0.103 | 0.909 | 0.523 | 0.433 |
| Eat in stand position | 0.822 | 0.536 | 0.565 | 0.892 | 0.426 | 0.577 | 0.892 | 0.311 | 0.701 |
| Walk toward each other | 0.696 | 0.061 | 0.063 | 0.667 | 0.121 | 0.300 | 0.947 | 0.278 | 0.333 |
| Kick | 0.866 | 0.532 | 0.454 | 0.822 | 0.624 | 0.522 | 0.968 | 0.615 | 0.483 |
| Carry object | 1.000 | 0.691 | 0.600 | 1.000 | 0.852 | 0.540 | 1.000 | 0.866 | 0.585 |
| Avg | 0.872 | 0.533 | 0.522 | 0.889 | 0.578 | 0.494 | **0.950** | **0.617** | **0.544** |

The decision results of frame sampling number are shown in Table 4.1, where $1L_{Sp}$, $1L_{Te}$, and $1L_{St}$ indicate a one-layer LSTM network trained by spatial features, temporal features, and structural features, respectively. Frame Sampling 10/15/20 respectively indicate the number of frame sampling as 10/15/20 frames. The recognition rates of F1-measurements are provided for single actions (listed in the blue region) and interactive actions (listed in the orange region). Avg refers to the average recognition rates of F1-measurement of each type of features with the corresponding number of frame sampling.

From Table 4.1, the average accuracy of frame sampling 20 had better recognition rates than the other options. This research also found that most actions can be finished within 20 frames. Therefore, the following experiments use subsampled human actions sequences to 20 frames to train the networks.

(2) *Preprocessing*

Preprocessing was implemented to determine whether it helps the system increase recognition rates. Preprocessing includes cropping and resizing the frames, and filling the skeletal joints. The number of training, testing, and action classes are the same as for the frame sampling selection. Table 4.2 shows the results of preprocessing. Similarly to Table 4.1, $1/2L^{p}_{Sp}$, $1/2L^{p}_{Te}$, and $1/2L^{p}_{St}$ indicate a one/two-layer LSTM network trained by the processed spatial features, temporal features, and structural features, respectively. Here, $1/2L^{w}_{Sp}$,$1/2L^{w}_{Te}$, and $1/2L^{w}_{St}$ indicate networks trained without using preprocessing.

Table 4.2 Results of preprocessing using 1/2-layer LSTM networks

| F1-Measur / Networks / Actions | Without Preprocessing | | | | | | Preprocessing | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $1L^{w}_{Sp}$ | $1L^{w}_{Te}$ | $1L^{w}_{St}$ | $2L^{w}_{Sp}$ | $2L^{w}_{Te}$ | $2L^{w}_{St}$ | $1L^{p}_{Sp}$ | $1L^{p}_{Te}$ | $1L^{p}_{St}$ | $2L^{p}_{Sp}$ | $2L^{p}_{Te}$ | $2L^{p}_{St}$ |
| Use laptop | 0.667 | 0.267 | 0.375 | 0.578 | 0.238 | 0.270 | 1.000 | 0.769 | 0.844 | 1.000 | 0.682 | 0.235 |
| Drink in stand position | 0.000 | 0.000 | 0.000 | 0.057 | 0.327 | 0.286 | 0.951 | 0.438 | 0.000 | 0.967 | 0.264 | 0.607 |
| Eat in stand position | 0.506 | 0.431 | 0.299 | 0.528 | 0.486 | 0.514 | 1.000 | 0.158 | 0.624 | 1.000 | 0.108 | 0.105 |
| Walk toward each other | 0.710 | 0.350 | 0.093 | 0.333 | 0.716 | 0.300 | 0.868 | 0.537 | 0.538 | 0.537 | 0.500 | 0.333 |
| Kick | 0.776 | 0.526 | 0.125 | 0.674 | 0.659 | 0.558 | 0.923 | 0.806 | 0.596 | 0.769 | 0.692 | 0.619 |
| Carry object | 0.594 | 0.469 | 0.000 | 0.044 | 0.468 | 0.520 | 0.984 | 0.896 | 0.467 | 0.984 | 0.844 | 0.700 |
| Avg | 0.588 | 0.400 | 0.227 | 0.45 | 0.516 | 0.450 | **0.955** | **0.650** | **0.567** | **0.889** | **0.572** | **0.483** |

From Table 4.2, the average accuracies with preprocessing were higher than without preprocessing. This proves that preprocessing does help the system to increase recognition rates. Thus, the following experiments all implemented preprocessing.

(3) *Action Classification Results of the CVIU Dataset*

This experiment uses all 16 human action classes in the CVIU dataset. Training data included 1696 sequences, evaluation data included 350 sequences and testing data included 1600 sequences. Table 4.3 lists the total human action sequences used for action classification experiments. Training, Validation, and Testing indicate the training data, validation data, and testing data, respectively. The human actions are represented by A01 to A16 with the amount of each sequence shown. Note that the actions “hug” and “kick” (highlighted in red) have more training data than the other actions. This is because these

two actions are more complicated. For example, in the action "kick", a person can kick with their left or right leg. For the action "hug", the person's hands can be in various positions.

Table 4.3 The total amounts of human action sequences used for action classification

| Sequences / Data Types / Actions | Training | Validation | Testing |
|---|---|---|---|
| A01: Drink in sit position | 89 | 22 | 100 |
| A02: Drink in stand position | 89 | 22 | 100 |
| A03: Eat in sit position | 88 | 22 | 100 |
| A04: Eat in stand position | 91 | 22 | 100 |
| A05: Play with a phone | 90 | 22 | 100 |
| A06: Read | 91 | 21 | 100 |
| A07: Sit | 93 | 22 | 100 |
| A08: Stand | 92 | 21 | 100 |
| A09: Use a laptop | 90 | 22 | 100 |
| A10: Walk horizontal | 95 | 22 | 100 |
| A11: Walk straight | 89 | 22 | 100 |
| A12: Carry object | 96 | 22 | 100 |
| A13: Walk away from each other | 94 | 22 | 100 |
| A14: Walk toward each other | 90 | 22 | 100 |
| A15: Hug | 223 | 22 | 100 |
| A16: Kick | 196 | 22 | 100 |
| Total | 1696 | 350 | 1600 |

As mentioned above, each type of feature has a proper LSTM network and we tested twelve types of LSTM networks trained by the three feature types. To evaluate which networks are appropriate for each feature type, we used an evaluation criteria, $R_r$ as shown in Equation (12), to evaluate the networks.

Assume the system obtains $n$ output classification results in a sequence, that is a set of action output probabilities, $\mathrm{X} = \{x_1, x_2, \ldots, x_n\}$. A threshold, $\theta$, can filter out the probabilities which are low while retaining the set of action output probabilities which are higher than the threshold, $\mathrm{X}' = \{x_i > \theta \mid x_i \in \mathrm{X}\}$, $\forall i \in n$. Each kind of network has a particular threshold value. The threshold values of spatial features, temporal features, and structural features are 80%, 60%, and 70%, respectively.

$$R_r = \frac{N_c}{N_g} \tag{12}$$

where $N_g$ indicates the cardinality of $\mathrm{X}'$, $N_g = |\mathrm{X}'|$. Further, $N_c$ indicates the number of correct classified human actions within $\mathrm{X}'$, $N_c = \left|\{x_j : Cor_a \mid x_j \in \mathrm{X}'\}\right|$, $\forall j \in n$, where $Cor_a$ indicates a correctly classified human action.

Tables 4.4, 4.5 and 4.6 show the action classification results using spatial features, temporal feature, and structural features, respectively. Here, $N_s$ and $R_r$ indicate networks and recognition rates, respectively, and $1/2/3L_{Sp/Te/St}$, $1/2/3/4B_{Sp/Te/St}$, $T1/2/3/4/5_{Sp/Te/St}$ are the aforementioned networks trained by spatial, temporal, and structural features, respectively. The training time of these networks are shown in Table 4.7. $N_s$ and $T_{trn}$ indicate networks and training time respectively. Time is measured in the unit of hour. Note that A01 to A11 are the single actions, and A12 to A16 are the interactive actions. Avg is the average $R_r$ of each type of feature with the corresponding networks and feature types. The unit is percentage (%).

Table 4.4 Action classification results of spatial features

| $N_s$ / $R_r$ / Actions | $1L_{Sp}$ | $2L_{Sp}$ | $3L_{Sp}$ | $1B_{Sp}$ | $2B_{Sp}$ | $3B_{Sp}$ | $4B_{Sp}$ | $T1_{Sp}$ | $T2_{Sp}$ | $T3_{Sp}$ | $T4_{Sp}$ | $T5_{Sp}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A01 | 98.65 | 99.32 | 71.00 | 99.21 | 97.96 | 98.30 | 98.98 | 0.00 | 98.72 | 0.00 | 0.00 | 0.00 |
| A02 | 99.38 | 99.08 | 93.79 | 98.73 | 94.34 | 96.65 | 96.05 | 0.00 | 99.42 | 0.00 | 76.46 | 68.62 |
| A03 | 95.79 | 92.75 | 64.00 | 95.18 | 97.43 | 96.15 | 98.62 | 0.00 | 95.56 | 0.00 | 0.00 | 0.00 |
| A04 | 98.64 | 93.24 | 55.00 | 96.89 | 97.93 | 98.45 | 97.87 | 0.00 | 98.21 | 0.00 | 85.13 | 33.65 |
| A05 | 94.22 | 97.00 | 7.00 | 85.00 | 96.00 | 94.57 | 94.40 | 0.00 | 95.00 | 0.00 | 0.00 | 0.00 |
| A06 | 99.41 | 98.88 | 11.00 | 100.00 | 98.89 | 98.32 | 99.37 | 0.00 | 99.07 | 0.00 | 0.00 | 0.00 |
| A07 | 93.37 | 91.66 | 93.00 | 95.67 | 95.31 | 92.04 | 92.55 | 0.00 | 95.98 | 0.00 | 93.80 | 0.00 |
| A08 | 85.32 | 85.64 | 34.55 | 81.25 | 81.82 | 81.00 | 83.93 | 0.00 | 84.37 | 0.00 | 0.00 | 0.00 |
| A09 | 100.00 | 99.88 | 98.00 | 100.00 | 100.00 | 100.00 | 100.00 | 0.00 | 100.00 | 0.00 | 0.00 | 98.00 |
| A10 | 95.66 | 97.73 | 98.00 | 96.40 | 97.41 | 98.20 | 97.20 | 10.00 | 99.27 | 0.00 | 89.35 | 67.57 |
| A11 | 95.06 | 95.93 | 99.00 | 99.47 | 96.88 | 95.75 | 94.89 | 0.00 | 97.15 | 0.00 | 79.27 | 14.33 |
| A12 | 99.21 | 99.20 | 100.00 | 100.00 | 99.88 | 99.28 | 99.55 | 63.08 | 99.64 | 0.00 | 47.84 | 97.32 |
| A13 | 95.72 | 97.14 | 90.80 | 98.50 | 97.82 | 95.67 | 96.27 | 0.00 | 95.88 | 0.00 | 81.07 | 98.00 |
| A14 | 93.13 | 86.48 | 95.33 | 93.15 | 94.51 | 91.32 | 91.83 | 0.00 | 90.13 | 0.00 | 63.37 | 33.33 |
| A15 | 98.00 | 97.89 | 91.00 | 100.00 | 100.00 | 100.00 | 100.00 | 84.61 | 98.00 | 85.11 | 96.02 | 89.73 |
| A16 | 97.27 | 96.00 | 73.67 | 100.00 | 100.00 | 100.00 | 100.00 | 85.40 | 96.67 | 39.40 | 94.83 | 97.17 |
| Avg | **96.18** | 95.49 | 73.45 | 96.21 | **96.64** | 95.98 | 96.34 | 15.19 | **96.44** | 7.78 | 50.45 | 43.61 |

For spatial features, the highest average recognition rates of LSTM, BiLSTM, and TE-LSTM networks were respectively achieved by the one-layer LSTM network (96.18%), two-layer BiLSTM network (96.64%), and TE-LSTM network with type 2 (96.44%), as shown in Table 4.4. The two-layer BiLSTM network had the highest recognition rate among all the networks, so we choose it to classify human actions that

are analysed using spatial features.

For temporal features, the highest average recognition rates of LSTM, BiLSTM, and TE-LSTM networks were respectively achieved by the one-layer LSTM network (71.58%), three-layer BiLSTM network (81.87%), and TE-LSTM network with type 2 (75.16%), shown in Table 4.5. The recognition rate of the three-layer BiLSTM network had the best results with a recognition rate 10.29% higher than that of the one-layer LSTM network. Thus, we choose the three-layer BiLSTM network to classify human actions that are analysed using temporal features.

Table 4.5 Action classification results of temporal features

| $R_r$ \ $N_s$ / Actions | $1L_{Te}$ | $2L_{Te}$ | $3L_{Te}$ | $1B_{Te}$ | $2B_{Te}$ | $3B_{Te}$ | $4B_{Te}$ | $T1_{Te}$ | $T2_{Te}$ | $T3_{Te}$ | $T4_{Te}$ | $T5_{Te}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A01 | 33.33 | 9.67 | 7.00 | 51.96 | 54.20 | 72.16 | 38.85 | 0.00 | 44.65 | 0.00 | 0.00 | 0.00 |
| A02 | 80.33 | 72.62 | 49.20 | 92.13 | 84.48 | 94.43 | 79.08 | 0.00 | 85.48 | 0.00 | 49.81 | 0.00 |
| A03 | 49.61 | 50.67 | 35.00 | 63.60 | 57.96 | 65.41 | 53.98 | 0.00 | 58.39 | 0.00 | 0.00 | 0.00 |
| A04 | 64.22 | 70.04 | 58.33 | 58.98 | 76.32 | 71.84 | 79.63 | 0.00 | 61.80 | 0.00 | 67.52 | 0.00 |
| A05 | 77.97 | 65.40 | 17.00 | 86.93 | 65.48 | 77.55 | 64.72 | 0.00 | 84.48 | 0.00 | 0.00 | 0.00 |
| A06 | 21.42 | 46.08 | 6.00 | 51.08 | 87.71 | 77.08 | 77.52 | 0.00 | 63.53 | 0.00 | 0.00 | 0.00 |
| A07 | 88.04 | 93.29 | 86.25 | 87.43 | 78.35 | 80.84 | 86.47 | 0.00 | 92.21 | 0.00 | 0.00 | 65.00 |
| A08 | 71.49 | 70.65 | 47.63 | 53.87 | 61.73 | 71.76 | 52.61 | 0.00 | 67.74 | 0.00 | 0.00 | 0.00 |
| A09 | 89.90 | 99.49 | 88.00 | 99.36 | 99.08 | 98.93 | 99.00 | 0.00 | 97.41 | 0.00 | 0.00 | 0.00 |
| A10 | 96.97 | 94.58 | 94.00 | 97.15 | 97.72 | 96.53 | 94.34 | 0.00 | 94.87 | 0.00 | 73.39 | 95.61 |
| A11 | 85.90 | 90.01 | 73.32 | 88.59 | 86.07 | 88.73 | 91.57 | 0.00 | 62.55 | 0.00 | 85.83 | 0.00 |
| A12 | 70.75 | 66.07 | 76.53 | 74.79 | 79.53 | 83.48 | 38.07 | 0.00 | 72.48 | 0.00 | 20.18 | 48.62 |
| A13 | 44.97 | 34.57 | 39.68 | 51.45 | 59.42 | 50.13 | 48.12 | 0.00 | 46.53 | 0.00 | 39.68 | 13.10 |
| A14 | 84.88 | 86.69 | 89.34 | 79.34 | 82.41 | 82.27 | 82.33 | 0.00 | 77.95 | 0.00 | 33.14 | 35.77 |
| A15 | 95.69 | 93.21 | 86.86 | 99.88 | 98.12 | 98.73 | 100.00 | 0.00 | 95.79 | 92.79 | 71.68 | 88.61 |
| A16 | 89.83 | 93.67 | 84.67 | 97.67 | 97.67 | 100.00 | 70.55 | 67.00 | 96.67 | 41.09 | 75.56 | 93.07 |
| Avg | **71.58** | 71.04 | 58.68 | 77.14 | 79.14 | **81.87** | 72.30 | 4.19 | **75.16** | 8.37 | 32.30 | 27.49 |

For structural features, the highest average recognition rates of LSTM, BiLSTM, and TE-LSTM networks were respectively achieved by the two-layer LSTM network (59.63%), three-layer BiLSTM network (60.35%), and TE-LSTM network with type 5 (68.10%), as shown in Table 4.6. The recognition rate of the TE-LSTM network with type 5 was 8.47% higher than that of the two-layer LSTM. Thus, we choose the TE-LSTM network with type 5 to classify human actions that are analysed using structural

features. Additionally, the TE-LSTM network with type 2 had the lowest recognition rate among other TE-LSTM networks, proving that the TE network enhances the temporal information of sequences.

Table 4.6 Action classification results of structural features

| $R_r$ / $N_s$ / Action | $1L_{St}$ | $2L_{St}$ | $3L_{St}$ | $1B_{St}$ | $2B_{St}$ | $3B_{St}$ | $4B_{St}$ | $T1_{St}$ | $T2_{St}$ | $T3_{St}$ | $T4_{St}$ | $T5_{St}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A01 | 0.00 | 35.72 | 23.33 | 0.00 | 30.50 | 50.50 | 2.00 | 73.00 | 61.04 | 55.52 | 55.84 | 81.79 |
| A02 | 0.00 | 60.65 | 82.79 | 0.00 | 83.35 | 80.53 | 48.70 | 83.25 | 73.94 | 72.56 | 92.28 | 81.79 |
| A03 | 0.00 | 2.25 | 4.00 | 0.00 | 1.00 | 0.00 | 0.00 | 6.08 | 2.25 | 0.00 | 59.37 | 15.48 |
| A04 | 0.00 | 68.09 | 53.39 | 0.00 | 43.57 | 84.10 | 19.00 | 75.10 | 58.03 | 62.58 | 52.91 | 92.41 |
| A05 | 0.00 | 66.38 | 57.08 | 0.00 | 36.47 | 39.00 | 26.00 | 64.67 | 71.40 | 92.80 | 38.92 | 68.38 |
| A06 | 0.00 | 1.00 | 1.00 | 0.00 | 1.50 | 1.00 | 0.00 | 3.40 | 0.00 | 0.00 | 35.08 | 0.00 |
| A07 | 0.00 | 85.18 | 92.14 | 0.00 | 94.95 | 87.83 | 68.77 | 90.17 | 87.79 | 65.74 | 93.01 | 84.17 |
| A08 | 0.00 | 46.82 | 67.05 | 0.00 | 60.68 | 56.83 | 35.50 | 75.47 | 55.44 | 46.42 | 57.68 | 42.23 |
| A09 | 0.00 | 59.46 | 4.00 | 0.00 | 29.00 | 10.00 | 0.00 | 17.22 | 29.36 | 32.92 | 63.87 | 55.00 |
| A10 | 4.50 | 74.68 | 75.13 | 73.48 | 80.33 | 92.19 | 90.06 | 85.71 | 69.95 | 81.75 | 91.24 | 89.45 |
| A11 | 0.00 | 78.00 | 37.86 | 0.00 | 52.23 | 60.17 | 84.25 | 90.50 | 42.65 | 71.85 | 64.75 | 91.65 |
| A12 | 33.85 | 52.72 | 44.50 | 35.44 | 47.24 | 67.43 | 57.23 | 49.62 | 58.64 | 50.52 | 46.88 | 67.16 |
| A13 | 51.60 | 62.47 | 68.10 | 56.07 | 58.57 | 72.50 | 63.83 | 89.98 | 67.42 | 52.57 | 63.35 | 63.58 |
| A14 | 27.06 | 69.47 | 75.39 | 66.84 | 70.20 | 75.70 | 74.62 | 71.73 | 68.79 | 65.60 | 69.33 | 71.22 |
| A15 | 72.78 | 93.66 | 95.24 | 82.73 | 93.95 | 89.27 | 94.25 | 90.01 | 91.08 | 88.88 | 90.08 | 94.20 |
| A16 | 87.47 | 95.95 | 95.23 | 89.17 | 95.00 | 98.50 | 96.50 | 98.00 | 97.17 | 99.33 | 97.90 | 91.17 |
| Avg | 17.33 | **59.53** | 54.76 | 25.23 | 54.91 | **60.35** | 47.54 | 66.49 | 58.43 | 58.69 | 67.03 | **68.10** |

Table 4.7 The training time of the twelve LSTM networks with the corresponding types of features (a) spatial feature, (b) temporal feature, (c) structural feature

(a)

| Ns | $1L_{Sp}$ | $2L_{Sp}$ | $3L_{Sp}$ | $1B_{Sp}$ | $2B_{Sp}$ | $3B_{Sp}$ | $4B_{Sp}$ | $T1_{Sp}$ | $T2_{Sp}$ | $T3_{Sp}$ | $T4_{Sp}$ | $T5_{Sp}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_{trn}$ (hr) | 2.77 | 2.76 | 2.75 | 2.77 | 2.81 | 2.78 | 2.84 | 2.79 | 2.80 | 2.79 | 2.83 | 2.89 |

(b)

| Ns | $1L_{Te}$ | $2L_{Te}$ | $3L_{Te}$ | $1B_{Te}$ | $2B_{Te}$ | $3B_{Te}$ | $4B_{Te}$ | $T1_{Te}$ | $T2_{Te}$ | $T3_{Te}$ | $T4_{Te}$ | $T5_{Te}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_{trn}$ (hr) | 2.76 | 2.76 | 2.75 | 2.77 | 2.77 | 2.80 | 2.80 | 2.78 | 2.78 | 2.79 | 2.81 | 2.86 |

(c)

| Ns | $1L_{St}$ | $2L_{St}$ | $3L_{St}$ | $1B_{St}$ | $2B_{St}$ | $3B_{St}$ | $4B_{St}$ | $T1_{St}$ | $T2_{St}$ | $T3_{St}$ | $T4_{St}$ | $T5_{St}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_{trn}$ (hr) | 2.76 | 2.75 | 2.75 | 2.77 | 2.76 | 2.76 | 2.79 | 2.75 | 2.75 | 2.75 | 2.78 | 2.78 |

## 4.4 Fusion Results

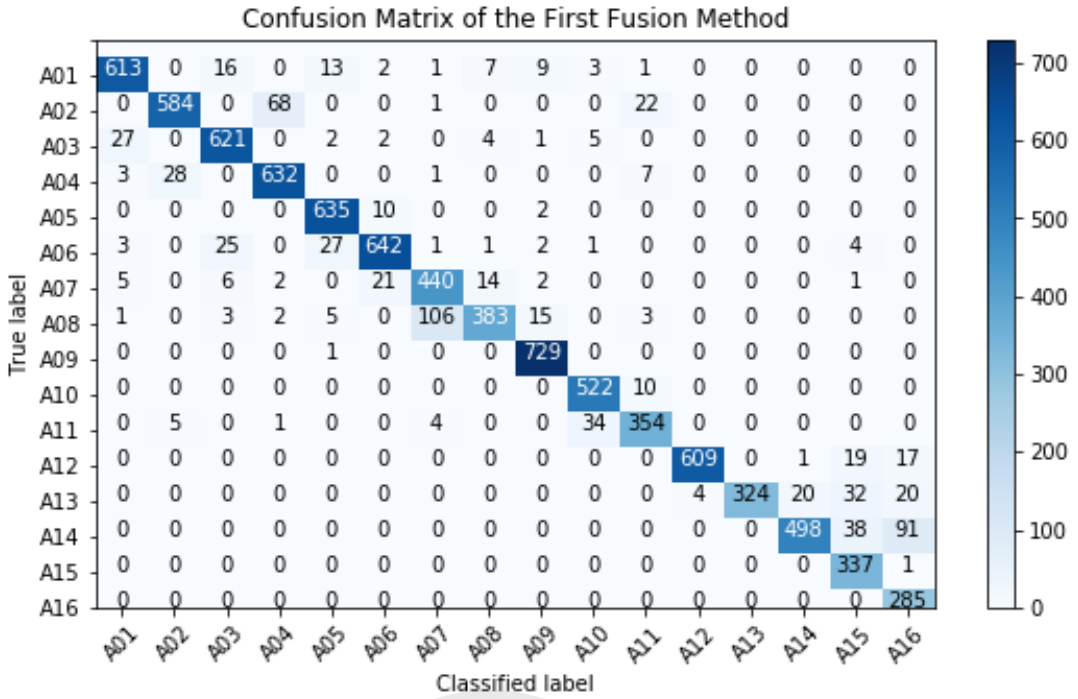
As mentioned above, two kinds of fusion methods were implemented and the classification results are shown in Table 4.8. Similarly to Table 4.4, Table 4.8 shows the recognition rates for the corresponding types of actions. Here, $2B_{Sp}$, $3B_{Te}$, $T5_{St}$, Fu1, and Fu2 respectively indicate the two-layer BiLSTM network that is trained by spatial features, the three-layer BiLSTM network that is trained by temporal features, the TE-LSTM with type 5 network that is trained by structural features, the first fusion method, and the second fusion method. Besides, $2B_{Sp}$/$3B_{Te}$/$T5_{St}$/Fu1/Fu2 takes about 2.4/3.7/2/4.3/4.2 seconds to output a classification result.

Table 4.8 Classification results of fusion methods and three types of features

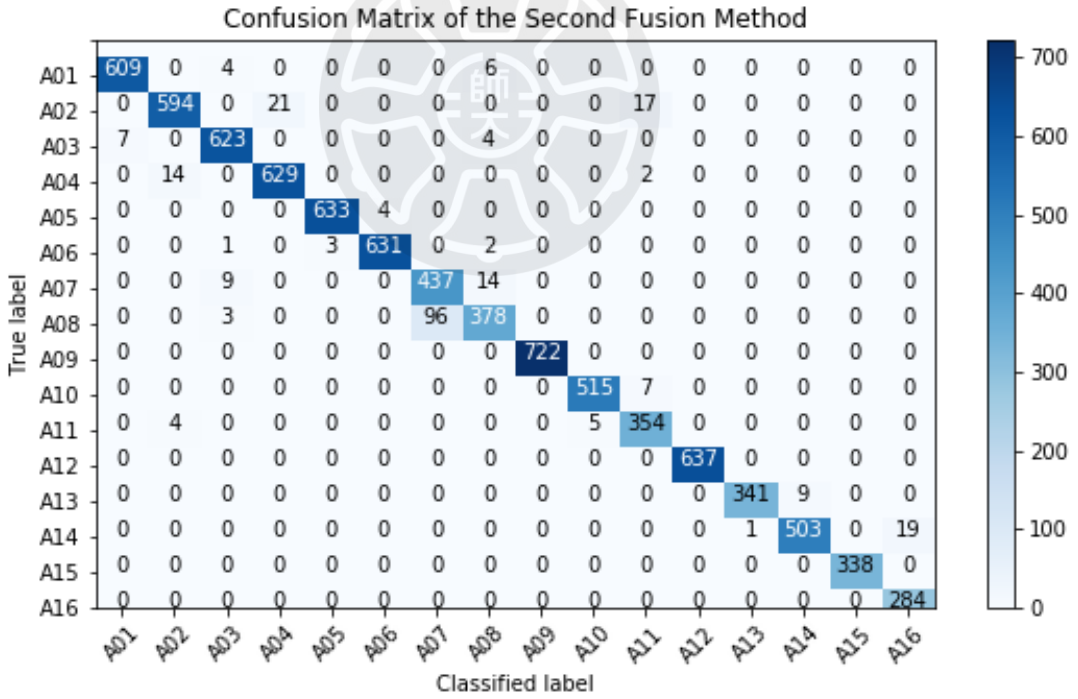
| $R_r$ Classification Methods / Actions | $2B_{Sp}$ | $3B_{Te}$ | $T5_{St}$ | Fu1 | Fu2 |
|---|---|---|---|---|---|
| A01 | 97.96 | 72.16 | 81.79 | 98.08 | 98.80 |
| A02 | 94.34 | 94.43 | 81.79 | 89.60 | 94.64 |
| A03 | 97.43 | 65.41 | 15.48 | 94.75 | 96.43 |
| A04 | 97.93 | 71.84 | 92.41 | 95.85 | 97.76 |
| A05 | 96.00 | 77.55 | 68.38 | 94.00 | 95.00 |
| A06 | 98.89 | 77.08 | 0.00 | 94.67 | 99.00 |
| A07 | 95.31 | 80.84 | 84.17 | 96.46 | 96.80 |
| A08 | 81.82 | 71.76 | 42.23 | 81.00 | 82.34 |
| A09 | 100.00 | 98.93 | 55.00 | 100.00 | 100.00 |
| A10 | 97.41 | 96.53 | 89.45 | 98.18 | 98.60 |
| A11 | 96.88 | 88.73 | 91.65 | 96.87 | 97.72 |
| A12 | 99.88 | 83.48 | 67.16 | 95.37 | 100.00 |
| A13 | 97.82 | 50.13 | 63.58 | 86.00 | 97.02 |
| A14 | 94.51 | 82.27 | 71.22 | 81.04 | 95.40 |
| A15 | 100.00 | 98.73 | 94.20 | 99.83 | 100.00 |
| A16 | 100.00 | 100.00 | 91.17 | 100.00 | 100.00 |
| Avg | 96.64 | 81.87 | 68.10 | 93.86 | **96.84** |

From Table 4.8, the average recognition rates of the second fusion method is higher than that of the first fusion method. Additionally, the average recognition rates of the second fusion method are the highest among all the classification methods. The second fusion method had the highest recognition rates for all actions except “eat in sit and stand positions, play with a phone, and walk away from each other”. The recognition rates of these actions was worse than that of $2B_{Sp}$. This might be because the recognition rates of these actions in $3B_{Te}$ and $T5_{St}$ reduce the recognition rates of these actions when

fusion is applied in the second fusion method.




(a)

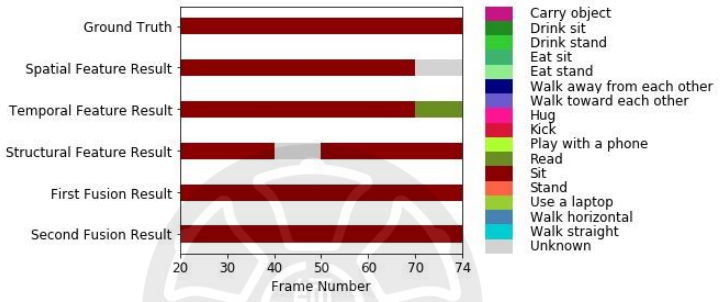



(b)

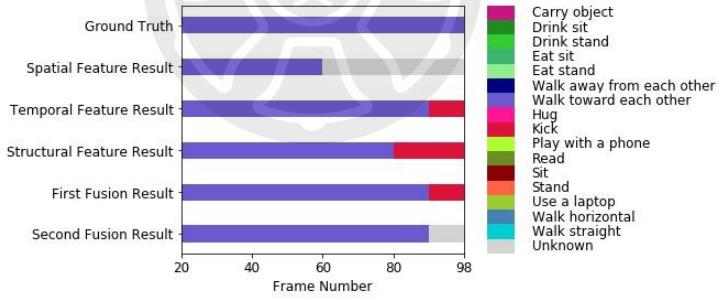
Figure 4.3 Confusion matrix for (a) the first fusion method (b) the second fusion method

Figure 4.3 shows the confusion matrices of fusion methods. In each matrix, the horizontal axis is the classified label, and vertical axis is the true label. The values in the matrix indicate the frame level output classification results. For example, assume the

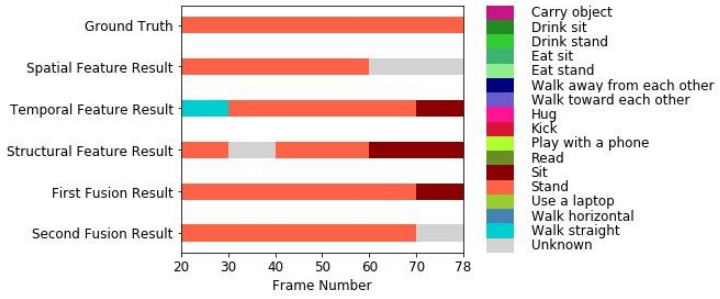
system outputs 5 classification results for an action sequence, and these results are added into the matrix. A higher value is indicated by a darker colour. Comparing the first fusion method, Figure 4.3 (a), to the second fusion method, Figure 4.3 (b), some actions are sometimes classified incorrectly in the first fusion method. For example, actions A11 to A13 are sometimes classified as other interactive actions in the first fusion method. Conversely, the second fusion method has fewer errors of this kind. However, the action "stand" (A08) is often classified as the action "sit" (A07) in both methods. This may be because in the last moment of the action "stand" (A08), the target person is in the stand position, and the system starts predicting that the target person is going to sit down. This causes the action "stand" (A08) to be classified as "sit" (A07). Figure 4.4 (c) shows the classification results of the action "stand".




(a)




(b)




(c)

Figure 4.4 Classification results of the online system (a) action "sit" (b) action "walk toward each other" (c) action "stand"

Figure 4.4 shows the classification results of the online human action recognition

system. The horizontal axis shows the frame numbers of a sequence. The vertical axis shows ground truth, spatial feature classification result, temporal feature classification result, structural feature classification result, classification result of the first fusion method, and classification result of the second fusion method. Each action is represented by a colour.

Figure 4.4 (a), (b) and (c) show the classification results of actions “sit”, “walk toward each other” and “stand”. Some output results using spatial features, temporal features and structural features are either unknown or classified incorrectly. Additionally, although the recognition rates of spatial features are high, the output results still have too many unknown classification results. However, by using the first and second fusion methods, some of these unknown items can be correctly classified. From Figure 4.3 (b) and (c), the classification results of the first fusion method sometimes still have incorrect classification results because the LSTM classifiers trained by temporal features and structural features affect the classification results and reduce the recognition rates.

## 4.5 Multi-Human Action Classification Results

Figure 4.5 shows the multi-human action classification results of the online human action recognition system using the same format as Figure 4.4. Figure 4.5 (a) shows a sequence containing two actions, “walk toward each other” and “carry object”. The recognition rate of the first fusion method is 90.90%, and that of the second fusion method is 100.00%. Figure 4.5 (b) shows a sequence containing three actions, “walk horizontal”, “sit”, and “drink sit”. The recognition rate of the first fusion method is 61.90%, and that of the second fusion method is 90.48%. In these examples, the second fusion method is better than the first fusion method. This is because the LSTM classifier trained by spatial features is more reliable than the other two classifiers and the LSTM classifiers trained by temporal features and structural features reduce the recognition rates of the first method. Figure 4.5 (c) shows a sequence that contains three action, “walk toward each other”, “kick”, and “walk away from each other”. The recognition rate of the first fusion method is 87.50%, and that of the second fusion method is 62.50%. In this example, the first fusion method is better than the second fusion method. This is because the second fusion method completely trusts the LSTM classifier trained by

spatial features; however, if that classifier recognises incorrectly, the output results lead the fusion to an incorrect classification result.

Although, the second fusion method has better recognition rates when sequences contain only one action class, the first fusion method sometimes has better recognition rates for sequences containing multiple actions. Overall, this research recommends using the second fusion method because it performs better in most cases.

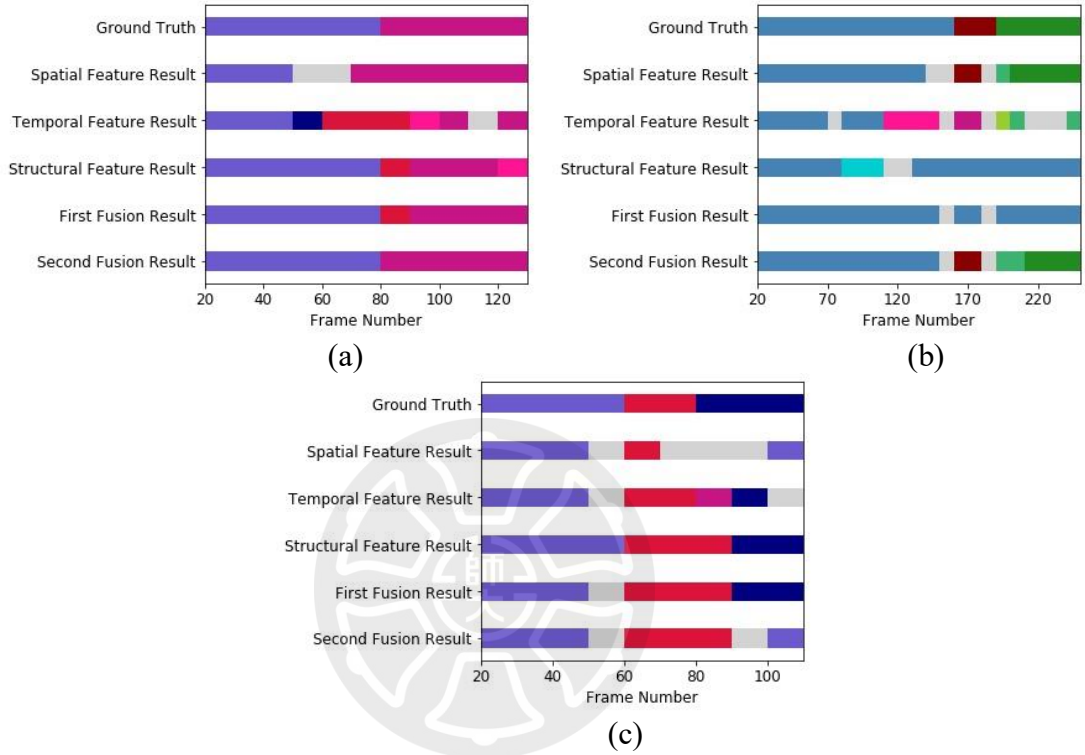



Figure 4.5 Multi-action classification results of the online system (a) actions “walk toward each other” and “carry object” (b) actions “walk horizontal”, “sit”, and “drink in sit position” (c) actions “walk toward each other”, “kick”, and “walk away from each other”

# Chapter 5 Conclusions and Future Works

## 5.1 Conclusions

This research proposes a vision-based online human action recognition system that can recognise human action under M-Video circumstances. The proposed system comprises five stages: human detection, human tracking, feature extraction, action classification, and fusion. Moreover, the system uses three types of input information for human action recognition: colour intensity, short-term dynamic information, and skeletal joints.

We adopted a 2D human pose estimator, OpenPose [Cao19], to detect humans and Deep SORT [Woj17] to track humans. We extracted three types of features, spatial-based features, temporal-based features, structural-based features, to analyse human actions. These three types of features were input into their corresponding LSTM networks for human action classification. Finally, we applied fusion methods to integrate the classification results of the LSTM networks to determine the final classification of the human action. In this study, we proposed a TE-LSTM network, composed of a TE network and a deep LSTM network. Experimental results show that the TE-LSTM network can increase the recognition rate based on structural features.

We also established the CVIU dataset, an M-Video dataset containing 11 types of single human actions and 5 types of interactive human actions. The CVIU dataset was used to train and to evaluate the proposed system. Experimental results showed that each type of feature has a suitable network among twelve kinds of LSTM networks. The two-layer BiLSTM network can obtain a 96.64% recognition rate of human action from spatial features. The three-layer BiLSTM network can obtain an 81.87% recognition rate of human action from temporal features. The TE-LSTM network with type 5 can obtain a 68.10% recognition rate of human action from structural features. Finally, the recognition rate of human action after integration was 96.84%.

## 5.2 Future Works

The CVIU dataset currently contains only 16 human action classes. However, the CVIU dataset could be extended in terms of both the amount of data and the number of action classes. This would let the system recognise more human actions in the future. This research only trained and evaluated twelve kinds of LSTM networks in the action classification stage. However, LSTM networks can be modified to more kinds of structures, such as by adding dropout layers and more fully connected layers. Modified LSTM networks might obtain a better recognition rate than the twelve networks we tested. The highest recognition rate of the LSTM networks trained by structural features was only 68.10%. Data augmentation methods could be implemented to enlarge the training data to help LSTM networks trained by structural features to perform better. In the evaluation stage, the threshold values used to filter out low output probabilities for action classification are currently non-automatic. An automatic threshold adjustment mechanism could be explored to get a more precise threshold value for the system.

This study used LSTM networks to classify human actions in the action classification stage. However, LSTM networks have many parameters because of the three gates that exist in an LSTM cell. Having too many parameters can cause problems, such as occupying too much memory, reducing the execution speed, and slowing down the network training. The recently proposed Gate Recurrent Unit (GRU) [Chu14] networks, another kind of RNN network, contain only two gates in a GRU cell. Therefore, the number of parameters in GRU networks is less than that of LSTM. Networks with fewer parameters might use less memory, execute quicker, and train faster. Consequently, GRU networks may be considered to replace LSTM networks in the proposed system.

Currently, the developed system only applies to indoor spaces. The system could be developed to apply to more diverse spaces, such as outdoor spaces. Compared with indoor spaces, outdoor spaces are more complicated because the illumination and environments are uncontrollable. However, we hope the proposed system could be enhanced to recognise human actions in both indoor and outdoor spaces in future.

# References


[Hoa12] M. Hoai and F. De la Torre, "Max-Margin Early Event Detectors," *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp. 2863-2870.

[De18] R. De Geest and T. Tuytelaars, "Modeling Temporal Structure with LSTM for Online Action Detection," *Proceedings of 2018 IEEE Winter Conference on Applications of Computer Vision*, Lake Tahoe, NV, 2018, pp. 1549-1557.

[Hoc97] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, 1997, pp. 1735-1780.

[Han18] Y. Han, S. Chung, A. Ambikapathi, J. Chan, W. Lin, and S. Su, "Robust Human Action Recognition Using Global Spatial-Temporal Attention for Human Skeleton Data," *Proceedings of 2018 International Joint Conference on Neural Networks*, Rio de Janeiro, 2018, pp. 1-8.

[Jun18] S. Jun and Y. Choe, "Deep Batch-Normalized LSTM Networks with Auxiliary Classifier for Skeleton Based Action Recognition," *Proceedings of 2018 IEEE International Conference on Image Processing, Applications and Systems*, Sophia Antipolis, France, 2018, pp. 279-284.

[Sha16] A. Shahroudy, J. Liu, T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 1010-1019.

[Son18] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection," *IEEE Transactions on Image Processing*, vol. 27, no.7, 2018, pp. 3459-3471.

[Tu18] J. Tu, H. Liu, F. Meng, M. Liu, and R. Ding, "Spatial-Temporal Data Augmentation Based on LSTM Autoencoder Network for Skeleton-Based Human Action Recognition," *Proceedings of 2018 25th IEEE International Conference on Image Processing*, Athens, 2018, pp. 3478-

3482.

[Li17] C. Li, Y. Hou, P. Wang, and W. Li, “Joint Distance Maps Based Action Recognition with Convolutional Neural Networks,” *IEEE Signal Processing Letters*, vol. 24, no. 5, 2017, pp. 624-628.

[Liu18] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, “Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, 2018, pp. 3007-3021.

[Soo19] K. Soomro, H. Idrees, and M. Shah, “Online Localization and Prediction of Actions and Interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, 2019, pp. 459-472.

[Wei16] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional Pose Machines,” *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 4724-4732.

[Ull18] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, “Action Recognition in Video Sequences Using Deep Bi-Directional LSTM with CNN Features,” *IEEE Access*, vol. 6, 2018, pp. 1155-1166.

[Ouy19] X. Ouyang, S. Xu, C. Zhang, P. Zhou, Y. Yang, G. Liu, and X. Li, “A 3D-CNN and LSTM Based Multi-Task Learning Architecture for Action Recognition,” *IEEE Access*, vol. 7, pp. 40757-40770, 2019.

[Hua19] J. Huang, N. Li, T. Li, S. Liu, and G. Li, “Spatial-Temporal Context-Aware Online Action Detection and Prediction,” *IEEE Transactions on Circuits and Systems for Video Technology* (Early Access), 2019, pp. 1-13.

[You19] Q. You and H. Jiang, “Action4D: Online Action Recognition in the Crowd and Clutter,” *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 11849-11858.

[Liu19] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, “Multi-Modality Multi-Task Recurrent Neural Network for Online Action Detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, 2019, pp. 2667-2682.

[Goe18] A. Goel, A. Abubakr, M. Koperski, F. Bremond, and G. Francesca,

"Online Temporal Detection of Daily-Living Human Activities in Long Untrimmed Video Streams," *Proceedings of 2018 IEEE International Conference on Image Processing, Applications and Systems*, Sophia Antipolis, France, 2018, pp. 43-48.

[Du18] W. Du, Y. Wang, and Y. Qiao, "Recurrent Spatial-Temporal Attention Network for Action Recognition in Videos," *IEEE Transactions on Image Processing*, vol. 27, no. 3, 2018, pp. 1347-1360.

[Ni11] J. Ni and J. Xu, "A Statistical Model Based on Spatio-Temporal Features for Action Recognition," *Proceedings of 2011 Seventh International Conference on Natural Computation*, Shanghai, 2011, pp. 1593-1597.

[Liu10] J. Liu, J. Yang, Y. Zhang, and X. He, "Action Recognition by Multiple Features and Hyper-Sphere Multi-Class SVM," *Proceedings of 2010 20th International Conference on Pattern Recognition*, Istanbul, 2010, pp. 3744-3747.

[Dol05] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features," *Proceedings of 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, 2005, pp. 65-72.

[Sch97] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, 1997, pp. 2673-2681.

[Kri12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 1, Nevada, 2012, pp. 1097-1105.

[Tra15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," *Proceedings of 2015 IEEE International Conference on Computer Vision*, Santiago, 2015, pp. 4489-4497.

[Sim14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556 [cs.CV], 2014.

[Bah14] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” arXiv:1409.0473 [cs.CL], 2014.

[Lin13] M. Lin, Q. Chen, and S. Yan, “Network in Network,” arXiv:1312.4400 [cs.NE], 2013.

[Liu16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot Multibox Detector,” *Proceedings of European Conference on Computer Vision*, arXiv:1512.02325 [cs.CV], Amsterdam, 2016.

[He16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 770-778.

[Li17] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, “Skeleton-Based Action Recognition Using LSTM and CNN,” *Proceedings of 2017 IEEE International Conference on Multimedia & Expo Workshops*, Hong Kong, 2017, pp. 585-590.

[Liu17] C. Liu, Y. Li, Y. Hu, and J. Liu, “Online Action Detection and Forecast via Multitask Deep Recurrent Neural Networks,” *Proceedings of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, 2017, pp. 1702-1706.

[Cio18] G. Ciocca, A. Elmi, P. Napoletano, and R. Schettini, “Activity Monitoring from RGB Input for Indoor Action Recognition Systems,” *Proceedings of 2018 IEEE 8th International Conference on Consumer Electronics - Berlin*, Berlin, 2018, pp. 1-4.

[Cha19] M. Chang, J. Hsieh, C. Fang, and S. Chen, “A Vision-Based Human Action Recognition System for Moving Cameras Through Deep Learning,” *Proceedings of the 2019 2nd International Conference on Signal Processing and Machine Learning*, Hangzhou, 2019, pp. 85–91.

[Wan16] P. Wang, C. Li, Y Hou, and W Li, “Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks,” *Proceedings of the 24th ACM international conference on Multimedia*, Amsterdam, 2016.

[Ijj14] E. P. Ijjina and C. K. Mohan, “Human Action Recognition Based on Recognition of Linear Patterns in Action Bank Features Using Convolutional Neural Networks,” *Proceedings of 13th International Conference on Machine Learning and Applications*, Detroit, MI, 2014, pp. 178-182.

[Ull19] A. Ullah, K. Muhammad, J. Del Ser, S. W. Baik, and V. H. C. de Albuquerque, “Activity Recognition Using Temporal Optical Flow Convolutional Features and Multilayer LSTM,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, 2019, pp. 9692-9702.

[Jag16] B. Jagadeesh and C. M. Patil, “Video Based Action Detection and Recognition Human Using Optical Flow and SVM Classifier,” *Proceedings of 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, Bangalore, 2016, pp. 1761-1765.

[Ilg17] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks,” *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, pp. 1647-1655.

[Cao19] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. (Early Access)

[Fan17] H. Fang, S. Xie, Y. Tai, and C. Lu, “RMPE: Regional Multi-Person Pose Estimation,” *Proceedings of 2017 IEEE International Conference on Computer Vision*, Venice, 2017, pp. 2353-2362.

[He17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *Proceedings of 2017 IEEE International Conference on Computer Vision*, Venice, 2017, pp. 2980-2988.

[Pap18] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, “Personlab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model,” *Proceedings of European Conference on Computer Vision*, Germany, 2018, pp. 282-299.

[Koc18] M. Kocabas, S. Karagoz, and E. Akbas, “MultiPoseNet: Fast Multi-Person Pose Estimation Using Pose Residual Network,” *Proceedings of European Conference on Computer Vision*, Germany, 2018, pp. 437-453.

[Woj17] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” *Proceedings of 2017 IEEE International Conference on Image Processing*, Beijing, 2017, pp. 3645-3649.

[Sze16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 2818-2826.

[He15] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *Proceedings of 2015 IEEE International Conference on Computer Vision*, Santiago, 2015, pp. 1026-1034.

[Iof15] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *Proceedings of the 32nd International Conference on Machine Learning*, France, 2015, pp 448-456.

[Sze15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015, pp. 1-9.

[Chu14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *Neural Information Processing Systems 2014 Workshop on Deep Learning*, Canada, 2014.

[1] Jenny Medeiros (2018, May 24). “LG Made An Airport Guide Robot and It's Going Places (Literally)” Available: https://www.voicesummit.ai/blog/lg-made-an-airport-guide-robot-and-its-going-places-literally-0. (Nov. 10, 2019)

[2] Margaret Rouse (2005, Sep.). “AIBO (Artificial Intelligence roBOt)”.

Available: https://searchcio.techtarget.com/definition/AIBO. (Jul. 30, 2019)

[3] Zenbo Design Guideline (Zenbo Introduction). Available: https://zenbo.asus.com/developer/documents/Design-uideline/Zenbo-Introduction/Basic-Functions. (Nov. 10, 2019)

[4] Hope Reese (2016, Aug. 11). “A 4-foot tall humanoid robot named Pepper, made by Aldebaran, a SoftBank company, interacts with people in hospitals, hotels, and homes. TechRepublic's comprehensive guide explains how Pepper works”. Available: https://www.techrepublic.com/article/pepper-the-robot-the-smart-persons-guide/. (Jul.30, 2019)

[5] Justin Kahn (2017, Jan, 4). LG tries to take over the world with new lineup of adorable life-size robots for the home, airport and more. Available: https://9to5toys. com/2017/01/04/lg-adorable-life-size-robots-home-airport/. (Apr. 4, 2020)

[6] Shine ding (2018, Jan. 12). Aibo resurgence is super cute ! Sony participates in AI network connection! Available: http://pc3mag.com/sony-ai-aibo/. (Apr. 4, 2020)

[7] Ausu. Zenbo Qrobot. Available: https://zenbo.asus.com.cn/product/zenbo/support/. (Apr. 4, 2020)

[8] Softbank Robotics Europe (2016, Feb. 12). File: Pepper the Robot.jpg. Available: https://fr.wikipedia.org/wiki/Fichier:Pepper_the_Robot.jpg. (Apr. 4, 2020)

[9] “Smart Robot Market – Global Industry Analysis and Forecast (2017-2026) _ by Components (Hardware, Software), by Industrial Application (Electronics, Automotive, and Others), by Service Application (Personal, Professional), and by Geography” (2018, Feb.). Available: https://www.maximizemarketresearch.com/market-report/smart-robot-market/2317/. (Jul. 30, 2019).

[10] “Global Indoor Robots Market – Industry Analysis and Forecast (2019-2026) – by Product, End User and Region” (2019, Jul.). Available: https://www.maximizemarketresearch.com/market-report/global-indoor-robots-market/33164/. (Apr. 4, 2020)

[11] “Mobile Robots Market by Operating Environment (Aerial, Ground,

and Marine), Component (Control System, Sensors), Type (Professional and Personal & Domestic Robots), Application (Domestic, Military, Logistics, Field), and Geography - Global Forecast 2023" (2019, Sep.). Available: https://www.marketsandmarkets.com/Market-Reports/mobile-robots-market-43703276.html?gclid=CjwKCA iA zuPuBRAIEiwAkkmOSNhzEu8csjHz WQaBt6UcIvepvuUAfmcJQhD ijRNQ0HZh_Xf630L-GBoCG20QAvD_BwE. (Nov. 23, 2019)

[12] "Service Robotics Market - Growth, Trends, and Forecast (2019-2024)" (2019, Apr.). Available: https://www.mordorintelligence.com/industry-reports/global-service-robotics-market-industry. (Nov. 23, 2019)

[13] "Robotics Market - Growth, Trends, and Forecast (2019-2024)" (2019,Feb.). Available: https://www.mordorintelligence. com/industry-reports/robotics-market. (Nov. 23, 2019)

[14] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Duan, and A. Kot, "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*), 2019. Available: http://rose1.ntu.edu.sg/ Datasets/actionRecognition.asp#C7. (Apr. 15, 2020)

[15] University of California, Berkeley. 2013. Berkeley Multimodal Human Action Database (MHAD). Available: https://tele-immersion.citris-uc.org/berkeley_mhad. (Apr. 15, 2020)

[16] KTH-dataset, Schuldt, Laptev and Caputo, *Proc. ICPR'04*, *Cambridge*, *U*K. 2005. Recognition of human actions. Available: https://www.csc.kth.se/cvap/actions/. (Apr. 15, 2020)

[17] Stony Brook University. 2012. SBU Kinect Interaction Dataset. Available: https://www3.cs.stonybrook. edu/~kyun/research /kinect_interaction/index.html. (Apr. 15, 2020)

[18] Spatial and Temporal Resolution Up Conversion Team, ICST, Peking University. 2017. PKU-MMD. Available: http://www.icst.pku.edu.cn/struct/Projects/PKUMMD.html. (Apr. 15, 2020)