ABSTRACT

XIE, ZIYANG. Improving Safety During Driving and Human-robot Collaboration through Deep Learning and Reinforcement Learning. (Under the direction of Dr. Xu Xu).

Enhancing safety in transportation and workplaces is crucial for safeguarding people against injuries and lowering medical expenses. In response to this, previous studies have introduced numerous safety standards and equipment to enhance safety. Nevertheless, statistics indicate that every year transportation accidents and occupational injuries claim many lives, and the expenses associated with them remain considerable. This serves as a warning that safety concerns persist as a pressing issue that requires immediate attention.

Machines were once regarded as 'blind' and 'deaf'. However, recent advancements in sensor technology and associated algorithms have enabled ubiquitous sensing, allowing machines to detect their surroundings and the actions of humans around them. This ability enables machines to assist people in avoiding risks. For example, algorithms have been developed to support human motion tracking through sensors like wearable inertial measurement units (IMU) and optical RGB cameras. These algorithms enable machines to perceive the presence and movement of humans, thereby enabling them to proactively react to unsafe human behaviors. Given these prospects, this thesis concentrates on devising approaches that enhance safety during 1) driving and 2) human-robot collaboration using ubiquitous sensors.

Distracted driving results in thousands of fatal vehicle accidents every year. Recognizing distractive driving behaviors in real-time can help mitigate driving distraction and consequently improve road safety. To achieve this, IMUs were affixed to drivers' right wrists to monitor their body movements. Motion data were then fed into a detection algorithm that could recognize a potential distraction activity. A deep neural network classifier, which incorporated convolutional layers for regional features and recurrent layers for temporal features, classified these potential distraction activities into different categories. The classifier's F1-score was found to be 0.87, indicating a high level of accuracy.

Human-robot collaboration (HRC) has emerged as a popular work arrangement in industries in recent years. However, safety concerns have been raised regarding collisions that may occur during such

collaboration. To address this issue, one possible solution is to assist robots in comprehending and reacting to surrounding workers' position and posture. To achieve this, a computer vision-based human pose reconstruction algorithm was created, along with a collision avoidance mechanism that allows robots to move away from workers actively. The preliminary evaluation of this method suggests that it is effective in preventing collisions.

In addition, a reinforcement learning algorithm was developed using computer vision and optimization to tackle the issue of musculoskeletal disorders related to HRC. The algorithm optimizes the point of operation position during HRC through a two-step process: Firstly, a 3D human skeleton reconstruction method was employed to determine workers' continuous awkward posture (CAP) score. Then, an online gradient-based reinforcement learning algorithm was created to dynamically enhance workers' CAP score by modifying the robot end effector's positions and orientations. Empirical experiments demonstrated that this approach significantly improves the CAP score, resulting in point of operation positions similar to those chosen subjectively by participants.

Improving Safety During Driving and Human-robot Collaboration through Deep Learning and Reinforcement Learning

by
Ziyang Xie

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Industrial Engineering

Raleigh, North Carolina
2023

APPROVED BY:

_____	_____
Dr. Xu Xu Committee Chair	Dr. Karen Chen
_____	_____
Dr. Tianfu Wu	Dr. Yunan Liu

DEDICATION

To my parents, advisor, and friends for their unwavering support.

BIOGRAPHY

Ziyang Xie was born and raised in Xi'an, China, where he also received his education. In 2017, he earned a bachelor’s degree in Vehicle Engineering from the School of Mechanical Engineering at Xi'an Jiaotong University. Subsequently, he moved to the United States and obtained his master's degree in Mechanical Engineering from North Carolina State University (NCSU) in 2019. Currently, he is pursuing his Ph.D. degree in Industrial Engineering from Edward P. Fitts Department of Industrial and Systems Engineering at NCSU. Upon completion of his degree, he plans to start his career in industry.

TABLE OF CONTENTS

LIST OF TABLES vi
LIST OF FIGURES v
CHAPTER 1 INTRODUCTION 1
1.1 Prevalence of Safety Issues. 1
1.1.1 Distracted driving. 1
1.1.2 Human-robot collision 2
1.1.3 Musculoskeletal disorder in human-robot collaboration 2
1.2 Ubiquitous sensing and active safety systems 3
1.2.1 IMU based motion tracking system 4
1.2.2 Computer Vision based motion tracking 5
1.3 Machine learning applications in safety 6
1.3.1 Human activity recognition 6
1.3.2 Optimization 6
1.4 Problem statement 7
1.5 Outline 9
CHAPTER 2 DRIVING DISTRACTION CLASSIFICATION METHOD 11
2.1 Motivation 11
2.2 Method 14
2.2.1 Experiment setup and data collection 14
2.2.2 Real-time hand motion detector design 18
2.2.3 Deep neural network structure 20
2.2.4 Within-participant and between-participant manual distraction recognition 23
2.2.5 Performance evaluation 24
2.3 Result and discussion 24
2.3.1 Real-time hand motion detector validation 25
2.3.2 Within-participant distraction recognition 25
2.3.3 Between-participant distraction recognition 26
2.3.4 Distraction-specific results 30
2.3.5 Comparison with other works in driving distraction recognition 32
2.3.6 Potential applications 34
2.3.7 Limitations 35
2.4. Conclusions 35
CHAPTER 3 AN INMAGE-BASED HUMAN-ROBOT COLLISION AVOIDANCE SCHEME 37
3.1 Motivation 37
3.2 Method 40
3.2.1 Apparatus 40
3.2.2 Real-time worker pose reconstruction 40
3.2.3 Coordinate transform 40
3.2.4 Collision avoidance scheme 45
3.3. Results of the preliminary validation 49
3.4. Discussion 51
CHAPTER 4: OPTIMIZATION OF ROBOT POSTURE IN HUMAN-ROBOT COLLABORATION USING REINFORCEMENT LEARNING 54
4.1 Motivation 54
4.2 METHOD 57
4.2.1 Automated continuous awkward posture (CAP) score estimation 57
4.2.2 Gradient-based online learning algorithm in HRC (GOLA-HRC) 59

4.2.3 Experiment setup 64
4.2.4 Statistical analysis 66
4.3 Results 66
4.3.2 CAP scores 67
4.4 Discussions 69
4.4.1 Effectiveness of GOLA-HRC 69
4.4.2 Adaptability of GOLA-HRC 70
4.4.3 Safety awareness of workers in HRC 71
4.4.4 Personalization of CAP 73
4.5 Limitations 74
4.6 Conclusion 75
CHAPTER 5 CONCLUSION AND FUTURE WORK 76
REFERENCES 78

LIST OF TABLES

Table 2.1 The results of within-participant evaluation using leave-one-out cross-validation..... 26
Table 2.2 The results of between-participant evaluation using leave-one-out cross-validation..... 28
Table 2.3 The distraction-specific results of between-participant evaluation. 31
Table 2.4 Comparison of driving distraction-related works. The greatest value is in bold..... 34
Table 4.1 The values of initialization-parameters and hyper-parameters 63

LIST OF FIGURES

Figure 1.1 Two human-robot collaboration tasks performed in our Automation Lab. (a): A co-robot holds a deformable assembly base and the student mounts parts using a drill. (b) the student uses a pneumatic device to carefully level an assembly base, and the co-robot mounts small parts on this base. Given the nature of collaborative tasks, physically isolating the human workers from the co-robots is not applicable. 3

Figure 1.2 Individuals' postures when interacting with the same collaborative robot. Left: a taller worker. Right: a shorter worker. The Green lines show the reconstructed skeletons. As the figure shows, although the robot arm holds the object at the same position, workers' postures can be very different due to the anthropometry difference. 9

Figure 2.1 Simulator and IMU attachment. Left: A screenshot from the recorded video from the side. The participant was driving on the highway scenario. Right: The direction of x, y, and z axes. 15

Figure 2.2 Distractions and their corresponding IMU pattern. The top figures are the camera-recorded images, and the bottom figures are the 3-D acceleration data collected by the wrist-mounted IMU. 17

Figure 2.3 The workflow of the detector. Top: A streaming window of 200 frames was moving along the acceleration data of x-axis. Middle: the value of P3 corresponds to the absolute value of P2-P1, and is plotted on the same frame of P2 on the top figure. The detector compared the value of P3 with a preset threshold (red line) in real time (bottom figure). If P3 value is greater than the threshold at time frame T, the detector would record T-100 to T+600 as start and end frames. Bottom: IMU data from T-100 to T+600 was sent to DNNs for recognition. 20

Figure 2.4 Neural Networks structures. Yellow blocks represent neural network layers. This model starts with two 1D convolutional layers that extract movement features of each reshaped subsequence. A dropout layer and a max-pooling 1D layer follow the Conv1d layers to reduce overfitting and output size. Afterward, the output of max-pooling is flattened to 1D time series. The LSTM layer receives the output of convolutional layers and extracting temporal features. LSTM is connected to a fully connected layer with Softmax function, which makes predictions on the input. 21

Figure 2.5 The structure of the convolution layers. The input data is first segmented to a 6 (subsequences) × 33 (step length) × 3 (x, y, and z channels) tensor. Two Conv1D layers (64 filters, kernel size = 3, stride = 1) perform convolutional feature extraction, and the output of the second Conv1d is a 6 (subsequences) × 64 (outputs determined by filter size) × 29 (features) tensor. Afterwards, a max-pooling layer (pooling size = 2, stride = 2) pools the features and a flatten layer outputs 6 (subsequences) × 896 (features) sequential data. 22

Figure 2.6 The structure of LSTM layer. X1 to X6 are subsequences of 896 × 1 vectors

from the previous layers, and Yi is the output of LSTM units and input to a fully connected layer that makes predictions. In this study, N is set to 100. A dropout layer was embedded between Yi and the fully connected layer with a dropout rate of 0.4. 23

Figure 2.7 Drinking water with and without holding cup. The same distraction can have different temporal patterns if check-road action is involved. 29

Figure 3.1 Lab setup. Three markers (Pm1, Pm2 and Pm3) are placed on the ground. Xr, Yr and Zr indicate the x, y, and z axes of the robot coordinate system. Xc, Yc and Zc indicate the x, y, and z axes of the camera coordinate system. The camera is placed 6 meters away from the robot. 41

Figure 3.2 (a) The markers' position in the robot coordinate system. (b) The reconstructed human skeleton from VideoPose3D. A worker's head position and ankle position are labelled. (c)-(e) A worker first stood on marker Pm1 , then marker Pm2 and marker Pm3 . (f)-(h) The corresponding reconstructed worker's pose 43

Figure 3.3 (a): The 3D coordinates of the head position in camera coordinate system. (b): The 3D coordinates of the head position in robot coordinate system after the transformation. 45

Figure 3.4 The sketch of the two-level hazard zone. Current worker's position Phead(T) and the predicted worker's location Phead(T+t) are marked on the figure. 47

Figure 3.5 The proposed collision avoidance scheme. Alarming and robot responses with respect to different hazard levels are described. 48

Figure 3.6 Planned end effector trajectory for collision avoidance. Phead is a worker's head position V is a worker's speed vector, and c is the vector pointing from the worker to the robot end effector. Le1, Le2 and Le are described in Equation 6 49

Figure 3.7 An operator walked from the left to the right side of a co-robot. (a) Four key frames were captured and shown in one figure. (b) Top and middle plots show the position of the robot's end effector and the operator in x- and z-axis, respectively. The bottom plot shows the relative distance between the robot's end effector and the operator. The blue belt indicates the radius of level 1 hazard zone, and the pink belt indicates the radius of level 2 hazard zone. Note that the camera used to take this picture is not the one connected to PC for worker pose reconstruction. 50

Figure 3.8 The demonstration of trajectories of an operator and an end effector under different approaching paths. (a): An operator walked from -x to +x. (b): An operator walked from +z to -z. The arrows indicate the moving directions of the end effector and the operator. 51

Figure 4.1 A set position may not be suitable for workers with different body features. A taller worker (left) has to bend forward compared to a shorter worker

(right) for the same HRC task. 55

Figure 4.2 The workflow of worker pose reconstruction. An RGB camera first captures an image of the workspace. The computer vision model then reconstructs the worker's pose. 58

Figure 4.3 Comparison of RULA and the proposed CAP scores of the neck, upper arm, trunk, and lower arm. 59

Figure 4.4 The collaborative task: A camera is placed three meters away from the collaborative robot. The origin of the robot coordinate system is set on the robot base. The directions of translational axes x, y, and z and rotational axes α,β, and γ are shown. 60

Figure 4.5 Gradient-based online learning algorithm in HRC (GOLA-HRC). Left: flow chart of GOLA-HRC. Right: the algorithm pseudocode 64

Figure 4.6 Postures over 13 iterations until the CAP score met the termination criterion. 67

Figure 4.7 (a) Means and quantiles of CAP score corresponding to different end effector positions. The red box plots show the 90%, 75%, 50%, 25%, 10% quantiles. (b) Means and standard deviation of CAP scores of specific body joint angles. The results of are annotated: 'ns' – no significance; '*' – $p < 0.05$ 68

Figure 4.8 Summary of the questionnaire. The questionnaire includes three aspects 'Partnership', 'Behavior' and 'Safety', and each of them is evaluated by three questions. The questions were randomized, and the participants did not know the aspects when they were answering the questions. The results of Steel-Dwass tests are annotated: 'ns' – no significance; '*' – $p < 0.05$. 69

Figure 4.9 The postures of workers gazing behavior. The green lines are reconstructed skeletons and red dashed lines indicate the sight of the workers' eyes. The eyesight lines are closer to be vertical to the head skeleton lines in (b) and (d). 71

Figure 4.10 A worker's postures in Learnt and Worker-selected positions. The Worker-selected position poses greater risk on the neck, while the Learnt position balanced the 'Neck' and 'Upper Arm' CAP scores. 73

Figure 4.11 Left: The optimized posture in original CAP. Right: The optimized posture with doubled weight on neck angle. 74

CHAPTER 1 INTRODUCTION

1.1 Prevalence of Safety Issues

Maintaining safe living and working environments is a top priority because accidents can cause severe consequences such as injury and death. Improving safety can also reduce the medicals cost associated with injuries. A report from the U.S. Department of Labor shows that the total number of fatal occupational injuries in 2019 is more than five thousand (U.S. department of labor, 2019), which is the highest number since 2017, and the report from Insurance Institute for Highway Safety (IIHS) indicates that more than 36,000 people died in 2019 within the U.S due to driving-related accidents. In addition, as human-robot collaboration is blooming in recent years, collaborating with robots may require workers to perform a task with an awkward posture, which is a major risk factor of musculoskeletal disorders (MSDs). MSDs caused 273,000 day-away-from-work cases in the U.S in 2017. All these numbers alert us that nowadays safety issues remain severe. The scope of this dissertation covers a few safety-related topics, including driving distraction recognition, collisions avoidance during human-robot collaboration and musculoskeletal injury prevention.

1.1.1 Distracted driving.

Driving distraction, defined as 'driver engagement in internalized thoughts,' is widespread and hazardous (Regan et al., 2011). As reported by the National Highway Traffic Safety Administration (NHTSA), 3.2% of US drivers use cellphones while driving in 2018 (National Center for Statistics and Analysis, 2019). In the Netherlands, 50% of drivers were reported using electronic devices while driving in 2005 (Jeanne Breen Consulting, 2009). In 2018, more than 3000 fatal crashes were distraction-affected in the United States (NHTSA, 2019), which indicates that driving distraction remains a significant problem.

Distractions impair a driver's response time to critical events (Jeanne Breen Consulting, 2009) and increase the risk of accidents (Klauer et al., 2006). There are three types of driving distractions: manual distractions, visual distractions, and cognitive distractions, which indicate 'hands off the wheel',

'eyes off the road', and 'mind off the task', respectively (Rebecca L. Olson et al., n.d.). Visual and cognitive distractions consume a driver's attention during driving and affect the driver's performance (Berti & Schröger, 2001; Rebecca L. Olson et al., n.d.). Manual distractions physically detach the driver's hand off the wheel, which prolongs driver's physical response time (Maciej & Vollrath, 2009). These distractions consume the driver's attention and are harmful to road safety.

1.1.2 Human-robot collision

Human-robot collaboration is a blooming work configuration in which human workers and collaborative robots (co-robots) work together in a shared workspace. It is expected that the market for co-robots will proliferate at a compound annual growth rate of 61% in the next four years (Sherwani et al., 2020). However, potentially hazardous collisions during human-robot collaboration raise safety concerns. Since a co-robot works with human workers, it is not an option to isolate workers from co-robots for collision avoidance. Therefore, multiple engineering design features, such as limited end effector speed, torque sensors (Schäffer et al., 2008), and rounded exteriors, have been implemented in the industrial co-robot design to improve working safety (Matthias et al., 2011). Yet, the National Institute of Occupational Safety and Health (NIOSH, 2017) reported that the overall risk of human-robot collaboration is still concerning. Particularly, as co-robots were just introduced to the industry in recent years, human workers may have limited safety awareness of working with co-robots. If a worker underestimates a co-robot's motion profile or omits the presence of a nearby co-robot, serious injuries due to collision can still occur, particularly when the co-robot is holding a sharp work piece.

1.1.3 Musculoskeletal disorder in human-robot collaboration

Assembly is a common task in human-robot collaboration. In plants, robots hold the parts and workers manipulate and guide the robot to finish manufacturing tasks (Figure 1.1). However, this work configuration raises safety concerns. For instance, robots are programmed to hold the objects at set positions. As human workers may have different body dimensions, human-robot collaboration may result in human workers working in awkward postures. Previous epidemiology studies have shown that

awkward posture is a risk factor of musculoskeletal disorder (Anita et al., 2014), a very common occupational injury that uses up a huge number of medical resources (Denis et al., 2008).

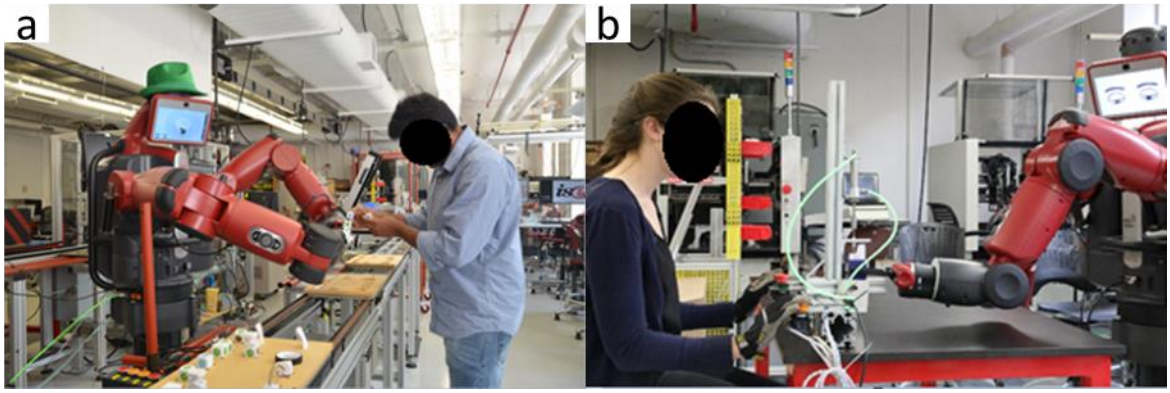


Figure 1.1 Two human-robot collaboration tasks performed in our Automation Lab. (a): A co-robot holds a deformable assembly base and the student mounts parts using a drill. (b) the student uses a pneumatic device to carefully level an assembly base, and the co-robot mounts small parts on this base. Given the nature of collaborative tasks, physically isolating the human workers from the co-robots is not applicable.

1.2 Ubiquitous sensing and active safety systems

In order to mitigate safety concerns and minimize economic losses, numerous safety-focused designs have been incorporated into the industrial and transportation sectors. For instance, to reduce the likelihood and impact of collisions between humans and robots, several engineering design features have been integrated into co-robot designs, including limited end effector speeds, torque sensors (Schäffer et al., 2008), and rounded exteriors, with the aim of enhancing workplace safety (Matthias et al., 2011). Moreover, to prevent driving distractions, automobile and cellphone manufacturers have disabled certain manipulations while driving.

In addition to the safety features of machines and automobiles, the advancement of human sensing technologies has spurred the development of active safety systems. The concept of ubiquitous sensing and intelligent systems has been proposed in recent years and has garnered significant research attention due to its potential to enhance safety (Paulovich et al., 2018). The fundamental purpose of

ubiquitous sensing and intelligent systems is to employ machine learning, big data, and sensing technology to create an intelligent environment. For instance, through sensing technology, machines can detect and analyze information about the behavior of humans in their surroundings, and then employ machine learning to make safety-related decisions. This read-and-decide process can be leveraged to improve safety with exceptional sensing accuracy.

To date, numerous accurate and dependable observational methods for human motion have been proposed, owing to the development of sensors and computer vision algorithms. These methods can be categorized based on their sensors as Inertial Measurement Unit (IMU) motion tracking systems (Flenniken et al., 2005) and computer vision (Colyer et al., 2018), which employ an RGB camera or an RGB-D camera to monitor human motion.

1.2.1 IMU based motion tracking system

IMUs have been utilized in navigation systems for a considerable time and are renowned for their precision, dependability, and real-time capabilities. Most commercially available IMUs today comprise accelerometers and gyroscopes, with each sensor possessing three degrees of freedom (DoF) in the x, y, and z-axes. By integrating the acceleration data from the accelerometers and angular velocity data from the gyroscopes, the motion trajectory of each IMU sensor can be calculated.

In recent years, IMUs have been utilized for the development of human motion tracking systems, thanks to their miniaturization to a few centimeters in size, making them easy to attach to the human body. Similar to the algorithm employed in navigation systems, an IMU can provide its real-time position and orientation data. By affixing a group of IMU sensors to human body joints, human motion and posture can be tracked. Compared to optical cameras, wearable IMUs are insensitive to lighting conditions or the field of view. Furthermore, IMUs raise fewer privacy concerns since they do not require cameras pointed at people and are preferred in private settings such as driving and home devices.

1.2.2 Computer vision based motion tracking

The domain of computer vision has witnessed an unprecedented boom in the last decade, mainly due to the remarkable success of AlexNet (Krizhevsky et al., 2012), a convolutional neural network that achieved unrivaled image recognition accuracy in the IMAGENET ILSVRC competition. Since then, the efficacy of deep convolutional neural networks (CNNs) in extracting information from images has been widely acknowledged, and several CNNs with varying architectures have been proposed in the field of computer vision. However, it has come to light that the computational cost of these networks is a major impediment, particularly for applications that require real-time image processing, which calls for innovative solutions (Al-Jarrah et al., 2015). Despite this limitation, the advent of graphical processing units (GPUs) has offered some relief. The capacity of these units has been increasing exponentially over the last decade, allowing for a wider range of applications for computer vision algorithms, especially in safety-related domains that necessitate real-time image recognition.

The potential of CNN has captured the attention of occupational practitioners who are interested in leveraging its capabilities for human pose reconstruction and activity recognition, tasks that traditionally require human labor and expertise (Cao et al., 2019; Pavllo et al., 2019; Yang et al., 2016). Furthermore, computer vision-based methods offer several advantages over IMU motion tracking systems. Notably, they do not necessitate the use of sensors attached to the human body, and are more cost-effective than IMU systems since they do not require any additional equipment beyond an optical camera. However, it is also important to acknowledge that computer vision-based motion tracking methods have their own set of limitations. Firstly, 3D human pose reconstruction techniques suffer from inaccuracies due to the loss of information from 2D images to 3D space, resulting in greater errors when compared to IMU-based motion tracking systems. Secondly, privacy concerns are raised since computer vision-based methods require at least one camera to be pointed towards the subject. Lastly, computer vision algorithms require the subject to remain within the field of vision of the camera, rendering it inconvenient for field applications.

1.3 Machine learning applications in safety

Machine learning has been applied to a wide range of safety-related topics, from predicting human errors while driving to identifying potential hazards on human robot collaboration. By learning from data and building analytical models, machine learning can help improve safety outcomes in various industries. The following subsections are elaborated topics related to human safety.

1.3.1 Human activity recognition

In recent years, there has been a growing interest in machine learning-based human activity recognition (HAR) tasks among researchers. With the availability of numerous open-source datasets (De-La-Hoz-Franco et al., 2018), various machine learning algorithms have been developed to recognize human motion, resulting in promising results. For example, Facebook's VideoPose3D model (Pavllo et al., 2019), can extract 3D information from a 2D image with an average joint estimation error of only 46.7mm. Building upon the motion tracking devices discussed in the previous section, current machine learning-based HAR methods can be categorized into sensor-based and image-based methods, depending on whether their raw data is temporal data or RGB images. Both sensor-based and image-based methods have been proven effective (Craye & Karray, 2015a; Jiang et al., 2018). With the ability to recognize human motion, machine learning-based methods have garnered much attention and have been applied to improve workplace safety (Koklonis et al., 2021), driving (Craye & Karray, 2015b; Jiang et al., 2018), and health care (Taylor et al., 2020; Torres-Huitzil & Alvarez-Landero, 2015). A key advantage of machine learning-based HAR is its ability to automatically recognize human activities without the need for human intervention, thereby increasing convenience and reducing costs.

1.3.2 Optimization

Over the last few decades, several optimization techniques have been proposed to obtain optimal solutions for linear and nonlinear problems. One such technique is continuous function approximation, which utilizes numerical calculations and approximation methods (Chong & Zak, 2004; Gill et al., 2019).

Recently, the reinforcement learning approach has gained popularity in solving optimization problems. The Markov decision process (MDP) is a well-established optimization technique that has been applied to a wide range of practical problems (Sutton & Barto, 2018). Along with MDP, other reinforcement learning algorithms like the Monte Carlo method, Q-learning, and artificial neural networks have also been developed to solve different problems. A significant advantage of modern reinforcement learning is its ability to operate without prior knowledge of a system, making it particularly useful in human-related problems where there is significant variation between individuals. Therefore, reinforcement learning has great potential to optimize safety-related problems effectively.

1.4 Problem statement

While machine learning has been utilized to improve worker safety, there exist several research gaps that hinder the implementation of these techniques in real-world scenarios. Within the scope of my dissertation, I will be addressing three such gaps, namely driving distraction recognition, human-robot collision avoidance, and optimization of the workers' posture during assembly tasks.

Driving distraction recognition Various studies have been conducted to recognize driving distractions using computer vision methods (Billah & Mahbubur Rahman, 2016; Craye & Karray, 2015a; N. Li & Busso, 2013) and wearable sensors (Abhayasinghe & Murray, 2014). However, privacy concerns have been a persistent issue in the application of computer vision-based distraction recognition, which hinders its widespread use (Martin et al., 2014; Prabhakar et al., 2003), On the other hand, sensor-based distraction recognition suffers from lower accuracy than computer vision-based methods (L. Li et al., 2020; Wollmer et al., 2011a) and higher computational costs (Streiffer et al., 2017). Nonetheless, recent research has proposed a novel deep neural network architecture comprising both convolutional and recurrent layers (Shi et al., 2015), that demonstrated high robustness and efficiency in classification (Hammerla et al., 2016). Thus, this new neural network architecture presents an opportunity to improve the performance of sensor-based driving distraction recognition.

Human-robot collision avoidance Prior investigations regarding human-robot collision avoidance have employed diverse sensing techniques, including capacitive proximity sensors (Schlegl et al., 2013), RGB-D sensors (Mohammed et al., 2017; Schmidt & Wang, 2013), and light projection (Vogel et al., 2017) to detect the position of a worker. Notwithstanding, the range of detection poses a common constraint among these methods, specifically, the aforesaid sensors' detection range falls short of 2.4 m × 3.4 m, rendering them unsuitable for industrial plant applications. However, recent strides in computer vision algorithms, exemplified by VideoPose3D (Pavllo et al., 2019), have made feasible the extraction of 3D positional information from a solitary RGB image, allowing collision avoidance with a standard camera. To overcome the limitations of detection range, we endeavor to leverage the VideoPose3D algorithm in Chapter 3.

Optimization of the workers' posture during assembly tasks As MSDs are predominantly caused by uncomfortable postures like extreme joint angles, it is important to improve the postures of workers to reduce the risk of these injuries. However, during human-robot collaboration, robots do not always adjust their end effectors to match the specific needs of each worker, which can result in less-than-ideal positions for some workers, leading to awkward postures and an increased likelihood of MSDs (Anita et al., 2014). Previous research and standards have aimed to find the optimal worker posture for lifting and handover tasks, such as the revised NIOSH lifting equation, which suggests that the end effector's height during a lifting task should be around 75 cm, approximately the average height of a worker's hand (Peternel et al., 2017; Waters et al., 1993). However, this method does not account for the variability of workers' body features and may place additional strain on workers who are above or below average height.

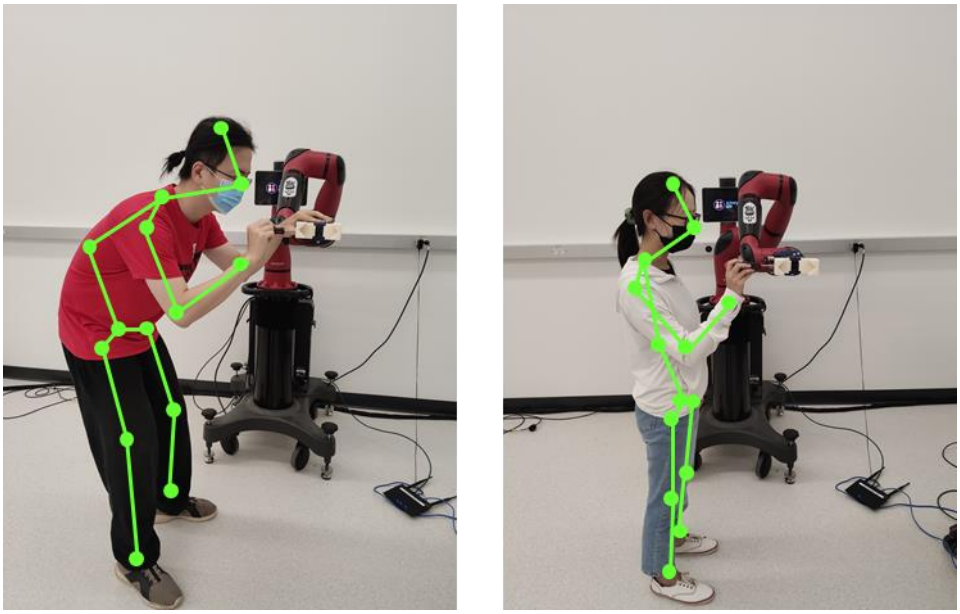


Figure 1.2 Individuals' postures when interacting with the same collaborative robot. Individuals' postures when interacting with the same collaborative robot. Left: a taller worker. Right: a shorter worker. The Green lines show the reconstructed skeletons. As the figure shows, although the robot arm holds the object at the same position, workers' postures can be very different due to the anthropometry difference.

In addition, extensive research has been conducted to optimize human-robot collaborative tasks through the use of biomechanical models, which calculate joint torque (Peternel et al., 2017). This approach involves creating a specific worker's biomechanical model and subsequently optimizing the position through numerical calculation. By accounting for variations in body features between workers, the method produces theoretically optimized postures. However, this technique has its limitations as it requires an accurate human body model construction, which necessitates the use of force plate and motion capture systems, and also demands expertise in biomechanics, leading to an increase in field application cost. Therefore, in this dissertation, a more readily applicable method will be presented in Chapter 4, which optimizes the robot's posture during collaboration through the use of reinforcement learning.

1.5 Outline

This dissertation is a compilation of three studies. Each study solves one problem mentioned above (Chapter 2 to 4). Specifically:

- Chapter 1 introduces the background of the problems, the motivation, existing literatures, current research gaps, and research objectives.
- Chapter 2 proposes a novel wearable sensor-based driving distraction recognition method.
- Chapter 3 presents a computer vision-based human-robot collision avoidance scheme.
- Chapter 4 proposes and validates a reinforcement learning approach to reduce MSD risks during collaborative assembly tasks.
- Chapter 5 summarizes the current findings and discusses future research directions.

CHAPTER 2 DRIVING DISTRACTION CLASSIFICATION METHOD

2.1 Motivation

Mitigation of distractions. Effectively detecting and mitigating driving distractions can help drivers avoid hazardous driving behaviors (Ayoub et al., 2019; Gallahan et al., 2013) and consequently reduce the likelihood of fatal car accidents (Y. Zhang et al., 2019). For example, as a distraction-mitigating system detects a driver is texting, the system can alarm the driver to focus back on the road. In autonomous driving, the distraction-detection system is a key component to evaluate the timing of transfer between manual and autonomous modes (Cummings et al., 2013). Therefore, effectively detecting distractions in real time is critical for driving safety.

Detection of distractions by cameras. Image-based distraction detection method is widely reported in recent years, but it also elicits concerns in applications. To date, video cameras (Billah & Mahbubur Rahman, 2016; Craye & Karray, 2015b; N. Li & Busso, 2013; Ragab et al., 2014; Streiffer et al., 2017) and the state-of-the-art computer vision algorithms have been widely used for distraction recognition. For example, the recognized driver's eye movement through video images can be used as a reliable indicator for monitoring visual and cognitive distractions (N. Li & Busso, 2013). However, camera-based distraction recognition also faces challenges in real-world implementations. Drivers may feel uncomfortable when they are videotaped by cameras (Martin et al., 2014; Prabhakar et al., 2003). In addition, recording video images needs a camera mounted in front of drivers, which may require retrofit of a car for continuous power supply and thus increase the cost. Because of these challenges, researchers are also seeking alternative ways to detect distracting behaviors.

Detection of distractions by motion tracking. Motion tracking is an ideal alternative to image-based detection methods. Note that many manual distractions are coupled with cognitive and / or visual distractions. For example, the physical movement of texting is coupled with both cognitive and visual distractions because texting requires visual and cognitive assistance. Thus, a variety of non-camera-based methods have been developed to infer driving distractions through driver's body motion (Prayudi & Kim,

2012; Voronin et al., 2012). For example, on-the-market distraction detection systems use a torque sensor mounted on steering wheel to detect whether a driver has both hands off the wheel (Zwicky et al., 2017). However, most manual distracting activities only involve one hand off the wheel. Therefore, a single torque sensor may be unable to provide the full picture of distracting activities, and further monitoring drivers' hand motion may be necessary.

A possible way to track drivers' hand motion is by using an inertial measurement unit (IMU). IMU is an electronic device that is comprised of accelerometers, gyroscopes, and sometimes magnetometers. To date, IMUs have been widely adopted in human activity recognition studies because they are low-cost and wearable (Abhayasinghe & Murray, 2014; Prayudi & Kim, 2012). In the context of driver's body motion tracking, wearable IMUs are individual-specific, not car-specific, so that they can measure the behaviors of drivers who drive multiple cars. On the other hand, using wearable IMU to infer distraction can be challenging because not all manual distraction activities are coupled with inattentive behaviors. For example, scratching one's face does not require cognitive or vision assistance. Alarming to non-coupled hand motion will not improve driver's attention status but could disturb drivers. Therefore, it is critical to develop a robust method that not only detects hands off the wheel output but also recognizes the type of behaviors from the IMU.

Algorithms for IMU-based driving distraction recognition. To date, different machine learning algorithms that are based on K-nearest neighbor (KNN), support vector machine (SVM), and deep neural networks (DNNs) have been developed and applied on wearable IMU data for human activity (Jiang et al., 2018; Voronin et al., 2012). Among these machine learning methods, KNN is less tolerant of noise and outliers (Kotsiantis et al., 2007). Thus, it is less suitable for the classification of driver's in-car motion as the movement features substantially vary from one individual to another (Park et al., 2010). SVM can separate features associated with a specific distracting activity (e.g., hand motion path of a touching head motion vs. a cellphone picking up motion) (Suykens & Vandewalle, 1999). However, compared to DNN,

SVM is less robust for the classification with activities that have similar body motion features (Wollmer et al., 2011b; Wöllmer et al., 2013).

A DNN is a type of artificial neural network with multiple layers between the input and output layers. Within the family of DNNs, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are the two most commonly used architectures for human activity recognition problems. CNNs are initially applied for image classification (Krizhevsky et al., 2012). It manipulates a single image in different ways and learns the global and local features of each category that needs to be classified. A well-trained CNN outperforms most of the other machine learning algorithms with respect to image classification (Iandola et al., 2016). A RNN consists of recursive nodes, which output the result back to their own. Such nodes help RNN nodes remember previous input. Modified from RNN, the long short-term memory (LSTM) solves the vanishing gradient problem in RNN (Gers et al., 1999) and makes RNN more robust on time-series data classification, such as natural language processing (Karafi & Cernock, 2010). LSTM has shown its advantages over CNN in temporal data classification as it could extract features across the time domain in a more effective way, which is essential in sequential data recognition (Karafi & Cernock, 2010). Very recently, researchers also sought to form ConvLSTM (Hammerla et al., 2016) by combining CNN and LSTM for those datasets with movement features and temporal features. Particularly, ConvLSTM outperforms existing architectures in human activity recognition problems in both accuracy and efficiency (Ordóñez & Roggen, 2016; Shi et al., 2015).

Research Gaps and Study Objective. Yet using a variety of DNNs to recognize driving distraction activities through drivers' body motion remains challenging in four aspects. First, as drivers are not distracted most of the time during driving, making predictions frame-by-frame can be computationally expensive and results in unstable recognition outcome in a real-time manner (Ragab et al., 2014). Second, there are great motion variabilities among people when performing the same manual distraction activity. Every driver imprints individual characters in IMU data spatially and temporally, which leads to difficulty in manual distraction recognition. Third, a driver can perform a variety of hand motions. Some hand

motions are not distraction-related (e.g., turning steering wheel), and some are distraction-related but not coupled with inattentive behavior (e.g., scratching face). The classification methods need to distinguish these hand motions from those coupled with inattentive behaviors. Fourth, considering the potential applications, the methods are preferred to be computationally affordable. In the study, we seek to fill these research gaps and develop a robust real-time driving distraction recognition method based on the acceleration data of driver's right wrist. It is hypothesized that ConvLSTM outperforms CNN and LSTM in distractive motion recognition since ConvLSTM is a more comprehensive feature extraction method.

2.2 Method

2.2.1 Experiment setup and data collection

For algorithm development purpose, we first conducted an experiment to collect driver's wrist motion during various distracted driving activities. Twenty participants (8 females and 12 males, all with valid driver's licenses) aged from 25 to 55 years old (mean = 38.8, SD = 10.2) were recruited. The average body mass index (BMI) was 26.9 kg/m^2 with SD = 5.8. The sample size is comparable to other widely used multimodal datasets in human activity recognition, such as Human 3.6M (11 participants) (Ionescu et al., 2014) and other reported works in Streiffer et al., (2017) (5 participants) (Streiffer et al., 2017) and Sztyler, (2019) (15 participants) (Sztyler, 2019). This research complied with the tenets of the Declaration of Helsinki and was approved by the New England Institutional Review Board. Informed consent was obtained from each participant.

Experiments were conducted on an RTI driving simulator (Ann Arbor, MI), which was a mock-up vehicle cockpit. The simulator had a fixed base with three displays placed approximately two meters away from the driver (Figure 2.1). Driving environments and traffic scenarios generated by RTI SimCreator and SimVista software were displayed in front of the driver. The simulated driving scenarios included both city roads and highways. City roads consisted of straight roads, intersections, stops with traffic lights, curved roads, while highways consisted of curved and straight roads. A cell phone, a touch panel, a water cup, and a few marker pens were placed near the driver's seat. A wearable IMU (MVN,

Xsens technologies, the Netherlands) system was attached to participants' upper body. The 3D linear acceleration of the wrist was collected by the wrist-mounted IMU at 100 Hz (Figure 2.1). Camcorders were placed around the driving simulator to record participants' driving behavior from back, side, and front during the experiment. The camcorder recorded videos at 29.97 Hz (standard NTSC scanning rate). The camcorder video and the recorded acceleration data were synchronized through a trigger event that can be identified from both the recorded video and the data acquisition system of the IMU sensor.

Figure 2.1 Simulator and IMU attachment. Left: A screenshot from the recorded video from the side. The participant was driving on the highway scenario. Right: The direction of x, y, and z axes.

Each participant was assigned to five experimental and one baseline driving trials. The experimental trials and baseline trial were assigned to each participant in a randomized sequence. Each trial lasted for approximately 15 minutes. In the baseline trial, the participants just performed regular driving without any instructions. In five experimental trials, verbal instructions were given to the participants, notifying participants to perform different distraction tasks approximately every 30 seconds. Five distractions were given to the driver in a random order in each experimental trial, and each distraction repeated five times per trial. These driving distraction tasks included talking on a cellphone (Phone), drinking water (Drink), taking a marker from the passenger r's seat and placing it into a cup holder (Marker), touching the infotainment system screen (TouchScreen), and texting (Texting). These five distraction tasks had been previously reported as those among the most common distraction tasks (Stutts et al., 2005). No instructions on driving pose or behaviors were given to the participants. Samples

of wrist acceleration showed that each type of distraction task had its unique IMU data pattern (Figure 2.2). The ground-truth timing of driving distractions was manually coded from the recorded video. Besides the five selected distractions, manual distractions that were not coupled with cognitive inattention during driving (e.g., scratching face) were also extracted from the experiment trials and labelled as 'Other'. Adding these non-coupled manual distractions can reduce the false alarm rate of the classifier.

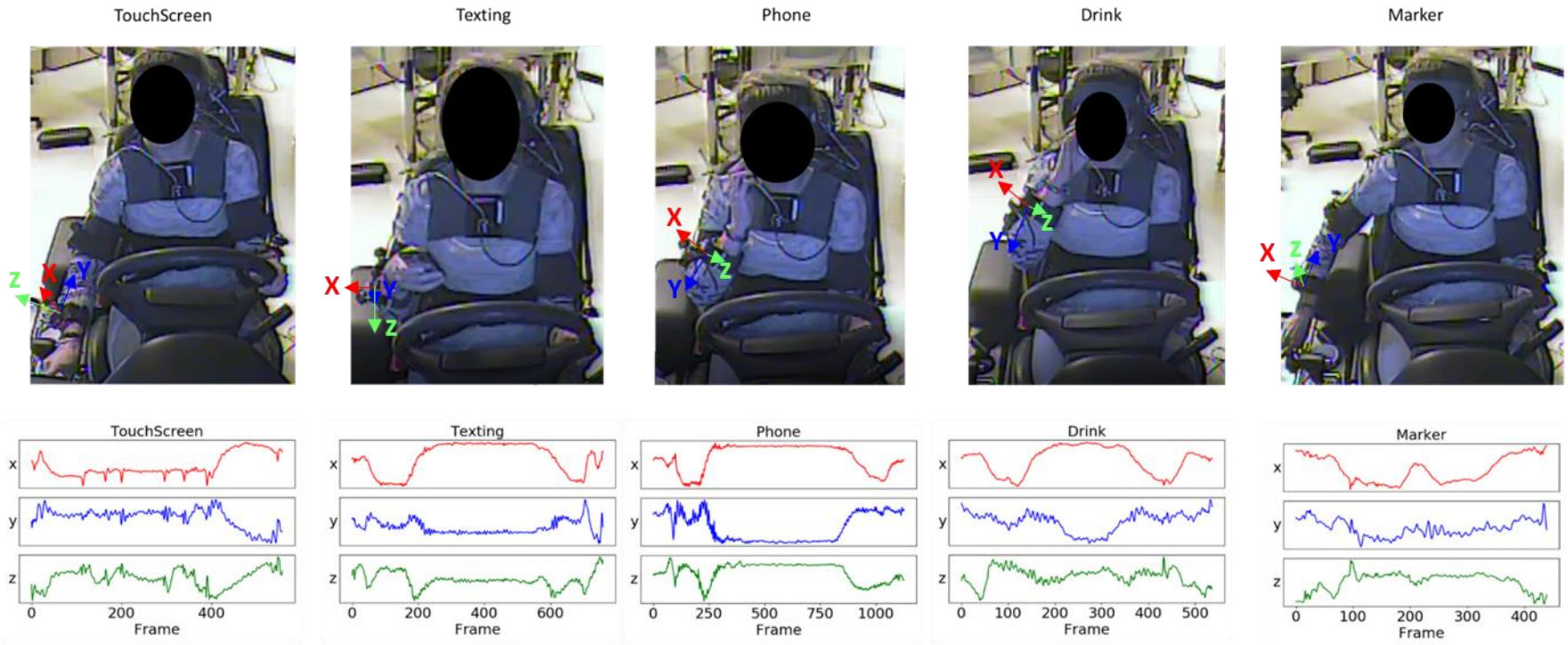


Figure 2.2 Distractions and their corresponding IMU pattern. The top figures are the camera-recorded images, and the bottom figures are the 3-D acceleration data collected by the wrist-mounted IMU.

2.2.2 Real-time hand motion detector design

A hand motion acceleration threshold based detector was first designed to detect a driver's distraction-like hand motion through streaming IMU data. The detector then sent the IMU data of potential manual distraction to the subsequent neural networks for recognizing driver's behavior. This way, the neural networks do not need to process IMU data all the time, and thus the computational cost can be reduced. The initial thought of detector design was to check the position of the right hand. However, calculating hand position over a long time by integrating the IMU-based acceleration would introduce substantial drift error (Flenniken et al., 2005). Note that the accelerometer readings were affected by the acceleration generated by the hand motion. Therefore, the raw accelerometer output could be used to indicate the initial hand motion during a potential manual distraction. As shown in Figure 2.1 and Figure 2.2, the x-axis of the wrist-mounted IMU was approximately perpendicular to the steering wheel axis and thus sensitive to hand detaching motion but less likely to be affected by a steering wheel-turning motion. In addition, as drivers move the wrist to perform eating, drinking, and other manual distractions, the projection of gravity on the x-axis also substantially changes. Therefore, in this study we chose x-axis acceleration data for the detection of potential manual distractions.

The hand motion detector monitored the driver's wrist motion through three steps. First, a moving window of 200 frames (two seconds) was adopted to scan the streaming data. Second, the first frame in the window was subtracted by the last frame, and the absolute value of the subtraction was compared with a preset threshold to infer whether there was a potential manual distraction (Figure 2.3). The time window length in the first step guaranteed the window can cover a substantial motion of driver's wrist, while the second step was designed to calculate the acceleration change between two points. The threshold of the detector determines whether a manual motion would be sent to the subsequent DNNs. A stricter threshold would filter out more distractive motions, and thus decreasing the number of false alarms but increasing the number of misses. Considering well-trained DNNs could further eliminate false alarms, the threshold needs to be set to a low level to send more potential distractive motions to the subsequent DNNs. A

small-scale independent pilot study was first conducted to decide the exact threshold value. From the pilot study, we found that a threshold of 5 m/s2 could detect more than 95% ground truth distractions while keeping a false alarm rate around 10%. We also found a higher or lower threshold did not help because they would miss more distractions or lead to more false alarms. Therefore, in this study, the threshold of the detector was set to 5 m/s2. Once the detector detected a potential manual distraction at time frame T, it would stop temporally and send IMU data from T-100 frame to T+600 frame to the subsequent DNNs. In other words, the period from one second before the triggering event and six seconds after was used for distraction recognition. This period of time included the initiation of the wrist movement, which was critical for the recognition of the type of distraction. Our preliminary test also showed that a shorter window could lead to difficulty of extracting enough unique features from the distracting activities with similarity, such as 'Texting' and 'Phone' where the initial hand motions were similar. On the other hand, a longer window does not further improve the performance of the detector but increases the computational cost. To further reduce the subsequent DNN computational cost, the data in this 700-frame window was down-sampled to 200 frames (from 100 Hz to 28 Hz). We chose 28 Hz because a previous study (Khusainov et al., 2013) had shown that this sampling frequency could collect enough information for activity recognition.

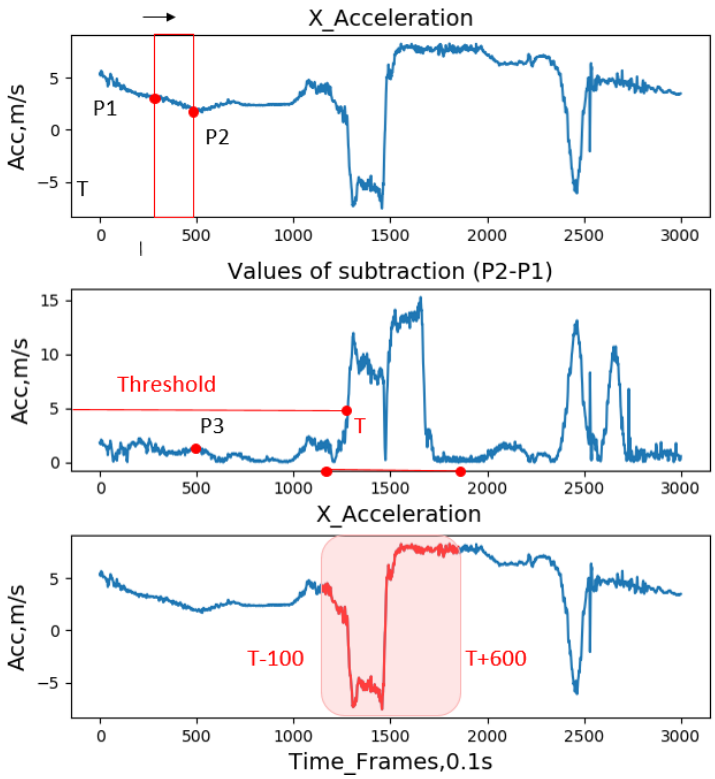


Figure 2.3 The workflow of the detector. Top: A streaming window of 200 frames was moving along the acceleration data of x-axis. Middle: the value of P3 corresponds to the absolute value of P2-P1, and is plotted on the same frame of P2 on the top figure. The detector compared the value of P3 with a preset threshold (red line) in real time (bottom figure). If P3 value is greater than the threshold at time frame T, the detector would record T-100 to T+600 as start and end frames. Bottom: IMU data from T-100 to T+600 was sent to DNNs for recognition.

2.2.3 Deep neural network structure

In this study, the network structure we mainly investigated was ConvLSTM, a deep neural network that included convolutional layers, recurrent layers, and fully connected layers (Figure 2.4). The IMU data was first reshaped to subsequences, which is essential in ConvLSTM because it converts input data from a matrix (channels × frames) to 3D time sequence matrices (subsequence × frames per subsequence × channels) that fit the LSTM structure. The key parameter of data reshaping is the step length (frames per subsequence), which determines each subsequence's time span. With a smaller step

length in a subsequence, convolutional layers make convolutional calculations on a smaller region in the raw data, which may lose relatively large movement patterns (e.g., move a cup from cup holder to mouth). On the other hand, as the input matrix's size is limited, and LSTM layer needs enough sequences to extract temporal features, the time span cannot be too large. This study reshaped the 200 frames to a 33 step-length × 6 subsequences matrix with the last two frames discarded. This way, each subsequence's step length is long enough for convolutional layers to extract motion features, and the six sequential subsequences can facilitate LSTM to extract the embedded temporal features.

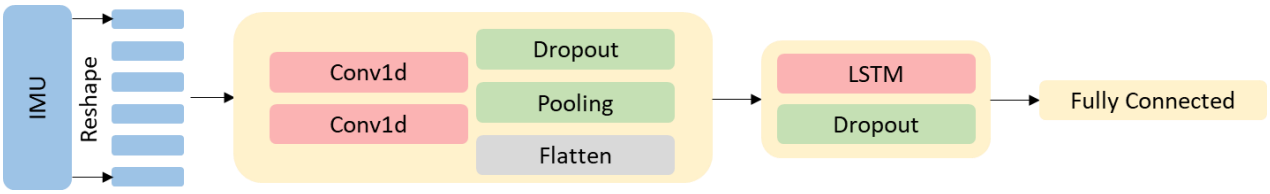


Figure 2.4 Neural Networks structures. Yellow blocks represent neural network layers. This model starts with two 1D convolutional layers that extract movement features of each reshaped subsequence. A dropout layer and a max-pooling 1D layer follow the Conv1d layers to reduce overfitting and output size. Afterward, the output of max-pooling is flattened to 1D time series. The LSTM layer receives the output of convolutional layers and extracting temporal features. LSTM is connected to a fully connected layer with Softmax function, which makes predictions on the input.

One-dimensional convolution layers (Ronao & Cho, 2016) were first adopted for extracting regional features from the IMU-based accelerations and decelerations of driver's right wrist (Figure 2.5). Considering the single and double integral of acceleration are velocity and position, respectively, the regional features extracted by convolutional layers corresponded to the movement features embedded in each subsequence. Two convolutional layers (64 filters, kernel size = 3, stride = 1) were stacked (Rawat & Wang, 2017), and followed by a dropout layer (dropout rate = 0.5) The dropout layer reduced potential overfitting and the computational cost (Srivastava et al., 2014). One flatten layer (Gulli & Pal, 2017) then flattened the tensor to one-dimensional data for subsequent LSTM layers. After flattening, the motion data contains 6 subsequences with 896 features.

LSTM layers took six subsequences sequentially. Each sequence included 896 elements which were movement features extracted from convolutional layers. LSTM layers analyzed these six subsequences temporally and connected movement features with their temporal relationship. Afterward, the temporally-connected movement features were sent to a Softmax activation function which normalized each element of the output vector to 0-1. The final prediction was made upon selecting the class represented by the element with the highest value in the output vector. The structure of LSTM layer and fully connected layer with Softmax function is demonstrated in Figure 2.6.

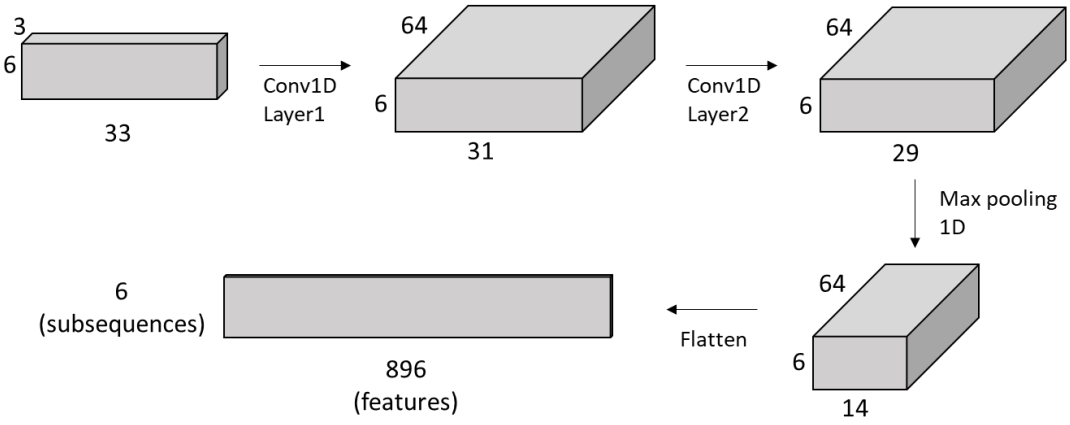


Figure 2.5 The structure of the convolution layers. The input data is first segmented to a 6 (subsequences) × 33 (step length) × 3 (x, y, and z channels) tensor. Two Conv1D layers (64 filters, kernel size = 3, stride = 1) perform convolutional feature extraction, and the output of the second Conv1d is a 6 (subsequences) × 64 (outputs determined by filter size) × 29 (features) tensor. Afterwards, a max-pooling layer (pooling size = 2, stride = 2) pools the features and a flatten layer outputs 6 (subsequences) × 896 (features) sequential data.

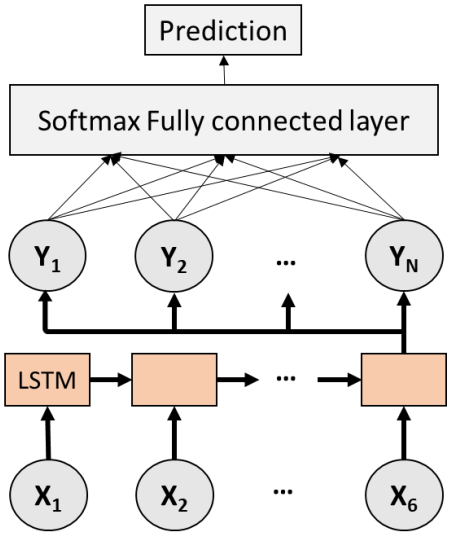


Figure 2.6 The structure of LSTM layer. X1 to X6 are subsequences of 896 × 1 vectors from the previous layers, and Yi is the output of LSTM units and input to a fully connected layer that makes predictions. In this study, N is set to 100. A dropout layer was embedded between Yi and the fully connected layer with a dropout rate of 0.4.

Besides ConvLSTM, two other baseline network structures, LSTM and CNN, were also designed and tested for the performance comparison in manual distraction recognition. The LSTM structure comprised recurrent and Softmax fully connected layers, which were similar to the LSTM part in the ConvLSTM structure. The CNN neural networks included a similar CNN structure in ConvLSTM and a Softmax fully connected layer.

2.2.4 Within-participant and between-participant manual distraction recognition

After the structural design, the DNNs were trained by the ground-truth data. As previously defined, the ground-truth data consisted of the five types of distractions as well as the sixth type of distraction named ‘Other’. Note that the training set consisted of ground-truth distractions, while the test set was the potential distractive motion identified by the detector. We evaluated the recognition performance in two ways: within a driver (within-participant) and between drivers (between-participant).

For within-participant evaluation, we investigated the performance of the DNNs when both the training set and the test set are from the same participant. Leave-one-out cross-validation was applied within each individual participant. For between-participant evaluation, we sought to understand the generalizability of the DNNs, and thus the leave-one-out cross-validation was applied across all participants.

2.2.5 Performance evaluation

The performances of the detector and the classifiers are evaluated by precision and recall. Precision is the number of correct detections over the total number of detected actions (Eq. 1), and recall is the number of correct detections over the number of ground-truth distractions (Eq. 2). Greater precision indicates fewer false alarms, while greater recall indicates less missing. The method's overall effectiveness is evaluated by F1-Score, which is a comprehensive evaluation considering both recall and precision (Goutte & Gaussier, 2005). An F1-score of 1.0 indicates the model prediction and ground-truth are identical.

$$Precision = \frac{tp}{tp+fp} \quad \text{Eq. 1}$$

where *tp* stands for true positive, *fp* for false positive, and *fn* for false negative.

$$Recall = \frac{tp}{tp+fn} \quad \text{Eq. 2}$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall+Precision} \quad \text{Eq. 3}$$

2.3 Result and discussion

During the experiments, some trials were unable to finish due to a variety of reasons. For instance, three participants experienced dizziness due to the discrepancy between the visual and vestibular feedback in a simulated environment and quit after one or two trials (participant #2 failed to record any of the experimental trials and was excluded from dataset). In some trials, the data was invalid due to technical issues like a loose IMU attachment and simulator software crash. In total, hand motion data of 1763 manual distractions was collected from the participants, including 363 ‘Phone’, 345 ‘Texting’, 347

'TouchScreen', 366 'Drink', and 342 'Marker'. In addition, there were 304 non-coupled manual distractions labeled separately as 'Other'. During the within-participant evaluation, only the data from the participants who finished at least five out six trials was analyzed to guarantee the size of the training set. Therefore, data from nine participants (1088 distractions, including 219 'Phone', 213 'Texting', 212 'TouchScreen', 222 'Drink' and 222 'Marker') was adopted for within-participant evaluation.

2.3.1 Real-time hand motion detector validation

Among the 1763 pre-defined ground-truth manual distractions, 1714 of them were successfully detected by the designed hand motion detector. There were 275 detected hand motions that were not included in the pre-defined distractions and were considered false alarms. These hand motions include scratching faces, touching arms, and other motions when drivers moved the hand off the wheel. The recall of our proposed detector is 0.97, and the precision is 0.86. Compared with a previously reported work (Bi et al., 2017), we achieved a similar recall value but lower precision. However, it should be noted that in the context of this study the false alarms can be further eliminated by the subsequent DNNs.

2.3.2 Within-participant distraction recognition

For within-participant evaluation, recall, precision, and F1-score of different DNNs are shown in Table 2.1. The Overall F1-scores of ConvLSTM, CNN, and LSTM are 0.87, 0.82 and 0.82, respectively. ConvLSTM outperforms CNN and LSTM on average F1-score, while CNN and LSTM are comparable. For some specific participants (e.g., participants #4 and #12), the F1-scores substantially differ between CNN and LSTM. This indicates the potential difference between the movement features and temporal features among these participants. A closer examination revealed that participant #4 demonstrated less clear movement features when performing distracting activities. For example, this participant tended to hold a cellphone in different positions while texting (e.g., on the side vs. in the front). This consequently caused discrepancies in the magnitude of acceleration data. It is difficult for CNN to learn this inconsistent movement feature so that CNN becomes less accurate on distraction recognition for participants #4. Besides, participants #12 showed substantial motion duration differences between city

road and highway, which can confuse LSTM model and make the LSTM model less accurate. Note that ConvLSTM consists of both CNN and LSTM structures. Therefore, ConvLSTM can extract features from both temporal features and movement features, resulting in a better recognition outcome than CNN or LSTM.

Table 2.1 The results of within-participant evaluation using leave-one-out cross-validation.

	ConvLSTM			CNN			LSTM		
Participant#	Recall	Precision	F1-score	Recall	Precision	F1-score	Recall	Precision	F1-score
4	0.96	0.88	0.92	0.73	0.86	0.79	0.85	0.86	0.86
6	0.89	0.74	0.81	0.72	0.80	0.76	0.88	0.72	0.79
8	0.89	0.84	0.86	0.88	0.86	0.87	0.85	0.81	0.83
9	0.92	0.85	0.86	0.75	0.83	0.79	0.79	0.80	0.79
11	0.84	0.86	0.85	0.82	0.84	0.83	0.82	0.84	0.83
12	0.97	0.90	0.93	0.98	0.91	0.94	0.89	0.82	0.86
16	0.86	0.93	0.89	0.83	0.89	0.86	0.88	0.78	0.83
17	0.91	0.82	0.86	0.82	0.84	0.83	0.81	0.89	0.85
19	0.80	0.84	0.82	0.79	0.70	0.74	0.81	0.75	0.78
Average	0.89	0.85	0.87	0.81	0.84	0.82	0.84	0.81	0.82

2.3.3 Between-participant distraction recognition

For between-participant evaluation, recall, precision, and F1-score of different DNNs are shown in Table 2.2. Overall F1-scores of ConvLSTM, CNN, and LSTM are 0.87, 0.76 and 0.85, respectively. Specifically, participants in the test dataset showed various F1-scores, which could be attributed to anthropometry variance among participants. For instance, we found that the participants with lower prediction accuracy (#20 and #17) had a greater body mass index value (BMI = 35 kg/m^2 and 40 kg/m^2, respectively). People with BMI greater than 30kg/m^2 is considered obesity (Deurenberg et al., 1998). Thus, it is possible that these two participants' IMU data patterns deviate from the major participants in training data due to limited joint range of motion as a result of obesity (Park et al., 2010). Body

anthropometry data has less influence on LSTM when compared with CNN. This is probably because LSTM extracts features on a larger scale where temporal information, such as a sequence of moving and duration of distraction, is dominant and less affected by anthropometry data.

Between ConvLSTM and LSTM, ConvLSTM outperforms LSTM in F1-score and computing speed. Further t-tests were performed. The results showed a significant difference between the F1-scores of ConvLSTM and LSTM ($p = 0.03$), and between the F1-scores of ConvLSTM and CNN ($p < 0.001$). In addition, ConvLSTM method has faster computational speed compared to LSTM in applications. This is because the input data only has three channels, but each channel contains a large number of frames. While LSTM needs to make recursive calculations on each time frame, ConvLSTM only makes recursive calculations on a subsequence. In our study, every distractive motion costs approximately 0.62ms for LSTM and 0.16ms for ConvLSTM with Intel Core i7-8700 CPU ($p < 0.001$).

Table 2.2 The results of between-participant evaluation using leave-one-out cross-validation.

	ConvLSTM			CNN			LSTM		
Participant#	Recall	Precision	F1-score	Recall	Precision	F1-score	Recall	Precision	F1-score
1	0.97	0.89	0.92	0.66	0.93	0.77	0.93	0.91	0.92
3	0.9	0.94	0.92	0.57	0.82	0.67	0.86	0.85	0.85
4	0.79	0.84	0.82	0.61	0.82	0.7	0.85	0.86	0.86
5	0.99	0.95	0.97	0.84	0.81	0.83	0.91	0.91	0.91
6	0.89	0.81	0.85	0.76	0.83	0.8	0.86	0.87	0.87
7	0.96	0.69	0.8	0.57	0.46	0.51	0.78	0.75	0.77
8	0.88	0.84	0.86	0.81	0.8	0.78	0.84	0.81	0.83
9	0.87	0.85	0.86	0.84	0.84	0.84	0.89	0.9	0.89
10	0.95	0.8	0.87	0.76	0.68	0.72	0.84	0.8	0.82
11	0.92	0.88	0.9	0.79	0.88	0.83	0.92	0.89	0.9
12	0.84	0.78	0.81	0.75	0.81	0.78	0.87	0.82	0.84
13	0.89	0.87	0.88	0.66	0.86	0.74	0.79	0.86	0.82
14	0.92	0.86	0.89	0.69	0.86	0.77	0.81	0.84	0.82
15	0.88	0.8	0.84	0.66	0.74	0.7	0.88	0.84	0.86
16	0.98	0.95	0.96	0.8	0.89	0.84	0.94	0.91	0.93
17	0.83	0.77	0.8	0.69	0.77	0.73	0.81	0.77	0.79
18	0.97	0.91	0.94	0.93	0.9	0.92	0.83	0.86	0.84
19	0.87	0.8	0.83	0.83	0.81	0.82	0.84	0.82	0.83
20	0.7	0.78	0.74	0.65	0.75	0.69	0.7	0.82	0.76
Average	0.9	0.84	0.87	0.73	0.8	0.76	0.85	0.85	0.85

Although temporal features are more consistent among the participants compared to the movement features, the variance of temporal features can also negatively affect the performance of DNNs. Some participants took a very short break during a distracting activity. For example, three steps needed to be done for drinking water: reach the cup, take the cup and drink, and put the cup back. Drivers could choose to finish these steps consecutively without a break. Alternatively, drivers can hold the cup and double-check the road to guarantee that it is safe to drink, especially when on city roads where traffic

conditions are complicated. Figure 2.7 shows two different ‘drinking strategy’ during driving. Because most participants adopted a consecutive way, there is a less accurate recognition of ‘drinking with a break’.

Other factors, such as random motions of hands and traffic conditions, can also influence the performance of DNNs. During driving, drivers do not always hold the steering wheel with two hands, but scratch their heads, trunk, or legs from time to time. These behaviors can trigger the detector, confuse the DNNs and cause false predictions. Besides, the traffic condition is also likely to influence motion patterns during distracting activities. When traffic condition is complicated, it requires more cognitive resources, which makes drivers finish distracting activities slower and consequently makes motion patterns less recognizable.

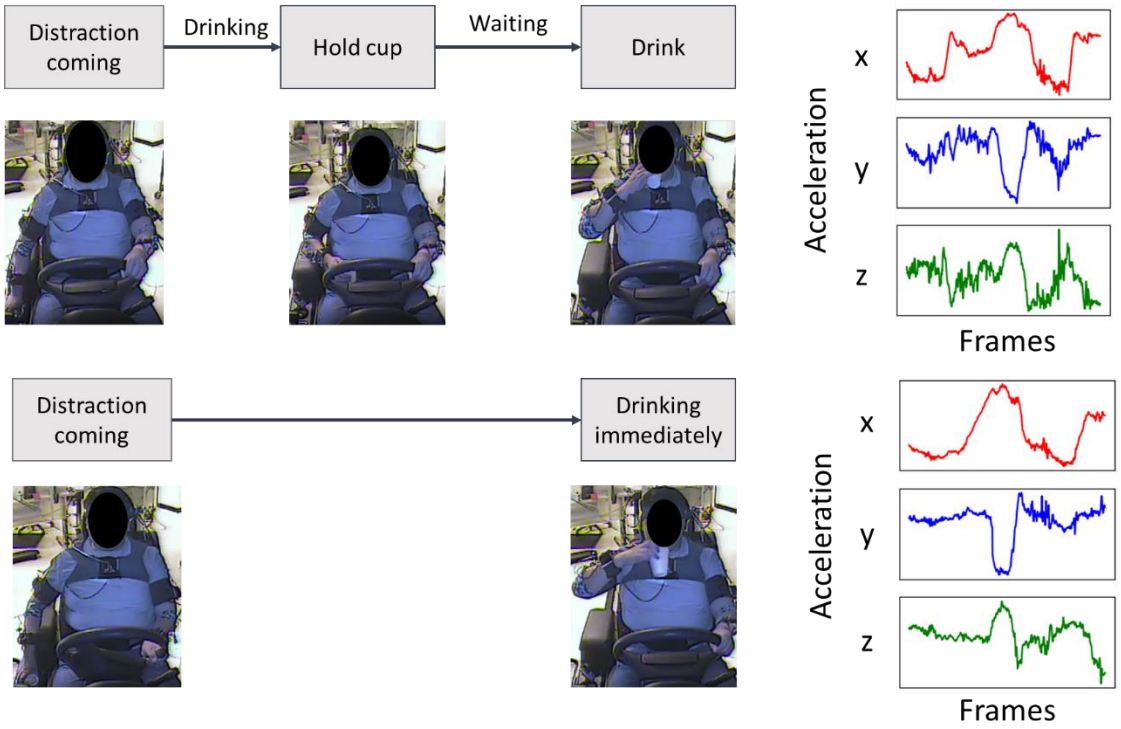


Figure 2.7 Drinking water with and without holding cup. The same distraction can have different temporal patterns if check-road action is involved.

Because repetitive body motions have a smaller within-participant variability than between-participant variability (Mathiassen et al., 2003; Perez & Nussbaum, 2006), human activity recognition is expected to perform better within a participant compared to between-participants. However, in this study the performance of ConvLSTM is similar between within-participant validation and between-participant validation. Since DNNs learn the embedded features of different distractions from training dataset, the size of training dataset determines how many samples DNNs can learn and plays an essential role in recognition accuracy. Note that there were only 90-120 distracting activities in a training dataset in within-participant validation, while there were 1600-1700 distracting activities in the training dataset in between-participant validation. Therefore, the lower-than-expected performance of DNNs during within-participants validation is likely due to the limited dataset size within a participant.

2.3.4 Distraction-specific results

Model performance varies in different distracting activities. Table 2.3 shows the recall, precision, and F1-score of each type of distracting activities. Figure 2.8 shows the confusion matrix of the detected distractions. Most of the non-diagonal elements in the confusion matrix are very small, which indicates most of the five pre-defined distracting activities can be correctly recognized. On the other hand, DNNs make more false predictions among those distracting activities with similar motions. For example, the motion of using touchscreen, texting and placing a marker pen are similar because drivers' wrists move slowly in the below-chest area. Therefore, greater values are shown in those corresponding matrix elements. Furthermore, based on the video, some participants tended to break their hand motions and temporarily placed phone or cup close to where marker pens were placed. Therefore, some of these distracting activities with a break are wrongly recognized as ‘Marker’.

The confusion matrix also reveals that DNNs show the worst performances on ‘Other’ categories. ‘Other’ includes all types of distracting activities other than pre-defined five (e.g., turning wheel, scratching head/leg). These distracting activities are introduced by the detector’s false alarm and considered as not coupled with high cognitive resources, so we do not consider them hazardous. Many

‘Other’ distracting activities are wrongly classified as ‘Marker’. Unlike the distractions that require drivers to manipulate an object, ‘Marker’ actions simply move a marker to a container without twisting the forearm or rotating the hand. Thus, ‘Marker’ shows less clear peaks and falls in IMU output, particularly in y- and z-axes (Figure 2.2). Also, ‘Marker’ shares similar patterns with some distractions in the category of ‘Other’, such as placing the hand on the knee or moving shift gear. In addition, ‘Marker’ takes a very short period, and this is temporally similar to the distractions in ‘Other’.

Table 2.3 The distraction-specific results of between-participant evaluation.

ConvLSTM	Phone	Texting	TouchScreen	Drink	Marker
Precision	0.85	0.86	0.87	0.87	0.95
Recall	0.92	0.92	0.85	0.93	0.68
F1-score	0.88	0.89	0.86	0.90	0.79
CNN	Phone	Texting	TouchScreen	Drink	Marker
Precision	0.80	0.79	0.86	0.71	0.64
Recall	0.93	0.86	0.87	0.87	0.57
F1-score	0.86	0.82	0.86	0.78	0.61
LSTM	Phone	Texting	TouchScreen	Drink	Marker
Precision	0.86	0.84	0.88	0.95	0.90
Recall	0.93	0.82	0.89	0.88	0.64
F1-score	0.89	0.83	0.88	0.91	0.75

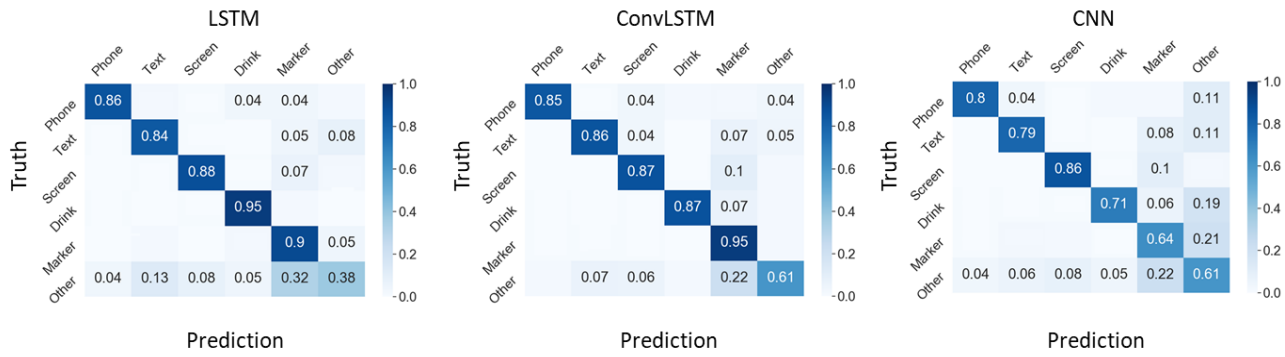


Figure 2.8 Confusion matrices of distractions by LSTM, ConvLSTM, and CNN. There are six types of distractions included. The 'Other's are distractions detected by the detector but not included in five pre-defined distractions.

2.3.5 Comparison with other works in driving distraction recognition

The accuracy level, type of input data, classification method, sample rate, and the number of recognition categories of this study and previous works in driving distraction recognition are listed in Table 2.4. This comparison shows that the accuracy of our method is comparable to Streiffer et al (2017), and greater than other previous works in multi-class distractive activity classification. This is potentially due to our proposed method and Streiffer's study both harnessing CNN for movement features and LSTM for temporal features, while others' works focus on image-based spatial features or IMU-based temporal features. In addition, compared with other works, our proposed method provides a hybrid solution to distractive driving recognitions that includes a detector and DNN classifier. Compared with frame-by-frame recognition, where DNNs need to recognize every frame of the data from a distraction's initiation to end, a hybrid method improves DNNs' accuracy by detecting the initiation of distractions. It should be noted that if the activity classification is limited to a binary classification (i.e., 'distracted' vs. 'not distracted'), applying LSTM on IMU data can reach an even greater accuracy (Wollmer et al., 2011a). On the other hand, the comparison in Table 2.4 should be taken cautiously because these works were conducted on the datasets with different participants, sensor type (cameras vs. IMUs) and noise level. Thus, a greater accuracy may not necessarily indicate one specific method outperforms another. Besides

DNNs, we also adopted a traditional classification algorithm, namely, linear discriminant analysis (LDA) for comparison. LDA is less computationally costly and could be an alternative to DNNs. The F1 score of LDA, however, turned to be only 0.38 in this application, which is much lower than the F1 score of ConvLSTM. This is possibly because LDA requires data in difference categories to follow Gaussian distribution with an equal variance. The human movement data in the current study, however, do not meet these requirements.

Table 2.4 Comparison of driving distraction-related works. The greatest value is in bold.

		classes	Accuracy	Input type	Method	Sample rate
Multi-class classification	(Streiffer et al., 2017)	6	**0.87**	Image+IMU	CNN+LSTM	4
	(Wollmer et al., 2011)	6	0.43	IMU	LSTM	2
	(L. Li et al., 2020)	6	0.59	image	Yolo	28
	Our proposed method	6	**0.87**	IMU	ConvLSTM	100
	(Ragab et al., 2014)	5	0.83	Image	RF model	20
	(Koesdwiady et al., 2017)	4	0.8	Image	CNN	-
Binary classification	(Wollmer et al., 2011)	2	**0.96**	IMU	LSTM	2
	(Atiquzzaman et al., 2018)	2	0.86	Image	RF model	1

2.3.6 Potential applications

Our proposed method demonstrated a good potential of using 3D acceleration data of the wrist to predict driving distraction. To date, most smartwatch has an integrated accelerometer for basic activity tracking (Guiry et al., 2014). Therefore, it is possible that in the future one can integrate this technique with a smartwatch for forming a more comprehensive driving distraction mitigation method. For instance, once a distracting driving activity is recognized through integrated accelerometers, a smartwatch can use the beeping sound to remind a driver to focus on driving. Regarding hardware feasibility, a prior research pointed out that the high computational cost of state-of-art DNNs can be a challenge in mobile devices due to up to millions of parameters one model can include (O. T.-C. Chen et al., 2020). However, the proposed ConvLSTM model in this study only has 79598 parameters in total and costs 0.16 ms for one recognition without utilizing a GPU, which makes this model feasible for mobile devices through TensorFlowLite, a DNN deployment environment developed by Google. In addition, the detector can filter out most of the normal driving activities, which further improves the calculation efficiency. In our

future studies, we will seek to conduct such integration and further evaluate its effectiveness in distractive driving mitigation.

2.3.7 Limitations

There are some limitations in this study that need to be addressed. First, the DNNs trained in this study are based on the wrist motion data collected in a driving simulator, which may be less representative of the actual wrist motion during real-world driving. In addition, the IMU system we adopted in this study only works when multiple IMU sensors are secured on participants' head, trunk, and upper arms and form a sensing system. Although we only used the raw acceleration data collected from the IMU sensor mounted on the right wrist, mounting additional IMU sensors using Velcro tapes on participants' upper body could slightly alter their natural behavior. In addition, the participants were instructed to perform distractions during the experiment. The body motion under verbal instruction might be slightly different than the natural impromptu motion. Due to the data-driven nature of DNN-based algorithms, the generalizability of the current DNN to the real-world driving needs to be taken with caution. Second, the proposed method requires drivers to wear an accelerometer on the wrist that is associated with the most distracting activities. If we later use a smartwatch to collect the wrist acceleration data, it is possible that a user prefers not to wear a watch on the distraction-related wrist, which could result in low user compliance. Third, in real driving, vehicles' acceleration during normal driving can introduce noise to the wrist-mounted IMU and negatively affect the recognition accuracy. Yet, an additional IMU rigidly fixed on the vehicle can be adopted to eliminate this noise by canceling out the vehicle acceleration. Fourth, our model shows limitations on activity recognition among the participants with greater BMI levels as discussed in section 3.3. However, this issue could be potentially solved by creating a more comprehensive data benchmark that includes overweight drivers.

2.4. Conclusions

In this study, we developed a method that can recognize driving distractions in real time based on wearable IMU sensor data. Only one IMU sensor was adopted in our method, which improves user

compliance and reduces the computational cost. The developed method consists of an acceleration-based hand motion detector and a DNN-based classifier. The detector can recognize 97% of driver's hand motion at the start of distracting activities. The DNN classifier further recognizes the type of distracting activities using seven seconds of IMU data. We selected ConvLSTM, LSTM, and CNN as DNN structures. Among these selected DNNs, ConvLSTM constantly outperforms CNN and LSTM, and shows an F1-score of 0.87 in within-participant and between-participant validation.

The proposed method shows its potential for the application of non-camera based real-time driving behavior monitoring and mitigation. This method could enable us to detect potentially hazardous driving behaviors in real time without camera-related privacy concerns. A driving distraction mitigation system can then use this information to remind drivers to refrain from performing distracting activities.

CHAPTER 3 AN INMAGE-BASED HUMAN-ROBOT COLLISION AVOIDANCE SCHEME

3.1 Motivation

In the previous chapter, the safety issue of driving distraction is solved using wearable sensors with neural network based classifiers. This chapter looks into the safety issues associated with the collisions in human-robot collaboration and proposes an easy-to-apply method to detect, warn. and avoid potential collisions.

One possible way to avoid the collision between a human worker and a co-robot is to inform the co-robot of workers' position and enable the co-robot to actively respond to worker's movement (Michalos et al., 2014). To achieve this goal, two important technical components need to be considered: human motion tracking and collision avoidance schemes.

Human motion tracking. A number of studies have proposed various methods or frameworks for human motion tracking (Halme et al., 2018). One of the most accurate measurement systems is laboratory-grade optical motion tracking system, which includes multiple cameras with markers attached to human body (Furtado et al., 2019). Despite its accuracy, applying such a motion tracking system in a plant for worker's body motion tracking can be challenging because of the bulky size and the high cost of the motion tracking system as well as the expertise required for running the system.

With the advancement in sensing technologies, a variety of sensors have also been applied to detect the existence of nearby human workers and to track their body motions in human-robot collaboration. For instance, capacitive proximity sensors can detect the change of electric field when a worker is approaching. Thus, they can be attached to a robot for helping the robot arm respond to potential collisions (Schlegl et al., 2013). However, the readings of capacitive proximity sensors can be affected by environmental factors such as temperature and humidity. Depth cameras (RGB-D sensors) have also attracted researchers' attention for collision avoidance (Mohammed et al., 2017; Schmidt & Wang, 2013). Depth cameras acquire depth information by analyzing the reflected structured light (Z.

Zhang, 2012). However, at least two depth cameras are needed to reconstruct surrounding 3D environments for collision avoidance purposes due to the limited view angle (Mohammed et al., 2017; Schmidt & Wang, 2013). The calibration process between multiple depth camera further increases the difficulty of field application. Researchers have also sought to project high-intensity light around a working area and used the continuity of reflected light to track workers' body motion (Vogel et al., 2017). This method, however, also requires calibrating multiple cameras, which makes it less practical in field applications.

In addition to the application difficulty, the detection range of sensors for human motion tracking is another concern that needs to be addressed. The detection range of the abovementioned sensors are typically smaller than $2.4\ m \times 3.4\ m$. Considering the average human walking speed is 1 m/s to 1.4 m/s (Bohannon & Andrews, 2011), a co-robot using abovementioned sensors would need to sense a fast approaching worker and complete collision avoidance actions in less than 1.5 second. On the other hand, the movement speed of a co-robot end effector is limited in accordance with co-robot safety standard (Matthias et al., 2011). For example, ISO/TS 15066:2016 proposed speed limits of a co-robot which depend on the distance between worker and a co-robot (ISO, 2016). Therefore, a small sensor detection range may incerase the risk of collision due to the lack of response time.

To address the application difficulty and the detection range limitation, in this study we propose to use a single regular RGB camera to track worker's body motion. In recent years, state-of-the-art open-source computer-vision algorithms, such as OpenPose (Cao et al., 2019), Detectron2 (Girshick, 2019) and VideoPose3D (Pavllo et al., 2019), have been developed to extract 2D and 3D human body posture from an image or a video clip captured by a regular camera. These computer-vision algorithms can also recognize humans from complicated backgrounds, which can help a co-robot to understand worker's existence in nearby areas. Additionally, using deep neural networks, the 3D pose reconstruction accuracy of Videopose3D is comparable with RGB-D sensors (Pavllo et al., 2019; Plantard et al., 2015).

Considering a single regular camera is less costly, easy to set up, and can monitor a large area, it may serve as an ideal alternative sensor for body motion tracking in human-robot collision avoidance.

Collision avoidance schemes. The collision avoidance schemes help co-robot evaluate the collision risk based on the tracked human motion and guide the co-robot to actively respond to an approaching worker. The risk of collision can be evaluated according to the distance between a worker and a co-robot. For example, a previous study defined a hazard zone of a co-robot (Wang et al., 2013). If a worker is in the hazard zone, the end effector of the co-robot would stop or retract. In addition, some studies further harness workers' moving speed to evaluate collision risks and proposed more comprehensive control strategies of end effectors (Chen & Song, 2018; Polverini et al., 2014; Mohammed et al., 2017). Besides moving the end-effector, one can also inform workers of potential collisions through auditory alarming (Stanton, 1994). Research on robot movement sonification indicates auditory signals can help a worker understand the motion of a robot (Frid et al., 2018; Zahray et al., 2020). In addition, visual alarms are also considered as an important warning channel and can improve the effectiveness of a warning system (Mendat & Wogalter, 2006).

In the current study, we propose a collision avoidance scheme that is in alignment with the previous studies. A two-level hazard zone will be first defined around a co-robot's end effector. Once a worker is detected in this hazard zone, depending on worker's position and walking speed output from computer-vision algorithms, a series of robot actions, such as applying visual and auditory signals as well as retracting robot arms, will be performed. The rest of the paper is organized as follows. The Method section describes the apparatus, data flow, and validation method of the proposed collision avoidance scheme. The Results section describes the outcome of a proof-of-concept validating process. The Discussion section discusses the validation outcomes, current limitations, and future works.

3.2 Method

3.2.1 Apparatus

A co-robot (Sawyer, Rethink Robotics) with 7 degrees of freedom was adopted in this study (Figure 3.1). The co-robot was connected to a workstation with an NVIDIA RTX 2080Ti GPU which supported the deployment of computer-vision algorithms. All computer-vision-related programming was developed using Python (ver. 3.6) on Linux Ubuntu (ver. 16.04). The robot control was realized by using Intera SDK platform (ver. 5.3), which is based on Robot Operating Systems (ROS Kinetic). A 100° wide-angle webcam (Model: MF920P, Spedal) was placed about 6 meters away from the co-robot and connected to the workstation (Figure 3.1). The camera placement ensured that worker's interactive motion with the co-robot can be fully covered.

3.2.2 Real-time worker pose reconstruction

The real-time worker pose reconstruction was realized by processing the successive image frames captured by the camera. Each frame was input to Detectron2 (Girshick, 2019), an open-source computer-vision package that is able to identify the 2D key joints of the human body in an image. The identified body joint positions in the image were then input to VideoPose3D (Pavllo et al., 2019), another open-source package that can reconstruct 3D human pose in camera coordinate system based on the 2D key body joint positions (Figure 3.2). With the workstation adopted in this study, analyzing one video frame takes 0.17 seconds, which is equivalent to a sampling rate of 5.8 Hz.

3.2.3 Coordinate transform

Note that Videopose3D outputs the 3D key body joint coordinates in the camera coordinate system (Figure 3.1). For robot communication, these coordinates need to be transformed to the co-robot's inherent coordinate system. Therefore, we developed a method to transform a coordinate from the camera coordinate system to the robot coordinate system.

The origin of the robot coordinate system is located on the top surface of the robot base, and the directions of each axis are shown in Figure 3.1. Three markers were labeled on the ground before the transforming process started (Figure 3.1). The first marker, $\mathbf{P_{m1}}$, was labeled at one meter in z+ direction (Figure 3.1) of the robot coordinate system. Thus, the coordinates of $\mathbf{P_{m1}}$ are $(0, H_{base}, 1)_r$ in the robot coordinate system, where H_{base} is the height of the robot base, and the right subscript "r" indicates this notation is with respect to the robot coordinate system. The other two markers, $\mathbf{P_{m2}}$ and $\mathbf{P_{m3}}$, were also labeled on the ground, whose coordinates in the robot coordinate system are $(1, H_{base}, 1)_r$ and $(0, H_{base}, 2)_r$, respectively (Figure 3.1).

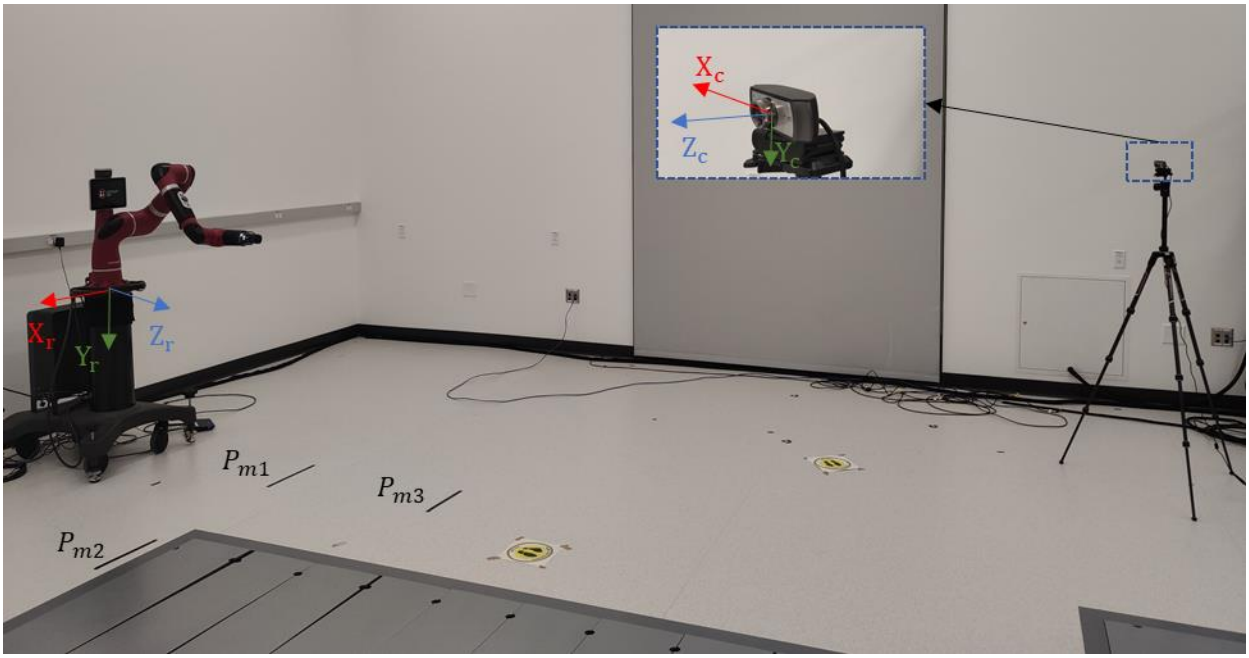


Figure 3.1 Lab setup. Three markers (P_{m1}, P_{m2} and P_{m3}) are placedon the ground. X_r, Y_r and Z_r indicate the x, y, and z axes of the robot coordinate system. X_c, Y_c and Z_c indicate the x, y, and z axes of the camera coordinate system. The camera is placed 6 meters away from the robot.

A worker then stood at marker one (Figure 3.2c and 3.2f). The coordinates of the mid-point of the worker's two ankles were recgonized as $\mathbf{P_{ankle\_m1}} = (X_{ankle\_m1}, Y_{ankle\_m1}, Z_{ankle\_m1})_c$ through VideoPose3D. The right subscript "c" indicates this notation is with respect to the camera coordinate system. The worker's head coordinates derived from VideoPose3D were $\mathbf{P_{head\_m1}} =$

$(X_{head\_m1}, Y_{head\_m1}, Z_{head\_m1})_c$, which corresponded to $\left(0, H_{base} - L_y, 1\right)_r$ in the robot coordinate system, where L_y is the worker's body height. The worker then moved to $\mathbf{P_{m2}}$ (Figure 3.2d and 3.2g), and the recorded head coordinates were $\mathbf{P_{head\_m2}} = (X_{head\_m2}, Y_{head\_m2}, Z_{head\_m2})_c$ in the camera coordinate system and $\mathbf{P_{head\_m2}} = \left(1, H_{base} - L_y, 1\right)_r$ in the robot coordinate system. Finally, the worker moved to $\mathbf{P_{m3}}$ (Figure 3.2e and 3.2h), and the recorded head coordinates were $\mathbf{P_{head\_m3}} = (X_{head\_m3}, Y_{head\_m3}, Z_{head\_m3})_c$ in the camera coordinate system and $\mathbf{P_{head\_m3}} = \left(0, H_{base} - L_y, 2\right)_r$. In the current lab setup, $L_x = L_y = 1m$ as markers are placed on the ground at 1 meter's distance from each other.

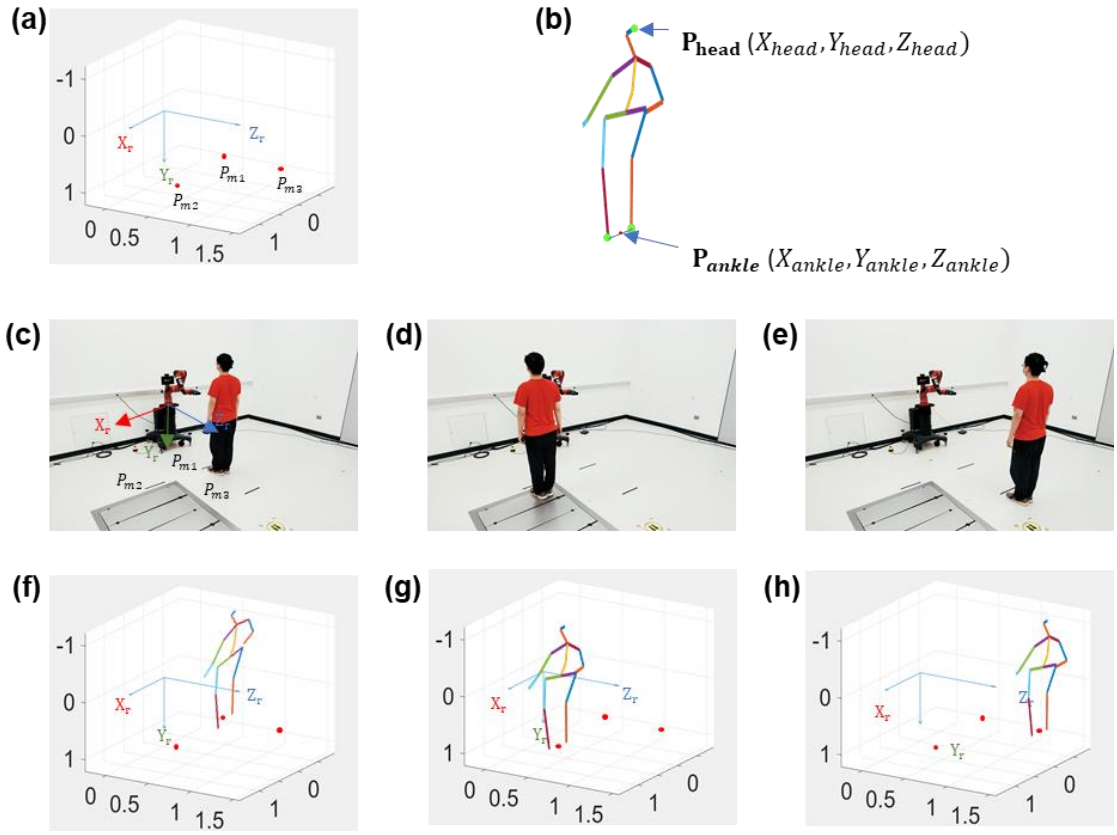


Figure 3.2 (a) The markers' position in the robot coordinate system. (b) The reconstructed human skeleton from VideoPose3D. A worker's head position and ankle position are labelled. (c)-(e) A worker first stood on marker P_{m1} , then marker P_{m2} and marker P_{m3} . (f)-(h) The corresponding reconstructed worker's pose

Suppose there is an arbitrary point **P**, whose coordinates are $(X, Y, Z)_c$ in the camera coordinate system, and $(X', Y', Z')_r$ in the robot coordinate system. The two coordinates can be bridged using Equation 1 and 2.

$$(X', Y', Z')_r = ((X, Y, Z)_c - (X_{ankle\_m1}, Y_{ankle\_m1}, Z_{ankle\_m1})_c) \cdot \mathbf{T}^{-1} + (0, H_{base}, 1)_r \qquad Eq.1$$

$$\mathbf{T} = \begin{pmatrix} \frac{|\vec{\boldsymbol{x}}_{\boldsymbol{c}}|}{L_x} & 0 & 0 \\ 0 & \frac{|\vec{\boldsymbol{y}}_{\boldsymbol{c}}|}{L_y} & 0 \\ 0 & 0 & \frac{|\vec{\boldsymbol{z}}_{\boldsymbol{c}}|}{L_z} \end{pmatrix} \cdot \begin{pmatrix} \frac{\vec{\boldsymbol{x}}_{\boldsymbol{c}}}{|\vec{\boldsymbol{x}}_{\boldsymbol{c}}|} \\ \frac{\vec{\boldsymbol{y}}_{\boldsymbol{c}}}{|\vec{\boldsymbol{y}}_{\boldsymbol{c}}|} \\ \frac{\vec{\boldsymbol{z}}_{\boldsymbol{c}}}{|\vec{\boldsymbol{z}}_{\boldsymbol{c}}|} \end{pmatrix}_{\boldsymbol{c}} \qquad Eq.2$$

where $\vec{\boldsymbol{x}}_{\boldsymbol{c}} = \left(\mathbf{P}_{\mathbf{head_{m2}}} - \mathbf{P}_{\mathbf{head_{m1}}}\right)_{\boldsymbol{c}}$, $\vec{\boldsymbol{y}}_{\boldsymbol{c}} = \left(\mathbf{P}_{\mathbf{ankle_{m1}}} - \mathbf{P}_{\mathbf{head_{m1}}}\right)_{\boldsymbol{c}}$, $\vec{\boldsymbol{z}}_{\boldsymbol{c}} = \left(\mathbf{P}_{\mathbf{head_{m3}}} - \mathbf{P}_{\mathbf{head_{m1}}}\right)_{\boldsymbol{c}}$

The first term on the right-hand side of Eq. 2 represents a scaling matrix, and the second term represents a rotational matrix. In addition, the mid-point of ankles is used to approximate the position of P_{m1} in this calibration process. It is worth noting that the above-mentioned process only needs to be performed once when a camera is initially placed by a co-robot.

To validate this coordinate transformation method, one performed the following body motions. First, this person moved from $\boldsymbol{P_{m1}}$ to $\boldsymbol{P_{m2}}$, and then moved back to $\boldsymbol{P_{m1}}$. The timing of this body motion is marked by dash lines 1 and 2 in Figure 3.3. This person then lowered the head by adopting a semi-squat pose and then stood up (line 3 and 4). Lastly, this person moved from $\boldsymbol{P_{m1}}$ to $\boldsymbol{P_{m3}}$, and then moved back to $\boldsymbol{P_{m1}}$ (line 5 and 6). The head motions during these body motions were aligned with the axes of the robot coordinate system. Before the coordinate transformation, the head motions output from the computer-vision algorithms in camera coordinate system were not aligned with any axis (Figure 3.3a). After the coordinate transformation, the head motions in the robot coordinate system were aligned with the main axes (Figure 3.3b).

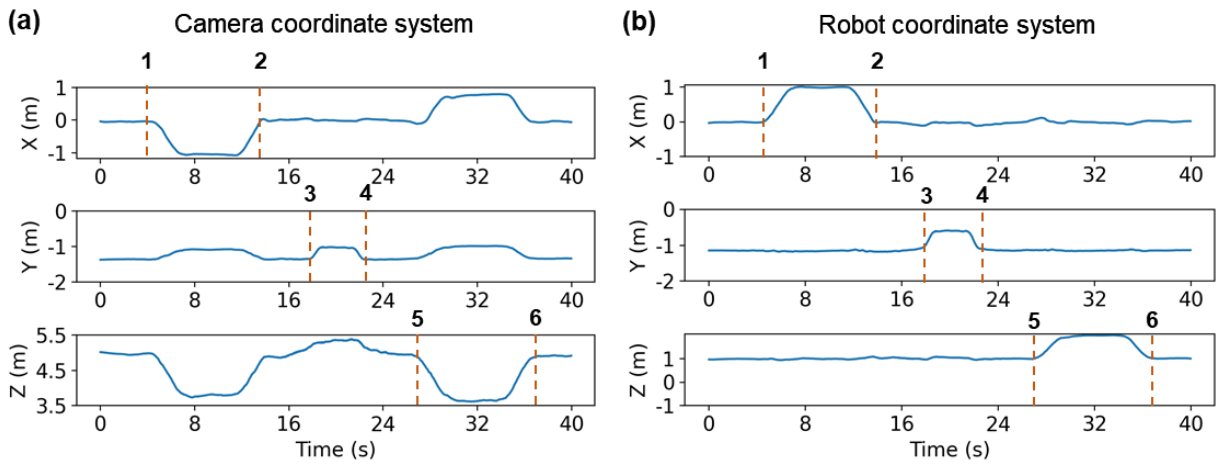


Figure 3.3 (a): The 3D coordinates of the head position in camera coordinate system. (b): The 3D coordinates of the head position in robot coordinate system after the transformation.

3.2.4 Collision avoidance scheme

3.2.4.1 Worker's position

To actively avoid human-robot collision, the robot should be aware of the worker's position and actively respond to potential collisions. In this study, the head position of a worker is adopted to indicate a worker's position. Given a worker's current position at time T is $P_{head}(T)$**,** the worker's head position after a short time t can be predicted as in Eq. 3:

$$\mathbf{P_{head}}(T+t) = \mathbf{P_{head}}(T) + t \cdot \vec{\mathbf{V}}_{\mathbf{head}} \qquad Eq.3$$

where $\vec{V}_{head}$ is the head's movement speed.

3.2.4.2 Defining hazard zone

In implementation, the parameter t should be adjusted based on the work configuration. In general, a longer time t can help a robot respond to a potential collision earlier, but the prediction accuracy can be negatively affected if workers frequently move at an inconstant speed or walk with

sudden change in direction. In our preliminary validation of the proposed collision avoidance scheme, we set t = 0.5 sec, a relatively short time that guarantees the prediction accuracy.

Based on the concept of hazard zone brought by ISO standards (ISO 10218) and previous studies (Mohammed et al., 2017; Wang et al., 2013), we propose a two-level hazard zone. Both levels are defined as a cylindrical volume whose center axis passes the end effector and is perpendicular to the ground (Figure 3.4). The second level hazard zone has a smaller radius compared to the first level hazard zone. One can calculate the ground-projected distance of $\boldsymbol{P_{head}}(T+t) - \boldsymbol{P_{effector}}(T)$ to determine whether a worker after a short time t will enter either level 1 or 2 of the hazard zone. The proposed scheme is to stop the robot motion and warn the worker through auditory and visual signals when $\boldsymbol{P_{head}}(T+t)$ is within level 1 zone but not within level 2 zone. If $\boldsymbol{P_{head}}(T+t)$ is within level 2 zone, robot end effector will retract to avoid collision in addition to auditory and visual signal.

Using predicted position $\boldsymbol{P_{head}}(T+t)$ rather than real position $\boldsymbol{P_{head}}(T)$ can provide some safety margin by alarming a worker earlier. If a worker is walking towards the robot's end effector, the robot can react t sec before the worker's real position $\boldsymbol{P_{head}}(T)$ enters the hazard zone (Figure 3.4). That means the robot has more time to retract its end effector. Although a larger hazard zone could provide additional safety margin, it could also lead to more false alarms (e.g., a worker is within a large hazard zone but not walking toward the robot).

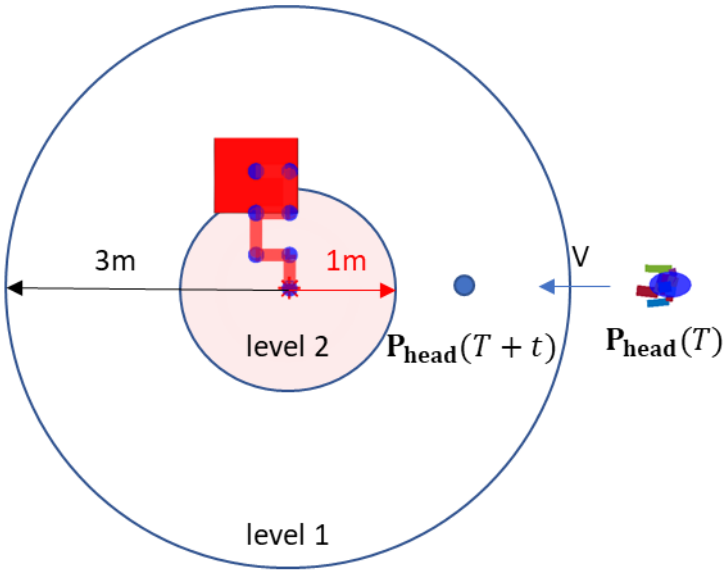


Figure 3.4 The sketch of the two-level hazard zone. Current worker's position $\boldsymbol{P_{head}}(T)$ and the predicted worker's location $\boldsymbol{P_{head}}(T+t)$ are marked on the figure.

To guarantee the robot can fully retracts its end effector before the potential collision, the radius of hazard zone can be estimated by Equation 4 and Equation 5.

$$R_{level\,2} \geq V_{worker} \cdot (T_{retract} - t) + R_{margin\,2} \qquad Eq.4$$

$$R_{level\,1} \geq R_{level\,2} + R_{margin\,1} \qquad Eq.5$$

where V_{worker} is the assumed worker's typical walking speed, $T_{retract}$ is the time for a robot to retract its end effector, $R_{margin1}$and $R_{margin\,2}$ are the optional additional safety margins whose values depend on the consquence of a collision.

In the current study we chose V_{worker} =1.4 m/s (Bohannon & Andrews, 2011), $T_{retract} = 1$ sec. (based on Saywer robot joint speed), $R_{margin2} = 0.3\ m$ and $R_{margin\,1} = 2\ m$. Thus, $R_{level\,2} = 1\ m$ and $R_{level\,1} = 3m$.

3.2.4.3 Auditory and visual alarm

Audio and visual alarms are considered as important alarming methods and widely used in alarming system design (Stanton, 1994). We designed a three-stage audio and visual alarm in the proposed collision avoidance scheme according to the Electronic Code of Federal Regulations (46 C.F.R. § 113.25-12) and American National Standards Institute code (ANSI 2535.1 safety color code) as shown in Figure 3.5. The audio alarms are pure tone sounds generated by a software (SoX, ver. 14.4.2) and played through a speaker. The visual alarm is displayed by a light bulb embedded on top of the robot.

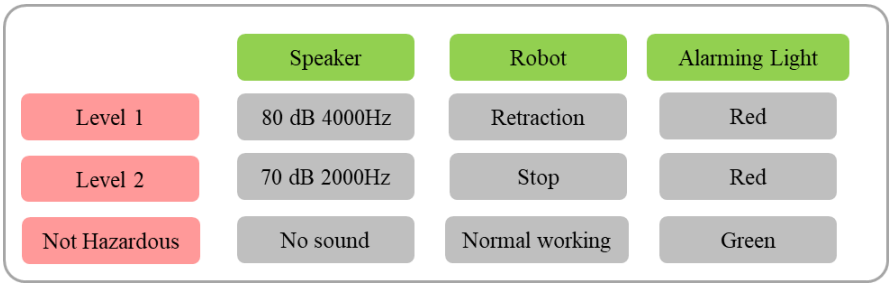


Figure 3.5 The proposed collision avoidance scheme. Alarming and robot responses with respect to different hazard levels are described.

3.2.4.4 End effector retraction

When a worker is in level 2 hazard zone, the robot will retract its end effector. Define, $\vec{\boldsymbol{c}} = |\boldsymbol{P_{effector}} - \boldsymbol{P_{head}}|_{ground}$, which is the vector from the worker's head to the robot's end effector projected to the ground. An end effector retraction motion parallel to **c** moves the end effector away from the worker (Figure 3.6) (J.-H. Chen & Song, 2018). $\vec{\boldsymbol{V}}_{\boldsymbol{ground}}$ is worker's walking speed vector projected on the ground. An end effector retraction motion parallel to $\vec{\boldsymbol{V}}_{\boldsymbol{ground}}$ moves the end effector away from worker's moving path (Mohammed et al., 2017). Because there are two possible end effector retraction directions, in this study we used the weighted summation to determine the end effector's retraction direction:

$$\vec{\boldsymbol{L}}_{\boldsymbol{e}} = \vec{\boldsymbol{L}}_{\boldsymbol{e1}} + \vec{\boldsymbol{L}}_{\boldsymbol{e2}} \qquad Eq.6$$

Where $\vec{L}_{e1} = \frac{P_{effector} - P_{head}}{|P_{effector} - P_{head}|} \cdot weigh\ factor\ 1$, $\vec{L}_{e2} = \frac{\vec{V}_{ground}}{|\vec{V}_{ground}|} \cdot weigh\ factor\ 2$

$weight\ factor\ 1$ and $weight\ factor\ 2$ can be set to different values depending on work configuration of the collaborative task. In this study, we tentatively set $weight\ factor\ 1 = 0.3$, and $weight\ factor\ 2 = 0.2$ according to our lab space setup and the direction a worker could approach. In addition, the end effector will also rotate in a way that its point of operation is pointing to the robot base during the retraction.

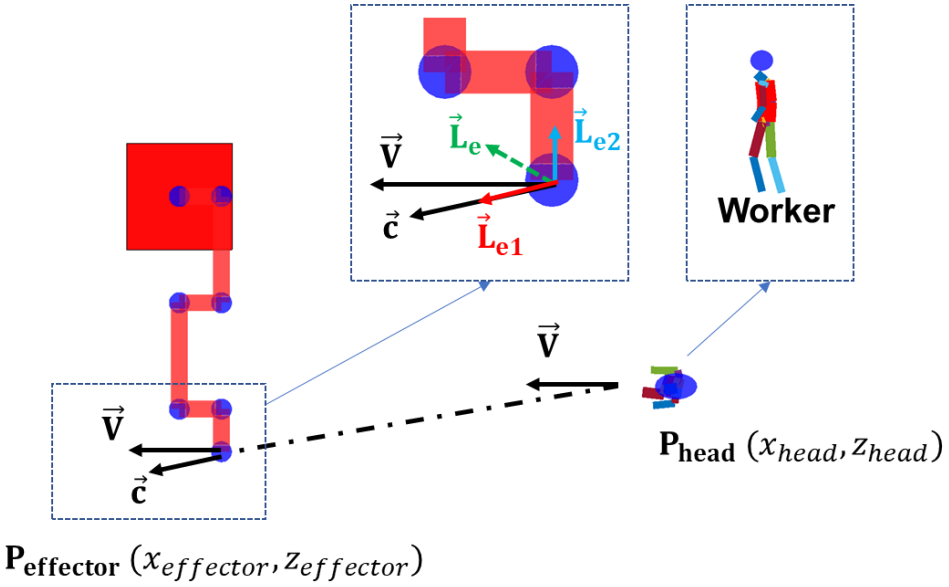


Figure 3.6 Planned end effector trajectory for collision avoidance. $\boldsymbol{P_{head}}$ is a worker's head position $\vec{\boldsymbol{V}}$ is a worker's speed vector, and $\vec{\boldsymbol{c}}$ is the vector pointing from the worker to the robot end effector. $\vec{\boldsymbol{L}}_{\boldsymbol{e1}}$, $\vec{\boldsymbol{L}}_{\boldsymbol{e2}}$ and $\vec{\boldsymbol{L}}_{\boldsymbol{e}}$ are described in Equation 6

3.3. Results of the preliminary validation

The functional validation of the proposed collision avoidance scheme is illustrated in Figure 3.7. An operator was walking from left to the right when the co-robot was in an expanded working posture and blocked the operator's moving path. The webcam was placed six meters away from the robot and its field of view was able to cover the operator's entire moving path.

The operator's initial position was at approximately three meters away from the robot's end effector. This operator then moved towards the co-robot end effector at time t = 0. The operator first entered level 1 hazard zone, and the co-robot's displayed red light, alarmed 2000 Hz tone at 70 dB, and stopped the end effector. As the operator kept walking and entered level 2 hazard zone at t = 1.0 s (shown in frame 1, Figure 3.7a), the intensity of the alarm sound increased to 80 dB and the frequency increased to 4000 Hz. In the meantime, the end effector started retraction (shown in frame 2 to frame 4, Figure 3.7a). Following the proposed collision avoidance algorithm, the co-robot successfully avoided a potential collision.

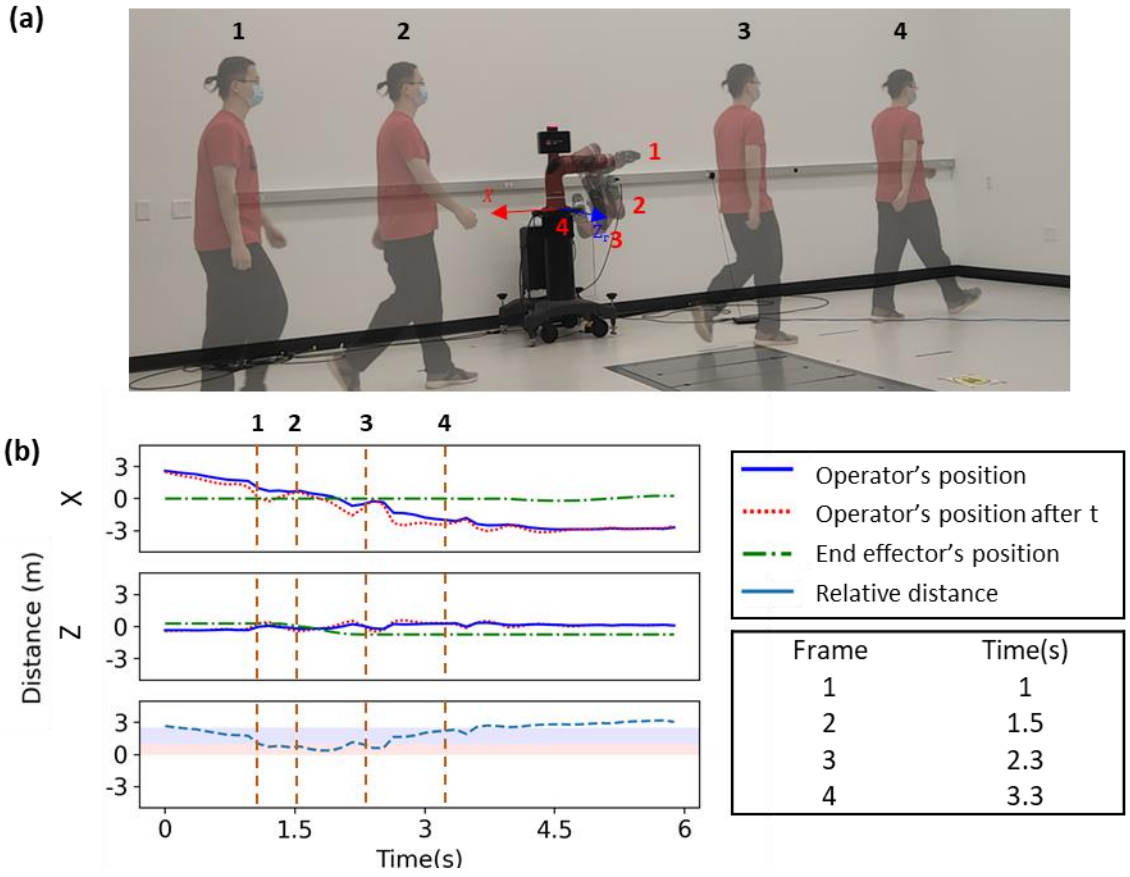


Figure 3.7 An operator walked from the left to the right side of a co-robot. (a) Four key frames were captured and shown in one figure. (b) Top and middle plots show the position of the robot's end effector and the operator in x- and z-axis, respectively. The bottom plot shows the relative distance between the robot's end effector and the operator. The blue belt indicates the radius of level 1 hazard zone, and the

pink belt indicates the radius of level 2 hazard zone. Note that the camera used to take this picture is not the one connected to PC for worker pose reconstruction.

In addition, we also tested two more scenarios where the operator approached from different directions. In the first scenario, the operator walked from right to left, and the co-robot end effector was at the middle of the planned walking path. In the second scenario, the operator walked towards the robot from the front. For both scenarios, the co-robot was able to retract the end effector in time and avoid collision. The trajectories of the operator and the co-robot's end effector in x-z plane of the robot coordinate system are shown in Figure 3.8.

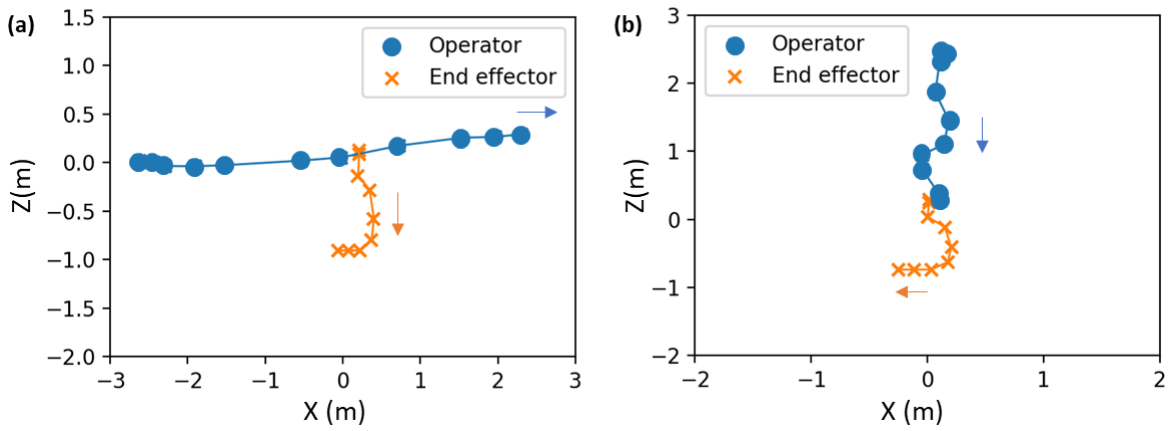


Figure 3.8 The demonstration of trajectories of an operator and an end effector under different approaching paths. (a): An operator walked from -x to +x. (b): An operator walked from +z to -z. The arrows indicate the moving directions of the end effector and the operator.

3.4. Discussion

This study proposed a collision avoidance scheme for human-robot collaboration using a single RGB camera and computer-vision algorithms. An alert mechanism and a collision avoidance algorithm were developed to mitigate the potential injuries due to collisions. The preliminary validation indicates the proposed method can help a robot to sense the existence of a nearby worker and to actively avoid potential collisions.

In the current study, the head position data is selected to represent a worker's position. This is because the computer-vision-based position data of upper and lower extremities, such as the hand and foot, are noisier then the position data of the head and trunk. Particularly, as the proposed method uses position and speed of a worker for robot action planning, a noise on worker's estimated position would lead to false alarm and trigger incorrect robot action. It should be noted, however, body segments other than the head, such as the shoulder and the elbow, may collide with a robot's end-effector first. Therefore, the radius of the hazard zone needs to be carefully set to ensure there is enough safety margin.

In addition, the current collision avoidance scheme assumes that a co-robot always needs to avoid an approaching worker. This might be in contradiction to the scenarios where a worker needed to approach a co-robot and pick items from its end effector, or stood by a co-robot and guided its motion. Therefore, the proposed collision avoidance method is mainly designed for the period of time when the interaction between a co-robot and a worker is not supposed to occur.

There are a few limitations that need to be addressed. First, our proposed scheme was applied to a co-robot whose end effector has 3 translational and 3 rotational degrees of freedom in the space. For those co-robots whose end effector motions are limited, the proposed end effector retraction algorithm may not apply. In such cases, one may pre-define a robot posture that minimizes the collision potential, and let the co-robot move its end effector to this posture if a worker approaches. Second, some camera lenses, such as a fisheye webcam, could result in excessive image distortions, and these distortions could lead to less accurate human pose reconstruction. While the distortion could be corrected through a calibration process (Z. Zhang, 2000), expertise in computer-vision would be required. Third, the reconstructed worker's posture maybe less accurate when the lower extremities of a worker are not presented in an image. Therefore, the camera should be placed at a high position where a worker's whole body can be observed. Yet, due to the workplace configuration, such a position for camera placement may not exist. Fourth, the position prediction is based on a worker's current walking speed, and assumes that the speed does not

significantly change in a short period of time. However, a sudden change in speed does occur in real-world (e.g., a worker makes a turn) and can introduce errors in the predicted worker's position.

CHAPTER 4: OPTIMIZATION OF ROBOT POSTURE IN HUMAN-ROBOT COLLABORATION USING REINFORCEMENT LEARNING

4.1 Motivation

The collision avoidance method developed in the previous chapter mitigates the occupational risks caused by collisions between human and robot. This chapter further investigates how to reduce the MSDs in human-robot collaboration scenario.

Musculoskeletal disorders (MSDs) are one of the most common occupational injuries in the industry (Kang et al., 2014; Stack et al., 2016). The major causes of MSDs include repetitive motions, awkward postures (Keyserling et al., 1992), and excessive force exertion (Stack et al., 2016). In 2017, approximately 273,000 day-away-from-work cases due to MSDs were reported in the U.S. (U.S. Bureau of Labor Statistics, 2020). One way to relieve workers from repetitive tasks and excessive force exertion is to adopt industrial robots (Gualtieri et al., 2021). For example, a variety of robots have been designed to transport heavy parts for human workers (Realyvásquez-Vargas et al., 2019; Vysocky & Novak, 2016), and consequently can help reduce the risk of low-back MSDs. Nevertheless, not all the tasks can be performed by robots alone. Some still require the inputs from human workers for quality control purposes, especially in advanced manufacturing and assembly (Bi et al., 2021; Pérez et al., 2020). This human-assisted work configuration is referred to as human-robot collaboration (HRC).

Yet, the risk of MSDs can still persist in HRC tasks. When robots are pre-programmed to work at a predetermined position, this position may not accommodate all workers due to individual differences such as body dimensions (Figure 4.1), preferences, and other personal characteristics. As a result, without customization, HRC can cause workers to adopt awkward postures and thus increase the risk of MSDs. (Anita et al., 2014). Specifically, awkward postures are defined as excessive joint bending and twisting outside a comfortable range of motion (Jaffar et al., 2011).

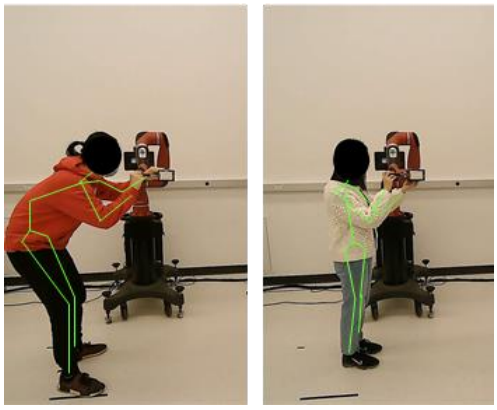


Figure 4.1 A set position may not be suitable for workers with different body features. A taller worker (left) has to bend forward compared to a shorter worker (right) for the same HRC task.

To reduce the risk of MSDs during HRC, the position of robot end effector should be adaptive and lead to a neutral posture for workers. To date, a number of optimization methods have been proposed to improve workers' postures during HRC by adjusting the position of robots' end effectors. Some studies have developed optimization methods based on joint loadings estimated through a biomechanical model or measured from electromyography (EMG) feedback (Kim et al., 2017; Peternel et al., 2017, 2018; van der Spaa et al., 2020). Yet, the estimated joint loading can be affected by the validity of the adopted biomechanical models. In some other studies (Busch et al., 2017, 2018; Roveda et al., 2020), optimizations are made to ergonomic assessment indicators. For example, Rapid Upper Limb Assessment (RULA) was used in a recent study as an indicator of workers' postural health during HRC (Liau & Ryu, 2020). RULA is a widely used and validated ergonomic tool in evaluating the overall MSD risks (Kee, 2020, 2021; McAtamney & Corlett, 1993a; Micheletti Cremasco et al., 2019). A greater RULA score indicates a greater risk of developing MSD.

Traditionally, assessment and calculation of RULA score require safety practitioners to manually code body postures. This process can be time-consuming and requires ergonomic expertise. In previous studies, researchers sought to improve this process by using wearable inertial measurement units (IMUs) to track workers' body postures and estimate the ergonomic score to optimize the position of a robot's

end effector and improve workers' postures (Busch et al., 2017, 2018). Nevertheless, wearable sensors may not be suitable in certain field applications because they may interfere worker's natural body motion during work. An alternative way to track body motion and automatically assess RULA score is to use camera and computer vision (Li & Xu, 2019; Manghisi et al., 2017; Massiris Fernández et al., 2020; Yazdani et al., 2021). Compared with IMUs, computer vision requires a camera set nearby a working area with good field of vision and an adequate computational power for pose reconstruction. On the other hand, computer vision does not require multiple IMU sensors attached on workers' body and thus does not interfere with workers' natural body motion. Considering the scenario of HRC, different workers may work with the same collaborative robot at different time in a day. Therefore, we choose computer vision to reconstruct each worker's pose and assess the corresponding RULA score. Otherwise, the HRC task could be interrupted by mounting IMU sensors on workers' body.

Once workers' ergonomics assessment scores are computed, different optimization algorithms can be applied to adjust positions of robots' end effector with an objective of minimizing workers' ergonomic assessment risk. Previous studies have proposed optimization methods where the ergonomic assessment was performed using simulated human pose during HRC tasks (Busch et al., 2017, 2018; Yazdani et al., 2021). However, simulation of human postures can be an ill-posed problem (Qu & Nussbaum, 2008) because 1) human body has a substantial amount of redundant degree of freedoms and 2) the range of motion limits may vary from person to person (Park et al., 2010).

In the current study, we will develop a novel model-free reinforcement learning (RL) method called Gradient-based Online Learning Algorithm in HRC (GOLA-HRC) to address these issues. This method is suitable in HRC tasks because the robot end effector can be programed to reach different positions from task to task, and from cycle to cycle. In contrast, a traditional workstation in automation only provides a fixed work position for workers. Even this position can be adjusted through ergonomics intervention, such adjustment is not as flexible as the movement of a robot end effector in an HRC task. This algorithm follows a trial-and-error method (Kober et al., 2013; Sutton & Barto, 2018) and optimizes

workers' awkward posture scores obtained from computer vision without the need for a whole-body biomechanical model. The data-driven nature of the proposed method allows the robot to “learn” the optimal effector position and provide personalized configurations for individual worker. Prior to this study, RL methods have been applied in robot action planning in different scenarios (Degris et al., 2012; Haarnoja, Ha, et al., 2018; Haarnoja, Pong, et al., 2018; Hu et al., 2019) and these applications exhibit promising potential in optimization tasks. The rest of this paper is organized as follows: the method section describes the adopted computer vision method, modified RULA score used in optimization, the proposed GOLA-HRC algorithm, and the experiment design for algorithm validation. The result section shows the validation outcomes. The discussion section discusses the key findings of this research, current limitations, and future works.

4.2 Method

4.2.1 Automated continuous awkward posture (CAP) score estimation

Worker posture reconstruction. The initial step is to equip the robot with the ability to determine the risk level of a worker's posture. To accomplish this, workers’ postures are first estimated with a computer vision model and the images collected from a single RGB camera. Specifically, the open-source computer vision model named VideoPose3D model (Pavllo et al., 2019) is used to predict 3D key joint positions of a worker based on frames captured by the camera (Figure 4.2). One advantage of the 3D computer vision model is that it allows for the analysis of off-plane movements, such as shoulder abduction. The mean absolute error of the estimated joint positions is reported to be 46.8 mm (Pavllo et al., 2019). Subsequently, the estimated joint positions are utilized to compute the relevant awkward posture score.

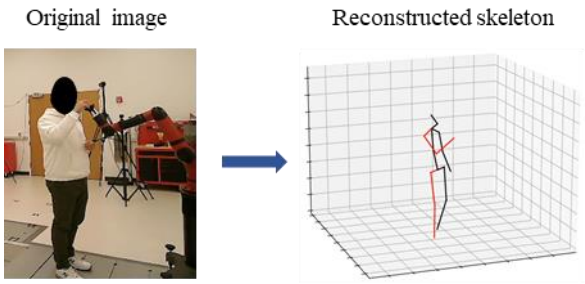


Figure 4.2 The workflow of worker pose reconstruction. An RGB camera first captures an image of the workspace. The computer vision model then reconstructs the worker's pose.

Continuous awkward posture score. Traditional RULA assessment evaluates overall scores of workers' postures by grouping the body part posture scores. The body part posture scores are step functions which are less sensitive when body joint angles are in the same category. For instance, trunk forward bending angles at 25 degree and 60 degree are treated as having the same body part score. Such a feature limits algorithms to find the optimization direction since there would be no gradient on RULA score. Thus, we propose to fit the RULA score into a continuous linear function (Figure 4.3), which is referred as continuous award posture (CAP) score hereinafter. The joint angles are calculated from the vectors of each body part. The flexion and extension is treated the same and This approach provides several benefits. The gradient of a linear function is constant, which helps stabilize the learning process, and the continuous nature of the CAP score allows for more precise optimization of workers' postures. In addition, it is important to note that we add a penalizing score of four to CAP whenever the workers' knee angle is over 20 degrees to penalize deep squatting postures. This penalizing score can also be implemented as a linear function. The joint scores are then summed with equal weighting factors to form the overall CAP score, which is used in the downstream reinforcement learning algorithm.

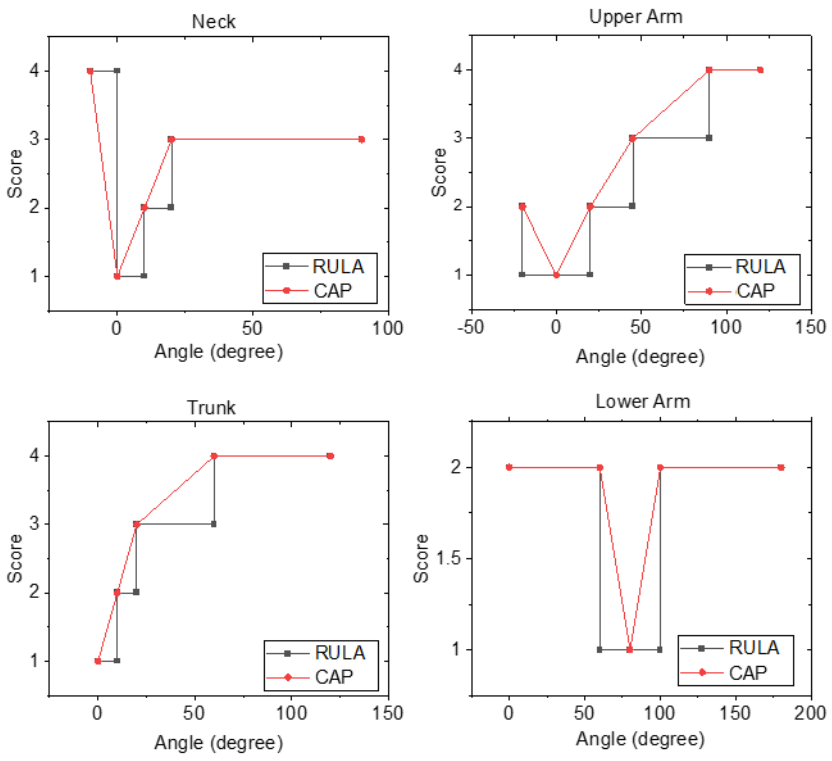


Figure 4.3 Comparison of RULA and the proposed CAP scores of the neck, upper arm, trunk, and lower arm.

4.2.2 Gradient-based online learning algorithm in HRC (GOLA-HRC)

Collaboration tasks. An HRC task was designed as a testbed for the development of the GOLA-HRC method. In this task, participants were asked to perform a fine wire insertion task where they were instructed to insert wires into a specific position on a breadboard gripped by a collaborative robot (Sawyer, Rethink Robotics). This task simulated a common human-assisted assembly task.

Robot degree of freedom to adjust. During the collaboration, workers' postures can be affected by the position and orientation of the robot's end effector. The adopted robot's end effector has six degrees of freedom (DoF), including three translational DoF (x, y, z, shown in Figure 4.4) and three rotational DoF (α, β, γ, Figure 4.4). Therefore, one can adjust the value of these six DoFs to affect workers' CAP scores. To reduce the state space (number of variables) and improve learning speed, some variables that are less likely to affect workers' posture (as workers can move in these directions freely) are excluded in

this study. Specifically, participants were encouraged to freely move during the experiment. Therefore, horizontal translational motions of the end effector (x and z) are less likely to affect workers' postures and could be excluded for the purpose of dimension reduction. Similarly, the rotational motion along the β-axis could be compensated by the horizontal movement of workers. In addition, the rotational motions along the γ-axis do not affect workers' postures due to the nature of the adopted fine wire insertion task, Therefore, there are two remained degree of freedom that can be adjusted for alternating workers' posture: the translational motion along the y-axis, which determines the height of the end effector, and the rotational motion along the α-axis, which determines the pitch angle of the end effector. The rest of the DoFs remain the same throughout the experiments.

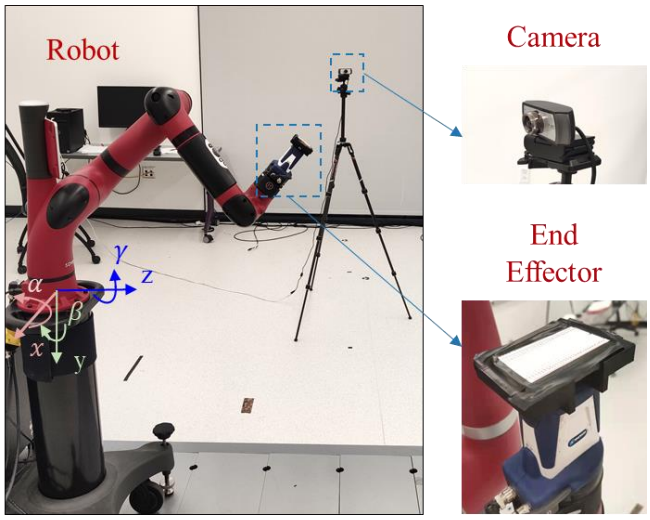


Figure 4.4 The collaborative task: A camera is placed three meters away from the collaborative robot. The origin of the robot coordinate system is set on the robot base. The directions of translational axes x, y, and z and rotational axes $\alpha, \beta, and\ \gamma$ are shown.

RL algorithm. The proposed GOLA-HRC algorithm searches for a set of values of the abovementioned DoF that minimizes the CAP score of a worker. The flowchart and algorithm are shown in Figure 4.5. With the computer vision, each time a worker finishes a fine wire insertion task at given end effector position and orientation, a robot posture-CAP score pair is generated (Figure 4.5). The

gradient indicating the optimization direction (Bottou, 2012; Duchi et al., 2011) can be then calculated by the robot posture-CAP pair. Following the gradient, the GOLA-HRC iterates until the termination conditions are met.

The gradients of CAP corresponding to the position changes along y-axis and orientation changes with respect to x-axis at each time step, ∇_{t_p} and ∇_{t_o}, are defined in Equation (1) and (2). These gradients indicate how the CAP can be improved or deteriorated by a translational motion or a rotational motion. Using the gradient, the end effector position and orientation at step t + 1 (P_{t+1} and O_{t+1}) are determined by the previous end effector position and orientation at time t (P_t and O_t) as well as the respective search step lengths (δh and δa) as described in Equation (3) ~ (6). In Equation (3) ~ (4), l_p and l_o are the learning rates set for adjusting the position and the orientation. We impose constraints s_{p} and s_{o} on the step lengths to guarantee a stable search process. A discount factor (d_α) is introduced to control the learning rate dynamically as a common practice in reinforcement learning (Sutton & Barto, 2018). A greater discounting factor improves the optimization accuracy but also increases the number of iterations. Learning rates l_p and l_o evolve after each iteration according to Equation (7) ~ (8). The termination condition is the condition algorithm stops iterating and terminates. In this study, the termination condition is defined in Equation (9), where ϵ is an error parameter that represents a tradeoff between convergence speed (i.e., how fast algorithm reaches optimal) and solution precision.

Note that the GOLA-HRC algorithm includes a set of initialization parameters ($\delta h_0, \delta a_0, l_{p0}, l_{o0}, P_0, O_0$) and hyper-parameters ($s_p, s_o, d_\alpha, \epsilon$) that constrain the behavior of the searching process, which is the nature of machine-learning algorithms (Yang & Shami, 2020). Initialization parameters are default values of the variables when the system initializes and are set before the training begins. The hyper-parameters define the behavior of the algorithm, such as how data is loaded (Snoek et al., 2012). In this study, hyper-parameters are manually initialized based on our previous experiences in computer vision and reinforcement learning. A grid search protocol was then adopted for tuning the hyper-parameters to improve the performance and convergence speed. Specifically, the termination

threshold ϵ should be carefully determined. We found a smaller value of ϵ may cause the algorithm fails to terminate. This is because there can be intra-individual posture difference even when the end effector stays at the same position. Such posture difference can introduce a certain level of noise to the gradients. On the other hand, when ϵ is greater, the algorithm converges faster but yields a less accurate solution. The initialization parameters are based on anthropometric data. Specifically, initial end effector position in the y-axis direction and orientation with respect to x-axis direction were set to 1.055 m and 0 degrees. With this setting, the breadboard is placed at the 50%ile elbow height of the entire population (male and female combined) (Freivalds & Niebel, 2008). The reason we choose the elbow height is that it is considered as a preferred height for assembly tasks in terms of reducing MSD risks (Freivalds & Niebel, 2008). The initial search step lengths were set as $\delta h_0 = 0.15m$ and $\delta a_0 = 20°$, and the initial learning rates were set as $l_{p0} = 1.1$ and $l_{o0} = 1.1$. These values were determined in a preliminary test for an adequate convergence speed. The values of the tuned hyper-parameters and initialization parameters are described in Table 4.1.

$$\nabla_{t_p} = (RULA_t^p - RULA_t) / \delta h \quad (1)$$

$$\nabla_{t_o} = (RULA_t^o - RULA_t) / \delta a \quad (2)$$

$$\delta h = \min\left(l_p \cdot \nabla_{t_p},\ s_p\right) \cdot \frac{\nabla_{t_p}}{|\nabla_{t_p}|} \quad (3)$$

$$\delta a = \min(l_o \cdot \nabla_{t_o},\ s_o) \cdot \frac{\nabla_{t_o}}{|\nabla_{t_o}|} \quad (4)$$

$$P_{t+1} = P_t + \delta h \quad (5)$$

$$O_{t+1} = O_t + \delta a \quad (6)$$

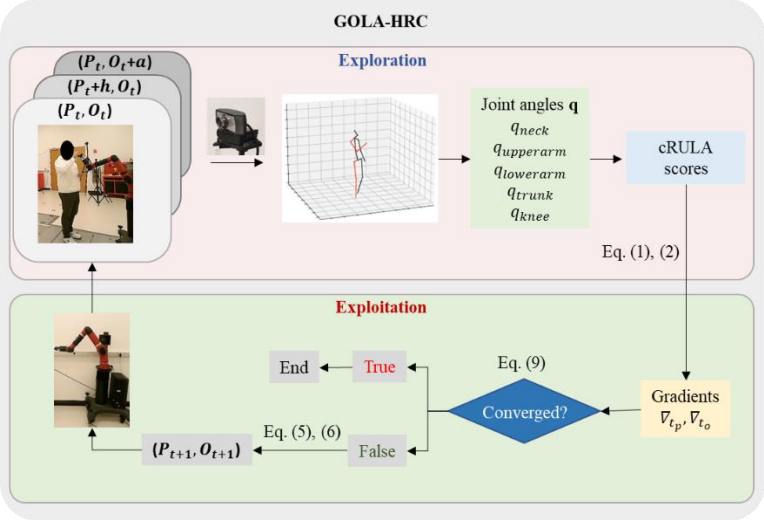
$$l_p = d_\alpha \cdot l_p \quad (7)$$

$$l_o = d_\alpha \cdot l_o \quad (8)$$

$$|CAP_t^p - CAP_t| < \epsilon \ and \ \ |CAP_t^o - CAP_t| < \epsilon \quad (9)$$

Table 4.1 The values of initialization-parameters and hyper-parameters

Parameters	In equation #	Values
$\boldsymbol{\delta h_0}$	1	0.15 m
$\boldsymbol{\delta a_0}$	2	20°
$\boldsymbol{P_0}$	5	1.055 m
$\boldsymbol{O_0}$	6	0°
$\boldsymbol{l_{p0}}$	7	1.1
$\boldsymbol{l_{o0}}$	8	1.1
$\boldsymbol{s_p}$	3	0.2 m
$\boldsymbol{s_o}$	4	30°
$\boldsymbol{d_\alpha}$	7, 8	0.9
$\boldsymbol{\epsilon}$	9	0.3



Algorithm 1: GOLA-HRC

Input: initial position and orientation P_0, O_0; searching search step lengths h_0, a_0; step limits s_p, s_o; learning rates l_p, l_o; discounting factor α

set $P = P_0$, O= O_0 move end effector to (P, O).

for t = 0,1,2,…..,N **do**

set $P_t = P_{t-1} + h_{t-1}, O_t = O_{t-1} + a_t$

Move to (P_t, O_t), Record $RULA_t$;

Move to $(P_t + h_t, O_t)$, record $RULA_t^p$;

Move to $(P_t, O_t + a_t)$, record $RULA_2^o$;

$\nabla_{t_p} = (CAP_t^p - CAP_t)/h_t$,

$\nabla_{t_o} = (CAP_t^o - CAP_t)/a_t$;

if $|\nabla_{t_p}| < \epsilon$ *and* $|\nabla_{t_o}| < \epsilon$ **then**:

end loop

$h_{t+1} = min\left(l_p \cdot \nabla_{t_p}, s_p\right) \cdot \frac{\nabla_{t_p}}{|\nabla_{t_p}|}$,

$a_{t+1} = min\left(l_o \cdot \nabla_{t_o}, s_o\right) \cdot \frac{\nabla_{t_o}}{|\nabla_{t_o}|}$;

$l_p = \alpha \cdot l_p, l_o = \alpha \cdot l_o$

Figure 4.5 Gradient-based online learning algorithm in HRC (GOLA-HRC). Left: flow chart of GOLA-HRC. Right: the algorithm pseudocode

4.2.3 Experiment setup

Apparatus and participants. As shown in Figure 4.4, a gripper was 3D printed and attached to the end effector (EGP-C 40, Schunk) of the collaborative robot to hold a breadboard (5.5 cm x 17 cm). The robot was connected to a workstation with a GPU (NVIDIA RTX 2080Ti) that supported the computer vision algorithms. All the computer vision algorithms, CAP score calculations, and the RL algorithm were programmed in Python (Ver. 3.6) on Linux platform (Ver. 16.04). Communication between the workstation and the robot was realized through Intera SDK (Ver. 5.3) based on Robot Operating Systems (ROS Kinetic). A webcam (Model: MF920P, Spedal) was placed three meters away from the robot to capture the images of the participants and then send the images to the workstation. Twenty participants were recruited (10 males and 10 females, with an average age of 27.3 ± 2.7 years old, height of 172 ± 11 cm, weight of 73 ± 15 kg) in this study. This research is in compliance with the tenets of the Declaration of Helsinki was approved by the Institutional Review Board at North Carolina State University (#24250). Informed consent was obtained from each participant.

Experiment process. To evaluate the effectiveness of the GOLA-HRC method, we performed a validation study. During the experiment, each participant was first asked to repetitively perform the fine wire insertion task until the estimated CAP scores converged with ϵ. After each repetition, the position and orientation of the robot's end effector was adjusted by applying the GOLA-HRC algorithm. The final position and orientation of the end effector with the converged CAP score was referred to as '*Learnt*' position hereafter. In addition, participants' CAP scores were also investigated when the end effector was placed at the 50%ile elbow height (referred to as '*Fixed*') as well as the computer-vision captured individual elbow height (i.e., the output of the Videopose3D right elbow position, referred to as '*Elbow Height*'). The orientations of the 'Fixed' and 'Elbow Height' are set to be 0°, which is the same orientation as shown in Figure 4.6 iteration 1. Furthermore, the participants were also asked to adjust the end effector to a position that they were most comfortable with, referred to as '*Worker-selected*' positions hereafter. This position was considered the subjectively optimal positions that resulted in minimal workload risk from a psychophysical perspective (Snook & Ciriello, 1991). During the experiment, the end effector was placed in each of the abovementioned four positions (*Learnt, Fixed, Elbow Height* and *Worker-selected*) with a random order. At each position, participants performed the fine wire insertion task three times and the CAP scores were calculated.

After inserting the fine wire at each tested end effector position, the participants were asked to complete a questionnaire. A questionnaire from a previous study on human-robot collaboration (Busch et al., 2017) was adopted in this study with some revisions. The participant was asked to rate whether the end effector position was appropriate from three aspects. Specifically, nine questions in 'Partnership', 'Performance' and 'Safety' were asked. The three 'Partnership' questions evaluating workers' feelings of a robot as a co-worker were: 'I reckon the robot is a good co-worker', 'I think the robot is not adapted to the handover task', and 'I prefer to finish the task alone'. The 'Performance' questions examining the extent to which the robot's behavior matched the worker's expectations were: 'I think the robot is trying to help me', 'I feel the robot's behavior is similar to my expectations', and 'I think the robot has helped

me keeping a good posture'. The questions on 'Safety' asking the workers to assess whether they felt the collaboration tasks were safe were: 'I believe I would feel exhausted repeating this task for hours', 'I feel comfortable picking up the object from the co-robot', and 'I would suffer from chronic pain working with the robot every day'. The Likert scale was used to quantify the response, with '0' representing 'extremely disagree' and '7' representing 'extremely agree'.

4.2.4 Statistical analysis

Nonparametric statistical tests were performed in this study considering the collected data does not follow a normal distribution ($p < 0.05$ in Shapiro-Wilk and Anderson Darling tests). Specifically, Kruskal–Wallis one-way analysis of variance was performed to investigate whether different end effector positions had significant effects on CAP and the subjective questionnaire scores. Steel-Dwass test was then performed for pairwise comparisons.

4.3 Results

4.3.1 CAP score vs iterations

Figure 4.6 shows how the GOLA-HRC improved the working posture of a participant step by step until the end effector moves to the *Learnt* position. In this figure, the initial position was too low for the participant and resulted in a forward trunk bending posture. By calculating the gradient of the CAP score, the algorithm adjusted the position and orientation of the end effector until the CAP score converged. In average, the proposed algorithm converges after 12.4 iterations.

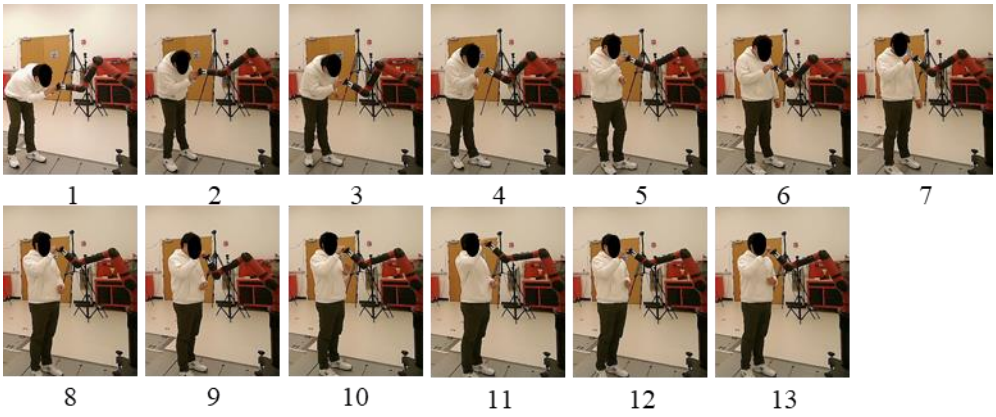


Figure 4.6 Postures over 13 iterations until the CAP score met the termination criterion.

4.3.2 CAP scores

The Kruskal-Wallis tests showed that the end effector positions had a significant impact on participants' CAP scores ($p < 0.0001$). As depicted in Figure 4.7, the results of the Steel-Dwass tests indicated that there was no significant difference between the Worker-selected and Learnt positions. However, significant differences were observed in all other paired comparisons. To better understand the exposure magnitude of each individual joint, we further analyzed the CAP scores for specific body parts (Figure 4.7b). The results showed that the end effector positions had a similar effect on neck and trunk bending angles, but no significant differences were observed in lower and upper arm angles.

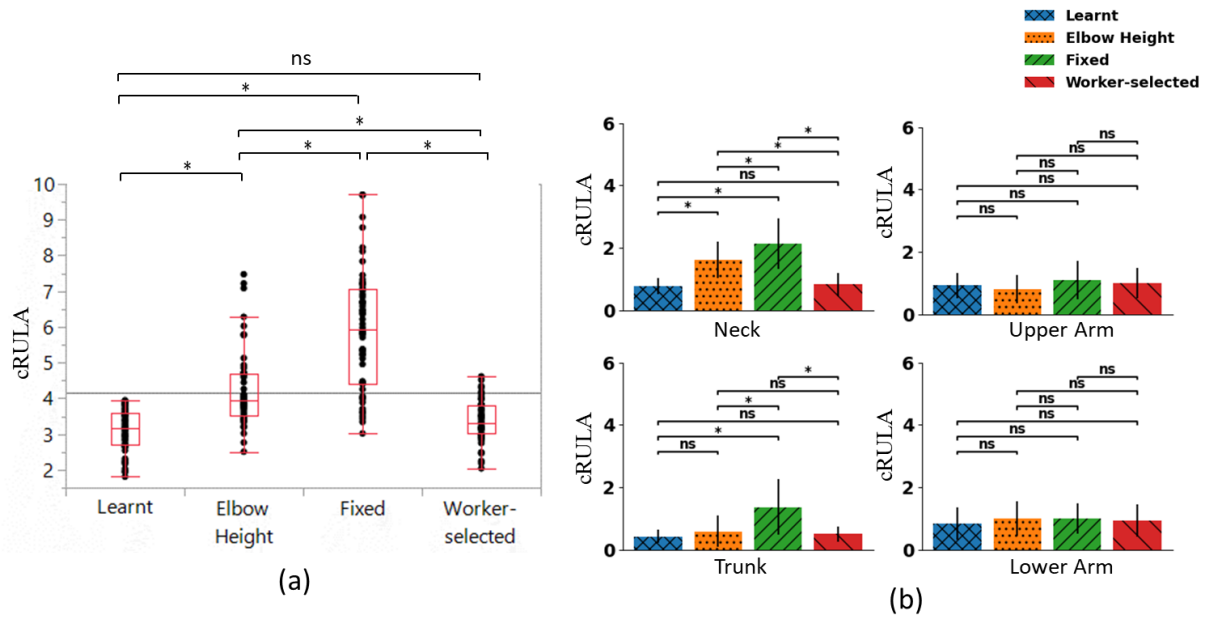


Figure 4.7 (a) Means and quantiles of CAP score corresponding to different end effector positions. The red box plots show the 90%, 75%, 50%, 25%, 10% quantiles. (b) Means and standard deviation of CAP scores of specific body joint angles. The results of are annotated: 'ns' – no significance; '*' – $p < 0.05$.

4.3.3 Subjective rating

The results of the subjective questionnaire are presented in Figure 4.8, with statistical significance marked. According to the Steel-Dwass tests, there is no significant difference between the Learnt and Worker-selected positions. Similarly, there is no significant difference between the Elbow Height and Fixed positions. These findings suggest that participants did not perceive a significant difference between the Learnt position and the Worker-selected position. Additionally, the results align with the findings from the CAP measurement, indicating that the Learnt position is significantly more preferred than both the Fixed and Elbow Height positions. The subjective questionnaire shows that the Fixed and Elbow Height positions have a similar effect on workers' perceptions, which differs from the CAP results.

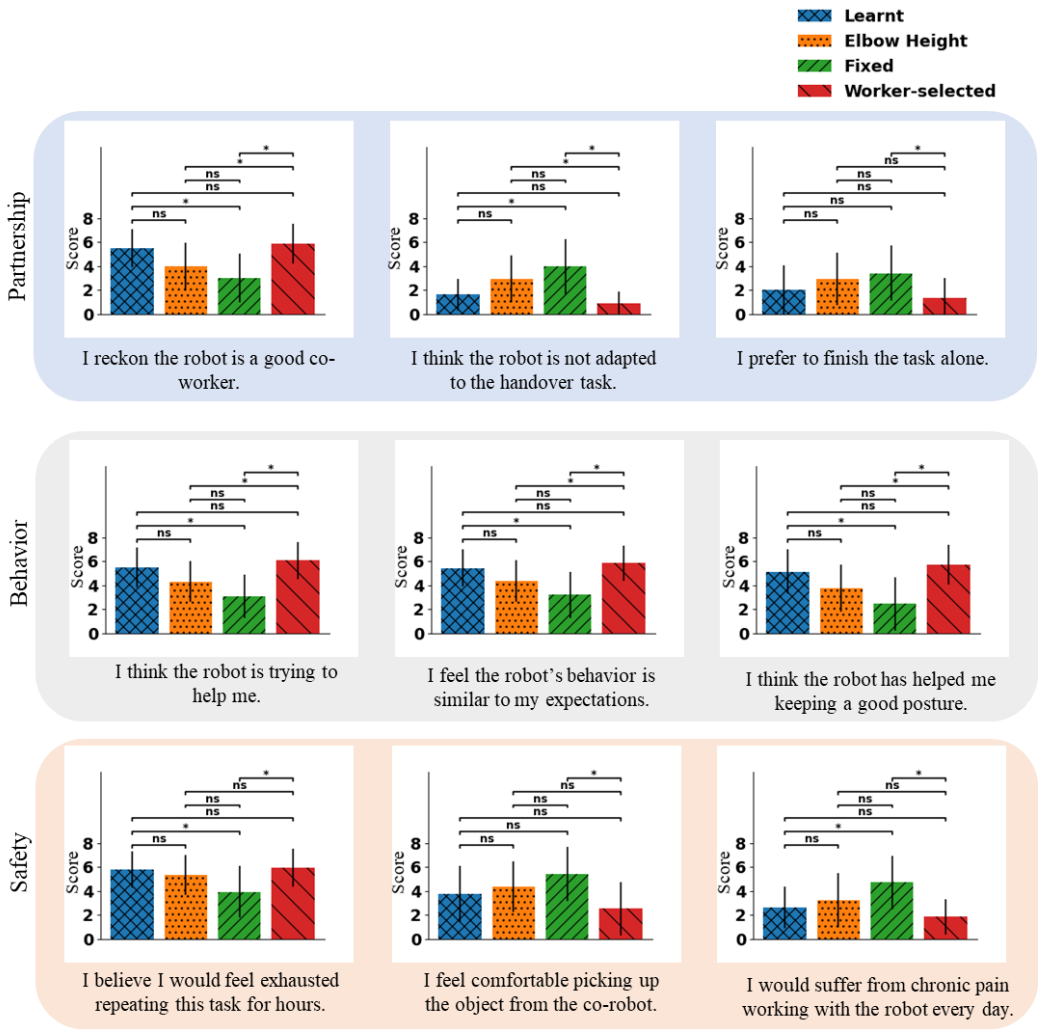


Figure 4.8 Summary of the questionnaire. The questionnaire includes three aspects 'Partnership', 'Behavior' and 'Safety', and each of them is evaluated by three questions. The questions were randomized, and the participants did not know the aspects when they were answering the questions. The results of Steel-Dwass tests are annotated: 'ns' – no significance; '*' – $p < 0.05$.

4.4 Discussions

4.4.1 Effectiveness of GOLA-HRC

As shown in the Results, the overall CAP scores of the *Learnt* positions are significantly smaller than those of the *Fixed* positions and *Elbow Height* positions. This result indicates that the GOLA-HRC can effectively improve the posture of a worker and thus reduce the MSD risks. Also, it was found that

the CAP scores of *Elbow Height* positions are significantly lower than the CAP scores of the Fixed positions, which indicates individualizing the robot end effector position helps improve the worker's posture.

When the performances of *Learnt* positions were further compared with the *Worker-selected* positions, it was found that there is no significant difference on CAP. A closer look on each component of CAP score further revealed that the effects of different end effector positions are more significant for the 'Neck' score and 'Upper Arm' score (Figure 4.6b). Particular, *Worker-selected* and *Learnt* positions significantly reduced neck flexion/extension angles and shoulder-elevation angles.

Such a result suggests that GOLA-HRC algorithm is able to converge at the positions that match the positions selected by workers. Meanwhile, there is no significant difference between *Learnt* and *Worker-selected* positions in any subjective ratings, which indicates that the *Learnt* position introduces the same level of physical workload as the baseline *Worker-selected* position. As the *Worker-selected* position is subjectively selected by the participants, this position can be considered as the optimal position from the psychophysical perspective (Snook & Ciriello, 1991). In psychophysics, it is assumed that people have the ability to be “self-protecting” from musculoskeletal injury by adopting relatively safe working postures and safe lifting weight (Acevedo & Ekkekakis, 2006). Although such an assumption has been criticized as workers, particularly new workers without receiving adequate training, could be unable to just the actual risk, psychophysics methods have been adopted for ergonomics interventions(Fernandez & Marley, 2014; Jiang et al., 1986).

4.4.2 Adaptability of GOLA-HRC

During the experiment, a postural difference among participants was visually observed by the experimenters. For example, some people tended to move their eyes and kept their necks straight during the tasks (Figure 4.9 left), while others tended to make neck flexions and gazed toward the breadboards without moving their eyes much (Figure 4.9 right). Yet, it should be noted our proposed GOLA-HRC algorithm would converge at a higher position for the participants who tend to bend their neck to a greater

extent. This is because the algorithm is more sensitive to the neck angles than the height of the breadboard. Another finding is that the optimized end effector orientation always points to the face of the participants since otherwise participants would have to greatly squat or bend their neck to look at the breadboard.

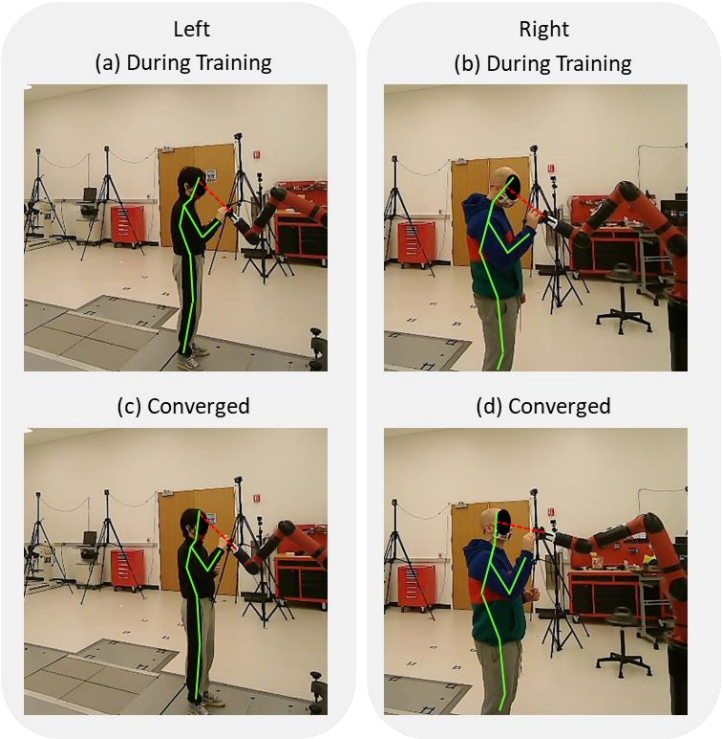


Figure 4.9 The postures of workers gazing behavior. The green lines are reconstructed skeletons and red dashed lines indicate the sight of the workers' eyes. The eyesight lines are closer to be vertical to the head skeleton lines in (b) and (d).

4.4.3 Safety awareness of workers in HRC

As collaborative robots have been newly adopted in recent years, limited research has been done in workers' safety awareness of long-term human-robot interaction. In particular, it is unclear whether the subjectively optimal position (i.e., 'worker-selected' position) is equivalent to the CAP-based optimal position. As described in the previous section, some participants have a significant neck flexion when they look at the breadboard, and therefore the GOLA-HRC algorithm moves the end effector to higher

positions to prevent high scores of the neck. According to the experiment, the majority of the participants (17 out of 20) did adjust the end effector to a position where they have a more neutral neck angle when they were asked to adjust the end effector positions. Yet, three participants selected a position with a greater neck flexion angle (Figure 4.10). For the participants in Figure 4.10, the *Worker-selected* position is below the worker's shoulder, which prevents the flexion of the upper arm but results in a greater neck flexion angle. Although there is a minor difference between the overall CAP scores for *Learnt* and *Worker-selected* positions, a 'Neck' CAP score of 1.57 for the Worker-selected position can increase the likelihood of developing neck and upper back MSDs (McAtamney & Corlett, 1993). The reason for choosing a posture with a greater neck flexion could be that some participants have personal preferences on a smaller shoulder flexion. While such a posture may not result in any discomfort over a short term, it may eventually lead to MSDs over a long period of time. Therefore, if workers are able to select the working location in an HRC task, ergonomics education is still necessary to ensure that they do not choose a posture that poses risk of MSDs on a specific joint.

Neck	1.57
UpperArm	0.3
Trunk	0.3
LowerArm	1.18
Sum	3.34

Worker-selected

Neck	0.59
UpperArm	0.83
Trunk	0.4
LowerArm	1.14
Sum	2.86

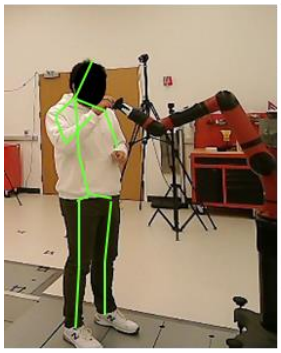
Learnt

Figure 4.10 A worker's postures in *Learnt* and *Worker-selected* positions. The *Worker-selected* position poses greater risk on the neck, while the *Learnt* position balanced the 'Neck' and 'Upper Arm' CAP scores.

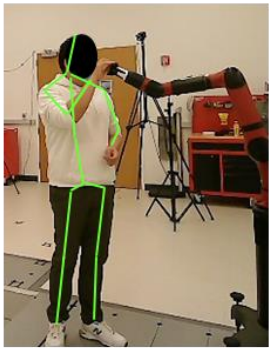
4.4.4 Personalization of CAP

One advantage of the proposed model-free RL method is that the optimization criteria can be conveniently personalized for workers by assigning different weighting factors to each joint, which can be helpful for individualized safety considerations. For instance, workers who already experience neck discomfort should ensure their neck is in an as much neutral posture as possible. In that case, one can apply a greater weighting factor to the 'Neck' CAP score before applying the RL algorithm. In a preliminary test, the robot's end effector moved to a higher position when the weighting factor of 'Neck' is doubled and thus resulting in a smaller neck flexion angle but a greater arm elevation angle (Figure 4.11). Similarly, we can easily modify the weighting factors of other joints to personalize the GOLA-HRC algorithm for specific needs In this way, we can proactively prevent workers from injuring a specific joint or reactively help workers avoid awkward posture of a joint with chornic pain. The utilization of electromyography sensors that measure fatigue levels presents the possibility of implementing a dynamic reward function. By doing so, the dynamic GOLA-HRC may serve as a viable solution to prevent the accumulation of fatigue.

Another advantage of GOLA-HRC is its flexibility in applications. There are various ergonomic accessment tools, such as REBA (Hignett & McAtamney, 2000) and NERPA (Sanchez-Lite et al., 2013) that show a variety of performance in different job configurations (Yazdanirad et al., 2018). The reward function of GOLA-HRC can be easily modified to fit other ergonomic assessment tools. Moreover, the overall CAP score in the current study is a linear summation of the CAP scores of each body part of a worker. If more evenly distributed scores among each joint are preferred, one can choose square or cubic summation which would penalize greater sub-scores.



Neck	1.0
UpperArm	0.7
Trunk	0.7
LowerArm	0.6
Sum	2.9



Neck	0.7
UpperArm	0.9
Trunk	0.9
LowerArm	0.7
Sum	3.2

Figure 4.11 Left: The optimized posture in original CAP. Right: The optimized posture with doubled weight on neck angle.

4.5 Limitations

There are a number of limitations that need to be addressed. First, the optimization process in the current study only considered two degrees of freedom of the robot end effector. In a real-world setting, a human-robot collaboration task can be more complicated and thus more robot end effector degree of freedom may need to be optimized. With a greater number of the degree of freedom to optimize, the state space, however, will exponentially grow. One possible solution is using sequential learning, which divide one complex problem with multiple parameters into several problems with fewer parameters, but the validity of sequential learning needs further investigation. Second, since the computer vision algorithm was used for pose reconstruction in this study, a camera needs to be placed where its field of view is not blocked. Otherwise, the error in pose reconstruction will be greater. To address this problem, one could consider using a multi-camera computer vision system. In addition, it was found that the output of the computer vision model could be less precise on worker's neck angle when the hair blocks the view of their neck. A headband may be needed for workers with longer hair. Third, the error in the reconstructed joint position can affect the value of the calculated CAP. To what extent this error affects the CAP score needs to be further investigated. On the other hand, we found the magnitude and the direction of the joint position error are not substantially affected by human postures. Thus, the gradient calculated in GOLA-HRC is less affected by this error as the gradient is calculated by the difference of CAP between each

iteration. Fourth, the original RULA scores is based on workers' joint angles as well as the weight of the external load (e.g., the weight of the material being handled). Given that this study focuses on improving workers' postures, the workload factors were not included in the assessment. Fifth, worker's loose clothes may occlude certain visual features for a precise human pose reconstruction. As suggested by a previous study (Liang & Lin, 2019), a multimodal database including the images of workers wearing a variety clothes as well as the ground-truth joint positions are further needed for training a more robust computer-vision algorithm. Sixth, the participants recruited in this study are not required to have professional work experience in HRC. Therefore, the postures adopted by these participants may differ from those of actual workers. Whether HRC-related working experiences affect the performance of the proposed method should be further examined.

4.6 Conclusion

In this study, we proposed a method to optimize workers' posture during HRC. A computer vision method was first adopted to recognize human posture and determine a CAP score. A model-free gradient descent optimization algorithm was then developed to lower the CAP score of a worker. The experimental result shows that GOLA-HRC effectively lowers workers' CAP during HRC tasks and thus reducing the risk of MSDs.

CHAPTER 5 CONCLUSION AND FUTURE WORK

The focus of this dissertation is on various aspects of human safety, including 1) recognizing driving distractions using machine learning and a wrist-mounted accelerometer, 2) developing a method to avoid collisions between humans and robots, and 3) enhancing workers' posture during human-robot collaborations through reinforcement learning and computer vision to minimize the risk of musculoskeletal disorders for workers. The outcomes from each of these topics are outlined below.

1. For driving distraction recognition, a neural network called ConvLSTM was developed and validated. It integrated convolutional neural networks and long short-term memory and demonstrates advantages in extracting both regional and temporal features from time series. Experiments with 20 participants were conducted on a driving simulator to collect driving distraction data, which was then labeled and used to train the ConvLSTM. The results showed that the ConvLSTM had an overall F1-score of 0.87 on a six-class classification task, indicating the great potential of using deep learning and wearable sensors to recognize driving distractions.

2. For human-robot collision avoidance, a computer vision model called VideoPose3D was implemented to create a collision avoidance method. This scheme first extracted 3D information of surrounding workers using a camera. A 3-level hazard zone was then defined in the workspace. If a worker steps into a hazard zone, a multi-channel alarm system will warn workers of potential danger while the robot proactively moves away and yields to the worker. Compared to previously proposed collision avoidance methods, this method requires only a low-cost single RGB camera and greatly expands the monitoring area. The preliminary test showed that this method can effectively avoid potential collisions between humans and robots.

3. For improving workers' posture and reducing the risks of MSD during human-robot collaborations, a model-free reinforcement learning approach was proposed. A new metric named Continuous Awkward Posture (CAP) score was introduced to evaluate the unnatural joint angle of the neck, trunk, arms, and legs. This CAP score is a continuous function that allows performing gradient

calculation between two different postures. The reinforcement-learning algorithm then used this gradient to iteratively alter the point of operation position during HRC and thus improve working postures. To validate this algorithm, 20 participants were recruited to perform wire-insertion tasks with a collaborative robot. The results showed that the proposed method significantly improved workers' posture and could accommodate workers with varying posture preferences.

Numerous potential areas of research merit attention in the future. First, despite the fact that distracted driving is a major contributor to thousands of fatal car accidents each year (U.S. department of labor, 2019), few driving assistance systems have been developed to address this issue. Given the advancements in autonomous driving systems that can take over control of a vehicle in certain circumstances, it would be beneficial to incorporate the driving distraction recognition system developed in this study as a trigger for a higher level of driving assistance.

Second, the current robot collision avoidance system, while designed to work in workplace environments, relies on VideoPose3D, which is trained by Human3.6m. This dataset captures the motion of actors performing daily life behaviors in an empty room, which is distinct from the real-world conditions of a workplace. This difference may impact the performance of VideoPose3D in actual production facilities. Although there are existing ergonomic practitioners who have created workplace datasets, the size of these datasets is still limited. To enhance the performance of VideoPose3D, it would be valuable to have a larger, labeled dataset captured in real workplaces and train the model using that dataset.

Third, the present reinforcement learning method for enhancing workers' postures solely takes into account two degrees of freedom (DoFs) of a robot. This is due to the fact that the state space grows exponentially as the number of DoFs increases. Such simplification may constrain the generalizability of this approach since, in less structured settings, optimizing more DoFs of a robot may be necessary. One possible solution is to merge data-driven generative models and optimization techniques to optimize human postures. This would enable fewer iterations to be computed while still ensuring convergence.

REFERENCES

Abhayasinghe, N., & Murray, I. (2014). Human activity recognition using thigh angle derived from single thigh mounted IMU data. *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. https://doi.org/10.1109/ipin.2014.7275474

Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., & Taha, K. (2015). Efficient machine learning for big data: A review. *Big Data Research*, *2*(3), 87–93.

Anita, A. R., Yazdani, A., Hayati, K. S., & Adon, M. Y. (2014). *Association between awkward posture and musculoskeletal disorders (MSD) among assembly line workers in an automotive industry*.

Ayoub, J., Zhou, F., Bao, S., & Yang, X. J. (2019). From manual driving to automated driving: A review of 10 years of autoui. *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 70–90.

Berti, S., & Schröger, E. (2001). A comparison of auditory and visual distraction effects: behavioral and event-related indices. *Cognitive Brain Research*, *10*(3), 265–273. https://doi.org/10.1016/s0926-6410(00)00044-6

Bi, C., Huang, J., Xing, G., Jiang, L., Liu, X., & Chen, M. (2017). SafeWatch: A wearable hand motion tracking system for improving driving safety. *Proceedings - 2017 IEEE/ACM 2nd International Conference on Internet-of-Things Design and Implementation, IoTDI 2017 (Part of CPS Week)*, 223–232. https://doi.org/10.1145/3054977.3054979

Billah, T., & Mahbubur Rahman, S. M. (2016). Tracking-based detection of driving distraction from vehicular interior video. *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2016*, 423–428. https://doi.org/10.1109/AVSS.2016.7738077

Bohannon, R. W., & Andrews, A. W. (2011). Normal walking speed: a descriptive meta-analysis. *Physiotherapy*, *97*(3), 182–189.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., & Sheikh, Y. (2019). OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(1), 172–186.

Chen, J.-H., & Song, K.-T. (2018). Collision-free motion planning for human-robot collaborative safety under cartesian constraint. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 4348–4354.

Chen, O. T.-C., Ha, M.-H., & Lee, Y. L. (2020). Computation-Affordable Recognition System for Activity Identification using a Smart Phone at Home. *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5.

Chong, E. K. P., & Zak, S. H. (2004). *An introduction to optimization*. John Wiley & Sons.

Colyer, S. L., Evans, M., Cosker, D. P., & Salo, A. I. T. (2018). A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system. *Sports Medicine-Open*, *4*(1), 1–15.

Craye, C., & Karray, F. (2015a). *Driver distraction detection and recognition using RGB-D sensor*. 1–11. http://arxiv.org/abs/1502.00250

Craye, C., & Karray, F. (2015b). Driver distraction detection and recognition using RGB-D sensor. *ArXiv Preprint ArXiv:1502.00250*.

Cummings, M. L., Mastracchio, C., Thornburg, K. M., & Mkrtchyan, A. (2013). Boredom and distraction in multiple unmanned vehicle supervisory control. *Interacting with Computers*, *25*(1), 34–47.

De-La-Hoz-Franco, E., Ariza-Colpas, P., Quero, J. M., & Espinilla, M. (2018). Sensor-based datasets for human activity recognition–a systematic review of literature. *IEEE Access*, *6*, 59192–59210.

Denis, D., St-Vincent, M., Imbeau, D., Jette, C., & Nastasia, I. (2008). Intervention practices in musculoskeletal disorder prevention: a critical literature review. *Applied Ergonomics*, *39*(1), 1–14.

Deurenberg, P., Yap, M., & Van Staveren, W. (1998). Body mass index and percent body fat: a meta analysis among different ethnic groups. *International Journal of Obesity*, *22*(12), 1164–1171. https://doi.org/10.1038/sj.ijo.0800741

Flenniken, W., Wall, J., & Bevly, D. (2005). Characterization of various IMU error sources and the effect on navigation performance. *Ion Gnss*, 967–978.

Frid, E., Bresin, R., & Alexanderson, S. (2018). Perception of mechanical sounds inherent to expressive gestures of a nao robot-implications for movement sonification of humanoids. *Sound and Music Computing*.

Furtado, J. S., Liu, H. H. T., Lai, G., Lacheray, H., & Desouza-Coelho, J. (2019). Comparative analysis of optitrack motion capture systems. In *Advances in Motion Sensing and Control for Robotic Applications* (pp. 15–31). Springer.

Gallahan, S. L., Golzar, G. F., Jain, A. P., Samay, A. E., Trerotola, T. J., Weisskopf, J. G., & Lau, N. (2013). Detecting and mitigating driver distraction with motion capture technology: Distracted driving warning system. *2013 IEEE Systems and Information Engineering Design Symposium*. https://doi.org/10.1109/sieds.2013.6549497

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). *Learning to forget: Continual prediction with LSTM*.

Gill, P. E., Murray, W., & Wright, M. H. (2019). *Practical optimization*. SIAM.

Girshick, Y. W. and A. K. and F. M. and W.-Y. L. and R. (2019). *Detectron2*.

Goutte, C., & Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. *Lecture Notes in Computer Science*, *3408*, 345–359. https://doi.org/10.1007/978-3-540-31865-1_25

Guiry, J. J., Van de Ven, P., & Nelson, J. (2014). Multi-sensor fusion for enhanced contextual awareness of everyday activities with ubiquitous devices. *Sensors*, *14*(3), 5687–5701.

Gulli, A., & Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.

Halme, R.-J., Lanz, M., Kämäräinen, J., Pieters, R., Latokartano, J., & Hietanen, A. (2018). Review of vision-based safety systems for human-robot collaboration. *Procedia CIRP*, *72*, 111–116.

Hammerla, N. Y., Halloran, S., & Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. *IJCAI International Joint Conference on Artificial Intelligence*, *2016-Janua*, 1533–1540.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *ArXiv Preprint ArXiv:1602.07360*.

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(7), 1325–1339. https://doi.org/10.1109/tpami.2013.248

ISO. (2016). *ISO/TS 15066:2016*.

Jeanne Breen Consulting. (2009). *Car telephone use and road safety Final Report An overview prepared for the European Commission*. *June*, 21p.

http://ec.europa.eu/transport/road_safety/pdf/car_telephone_use_and_road_safety.pdf%5Cnhttp://trid.trb.org/view/1149617

Jiang, L., Lin, X., Liu, X., Bi, C., & Xing, G. (2018). SafeDrive. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(4), 1–22. https://doi.org/10.1145/3161179

Karafi, M., & Cernock, J. H. (2010). *Recurrent neural network based language model*. *September*, 1045–1048.

Khusainov, R., Azzi, D., Achumba, I. E., & Bersch, S. D. (2013). Real-time human ambulation, activity, and physiological monitoring: Taxonomy of issues, techniques, applications, challenges and limitations. *Sensors*, *13*(10), 12852–12902.

Klauer, S. G., Klauer, S. G., Dingus, T. a., Dingus, T. a., Neale, V. L., Neale, V. L., Sudweeks, J. D., Sudweeks, J. D., Ramsey, D. J., & Ramsey, D. J. (2006). The Impact of Driver Inattention On Near Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data. *Analysis*, *April*, 226. https://doi.org/DOT HS 810 594

Koklonis, K., Sarafidis, M., Vastardi, M., & Koutsouris, D. (2021). Utilization of Machine Learning in Supporting Occupational Safety and Health Decisions in Hospital Workplace. *Engineering, Technology & Applied Science Research*, *11*(3), 7262–7272.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, *160*(1), 3–24.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Li, L., Zhong, B., Hutmacher, C., Liang, Y., Horrey, W. J., & Xu, X. (2020). Detection of driver manual distraction via image-based hand and ear recognition. *Accident Analysis & Prevention*, *137*, 105432. https://doi.org/10.1016/j.aap.2020.105432

Li, N., & Busso, C. (2013). Analysis of facial features of drivers under cognitive and visual distractions. *Proceedings - IEEE International Conference on Multimedia and Expo*, 1–6. https://doi.org/10.1109/ICME.2013.6607575

Maciej, J., & Vollrath, M. (2009). Comparison of manual vs. speech-based interaction with in-vehicle information systems. *Accident Analysis & Prevention*, *41*(5), 924–930. https://doi.org/10.1016/j.aap.2009.05.007

Martin, S., Tawari, A., & Trivedi, M. M. (2014). Toward privacy-protecting safety systems for naturalistic driving videos. *IEEE Transactions on Intelligent Transportation Systems*, *15*(4), 1811–1822. https://doi.org/10.1109/TITS.2014.2308543

Mathiassen, S. E., Möller, T., & Forsman, M. (2003). Variability in mechanical exposure within and between individuals performing a highly constrained industrial work task. *Ergonomics*, *46*(8), 800–824.

Matthias, B., Kock, S., Jerregard, H., Kallman, M., Lundberg, I., & Mellander, R. (2011). Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept. *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 1–6.

Mendat, C. C., & Wogalter, M. S. (2006). WARNING CHANNEL: MODALITY AND MEDIA. *Handbook of Warnings*, 123.

Michalos, G., Makris, S., Spiliotopoulos, J., Misios, I., Tsarouchi, P., & Chryssolouris, G. (2014). ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future. *Procedia CIRP*, *23*, 71–76.

Mohammed, A., Schmidt, B., & Wang, L. (2017). Active collision avoidance for human–robot collaboration driven by vision sensors. *International Journal of Computer Integrated Manufacturing*, *30*(9), 970–980.

National Center for Statistics and Analysis. (2019). *Driver Electronic Device Use in 2018 (Traffic Safety Facts Research Note. Report No. DOT HS 812 818)*. *2018*(October 2019), 1–9.

NHTSA. (2019). Distracted driving in fatal crashes, 2017. *Traffic Safety Facts*, *DOT HS 812*(April 2019), 1–8. https://doi.org/DOT HS 811 379

NIOSH. (2017). *NIOSH Presents: An Occupational Safety and Health Perspective on Robotics Applications in the Workplace*. Health, National Institute of Occupational Safety And.

Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors (Switzerland)*, *16*(1). https://doi.org/10.3390/s16010115

Park, W., Ramachandran, J., Weisman, P., & Jung, E. S. (2010). Obesity effect on male active joint range of motion. *Ergonomics*, *53*(1), 102–108. https://doi.org/10.1080/00140130903311617

Paulovich, F. V, De Oliveira, M. C. F., & Oliveira Jr, O. N. (2018). A future with ubiquitous sensing and intelligent systems. *ACS Sensors*, *3*(8), 1433–1438.

Pavllo, D., Feichtenhofer, C., Grangier, D., & Auli, M. (2019). 3d human pose estimation in video with temporal convolutions and semi-supervised training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7753–7762.

Perez, M. A., & Nussbaum, M. A. (2006). Posture and motion variability in non-repetitive manual materials handling tasks. *Human Movement Science*, *25*(3), 409–421. https://doi.org/10.1016/j.humov.2006.02.003

Peternel, L., Kim, W., Babič, J., & Ajoudani, A. (2017). Towards ergonomic control of human-robot co-manipulation and handover. *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 55–60.

Polverini, M. P., Zanchettin, A. M., & Rocco, P. (2014). Real-time collision avoidance in human-robot interaction based on kinetostatic safety field. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4136–4141.

Prabhakar, S., Pankanti, S., & Jain, A. K. (2003). Biometric recognition: security and privacy concerns. *IEEE Security & Privacy*, *1*(2), 33–42. https://doi.org/10.1109/msecp.2003.1193209

Prayudi, I., & Kim, D. (2012). Design and implementation of IMU-based human arm motion capture system. *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012*, 670–675. https://doi.org/10.1109/ICMA.2012.6283221

Ragab, A., Craye, C., Kamel, M. S., & Karray, F. (2014). A visual-based driver distraction recognition and detection using random forest. *International Conference Image Analysis and Recognition*, 256–265.

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, *29*(9), 2352–2449.

Rebecca L. Olson, R., J. Hanowski, Jeffrey S. Hickman, A., & Bocanegra, J. (n.d.). *DRIVER DISTRACTION IN COMMERCIAL VEHICLE OPERATIONS*. https://doi.org/10.21949/1502647

Regan, M. A., Hallett, C., & Gordon, C. P. (2011). Driver distraction and driver inattention: Definition, relationship and taxonomy. *Accident Analysis and Prevention*, *43*(5), 1771–1781. https://doi.org/10.1016/j.aap.2011.04.008

Ronao, C. A., & Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, *59*, 235–244. https://doi.org/10.1016/j.eswa.2016.04.032

Schäffer, A. A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimböck, T., Wolf, S., & Hirzinger, G. (2008). Soft robotics, from torque feedback-controlled lightweight robots to intrinsically compliant systems. *IEEE Robotics and Automation Magazine*, *15*(3), 20–30.

Schlegl, T., Kröger, T., Gaschler, A., Khatib, O., & Zangl, H. (2013). Virtual whiskers—Highly responsive robot collision avoidance. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5373–5379.

Schmidt, B., & Wang, L. (2013). Contact-less and programming-less human-robot collaboration. *Procedia CIRP*, *7*, 545–550.

Sherwani, F., Asad, M. M., & Ibrahim, B. (2020). Collaborative robots and industrial revolution 4.0 (ir 4.0). *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 1–5.

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, *2015-Janua*, 802–810.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Stanton, N. A. (1994). *Human factors in alarm design*. CRC Press.

Streiffer, C., Raghavendra, R., Benson, T., & Srivatsa, M. (2017). DarNet: A deep learning solution for distracted driving detection. *Middleware 2017 - Proceedings of the 2017 International Middleware Conference (Industrial Track)*, 22–28. https://doi.org/10.1145/3154448.3154452

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300.

Sztyler, T. (2019). *Sensor-based human activity recognition: Overcoming issues in a real world setting*. Universitaet Mannheim (Germany).

Taylor, W., Shah, S. A., Dashtipour, K., Zahid, A., Abbasi, Q. H., & Imran, M. A. (2020). An intelligent non-invasive real-time human activity recognition system for next-generation healthcare. *Sensors*, *20*(9), 2653.

Torres-Huitzil, C., & Alvarez-Landero, A. (2015). Accelerometer-based human activity recognition in smartphones for healthcare services. In *Mobile health* (pp. 147–169). Springer.

U.S. department of labor. (2019). *NATIONAL CENSUS OF FATAL OCCUPATIONAL INJURIES IN 2019*.

Vogel, C., Walter, C., & Elkmann, N. (2017). Safeguarding and supporting future human-robot cooperative manufacturing processes by a projection-and camera-based technology. *Procedia Manufacturing*, *11*, 39–46.

Voronin, A., Palagin, A., & Milani, A. (2012). Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer. *Journal of Informatics Models and Analysis*, *1*, 101–200.

Wang, L., Schmidt, B., & Nee, A. Y. C. (2013). Vision-guided active collision avoidance for human-robot collaborations. *Manufacturing Letters*, *1*(1), 5–8.

Waters, T. R., Putz-Anderson, V., Garg, A., & Fine, L. J. (1993). Revised NIOSH equation for the design and evaluation of manual lifting tasks. *Ergonomics*, *36*(7), 749–776.

Wollmer, M., Blaschke, C., Schindl, T., Schuller, B., Farber, B., Mayer, S., & Trefflich, B. (2011a). Online driver distraction detection using long short-term memory. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 574–582.

Wollmer, M., Blaschke, C., Schindl, T., Schuller, B., Farber, B., Mayer, S., & Trefflich, B. (2011b). Online driver distraction detection using long short-term memory. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 574–582. https://doi.org/10.1109/TITS.2011.2119483

Wöllmer, M., Kaiser, M., Eyben, F., Schuller, B., & Rigoll, G. (2013). LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, *31*(2), 153–163. https://doi.org/10.1016/j.imavis.2012.03.001

Yang, J., Shi, Z., & Wu, Z. (2016). Vision-based action recognition of construction workers usicng dense trajectories. *Advanced Engineering Informatics*, *30*(3), 327–336.

Zahray, L., Savery, R., Syrkett, L., & Weinberg, G. (2020). Robot Gesture Sonification to Enhance Awareness of Robot Status and Enjoyment of Interaction. *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 978–985.

Zhang, Y., Wu, C., Qiao, C., & Hou, Y. (2019). The effects of warning characteristics on driver behavior in connected vehicles systems with missed warnings. *Accident Analysis and Prevention*, *124*(September 2018), 138–145. https://doi.org/10.1016/j.aap.2018.12.018

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE Multimedia*, *19*(2), 4–10.

Zwicky, T. D., Nave, C. S., Us, M. I., Hsia, A. C., Arbor, A., Us, M. I., Tellis, L., & Us, M. I. (2017). *Hands-off steering wheel governed by pedestrian detection*. *2*(12).