Swansea University
Prifysgol Abertawe

DOCTORAL THESIS

Development and Implementation of Intuitive Human-Machine Interaction System

---

by:

Shuo Zhu

Supervised by Dr. Chunxu Li
A thesis submitted in fulfillment of the requirements
for the Doctor of Philosophy

of the

Department of Mechanical Engineering,
Faculty of Science and Engineering,
Swansea University

June 12, 2025

Declaration of Authorship

I, Shuo Zhu, declare that this thesis titled, “Development and Implementation of Intuitive Human-Machine Interaction System” and the work presented in it are my own. I confirm that:

- This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.
- This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.
- I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.
- The University’s ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Name: Shuo Zhu

Date: 13/11/2024

Abstract

Robotics has been well-developed in various fields in recent years. With the widespread application of intelligent machines, the interaction between humans and them is particularly important. Simple and reliable interactive systems are the general trend of future development. This thesis focuses on the development and implementation of serious intuitive human-machine interaction systems and explores the application of technology in graph classification, robot learning, droplet detection and robot control.

First of all, the real-time observation of the intelligent machines to the outside world is particularly important. A real-time double emulsion droplet detection system based on colour space segmentation and Hough transform is proposed, which can accurately detect the contour, size and generation frequency of droplets, and provide support for quality monitoring in industrial production.

In terms of graph classification, the research introduces a target-unbiased meta-learning algorithm to solve the graph structure classification problem under small sample data, demonstrates its excellent generalization ability, and improves the efficiency of image processing and recognition in human-machine interaction systems. In robot learning, the model optimized by extreme learning machine(ELM) combined with beetle antennae search(BAS) is studied, which significantly improves the accuracy and learning speed of trajectory prediction of KUKA iiwa robot and enhances the operational flexibility and real-time performance of the robot. The research also proposed robot path planning based on a fault-tolerant motion planning algorithm to achieve automatic drawing tasks. These studies demonstrate the important role of machine learning algorithms in robot control and planning, and significantly improve the robot's ability to interact with the environment.

In addition, the research also developed a flexible sensor data glove that can be used for remote control, further improving the naturalness and real-time nature of human-machine interaction. By integrating manual algorithms with machine learning technology, the thesis provides innovative solutions for the development of intelligent human-machine interaction systems and promotes technological progress in this field.

Acknowledgements

As this thesis is completed, I would like to extend my heartfelt gratitude to all those who have supported me throughout my PhD journey. This experience has been filled with challenges and rewards, and it is through your companionship and encouragement that I have been able to persevere and achieve these results.

First and foremost, I am deeply grateful to my primary supervisor, Dr. Chunxu Li, for his unwavering guidance, patience, and expertise. His insightful feedback and encouragement have been invaluable at every stage of this project. Dr. Li's dedication to excellence and his passion for robotics has continually inspired me to strive for greatness and persist in the face of challenges.

I would also like to express my appreciation to the School of Mechanical Engineering at Swansea University for providing the necessary resources and support that facilitated my research. My second supervisor, Professor Cinzia Giannetti, has inspired me to keep moving forward with her success. I am equally thankful to Professor Zhan Li for his substantial support during his time as my supervisor; I will always cherish their contributions to my academic journey.

Lastly, I am immensely grateful for the understanding and encouragement provided by my wife, Liyuan Wang, and my parents. Their unwavering support has been my driving force throughout this process. I hope that my beloved wife will also successfully achieve her PhD in the future.

List of Publications

1. C. Li, **S. Zhu**, Z. Sun and J. Rogers, BAS Optimized ELM for KUKA iiwa Robot Learning, *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2023

2. **S. Zhu**, A. Stuttaford-Fowler, A. Fahmy, C. Li and J. Sienz, Development of a Low-cost Data Glove using Flex Sensors for the Robot Hand Teleoperation, *2021 3rd International Symposium on Robotics and Intelligent Manufacturing Technology (ISRIMT)*

3. Z. Li, C. Li, S. Li, **S. Zhu**, H. Samani, A sparsity-based method for fault-tolerant manipulation of a redundant robot. *Robotica*. 2022

4. E. Hefter, C. Perry, N. Coiro, H. Parsons, **S. Zhu**, C. Li, Development of a Multi-sensor Emotional Response System for Social Robots, *Interactive Collaborative Robotics*. 2021

5. **S. Zhu**, C. Li, J. Rogers, M. Gianni and I. Howard, A Real-time Double Emulsion Droplets Detection System using Hough Circle Transform and Color Detection, *2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*

6. M. Li, **S. Zhu**, C. Li, W. Zhao, Target unbiased meta-learning for graph classification, *Journal of Computational Design and Engineering*. 2021

7. C. Li, **S. Zhu**, Home-Automated Robot Massaging for Pandemics, *Robotics for Pandemics*. 2021

8. R. Eshaq, **S. Zhu**, C. Li, PIDC-NN for Binary Classification Tasks in a Coal Preparation Plant(under review)

9. **S. Zhu**, C. Li, L. Wang, C. Giannetti, A Robot Drawing System Based on Image Segmentation of Fault Tolerant Algorithm(under review)

Acronyms

- AI - Artificial Intelligence
- ANN - Adaptive Neural Network
- AR - Augmented reality
- BAS - Beetle Antennae Search
- BP - Backpropagation Algorithm
- CGI - Computer-Generated Imagery
- CNN - Convolutional Neural Networks
- DoFs - Degrees of Freedom
- EGNN - Equivariant Graph Neural Network
- ELM - Extreme Learning Machines
- GA - Genetic Algorithm
- GNN - Graph Neural Network
- HMI - Human-Machine Interaction
- MSE - Mean Squared Error
- R-CNN - Region-based Convolutional Neural Network
- RNN - Recurrent Neural Network
- ROS - Robot Operating System
- SGD - Stochastic Gradient Descent
- SVM - Support Vector Machine
- VR - Virtual Reality
- YOLO - You Only Look Once

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Definition and Evolution of Intuitive Human-Machine Interaction System

Human-machine interaction (HMI) is a field that explores the interface between humans and machines, encompassing areas such as hardware and software design, user experience, and cognitive psychology[1]. Its primary goal is to make machine operation more efficient, safe, and intuitive through optimised design, thereby enhancing user experience and improving work efficiency[2].

The essence of human-machine interaction is to design systems or interfaces that maximise machine effectiveness while minimising the complexity of user operations. This field not only covers visual interfaces but also includes tactile feedback, voice interaction, gesture control, and other interactive methods. Through these mechanisms, users exchange information and control instructions with machines to complete specific tasks[3].

HMI is an interdisciplinary domain involving computer science, psychology, design, and engineering. It demands both technical skills and a deep understanding of user behaviour and cognition. Its research findings are applied to various platforms beyond computers and mobile devices, including robots, autonomous vehicles, smart homes, and medical equipment[4]. Human-computer interaction is also a kind of HMI.

In the initial phase of human-computer interaction, command line interfaces were the primary means of interaction. Users entered text-based commands via the keyboard, and computers executed the corresponding operations. This method required a high level of technical proficiency and understanding of command syntax, limiting its accessibility. Despite this, the Command Line Interface still plays a role in modern computing, especially in environments that require precise control[5].

With advancements in computer graphics and display technologies, graphical user interfaces became the dominant mode of interaction. Rise of Graphical User Interfaces utilises visual elements such as icons, windows, and buttons, allowing users to interact with machines intuitively without needing to memorise complex command syntax. This shift significantly lowered the barrier to entry, making computers more accessible for everyday use and work[6].

In recent years, natural user interfaces have emerged as a key area of research in HMI. NUI enables interaction methods that align with human natural behaviours, such as touch, gestures, voice, and eye tracking[7]. The combination of artificial intelligence and sensor technologies has made the Natural User Interfaces implementation increasingly viable. This type of interface enhances both the intuitiveness and efficiency of interactions, as well as the overall immersion and user experience[8].

1.2 Necessity of Intuitive Human-Machine Interaction System

1.2.1 Motivation

As technology continues to advance and become more widespread, users' expectations for human-machine interaction (HMI) systems have been steadily rising. There is a growing desire for interactions with machines to occur in a more natural manner, eliminating the need for complex learning processes [9]. Intuitive HMI systems hold particular significance, especially within the domains of consumer electronics, smart home devices, and social robotics. Users anticipate that these systems will possess capabilities such as natural language understanding, gesture recognition, and emotion perception, aiming to make human-machine interaction as seamless and instinctual as interpersonal communication [10].

Cognitive load, defined as the mental effort exerted by users during system interaction, has been the focus of extensive research. Findings indicate that a high level of cognitive load can lead to a deterioration in user experience, potentially even inducing feelings of frustration and resistance during usage [11]. Consequently, the design of intuitive interaction systems that effectively reduce cognitive load is of paramount importance for enhancing user experience and promoting system acceptance. Practical approaches include optimising interface layouts and integrating natural interaction modalities, both of which have been proven to significantly shorten users' initial learning curves when engaging with a new system [12].

Traditional HMI systems typically require users to possess a certain level of technical expertise and prior usage experience, which inherently restricts their accessibility to a broader audience. With the increasing proliferation of computer technology, an ever-growing number of ordinary users are beginning to interact with various electronic devices and systems. To meet the needs of this expanding user base, the development and implementation of intuitive HMI systems have become increasingly critical. These systems enable users to quickly become proficient, lower the barriers to entry, and thereby facilitate the wider dissemination and adoption of technological innovations [13].

1.2.2 Application Fields

In the **medical field**, intuitive HMI systems can help doctors and medical staff operate medical equipment and systems more efficiently. For example, through gesture recognition and voice control, doctors can interact with surgical robots in a sterile environment, or free their hands when viewing patient information[14]. These systems can also help the elderly and people with disabilities use medical devices more easily, improving the accessibility and convenience of medical services[15].

In **industrial automation**, intuitive HMI systems can help workers operate and monitor production equipment more efficiently. For example, augmented reality technology can help workers more intuitively understand equipment status and operation steps in complex production environments, reduce errors and improve production efficiency[16]. In addition, voice and gesture control systems can enable workers to operate machines more conveniently and reduce tedious key operations[17].

In the field of **smart homes and social robots**, intuitive HMI systems can significantly enhance user experience. For example, through voice assistants, users can naturally control smart devices at home, perform voice queries, or play music[18]. Social robots, by recognising users' facial expressions and vocal emotions, can respond in a more human-like manner, thereby enhancing the emotional experience and user acceptance[19].

In the field of **transportation and travel**, multimodal fusion can improve driving safety and experience. The driver can adjust the air conditioning temperature by circling in the air, switch the instrument panel display mode by waving, and combine voice commands (such as "navigate to the nearest charging station") to achieve safe operation of "keeping eyes on the road and hands off the steering wheel". The windshield is used as a projection interface to directly superimpose navigation arrows on the real road conditions, reducing the frequency of drivers looking down at the screen[16].

In addition, HMI is widely used in many fields such as **education, military and aerospace, retail and service industries**[17].

1.3 Existing Technologies and Challenges

1.3.1 Current State of Technology

Speech recognition technology is one of the core technologies of natural user interfaces and is widely used in smart assistants (such as Siri, Alexa) and smart home devices. Modern speech recognition systems can recognise voice commands from different users in noisy environments through deep learning models[20]. However, speech recognition technology still faces challenges in recognition rate and response speed, especially in multi-context and multi-dialect environments[21].

Computer vision technology enables machines to "see" and understand the visual information around them. This technology has a wide range of applications in autonomous driving, security monitoring, face recognition and other fields. Through advanced algorithms such as convolutional neural networks (CNN), computer vision systems can perform tasks such as image classification, target detection, and pose estimation[22]. However, in complex real-world scenarios, such as lighting changes, occlusion and multiple targets, the robustness of the visual system still needs to be improved[23]. As an important part of the development of computer vision, **Augmented reality (AR) and virtual reality (VR) technologies** have changed the way humans and computers interact by creating an immersive user experience. In AR systems, users can interact with augmented reality environments through wearable devices (such as Microsoft HoloLens), while in VR systems, users are fully immersed in a virtual environment[24]. These technologies have great application potential in education, training, medical care, games and other fields. However, the popularisation of these systems still faces challenges such as high equipment costs, lack of comfort and content richness[25].

Haptic feedback technology enhances the user experience by providing physical feedback (such as vibration, force feedback) when the user interacts with the system. This technology has been widely used in virtual reality and surgical robots to provide a more realistic sense of operation[26]. However, the design and implementation of tactile feedback systems are complex and are subject to cost, weight and power consumption limitations, which are the main obstacles to their widespread application[27].

1.3.2 Technical Challenges

Although speech recognition and computer vision technologies(including VR and AR) have achieved high accuracy under laboratory conditions, the **accuracy and robustness** of the system remain a challenge in real-world applications. Different background noise, lighting conditions, and user characteristics (such as dialects and accents) can affect system performance. The existence of these problems limits the popularity of HMI technology in diverse application scenarios[28].

In many interaction scenarios, the response time of the system is crucial. Users expect the system to respond to their operations in **real time** and smoothly. However, complex computing tasks (such as inference of deep learning models) often consume a lot of computing resources and time, resulting in less than ideal system response speed. This is particularly important in some time-sensitive application scenarios (such as autonomous driving and remote surgery)[29].

Different user groups have different needs and habits, which brings great challenges to the design of human-machine interaction systems. A good system not only needs to adapt to the usage habits of different users, but also needs to be able to learn and adapt to user changes[30]. Although machine learning and personalised recommendation systems are helpful in this regard, how to effectively achieve **user adaptability** in practical applications remains a difficult problem[31].

The **cost** of many human-machine interaction technologies is still high, which to some extent limits their widespread application. Especially in the consumer market, cost reduction is the key to technology popularization. How to reduce costs without sacrificing performance is still an important issue that technology developers need to face[32].

Robots may encounter hardware failures when performing tasks, so effective **fault tolerance** mechanisms are needed to ensure the smooth completion of tasks. Current fault tolerance strategies are often targeted at specific types of failures and are not general enough.

The technical challenges of haptic feedback are currently mainly focused on the hardware limitations of fine tactile simulation, which makes it difficult to restore complex tactile textures. Haptic feedback needs to be integrated with force feedback, temperature feedback, etc. (such as synchronous low-temperature stimulation when virtually touching ice cubes), but existing devices (such as tactile gloves) are limited by volume and it is difficult to integrate multiple types of actuators[33]. Due to individual differences in human perception thresholds, the same vibration intensity may be perceived as different results by different users[34].

1.4 Research Objectives and Thesis Structure

1.4.1 Research Objectives

The main goal of this study is to develop and implement an intuitive HMI system, including:

- Develop a real-time double emulsion droplet detection system based on colour space segmentation and Hough transform: the system can accurately identify the contour, size, and generation frequency of droplets, providing a reliable data basis for subsequent processing.
- Explore the application of neural networks in image classification: study the performance of machine learning in processing image classification and propose improvement strategies to optimise its performance.
- Study the application of extreme learning machines in robot trajectory prediction: explore how to use extreme learning machines for robot trajectory prediction and improve prediction accuracy through optimisation algorithms.
- Propose a fault-tolerant algorithm: use to handle drawing tasks in the case of robot joint failures, ensuring that the robot can maintain normal movement of the end effector even when multiple joints fail.
- Design a low-cost data glove: use bending sensors to achieve remote operation of the robot hand, further improving the naturalness and real-time nature of human-computer interaction.

1.4.2 Overview of the Thesis Structure

Chapter 2 - Background Research: reviews the existing literature in related fields, analyses the advantages and limitations of current technologies, and provides theoretical support and technical background for the development of this study.

Chapter 3 - A Real-Time Double Emulsion Droplet Detection System Using Hand-Crafted Algorithms: describes in detail the real-time double emulsion droplet detection system based on colour space segmentation and Hough transform, and demonstrates the effectiveness of the system under different experimental conditions.

Chapter 4 - Target-Unbiased Meta-Learning for Few-Shot Graph Classification Using Graph Neural Networks: discusses the mathematical principles and learning methods of neural networks in graph classification tasks.

Chapter 5 - BAS Optimised ELM for KUKA iiwa Robot Learning: discusses the application of machine learning to robot trajectory prediction(extreme learning machines).

Chapter 6 - A Robot Drawing System Based on Image Segmentation of Fault Tolerant Algorithm: explores the role of the fault-tolerant algorithm in redundant robot operation, especially the method of maintaining normal movement of the end effector in the case of multi-joint failure.

Chapter 7 - Development of a data glove using flex sensors for the robot hand teleoperation: describes the design process of a low-cost data glove and its application in the remote operation of robot hands.

Chapter 8 - Conclusion and Prospect: summarises the research results of the whole research and proposes future research directions.

Chapter 2

Background Research

2.1 Current Status of Human-Machine Interaction Systems

2.1.1 Development of Human-Machine Interaction Technology

Human-machine interaction (HMI) technology has evolved significantly, transitioning from mechanical control interfaces, such as buttons and levers, to advanced sensor-based systems that facilitate natural, multimodal interactions. Early HMIs relied on basic controls for industrial and manufacturing applications, primarily focusing on machine operation and safety monitoring. With advancements in microprocessors, sensor technology, and artificial intelligence, HMI systems have integrated increasingly sophisticated interfaces[35]. Modern HMIs frequently utilize touchscreens, voice recognition, and gesture control to offer more intuitive and responsive interactions. Additionally, technologies like augmented reality (AR) and virtual reality (VR) are transforming HMI, particularly in sectors such as automotive, where AR displays provide real-time information directly within the driver's field of view, and in healthcare, where VR-based interfaces support complex surgical procedures. The development of machine learning algorithms has further enhanced HMI systems, enabling personalized, context-aware interactions that adapt to user behaviour and preferences. Today's HMI technologies are progressing towards fully autonomous systems, improving not only user experience but also operational efficiency and safety across a wide range of applications[36]. Figure 2.1 shows a schematic sketch of the working of Human-machine interaction.

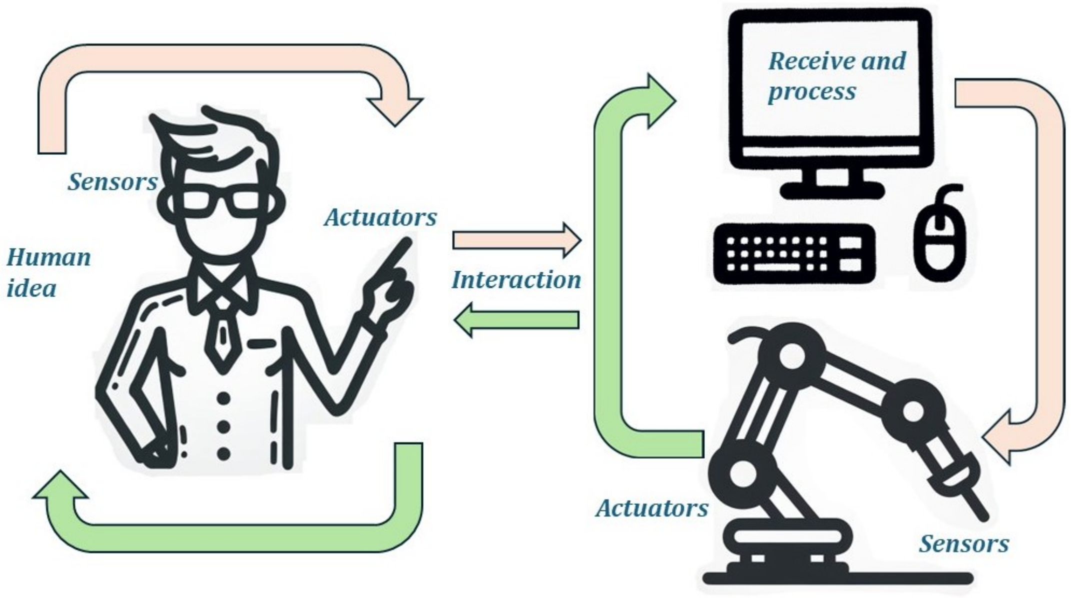


Figure 2.1: An overview of Human-Machine Interaction.

2.1.2 Demand for Intuitive Human-Machine Interaction Systems

The need for intuitive HMI systems has grown substantially, driven by users across diverse industries who seek more natural, efficient interaction methods with machines. In consumer technology, such interfaces enhance accessibility and user satisfaction, as exemplified by voice-controlled smart home devices that enable hands-free management of household tasks. In industrial automation, intuitive HMIs streamline complex processes, reduce errors, and improve safety by allowing operators to monitor and control equipment with minimal cognitive load[37]. This demand is especially pronounced in fields with stringent real-time requirements, such as autonomous vehicles and healthcare, where operators must interact rapidly and accurately with systems. Users increasingly expect HMI systems capable of understanding and responding to natural language, gestures, and even emotional cues[13]. Developing such intuitive systems requires ongoing advancements in artificial intelligence, sensor accuracy, and robust security measures to safeguard user privacy. As technology progresses, intuitive HMIs are expected to become the norm, seamlessly integrating intelligent systems into daily life and specialized fields, fundamentally transforming human-machine interaction[4].

2.1.3 Key Technologies and Limitations

Visual tracking technology uses cameras to capture user movements is widely used in games, virtual reality and other fields. It can identify information such as human posture and facial expressions, making the interaction more vivid and natural[38]. Deep learning-based methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have made significant progress in this field [39]. For example, OpenPose is an open-source human posture estimation library that can detect key points of multiple people in real-time, thereby supporting complex HMI applications[40].

Tactile feedback can provide feedback information to users through vibration or other physical stimulation to enhance the realism of interaction[41]. It is very important for simulating the sense of touch in the real world, especially in medical training and remote operations. The latest research shows that combining force feedback and temperature feedback can further enhance the user experience. For example, HaptX gloves provide high-resolution tactile feedback through microfluidics technology, allowing users to feel the shape, texture and temperature of objects[42].

Voice recognition allowing users to communicate with devices through voice commands is particularly suitable for mobile devices and smart homes[43]. Advances in voice recognition technology enable devices to better understand users' intentions and improve the convenience of interaction. Currently, end-to-end deep learning models such as Transformer have performed well in speech recognition tasks [44]. Google's Transducer model is a typical example, which has achieved advanced performance in speech recognition tasks in multiple languages[45].

Biosignal monitoring, such as electromyography, electroencephalography, etc., can be used to identify the physiological state and intention of the human body. These technologies have broad application prospects in rehabilitation training, emotion recognition, etc[46]. For example, by analysing electroencephalography signals, researchers have been able to achieve real-time monitoring of brain activity and develop brain-computer interface systems based on this[47]. Emotiv EPOC+ is a commercial portable EEG headset that can be used for a variety of applications, including mental state monitoring and simple mind control[48].

The limitations and challenges of **robot trajectory planning** technology are rooted in its unique mechanical structure, application scenario constraints and control theory difficulties[49]. There are multiple sets of joint angle solutions in the inverse kinematics solution of the serial robot, and it is necessary to introduce additional optimisation targets to screen feasible solutions. For example, when the robot is in a singular configuration, a slight change in the end position may cause a surge in joint speed and cause loss of control[50]. When the robot is working, the 20ms delay of the visual sensor may cause the target position prediction error to reach 10mm, which is difficult to compensate for by traditional PID tracking control. It is necessary to combine model predictive control with Kalman filtering to achieve advance planning[49].

Although the above technologies have their own advantages, they also have some common limitations:

High cost: High-quality sensors and computing resources are usually expensive, which limits the popularity of the technology. For example, high-end tactile feedback devices and high-precision visual tracking systems often require a lot of investment[51].

High computational complexity: Many advanced algorithms require a lot of computing resources, which may be difficult to implement on embedded systems or low-power devices. Especially in real-time application scenarios, efficient computing is crucial[52].

Environmental sensitivity: Changes in the external environment (such as changes in lighting) may affect the quality of sensor data, thereby reducing the reliability of the system. For example, a visual tracking system may fail in strong or low light conditions[53].

Different user behaviour: Differences in behaviour patterns of different users may cause unstable system performance, so stronger adaptive capabilities are needed. Personalization settings and adaptive learning algorithms can help alleviate this problem[54].

2.2 Image Processing Based on Machine Vision

Machine vision enables machines to “see” and understand the content of images by simulating the human visual system. Image processing technology is the basis of machine vision. The main tasks include image acquisition, preprocessing, analysis and processing in specific application scenarios. With the development of deep learning and big data technology, the image processing algorithm of machine vision has been significantly improved, but it still faces many challenges in terms of real-time performance, stability and adaptability to complex environments[55].

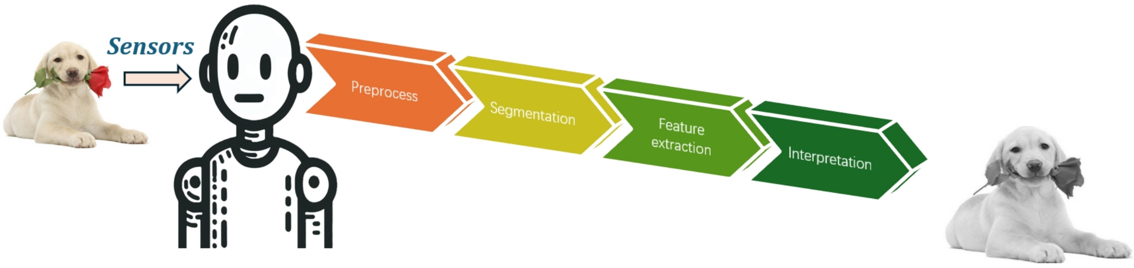


Figure 2.2: Representative machine vision processing processes.

Image preprocessing plays a critical role in machine vision systems by enhancing image quality, which is foundational for subsequent tasks like feature extraction and object recognition. Key techniques in image preprocessing include denoising, image enhancement, grayscale conversion, and edge detection. For denoising, methods like Gaussian filtering, median filtering, and bilateral filtering are commonly used to reduce noise while preserving essential features such as edges. Edge-preserving denoising is especially important in applications where both noise reduction and structure retention are crucial, such as object recognition and biometric systems. Advanced algorithms, such as edge-aware and CNN-based denoising filters, can selectively process high-frequency components, thus preserving important structural details without compromising computational efficiency. Similarly, common enhancement techniques like histogram equalization and gamma correction improve contrast and pixel intensity distribution, aiding feature detection and classification tasks. Adaptive methods, such as the adaptive gamma correction approach, further fine-tune image enhancement by adjusting sensitivity to darker image regions, which is beneficial in low-light conditions[56].

Neural networks (NN) are crucial in the field of image processing and machine vision. Their core value lies in automatically learning visual features in a data-driven way, breaking through the limitations of traditional manual feature engineering. Through layer structures such as convolution and pooling, neural networks can automatically extract hierarchical representations from data from low-level features (edges, corners) to high-level semantics (objects, scenes)[57]. Neural networks support joint optimization of the entire process from raw pixels to task outputs (such as target boxes, category labels) without splitting the feature extraction and classification stages. Images are high-dimensional data. Neural networks efficiently capture spatial correlations through the weight sharing and local receptive field mechanism of the convolutional layer, reducing the number of parameters while retaining contextual information. Neural networks have built a bridge between raw pixels and semantic understanding by “learning from data” rather than “explicit rule design”, becoming the cornerstone of modern machine vision[58].

Edge detection is another foundational step, essential for identifying boundaries and regions of interest within images. Techniques like the Canny edge detector and Sobel operator are widely applied in extracting edge features, which helps simplify images and focus computational resources on significant structural elements. Recent advancements also include machine learning-driven preprocessing techniques, where algorithms like k-nearest neighbour and support vector machines are utilized to dynamically adjust image brightness and contrast, optimizing images for specific recognition tasks[59]. Figure 2.2 shows how machine vision receives and processes images.

Feature extraction is central to image processing, aiming to condense the essential information within images to lower-dimensional representations that facilitate tasks like detection and classification. Traditional approaches rely on colour, shape, texture, and methods like the scale-invariant feature transform to identify unique image features. In recent years, deep learning-based methods, particularly convolutional neural networks (CNNs), have revolutionized feature extraction by enabling automatic learning of hierarchical features. CNNs utilize convolution kernels to identify edges, corners, and more complex patterns, providing a more expressive and adaptable representation of image characteristics than conventional methods[60].

For instance, CNN-based extraction techniques have shown remarkable accuracy in distinguishing complex shapes and textures, which is essential for tasks involving object and scene recognition. These deep learning models can be further enhanced with optimization techniques to improve their adaptability and efficiency when applied to specific image datasets. This approach allows for a more flexible and robust feature extraction process that enhances overall performance in recognition and classification tasks[61].

Object detection is an important application direction in machine vision and is often used to identify target objects in images. Traditional object detection algorithms include Haar feature cascade classifiers and HOG+SVM methods, while deep learning-based algorithms such as YOLO and Faster R-CNN have greatly improved detection accuracy and speed by introducing convolutional neural networks[57]. Image segmentation is to divide an image into regions with specific meanings. Common methods include threshold segmentation, region growing, and edge-based segmentation algorithms[62]. In terms of deep learning, network architectures such as U-Net and Mask R-CNN have significantly improved the segmentation accuracy of application scenarios such as medical imaging and scene understanding. Through effective segmentation technology, machine vision systems can achieve more refined target area extraction[58].

2.3 Artificial Neural Networks in Graph Classification

2.3.1 Basic Principles of Artificial Neural Networks

Artificial neural networks are a computational model that simulates the connection structure of neurons in the human brain and are widely used in tasks such as data classification, regression, and pattern recognition. The core of artificial neural networks is to learn through a large amount of training data, so that they can automatically extract data features and perform complex mapping[63].

Multilayer Perceptron is one of the earliest neural network models, mainly used to deal with simple linear separable problems. Although Multilayer Perceptron has limited performance in dealing with complex nonlinear problems, it laid the foundation for the subsequent development of deep learning[64].

Convolutional Neural Network(CNN) extracts local features of images through convolution operations and is widely used in image classification and target detection tasks. Its hierarchical feature extraction method enables the model to capture image features at different levels, significantly improving the accuracy of image recognition[61].

Graph Neural Network(GNN) is a type of neural network model that has developed rapidly in recent years. It is specifically used to process graph-structured data (such as social networks, molecular structures, etc.). GNN learns the representation of nodes and edges through a message-passing mechanism to achieve classification and regression of graph data[65].

2.3.2 Challenges and Existing Methods of Graph Classification

Graph classification tasks involve classifying graph-structured data. The challenge is how to effectively capture the complex relationships and structural information between nodes. Traditional machine learning methods often require a lot of manual feature extraction when processing graph data, while neural networks greatly simplify this process by automatically learning features.

Graph Convolutional Networks capture local neighbourhood information of nodes by performing convolution operations on graph structures and have been widely used in graph classification tasks[66]. One of the main advantages of Graph Convolutional Networks is its parameter-sharing mechanism, which enables the model to effectively process large-scale graph data.

Graph Attention Network introduces an attention mechanism to adaptively assign different weights to each edge in the graph, thereby improving the flexibility of the model in processing heterogeneous graphs[67]. Graph Attention Network performs well in node classification and edge prediction tasks, especially when processing graph data with nodes of different importance.

In graph classification tasks, target-unbiased meta-learning optimizes the meta-model so that the model can quickly adapt to new tasks and new data distributions. Unlike traditional supervised learning methods, meta-learning does not rely on a large amount of labelled data, so it has unique advantages in small sample learning and transfer learning[68]. The core idea of Target Unbiased Meta-Learning is to learn a model parameter that can quickly adapt to new tasks through meta-task training. Specifically, the model continuously optimizes the loss function on the meta-task set during training, so that the model can quickly update parameters and obtain better performance when encountering new tasks[68]. In my paper "Target Unbiased Meta-Learning for Graph Classification", researchers proposed a new meta-learning algorithm that enables the model to better adapt to data with different graph structures by training on multiple graph classification tasks. In experiments, the algorithm performed well on multiple graph classification benchmark datasets, proving its advantages in processing heterogeneous graph data.

2.4 Extreme Learning Machine in Robot Learning

2.4.1 Basic Principles of Extreme Learning Machine

Extreme Learning Machine (ELM) is a single hidden layer feedforward neural network that has attracted widespread attention for its fast training and good generalization ability. Unlike traditional neural networks, the hidden layer weights and biases of ELM are randomly generated, and the training process only needs to solve the output layer weights by the least squares method, which greatly simplifies the computational complexity[69].

ELM maps input data to a high-dimensional space through random projection, and then uses the least squares method to solve the output layer weights to complete the training. This process avoids the tedious gradient descent optimization, giving ELM a significant speed advantage when processing large-scale data[70]. Figure 2.3 shows a schematic diagram of the structure of the Extreme Learning Machine.

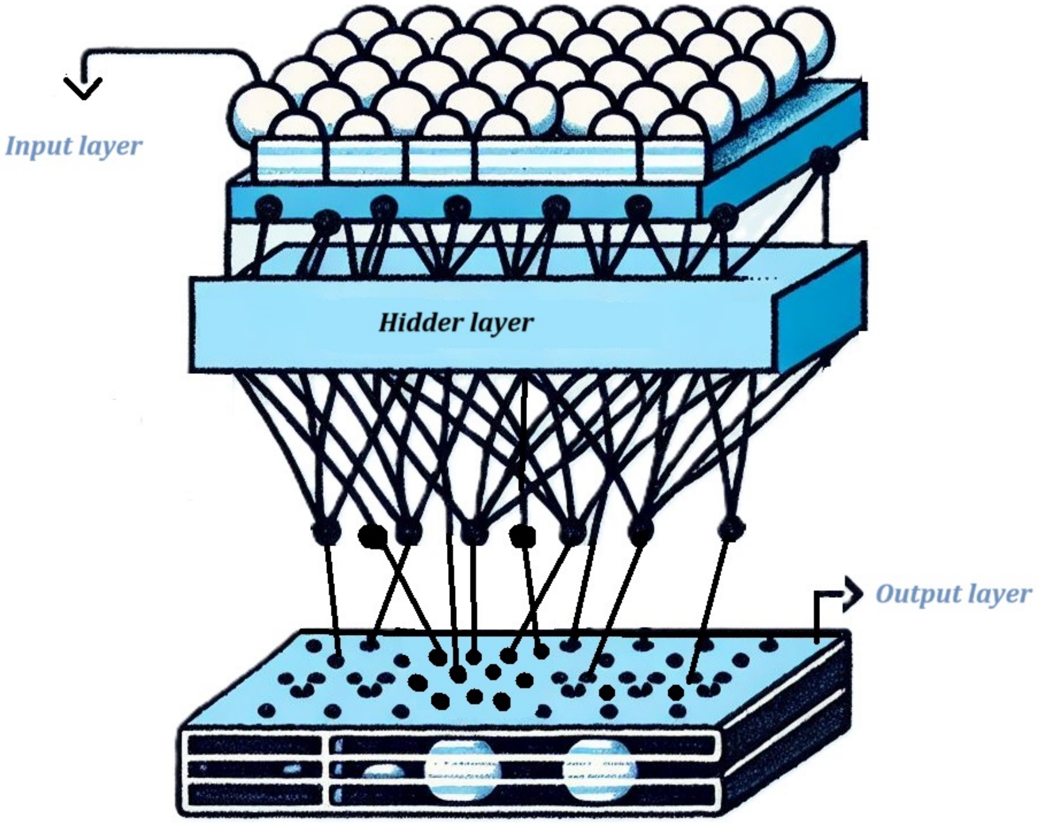


Figure 2.3: Basic principles of the Extreme Learning Machine.

ELM is widely used in tasks such as classification, regression, and time series prediction, especially in real-time applications that require fast response. Studies have shown that ELM has high application potential in image recognition, speech recognition, and robot control[69].

2.4.2 Application of ELM in Robot Trajectory Prediction

Robot trajectory prediction involves predicting the future trajectory of a robot based on historical motion data. Traditional methods usually rely on physical modelling and motion equations, which are susceptible to noise and nonlinear factors. With the development of machine learning technology, data-driven methods have gradually become mainstream [71].

ELM has become an effective method in robot **trajectory prediction** due to its fast learning and good generalisation ability. In our paper “BAS Optimised ELM for KUKA iiwa Robot Learning”, we used the ELM model to predict the trajectory of the KUKA iiwa robot, and optimised the hyperparameters of ELM through the Beetle Antennae Search (BAS) algorithm to further improve the prediction accuracy of the model. Figure 2.4 shows how the Beetle Antennae Search algorithm works.

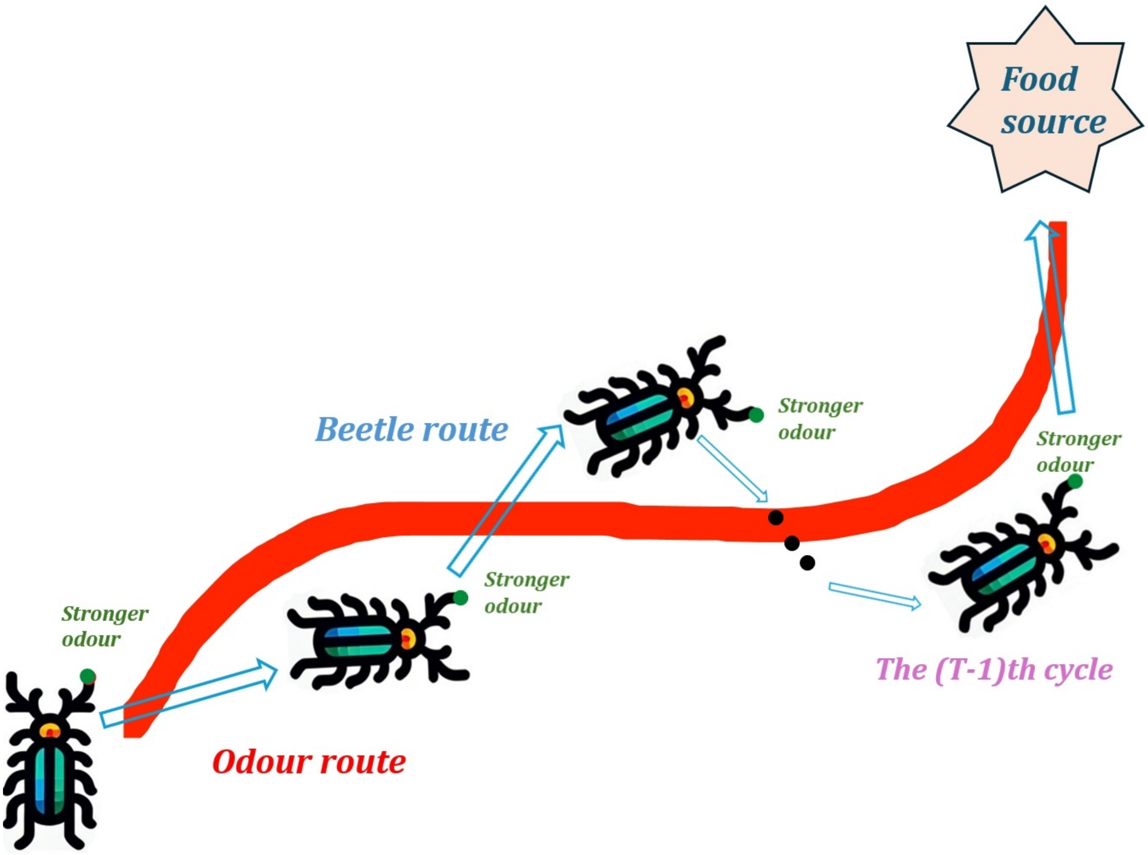


Figure 2.4: Schematic diagram of the Beetle Antennae Search algorithm. When foraging, the beetle relies on its two whiskers to sense the difference in odour intensity and decides the direction of movement. Through a simple bionic mechanism, the biological perception behaviour is transformed into efficient mathematical iteration. Its core logic of “perception-movement” reflects the powerful inspiration of natural behaviour in solving complex problems.

The study of the optimisation method and experimental results shows that the ELM model optimised by BAS performs well in the trajectory prediction task of the KUKA iiwa robot, and its prediction accuracy and computational efficiency are better than traditional methods. In experiments, the BAS-ELM model was able to quickly adapt to different trajectory patterns, demonstrating its potential in real-time robotic control.

2.5 Fault-Tolerant Algorithms in Redundant Robots

Fault-tolerant algorithms play a crucial role in redundant robotic systems, enabling continued operation and ensuring safety even when one or more components fail. These algorithms are particularly valuable in high-reliability applications such as industrial automation, space exploration, and robotics for hazardous environments. Recent advancements in fault-tolerant control focus on maintaining task performance by compensating for failures in joints or actuators[72]. Figure 2.5 shows a schematic example of a redundant robotic arm reaching its intended destination despite joint failure.

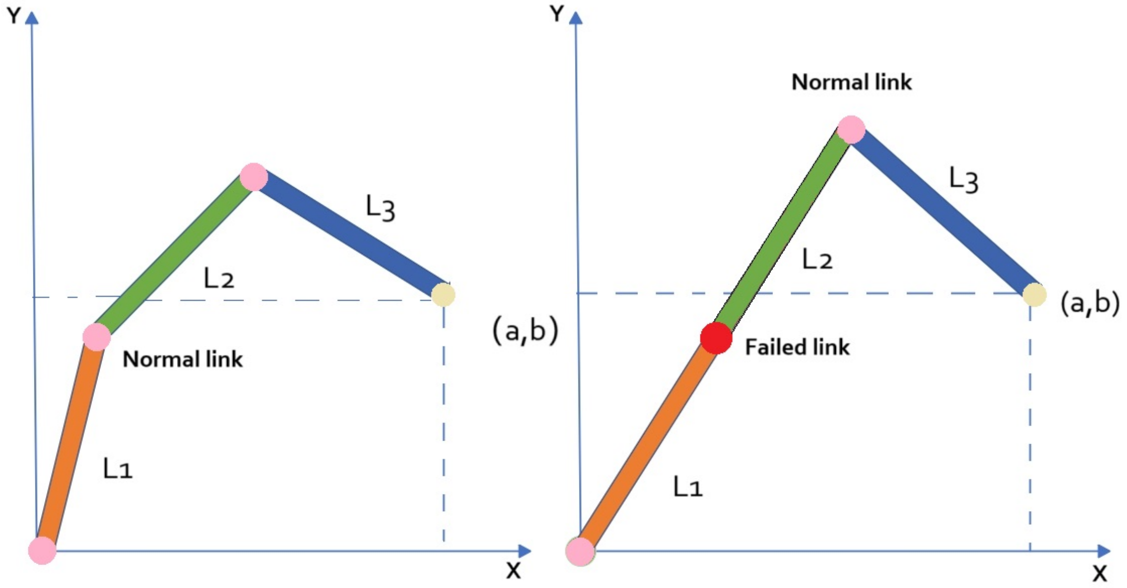


Figure 2.5: The fault-tolerant algorithm of the redundant robotic arm aims to utilize the redundant degrees of freedom of the system (i.e., the number of degrees of freedom exceeds the minimum degrees of freedom required to complete the task) to maintain the task execution capability of the end effector by reallocating joint motion instructions when a joint fails.

An effective technique is model-free fault-tolerant control, which adapts to failures without requiring a detailed dynamic model of the robot[73]. Some algorithms utilise real-time sensory feedback and statistical modelling to detect faults, adjusting joint velocities to mitigate positional or trajectory deviations. This flexibility makes model-free approaches highly adaptable, especially in applications where system parameters may vary unpredictably.

Kinematically optimised redundancy management is also employed to maintain a fault-tolerant workspace by reorganising redundant joints and prioritising reachable configurations. This strategy involves configuring the robot's kinematics to maximise a "fault-tolerant workspace," ensuring it can achieve its goals despite joint-level failures. Optimising both workspace and joint configurations is particularly effective in spatially constrained environments, such as compact industrial settings[74].

Overall, fault-tolerant algorithms in redundant robotics emphasise robustness, adaptability, and efficiency, addressing both anticipated and unexpected failures. Future research may focus on integrating these techniques with advanced machine learning methods to enhance real-time fault detection and adaptive response capabilities in dynamic or unpredictable environments.

2.6 Robot Platform Introduction

2.6.1 KUKA iiwa Robot Manipulator

The KUKA LBR iiwa (Intelligent Industrial Work Assistant) is a lightweight, collaborative robotic arm optimised for human-machine interaction applications. As the first widely produced robot specifically designed for safe human collaboration, the LBR iiwa is equipped with advanced joint torque sensors across all seven axes. These sensors enable the robot to detect contact instantly, allowing for rapid force and speed reduction to prevent accidents, making it ideal for high-precision tasks in manufacturing, healthcare, and other safety-critical environments[75]. Figure 2.6 shows the KUKA LBR iiwa 7 R800 robot arm.

The iiwa's touch responsiveness and ability to learn programmed paths through simple commands facilitate the implementation of complex automation sequences with minimal programming. Powered by KUKA's Sunrise Cabinet controller, the iiwa offers intuitive setup and control for a broad range of applications[75].

Research and applications demonstrate the iiwa's capability for precise and compliant control, enabling it to handle repetitive, ergonomically challenging tasks independently, thereby reducing human operators' exposure to monotonous work. Its modular design and compatibility with accessories, such as pneumatic and electrical media flanges, enhance its versatility across industries, from electronics assembly to cleanroom operations in semiconductor production[75].

In summary, the KUKA iiwa robotic arm represents significant advancements in collaborative robotics, delivering safe, adaptable, and efficient automation solutions for both industrial and research environments. This flexibility and focus on safety reinforce the KUKA iiwa's prominent role in the Industry 4.0 paradigm, establishing a foundation for evolving human-robot workspaces.

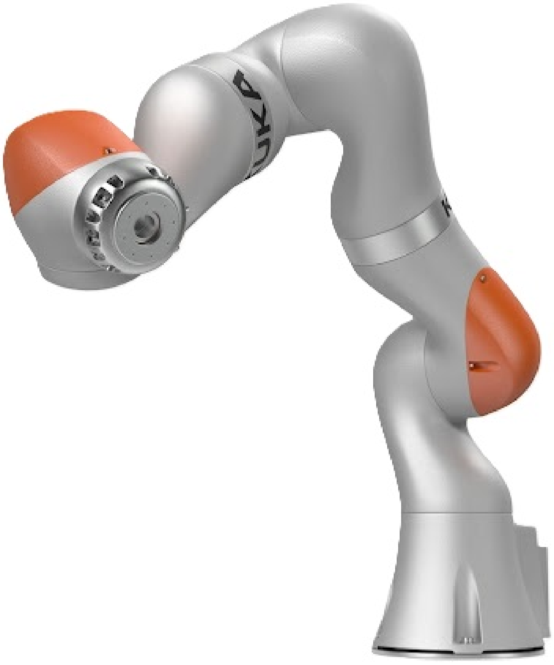


Figure 2.6: KUKA LBR iiwa 7 R800 robot arm. Adapted from reference 76.

2.6.2 Control Systems and Communications

The KUKA iiwa robotic arm integrates a range of control options, including real-time motion adjustments and compliance control, enabling it to adapt to environmental changes and interact safely with human operators. This flexibility is supported by the KUKA Sunrise Cabinet, the robot's main controller, which accommodates multiple communication protocols, such as EtherCAT and the Robot Operating System (ROS), thus providing robust interfaces to meet diverse application requirements[75]. Figure 2.7 shows how the KUKA Sunrise Cabinet works.

To enhance versatility, KUKA iiwa's communication system is compatible with ROS and ROS2 through frameworks like libiiwa, which streamline communication across platforms. This compatibility is particularly valuable to researchers and engineers seeking precise control over joint velocities, positional adjustments, and compliance mode switching, enabling the iiwa to perform adaptive tasks, including grasping and autonomous navigation[76]. The iiwa's control structure also supports impedance and compliance controls, allowing fine adjustments to stiffness and damping—a crucial feature for tasks involving unpredictable objects or environments[77].

Overall, the KUKA iiwa stands out in research and industry as a top choice for applications requiring advanced control and seamless communication, promoting safe, adaptive,

Figure 2.7: Control and communication of the robot via the KUKA Sunrise Cabinet.

and efficient robotic deployment across various fields.

2.7 Summary and Research Gaps

Although neural networks, extreme learning machines, and fault-tolerant algorithms have made significant progress in their respective fields, there are still some challenges and limitations in the practical application of intuitive HMI systems:

The complexity of deep neural networks makes their decision-making process often difficult to explain, which is particularly important in applications involving safety and reliability[78]. Although ELM performs well in some applications, its randomly generated hidden layer parameters may lead to unstable model performance, especially when processing high-dimensional data[70]. Existing fault-tolerant algorithms are often optimised for specific fault types and lack the ability to comprehensively deal with multiple faults. How to design a general and efficient fault-tolerant strategy remains an open problem[74].

Based on the above analysis, future research can be further expanded in the following aspects:

- Enhance the interpretability of neural networks: study interpretable neural network structures or methods that integrate interpretable models with black box models to improve the transparency and credibility of the system.
- Improve the robustness and stability of ELM: explore more effective hidden layer weight initialisation methods or combine other optimisation algorithms to improve the performance of ELM in different application scenarios.
- Better combine fault-tolerant algorithms with practical applications of robots: design fault-tolerant strategies that adapt to multiple fault types, making the algorithm more universal.
- Improving image processing algorithms: by introducing more advanced feature extraction modules, the impact of interference factors on image processing results is reduced. For example, multi-scale feature fusion and self-attention mechanisms are used to improve the robustness and accuracy of droplet detection.
- Introducing more efficient machine learning models: using target-unbiased meta-learning methods to optimise graph classification tasks, and using BAS-optimised extreme learning machines to improve the accuracy of robot trajectory prediction. The effectiveness of these methods is verified through experiments and applied to practical scenarios.
- Designing innovative hardware interfaces: developing low-cost data gloves, using bending sensors to accurately capture hand movements, thereby realising remote con-

trol of the robot hand. At the same time, tactile feedback should be integrated into the gloves to provide two-way communication functions and enhance user experience.

Chapter 3

A Real-Time Double Emulsion Droplet Detection System Using Hand-Crafted Algorithms

This chapter is based on our published paper "A Real-time Double Emulsion Droplets Detection System using Hough Circle Transform and Colour Detection" and proposes a real-time double emulsion droplet detection system based on a manual algorithm, aiming to accurately detect and analyse the contour, size and generation frequency of droplets.

3.1 Introduction

Double emulsion droplet is a highly structured fluid in which smaller droplets are wrapped in dispersed phase droplets[79]. The outer droplet forms a shielding layer around the inner droplet, which can effectively isolate the inner droplet from the continuous phase. Double emulsion droplets as a closed reaction system, avoiding the change of reactant concentration and cross-contamination between different reactions; the inside of the double emulsion droplet can carry out a variety of biological and chemical reactions, and has the advantages of the small amount of sample reagents and low consumption[79]. With these specific advantages, double emulsion droplets are widely used in cosmetics, pharmaceutical production, cell medicine, food science, the petroleum industry, chemical synthesis, environmental monitoring and some other fields[80].

In the specific application process, sometimes it is necessary to detect the double emulsion droplet to obtain the droplet size, frequency and other parameters. In this case, it is not easy to identify the edge of the inner droplet, mainly because the inner droplet will not exactly at the centre of the wrap; and even if the inner droplet is in the centre of the wrap, the traditional Canny algorithm is not that easy to recognize it[81]. However, the size and frequency of droplets must be accurately monitored during the production process. Therefore, this chapter proposed a double-droplet detection method based on colour space, which can get information of the contour, size and frequency of the double emulsion at one time.

Redmon et al. proposed a real-time object detection method, the principle of which is that a single neural network can directly predict bounding boxes and class probabilities from the complete image in one evaluation. This unified system has a fast detection speed, but the YOLO model used may produce some localisation errors [82]. Bao et al. analysed the proportional multiplication technology under the framework of Canny edge detection, which can greatly improve the positioning criteria through proportional multiplication when the loss of the detection criteria is small[83]. This technique is more accurate than the traditional Canny algorithm, but the detection speed is relatively slow. Zhan et al. proposed an improved algorithm based on the frame difference method and edge detection for the detection of moving targets. This method can mark the connected components of the block to obtain the smallest rectangle that contains the moving object. It has a fast detection speed and a high recognition rate, but it cannot obtain accurate object edges[84].

In the traditional droplet recognition method, the colour image will be converted into a grey image, and then the grey image will be preprocessed to remove noise [85]. The next step is to use the canny operator or threshold segmentation method to process it and then go through suitable post-processing to get the information of the droplets[86]. Zhao etc. from Beijing Information Science and Technology University introduced four algorithms through Open CV programming, which can visually display the detection results under different thresholds to facilitate image edge detection[87]. However, the non-maximum suppression and dual threshold processing adopted by it require high time complexity, and it takes a long time to process on high-resolution images, which cannot meet the real-time scenario. Moreover, the fixed threshold solution demonstrated by the team cannot adapt to changes in illumination. In images with an illumination gradient greater than 50 pixels per centimetre, the edges of dark parts may be completely lost, and a large number of pseudo edges will be generated in bright parts. This research intends to find a way to directly extract and divide the colour of the colour image, then the information can be obtained after post-processing. Compared with the traditional method, the pre-processing and post-processing work of this method is relatively simple, which can greatly increase the processing speed. This method requires the use of a colour camera and dyeing the droplets. With the development of imaging technology, the cost of colour cameras is not expensive, and the algorithm itself has a strong tolerance for chromatic aberration.

In computer vision, colour components are best separated from intensity due to reasons such as lighting changes or eliminating shadows. YUV is a colour coding system, mainly used in video and graphics processing pipelines. Y stands for the luma component (the brightness), U and V are the chrominance (colour) components[88]. It can make the encoding and transmission more convenient, reducing bandwidth occupation and information errors[89]. The conversion matrix of YUV and RGB space is shown as follows, wherein the YUV is an advanced option for colour processing while the RGB is the most commonly used colour display method[90]. OpenCV is a computer vision library developed by Intel. Its main role is for real-time image processing, computer vision and pattern detection programs. It is completely open-source and can be used well in engineering practice[91].

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \begin{matrix} Y \in [0, 255] \\ U \in [-111, 111] \\ V \in [-157, 157] \end{matrix} \tag{3.1.1}$$

Hough circle transform is an image transform that can extract circular objects from the image. This transformation can effectively search for objects with high radial symmetry, with each degree of symmetry receiving one “vote” in the search space[92]. Grycuk and etc. used the Canny edge detection method and quadrilateral detection algorithm to obtain a list of objects of interest, but this method can only be applied to a single picture[93]. Therefore, this research intends to find a double emulsion detection method that can be applied to fast cameras and can be monitored in realtime. The contribution of our proposed tracking system can be highlighted as follows:

1. There is no need to determine the colour of the droplets being tracked, the first two peaks of colours will be grabbed all at once, one is for the inner droplet the other is for background, which makes a faster detection speed and fulfils the district requirement of microfluidics detection.

2. Combining colour thresholding with Hough circle algorithms, this research provides an opportunity to online monitor the quality of microfluidics formation by selecting those failed formed droplets which do not have the inner coloured droplets.

3. The novelty is primarily in bringing together existing technologies to a complex task. The novelty and innovation required in this kind of work are not to be underestimated, whilst the system will be based on a number of existing technologies, combining these in a way that makes a reliable, coherent and effective system is a complex research challenge in its own right.

The Canny algorithm is a relatively straightforward algorithm, but the degree of freedom is relatively low, which requires manual debugging of a large number of parameters (two thresholds) to select the edge[87]. The first step of Canny edge detection is to use Gaussian blur to remove noise, but this step will smooth the edges, making the edge information weaker, which may cause some required edges information to be missed in the subsequent steps, especially weak edges and isolated edges[94]. It can be eliminated in the calculation of the double threshold[95]. The Sobel operator is not very accurate in positioning edges, and the edges of an image are more than one pixel[96]. The detection is not stable enough when the traditional edge detection algorithm is applied to high-speed cameras[97]. The detection algorithm we proposed can reach the point directly and has a strong purpose. At the same time, the detection speed and accuracy can also be guaranteed.

3.2 Problem Description

In order to detect double emulsion droplets through colour space, the droplets need to be dyed first. The advantage of this method is that it can detect the edges of both inner and outer droplets at the same time. When detecting the edge of the outer droplet, multiple rays (vectors) are emitted from the centre of the droplet outward (especially toward the left side of the nozzle). Along each ray, the darkest point is identified, and the collection of these points preliminarily constitutes the edge contour of the outer droplet.

However, this method may be subject to two interferences that lead to false detection: one is that there may be dark particles in the outer area of the pipe, and the other is that the inner droplet is darker than its wrapping layer (outer droplet). In order to solve the interference of the outer pipe, a white background can be covered in this area. For the problem that the inner droplet is too dark in color, which may be caused in the detection of the outer droplet edge, since the overall mask of the droplet has been obtained in the previous step, the part within the mask area can be covered with white, thereby shielding the interference of the inner dark area.

Finally, the complete outer droplet edge is formed by merging the darkest points identified on all rays. Since this preliminary edge may have errors, a standard window filter needs to be applied to smooth and optimise it.

After converting the image to YUV format, the frequency of each tone is recorded on the entire image. In this way, an image with two peaks can be obtained, where the first peak is the background, and the second peak is the inner droplet. Since the background creates many close peaks, the image must be filtered by the average window. However, multiple peak detections are still triggered in the background. Find the next peak through the points with a descending gradient on both sides of the first peak. When the largest peak has been found, a dead zone should be set around it, and this dead zone should be ignored after the second peak is found. After locating the second peak, consider it as the chroma of droplets. All pixels whose colours are closer to the chroma of the droplet than the background colour are marked as white on the droplet mask. In the next step, the image is morphologically filtered to remove noise.

The number of double emulsions needs to be detected. This method is mainly aimed at the situation where a large droplet wraps a small droplet. In the situation where one droplet wraps multiple droplets, it is easy to overlap during detection, which will lead to errors. While determining the number of double emulsions, it can also identify the edges of the two droplets inside and outside, then get the size of the droplets.

The frequency of droplet generation can also be directly derived. It can measure the production time of a single droplet without the need for similarity conversion. By achieving the above goals, the production of the double emulsion can be well monitored, so that it can stably produce large droplet wrap small droplet.

3.3 Proposed Methodologies

3.3.1 Principle of Droplet Detection

Compared with the traditional method, this method extracts a specific colour by extracting the value of U and V, reference data needs to be checked to find the U, V values range of a certain colour. This method needs to manually set an initial value and a value range. The first step is to use chroma values to perform colour segmentation on the target. The algorithm associates different colour pixels in the visible range of the field of view with their corresponding colours. As shown in Figure 3.1, the most common distribution is taken as the background colour; the second most common distribution is taken as the colour of the droplet. This method can quickly extract the droplet colour and background colour no matter what colours they are.

The second step is to perform droplet detection through binarisation. As shown in Figure 3.2. First, randomly select a pixel in the visible range of the field of view and determine whether the point is a background colour or a droplet colour. If it is the background colour then give it black colour, while the droplet colour gives it white colour. So droplet detection can become spot detection. Before this process, noise reduction is required through filtering. Then by determining the centre of the mass point to identify the centre point of the spot (inner droplet), this centre point is a necessary condition for subsequent identification of the outer edge and calculation of the size.

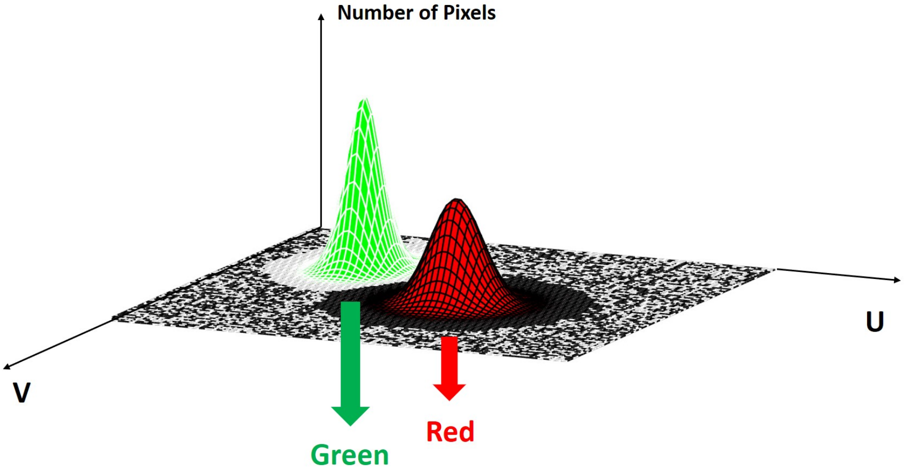


Figure 3.1: YUV space diagram. Green is the most distributed chroma, which is the background colour; red is the second most distributed chroma, which is the droplet colour.

3.3.2 Determine If the Droplet is Successfully Formed

If the double emulsion has successfully formed, its shape is close to a circle; if it fails, it has an irregular shape. First, find the centre point of the inner droplet, then find the farthest point on its outer droplet from this centre point. Calculate the area of the fitted circle with the connecting segment of these two points as the radius, and compare this area with the actual area obtained by calculating the sum of pixels. In Figure 3.3, if the droplet is formed successfully, the area obtained by fitting is not much different from the actual area; if it fails, the area obtained by fitting will be much larger than the actual area. Assuming S_0 is the size of the outer droplet, obtained by counting the pixels numbers. The following equation can be derived:

$$\xi = \pi r^2 - S_0 \tag{3.3.1}$$

$$sign\left(\xi\right) = \begin{cases} -1, & \xi < -a \\ 0, & -a < \xi < a \\ 1, & \xi > a \end{cases} \tag{3.3.2}$$

Where r is the radius of the fitted circle, which is shown in Figure 3.3 as red circle; a is the preset gap factor, which can be adjusted according to the difference allowed by different experiments. When the output of the $sign$ function is 0, the droplet is successfully formed.

3.3.3 Wrap Detection Principle

The traditional method is to determine the centre point of the entire double emulsion droplet through the droplet inside[97]. In this research, we present two methods to detect

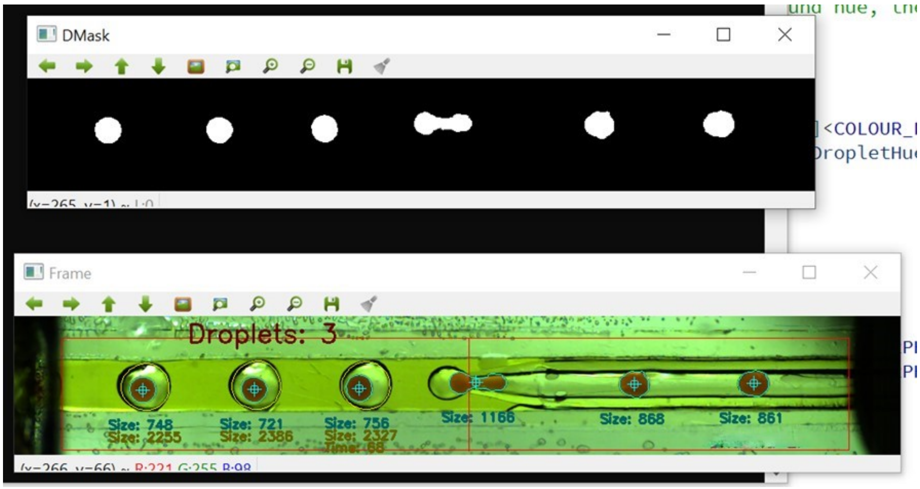


Figure 3.2: Field of view obtained by binarisation. Give the background colour black and the droplets colour white.

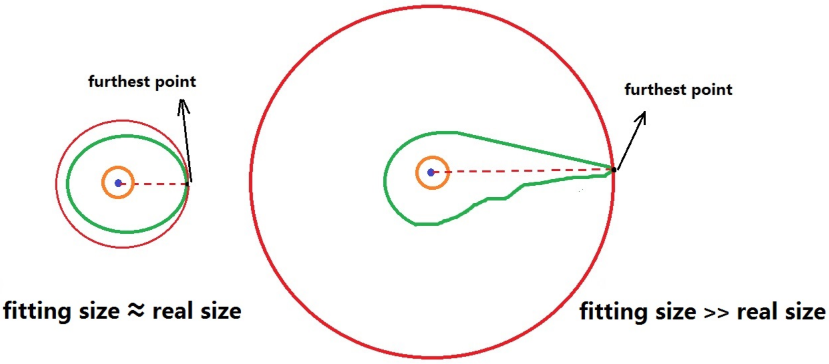


Figure 3.3: The area of the fitted circle has been calculated and compared with the real size. The green line represents the true outer edge of the droplet, and the red line represents the circle formed by the fitting.

the outer edge of the droplet. First, as shown in Figure 3.4, the centre of the inner droplet should be taken as the starting point, and use a vector of a certain length to make the half-line. The length of the half-line can be defined by yourself. Find the darkest point on each ray and circle it to form an external contour detection. Then the inner and outer edges of the droplet can be identified and extracted. In this case, the formula can be expressed as:

$$\vec{S_n} = \vec{j} \cdot r \cdot sin\left(n \cdot \theta\right) + \vec{l} \cdot r \cdot cos\left(n \cdot \theta\right) \tag{3.3.3}$$

Where r is the preset vector length; n is a geometric multiple, which represents the number of iterations. The unit vectors $\vec{j}$ and $\vec{l}$ can be synthesized into the required vector $\vec{S_n}$ when the angle θ given, where angle θ is the angle between the sum vector and the positive direction of the vector $\vec{j}$.

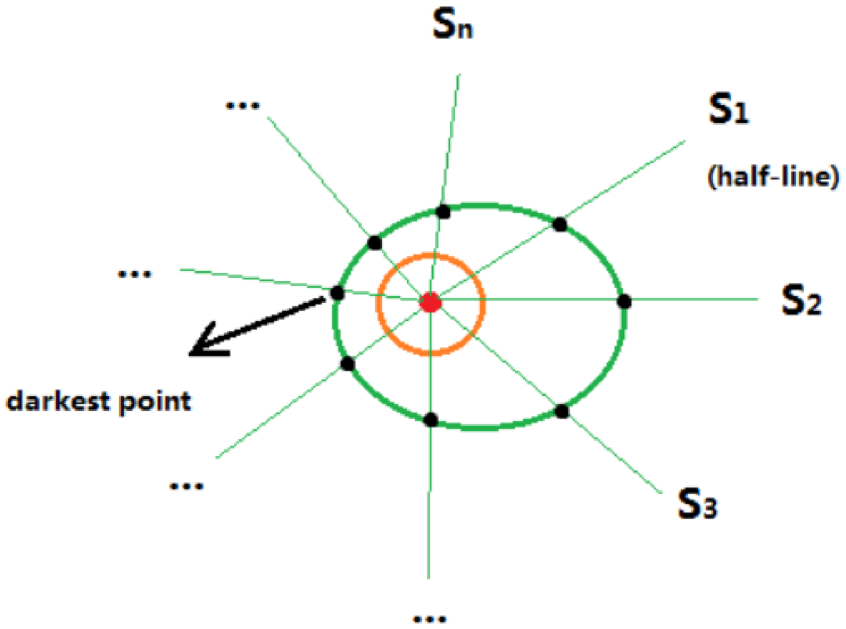


Figure 3.4: The yellow circle indicates the contour of the inner droplet that has been identified, and the red point is the centre point of the inner droplet. The half-line is emitted with the red point as the centre, and the collection of the darkest points on each half-line is the outline of the outer droplet, which is represented in green.

However, if the inner droplet did not successfully enter the outer droplet, the detection would not succeed. The second method is to apply the Hough transform algorithm for detection. Specifically, it is the curve transformation applied to the Hough line transformation. For a certain point in the pixel space, through the Hough transform, it is possible to find all the specific linear shapes of this point. In this experiment, the edge shape is close to a circle, so find all n circles passing through this point. As shown in Figure 3.5, if the point is on the real edge of the outer droplet, then there is one and only one of these n circles, which is the real edge of the outer droplet. So we need to constantly change the position of the point to find the circle with the highest repetition rate, which is the outer droplet edge.

For the known circle, the general equation in the rectangular coordinate system is:

$$(x-a)^2 + (y-b)^2 = r^2 \tag{3.3.4}$$

Where (a, b) is the coordinates of the centre of the circle, and r is the radius of the circle[98]. Then the parameter space can be expressed as (a, b, r), and a circle in the image coordinate space corresponds to a point in the parameter space. The calculation process is to increase a and b within the range of values, and solve the r value that satisfies the above formula.

Each time a value of (a, b, r) is calculated, the corresponding array element $A(a, b, r)$ plus 1. After the calculation, the (a, b, r) corresponding to the largest $A(a, b, r)$ are the parameters of the desired circle[99].

The Hough transform algorithm has been used to scan all points within the visible range of the screen, then the edges of the outer droplets can be extracted.

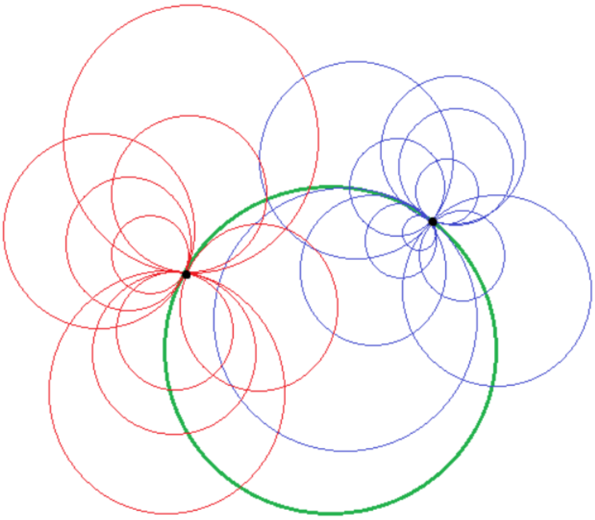


Figure 3.5: The Hough transform has been used to find the circle with the highest repetition rate is the outer droplet edge, which is represented in green.

This circle is obtained by the Hough transform fitting. However, this circle is still different from the real shape of the droplet. AS shown in Figure 3.6. The next step is to use the fitted circle to extract its true outer edge. The method is similar to the previous method. Two thresholds are added so that the vector only scans the area within the threshold to get the set of the darkest points, which is the real outer droplet edge. This method can extract empty droplets, which is the inner droplet did not successfully enter the outer droplet. The size of the droplets can be measured by counting the total number of pixels in the edge that has been identified, the total number of pixels can be converted into area units after getting the scale factor data. In this case, the equation is expressed as:

$$\vec{S_n}' = \vec{j} \cdot \left(R - r'\right) \cdot sin\left(n \cdot \theta\right) + \vec{l} \cdot \left(R - r'\right) \cdot cos\left(n \cdot \theta\right) \tag{3.3.5}$$

where the large threshold is represented by R, small threshold is represented by r'. The meanings of other parameters are the same as in equation 3.3.4.

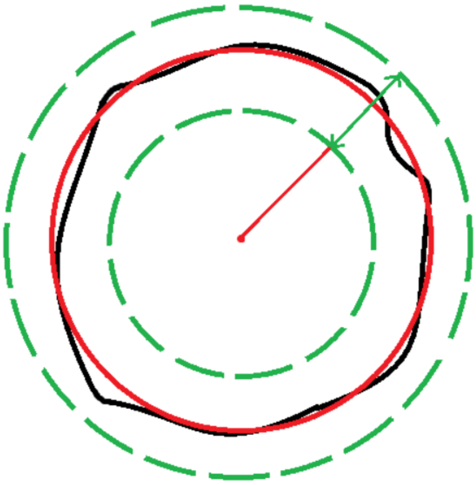


Figure 3.6: The red circle is the fitted circle, and the green dashed line is the two thresholds based on the fitting circle after scaling and expansion. By scanning the area within the threshold, the true edge of the outer droplet can be obtained, which is represented in black.

3.3.4 Principle of Frequency Identification

The frequency of droplet generation is obtained by calculating how long it takes for the droplet to pass through the nozzle. A line has been drawn from the centre of the inner droplet to the nozzle, then check if there is an edge that exists along the line to judge the rupture. The frequency of droplet generation has been determined by counting the number of frames. The start frame is when the droplet begins to form, and the end frame is when the droplet just formed. Therefore, the key point of this algorithm is to determine when the droplet begins to form and when the droplet has just formed. If the droplet is in the forming state, the number of frames is increased by 1; if it is already formed, the number of frames is increased by 0, and the number of frames should be output as the frequency at the same time.

The state of the droplet is judged by the leading point and the tailing point. In Figure 3.7 to 3.10, the m-line is the line segment where the droplet nozzle is located, perpendicular to the direction in which the droplets are generated. The following situations can be analysed:

1. In Figure 3.7, the leading point has not reached the droplet nozzle yet, the number of frames should be output as the frequency. if the leading point has passed the droplet nozzle, the number of frames should be increased by 1.

2. In Figure 3.8, the leading point has passed the droplet nozzle and tailing point has not reached the droplet nozzle, the number of frames should be increased by 1.

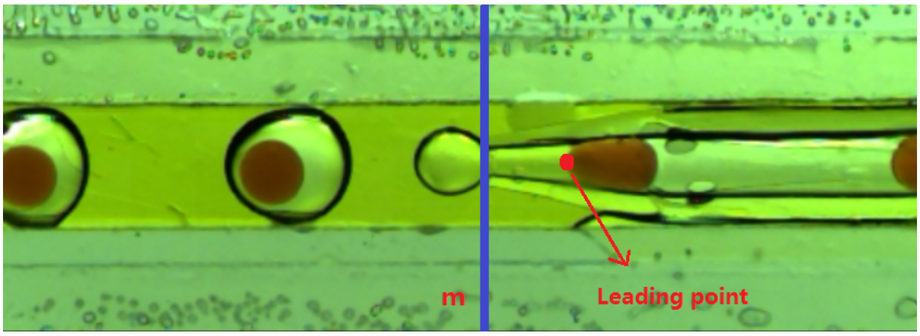


Figure 3.7: Leading point has not reached m-line.

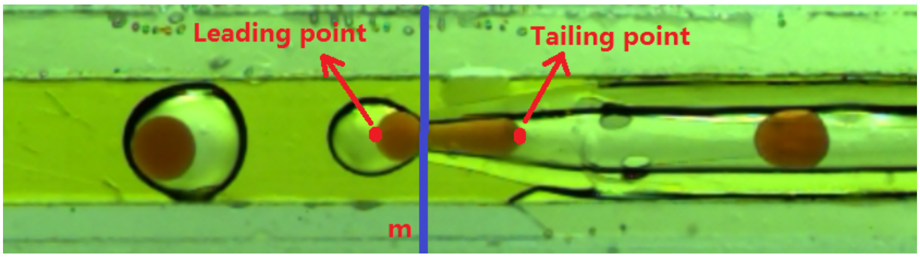


Figure 3.8: Leading point on the left side of m-line and tailing point on the right of m-line.

3. In Figure 3.9, if the second leading point can be found between the previous leading point and droplet nozzle, the number of frames should be increased by 1. If the second leading point cannot be found, it may also be in the forming state, so further verification statements need to be added.

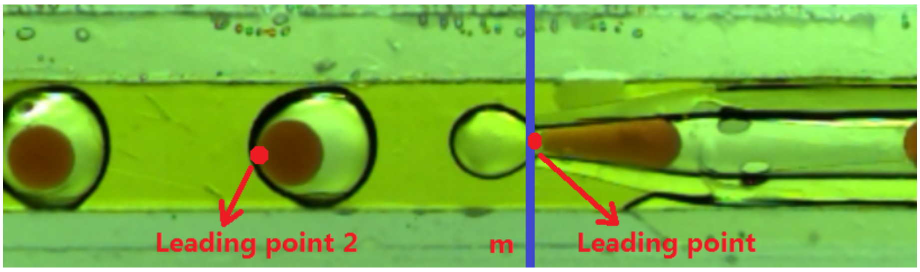


Figure 3.9: Whether there is a new leading point between leading point 2 and m-line. Leading point 2 means the previous leading point.

4. In Figure 3.10, the centre of the inner droplet to the midpoint of the m-line has been connected to see if there are two edges on this line segment. If yes, it means that the droplet has been formed, the number of frames should be output as the frequency; if there is only one edge on the line segment, it means that the droplet is still forming, so the number of frames should be increased by 1.

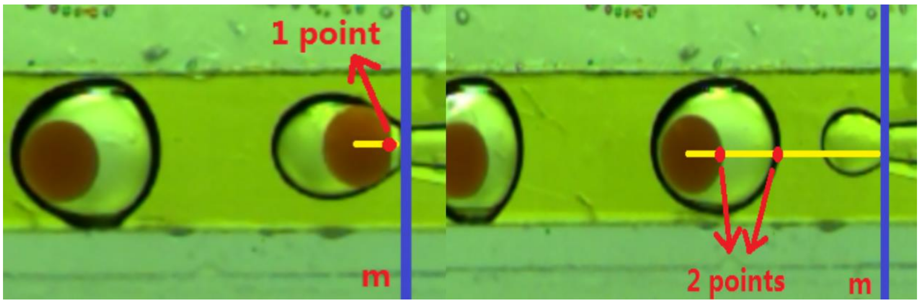


Figure 3.10: Whether the droplet is formed should be determined by the number of edges on the line segment, which connects the centre point of the inner droplet and the midpoint of the m-line.

3.4 Experimental Results and Analysis

The video in the experiment runs at a frame rate of 30fps, and the program needs about 7-11 milliseconds per frame to track all droplets, so it can track in real-time at about 100 fps. This can be optimised through better hardware, optimisation, GPU acceleration, etc. In the experiment, the accuracy of the system was close to 100%, but the situation of the experiment video was relatively ideal.

When extracting the colour space, a one-step process method has been considered in this research, which is to determine the background colour and the colour of the droplet by catching the two peaks of the ordinate. This method does not need to consider what colours they are but only needs to find the peaks. Another advantage of this method is that the detection is accurate, and multiple colours can be extracted in the user interface, which is very flexible. Different microfluidic devices have been used to validate our proposed detection system.

3.4.1 Transparent Background with coloured Inner Droplet

In this case, the statistically calculated YUV curve has only one peak. In Figure 3.11, the background is nearly transparent and the droplets are blue. To extract the colour blue, the U value needs to exceed the peak. Its value range is [133,255]. If a cleaner extraction is needed, the V value can be reduced to meet the requirements, then the effect of red can be eliminated. The value does not include the main peak and the range is [1,123].

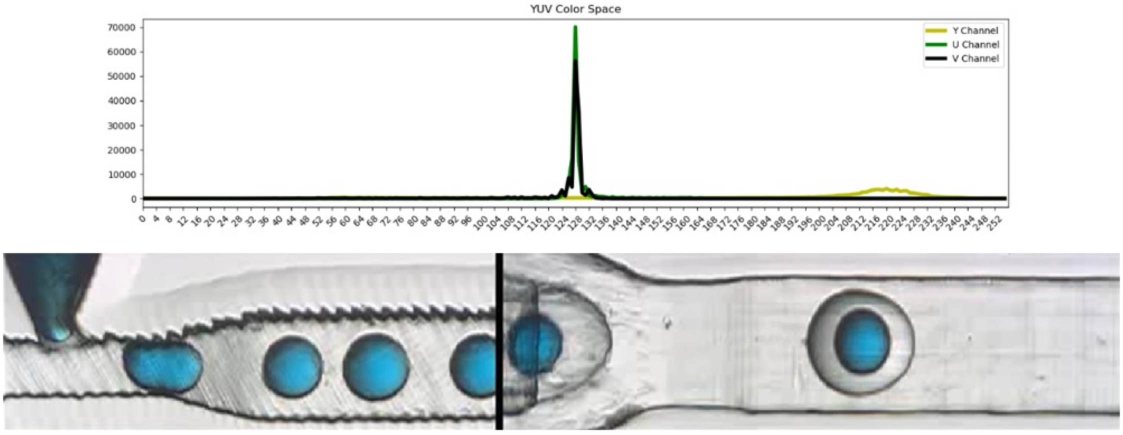


Figure 3.11: The background is nearly transparent and the inner droplet has one colour.

3.4.2 Dyed Background with Darker coloured Inner Droplet

The background colour is caused by the dyeing of the outermost liquid, which interferes with the colour extraction of the inner droplets. In Figure 3.12, the red colour of the inner droplet needs to be extracted, which is relatively darker. Then the V channel should be focused, there can be two peaks. The first peak is the green of the overall pipe background, and the second peak is the red of the inner droplet. The colour green can be composed of red and blue, so red can also be detected in the background. If the second peak value is taken, the inner droplet size will be underestimated because the red colour inside the droplet is uneven. The refraction of light caused by the droplet curve will also lead to uneven red, which will cause the red at the edge of the droplet to be less red, resulting in the identified droplet size being smaller than the actual size. It can be known from the experiment, the detection effect is better when the middle value of the two peaks has been taken.

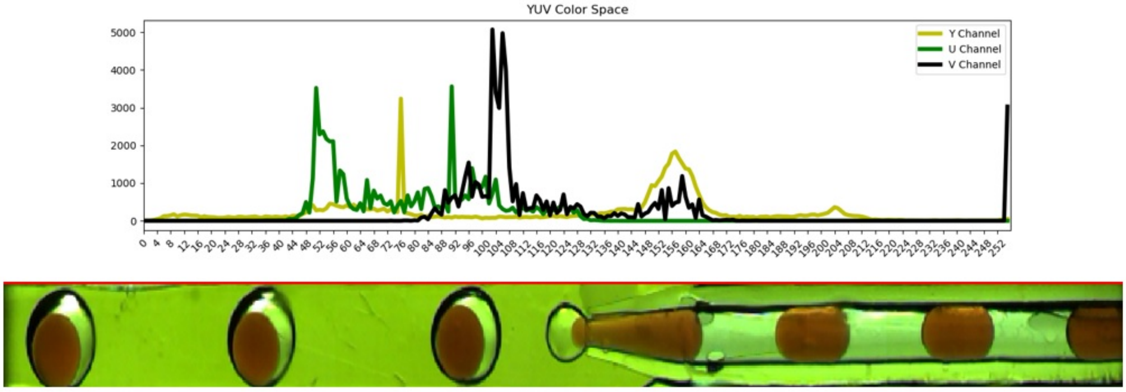


Figure 3.12: The background has been dyed, and the inner droplet has one darker colour.

3.4.3 Dyed Background, coloured Inner Droplet with Unevenly Distributed Illumination

In this case, the V channel should be focused on extracting the red colour and the YU channel should be ignored. As shown in Figure 3.13, two peaks have appeared. Regarding the value of V, the intermediate value of 130 between the two peaks has been taken. Two methods can be used in the brighter area to eliminate the impact, increasing the U value or decreasing the Y value. The interval of U value must cross the second peak, which can be determined as [56, 255]; the interval of Y value can be determined as [0, 196], because it belongs to the brighter area, only the two largest peaks need to be avoided.

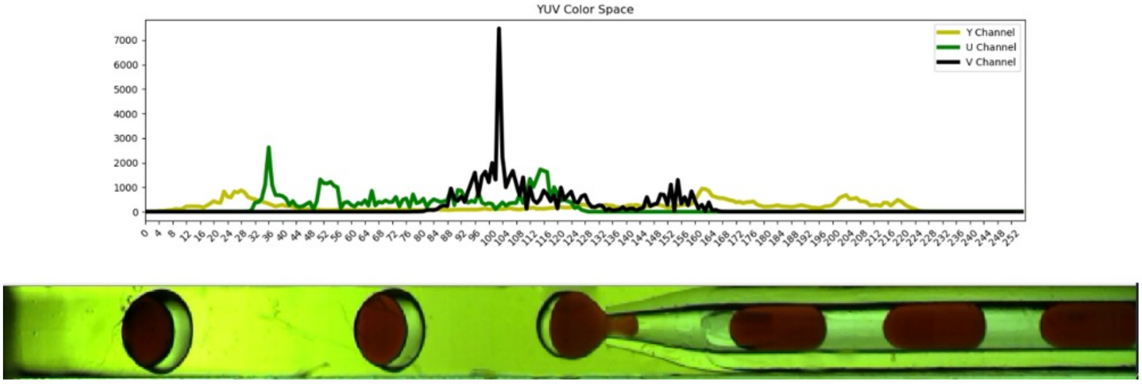


Figure 3.13: The light field is relatively uneven.

3.4.4 Dyed Background with Lighter coloured Inner Droplet

In Figure 3.14, the inner droplet is red, and the colour is relatively lighter. To extract the red colour, the V channel should be focused on. There are two peaks in Figure 3.14, but the peaks are not very clear. The value interval of V is [104, 255], in the middle of the two peaks. However, there are shadows in the outer droplets. This problem can be solved by three methods: debugging U value separately, debugging Y value separately, and debugging U value and Y value at the same time. The U value range should include the main peak, so take the value just past the main peak. In this figure, the U value can be debugged in [0,110]; debugging the Y value alone is not very effective in this example, because the pixels of the droplet will be lost; debugging the U value and Y value at the same time can get a cleaner image, but the difference between this result and the result when debugging the U value alone is not much different.

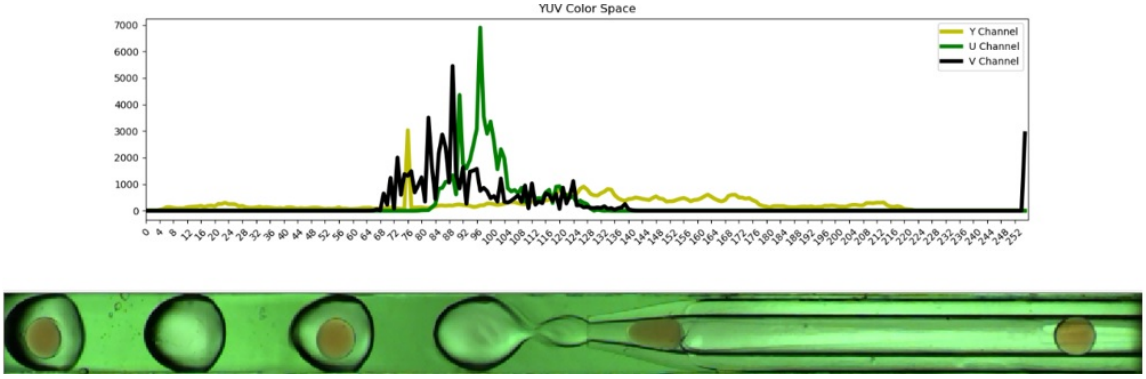


Figure 3.14: The inner droplet has one lighter colour.

3.5 Conclusion

Through the analysis of the experimental results, it can be concluded that this method can detect double emulsion and obtain accurate edges about inner and outer droplets. At the same time, information such as the size and frequency of droplets can also be accurately acquired. And it can effectively detect and mark the droplets that have failed to form. In the high-speed production process, this method is still very effective. This system currently only uses video for offline detection, the next step is to embed the model into a high-speed camera to achieve real-time monitoring. Our future work will also focus on the implementation of colour tracking-based algorithms into trajectory mapping of a Baxter robot with a moving coloured item, which is also a further expansion of our previous work, to achieve user experience enhanced human-robot interaction[100]. This method can play an important role in a production environment that requires real-time monitoring.

Chapter 4

Target-Unbiased Meta-Learning for Few-Shot Graph Classification Using Graph Neural Networks

This chapter is based on the published paper "Target Unbiased Meta-Learning for Graph Classification" and explores the mathematical principles and learning methods of neural networks in graph classification tasks.

4.1 Introduction

Even though the algorithms based on deep learning have powerful feature extraction and knowledge expression abilities[101], there are still challenges. On the one hand, deep learning methods fully depend on datasets that require considerable labelled samples for training purposes[102]. Conversely, there is plenty of untagged data that needs to be artificially tagged, which is expensive and time-consuming (a few-shot learning problem). On the other hand, deep learning is merely for a specific task. In other words, models are only trained for the current tasks. Hence, the performance of trained models needs to be promoted while doing new tasks that have never been seen[103]. In addition, owing to lacking training samples, the trained models may work well on the training set but encounter parameter overfitting problems on the testing set. Therefore, another key point of solving the few-shot learning problem is to overcome the problem of overfitting. Due to these inherent defects of deep learning, it remains gravely difficult to realise artificial intelligence by using deep learning alone[104].

Meta-learning, often characterised as "learning to learn," has garnered significant research interest. Distinct from conventional artificial intelligence approaches, its primary objective is the enhancement of learning algorithms themselves by leveraging accumulated experience across multiple learning episodes. This paradigm offers potential solutions to persistent challenges in deep learning, such as limitations imposed by data scarcity and computational demands, alongside fundamental issues of generalisation. Crucially, meta-learning shifts focus from solving specific tasks to mastering the process of learning itself, utilising previously encountered tasks. While meta-learning constitutes a subset of machine learning, a key distinction exists: whereas traditional machine learning aims to enable machines to learn specific tasks, meta-learning seeks to enable machines to learn how to learn effectively. Fundamentally, meta-learning extensively utilises prior knowledge and experience to guide and accelerate the acquisition of new tasks, embodying the ”learning to learn” capability. It is arguably essential for achieving more robust and human-like artificial intelligence. Consequently, meta-learning has emerged as a prominent and critical research direction.

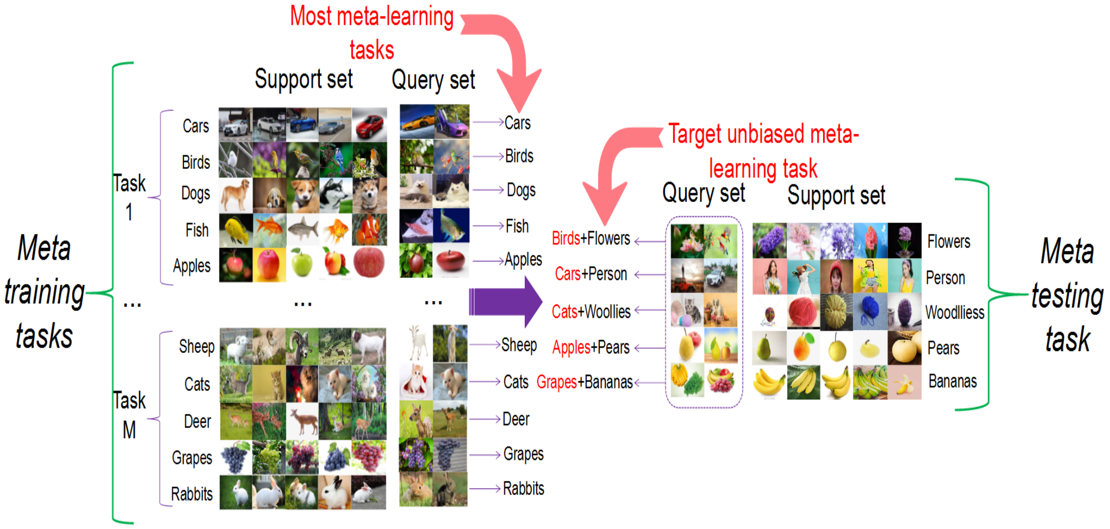


Figure 4.1: **The overview of meta-learning tasks and target bias.** To simulate a meta-testing task, we derive M 5-way 5-shot tasks applied for meta-training. What is worth noting is that not only do the samples of the query set used for meta-testing contain the classification targets(flower, person, woollies, pears, bananas), but also include the targets(birds, cars, cats, apples, grapes) seen in the meta-training stage. Under these circumstances, the trained model is more inclined to identify the target that has been seen and produce the target bias problem, which will affect classification accuracy.

However, although the existing meta-learning methods have achieved excellent results with broad prospects, they all extract feature vectors from the support set and the query set separately, which fail to consider the difference between the sample targets in the support and query set[105]. The emerging meta-learning method can be used with the training set to derive multiple few-shot classification tasks and simulate classification tasks with a few labelled samples of the testing set. However, when most existing meta-learning methods extract feature vectors, they only take the benefits of making full use of data information caused by prior knowledge into account, but fail to consider the preference of the training model for targets that have already been seen. Also, meta-learning tasks coexist with multiple tasks; they have a certain memory ability for prior knowledge. It will lead to the extracted feature vectors not being conducive to the current classification task if this preference is not considered, and we define this preference as "target bias" (representing the preference of the model for the meta-training targets that have been seen in the process of training the classification model). As illustrated in Figure 4.1, the labelled dataset in the training task is images of cars, birds, etc. The testing task, unfortunately, keeps classifying flowers, people, etc. If there are cars or birds in the image to be classified besides flowers or persons, the trained model may be more inclined to regard cars or birds as classification targets, other than flowers or persons, which may lead to deviations in the classification results. In order to alleviate this deviation, this research proposes an interactive attention extraction module in the feature extraction process, which has been exploited to improve the separability of feature vectors and reduce the model's preference for a certain target. It is worth noting that our interactive attention extraction module can be used as a supplement to any feature extraction module, reducing the model's preference of the "seen" targets to improve the classification accuracy.

Also, most of the data in the real world have complex interactive relationships, which can be represented by graph structures, and GNNs are created to fully explore the relationships between samples. In order to make full use of the relationships between samples, a neural network has been introduced as a classification module to map the image-level classification task to the node level, and obtain the implicit relationship between the samples through the node feature updating module and the edge feature updating module, where each feature vector was treated as a node, and the GNN was utilized to fully express the relationship in between. Simultaneously, in order to solve the issues that the distribution of samples cannot truly represent the distribution characteristics of real data triggered by few training samples, and to further deal with the amount of data that is incapable of meeting the requirements of training and learning, we took the advantages of semisupervised learning[106] and transductive inference[107] to enhance the effect.

In summary, our contributions are three-fold: **First**, a simple and well-behaved target unbiased meta-learning method for graph classification has been developed; it operates directly on graphs and can quickly lock the target domain of the current classification task and solves the target bias problem of other graph classification methods. **Second**, we demonstrate how this form of a graph-based neural network model can be used for the few-shot classification of nodes in a graph. **Third**, we demonstrate the superior performance of the proposed approach on the standard few-shot learning benchmark datasets and consider new few-shot learning settings (such as transductive and semisupervised learning) that are also important real-world use cases and results show that our approach can outperform baseline algorithms under these settings too.

4.2 Related Works

Aiming at solving the problem of few-shot learning, a large number of recent studies have focused on meta-learning, because it can quickly adapt to new tasks and transfer useful knowledge between tasks with fewer samples. Models and algorithms of meta-learning, in a general way, can be basically divided into the following three types: optimisation-based methods, metric learning-based methods, and memory-based methods. The following is a specific introduction to these distinguished methods.

The optimisation-based methods are designed to quickly update the parameters on scarce samples through the design of the model structure, and directly establish the mapping function between the input and the predicted value. Chelsea Finn and Sergey Levine proposed the MAML algorithm in 2017[108], which is currently one of the most elegant and promising meta-learning algorithms. In 2018, OpenAI released the simple meta-learning algorithm Reptile[109], which repeatedly samples a task, performs stochastic gradient descent(SGD), and updates the initial parameters until the final parameters are learned. The authors of Jamal and Qi successfully trained a task-agnostic meta-learning algorithm, alleviating the problem of task preference[110]. Guo and Cheung etc. proposed a few-shot learning algorithm (AWGIM) based on weight generation[111]. The above methods have achieved extreme success in tackling the problem of few-shot learning. Unfortunately, due to the limitations of optimiser selection (such as SGD, Adam, etc.) and learning rate, optimization-based meta-learning methods usually have to update multiple steps to reach a better point, so when models face new tasks, the learning process remains extremely slow. Worse, they are easily overfitted when updating the weights in the case of trains.

The basic idea of methods based on metric learning is to compare the distance between query set samples and support set samples, and then utilise the means of nearest neighbours to achieve classification. The research works[112] are inspired by the Siamese Network[113], the Matching Network[114], and the Prototypical Network[115] and then use metric learning ideas to implement few-shot classification tasks. Metric-based meta-learning methods, also known as non-parametric methods, perform well for few-shot classification, but their effectiveness in other meta-learning fields, such as regression or reinforcement learning, has yet to be proven. Also, when the differences between the testing and training tasks are evident, the effect fails to appear. Furthermore, when the task becomes larger, the pairwise comparison may result in expensive computational costs.

Memory-based methods are to construct an external memory, introduce prior knowledge into it, and then use it to realise the classification of few-shot. The MetaNet[116] proposed in ICML2017 is a method that realizes rapid parametrization of generalisation tasks. CVPR2018 proposes a few-shot learning algorithm MM-Net[102] based on external memory. The innovation of this article is to use external memory modules to achieve few-shot learning tasks. Memory-based methods are one of the most commonly used methods, and have been implemented in few-shot classification, regression tasks, and meta-reinforcement learning. Although these methods are more flexible, due to the learner networks needing to design a learning strategy from scratch, they are slightly less efficient than other meta-learning methods. In addition, they impose restrictions on the model (RNN), which may prevent its development and application to some extent.

The above meta-learning methods have completely affected an immense number of learning tasks, from image classification, and video processing to speech recognition, natural language processing, etc. Data, on the other hand, are non-linear and expressed as a graph structure of complex relationships and interdependence in between[117]. Hence, quite a few indispensable operations such as convolution are not suitable for graph data[103]. To sort it out, the GNN is used to embed data into a suitable Euclidean space. Victor Garcia et al. used the few-shot learning algorithm of GNN to transfer the distance measurement from Euclidean space to non-European space[118], and calculated the relationship between images by graph model, although the thought is very simple, but obtaining an advanced effect. Moreover, some new network architectures have been proposed recently, such as Graph attention networks[119], Graph generative networks[120], and Graph spatial-temporal networks[121]. In addition, Luo et al. proposed a deep relationship network to capture the relationship between different samples and constructed a knowledge graph by linking images with tags[122]; Lin et al. proposed a network (HOSP-GNN) for few-shot learning, which can not only use relative metrics in multiple samples to describe higher-order structural relationships, but also re-formulate the update rules of the graph structure through alternate calculations between vertices and edges based on higher order structures[123]; Ding et al. introduced a new graph element learning framework-Graph Prototype Network to solve the problem of the few-shot node classification on the attribute network[124]; Ma et al. proposed a direct method that using good initial value to capture the substructure[125]. This research focuses on the classification of few-shot learning and proposes a new framework AS-MAML. What is worth noting is that Yanbin Liu et al. proposed a transductive method that carries on the label propagation to the training set and the testing set in the episode, by establishing an undirected graph together with all the unlabeled data and the annotated data, solving the problem that the classifier is unreliable, due to the lack of training data. Compared with inductive inference methods, it has achieved advanced results[126]. In this research, similar transductive inference methods are used to make full use of the data without labels. The experimental results show that the classification ability of the model is improved after the addition of transductive inference. However, these methods also have the following shortcomings: They only take the benefits of making full use of data information caused by prior knowledge into account, but fail to consider the preference of the training model for samples that have already been seen. To a certain extent, it will interfere with the model's classification task. In order to alleviate this deviation, this research proposes an interactive attention extraction module.

Considering the nature of metric learning, the difference between our work and the above methods is that, through employing the interactive attention extraction module during the process of feature extraction, the trained model is able to quickly locate the target to be identified, and naturally reduces the interference of other objects in the same situation. Furthermore, in order to improve the accuracy of classification, the advanced GNN is therefore combined with the classification module. The specific implementation process is described in Section 4.4.

4.3 Problem Definition

Meta-learning: It introduces a series of concepts, including N-way K-shot, meta-training, meta-testing, base class and novel class, support set and query set, etc. Wherein base class and novel class represent the source domain data used in the meta-training phase and the target domain data to be learned in the meta-testing stage, respectively, in which base class and novel class have no intersection[127]. The so-called $N - way\ K - shot$ classification problem means that the samples in the testing set desired to be classified are all in N categories, and only K samples in each category are labelled. What to prove is that using these $N \times K$ samples to realise the classification of unlabeled samples in the testing set[128].

It is known that meta-learning demands learning many tasks to learn a model algorithm with strong generalisation ability. Therefore, meta-training is conducted by constructing multiple classification tasks similar to the testing set to ensure the adaptability of the model. Since meta-learning uses tasks as its own training data, the actual training set and testing set of meta-learning are composed of multiple tasks. In order to distinguish, the training set within each task was renamed to support set $S = \{(X_1, Y_1), ..., (X_n, Y_n)\}$, where $n = N \times K$, and the testing set was renamed to query set $Q = \{x_1, ..., x_q\}$. The formation process of the support set and the query set is: each iteration randomly selects N classes from all the classes in the training set, extracts K samples from all the samples in each class as the support set, and the rest as the query set. The process of meta-training is to use the samples in the support set to train the model. Moreover, the process of meta-testing is to randomly select a few instances from the query set, and let the model query which category it belongs to in the support set, namely the label Y_i, where $i \in (1, n)$, if the classification is correct, it can be proved that the model performance is excellent. Repeating the above meta-learning process can finally get the accuracy of the task model which will be used to further optimise the parameters of the model. When the training tasks are all completed, the trained model will be applied to classify the samples in the testing set to achieve fine-tuning of the model, which will be used to classify the samples with unknown labels in the testing set[129]. As illustrated in Figure 4. 1, they are five-way five-shot tasks, and the number of meta-training tasks is M which is formed by repeating the above meta-learning process M times.

During the meta-learning training process, various problems and challenges will be encountered. To address these critical limitations, we propose a novel meta-learning framework centred on structured relational reasoning and explicit cross-set interaction. Our core contributions are threefold:

Graph classification: in the real world, there are complex interactive relationships between most data, and this relationship can be represented by a graph structure, which is composed of nodes and edges, where the nodes represent the objects of task processing and edges represent the relationship between them. In this research, we regard the support set categories or query set samples cascaded with tags as nodes, and the similarity among categories or between categories and query set samples as edges, and the matrix composed of the weights on the edges is regarded as an adjacency matrix (see Section 4.4 for details). That is to say, our classification task converts the comparison methods at the image level into the comparison methods at the graph (node) level. The purpose is to compose the graph structure through the image, and use the GNN to update the node features and edge features to fully mine the relationship between them, thereby increasing the accuracy rate.

Interactive: most of the existing meta-learning methods are multitasking and independently extract the feature vectors of the training set and the testing set samples, which leads to the low resolution of the extracted features (the extracted feature vectors are easily affected by prior experience and blindly treat the trained targets as the targets of the current classification task), while our model can extract the interactive attention R^c (which represent the relationship between the prototype and the query set sample feature map) and R^x (which represent the relationship between the query set sample feature map and the prototype) between the training set samples and the testing set samples, making the feature vectors more favourable for classification. In short, the "interaction" in this research means that the feature extraction network no longer independently extracts the feature vectors of the support set and the query set samples, instead, the feature vectors related to the current task are generated by extracting the interaction between support

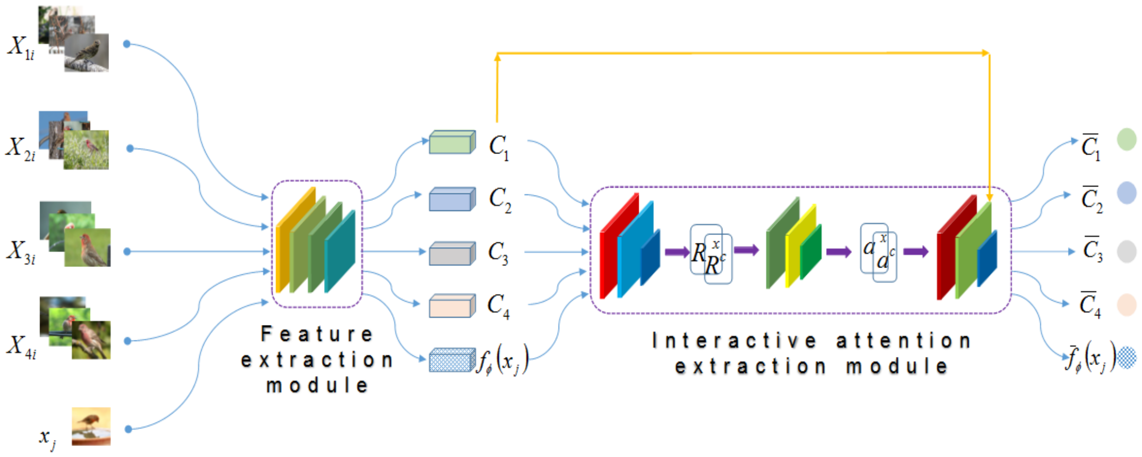


Figure 4.2: **Overview of unbiased feature vectors extraction process in a meta-learning task.** The feature extraction module has been used to acquire class centre feature vectors(C_1, C_2, C_3, C_4) of support set and sample feature vectors($f_\phi(x_j)$) ($j = 1, ..., q$) of query set. Afterwards, interactive attention extraction module is used for obtaining unbiased feature mappings $\bar{C}_i and \bar{f}_\phi(x_j)$.

set samples and query set samples (locking the target domain quickly through interactive attention).

Bias: most meta-learning tasks coexist with multiple tasks, and they have a certain memory ability for prior knowledge. Therefore, if not considered the preference of the trained model for the training target that has been executed, it will lead to the extracted feature vectors not being conducive to the current classification task, and we define this preference as “target bias”. Our proposed interactive attention extraction module can quickly lock the classification target by generating interactive attention, so that the meta-learning model reaches the target unbiased (we still retain the prior knowledge while interfering with the past goal). In short, bias represents the preference of the model for the classification targets (meta-training targets) that have been seen in the process of training the classification model.

4.4 Mathematical Principles of Meta-Learning

From the theory of previous sections, we know that the meta-learning task is composed of multiple N-way K-shot tasks, and this section will expound the execution process of a certain meta-learning task, and the theory of Section 4.3 has explained that the GNN contains a node update module and an edge update module, and in this section, we shall explain that how the distance between the sample and categories be continuously updated to obtain the final similarity through these two modules. We shall first briefly describe how

Algorithm 1: Training strategy of target unbiased meta-learning

```
Inputs: $S = \{(X_1, Y_1), ..., (X_n, Y_n)\}$, $Q = \{x_1, ..., x_q\}$, where $n = N \times K$;
Outputs: labels $y$ of images in query set;
For all $M$ tasks do:
    Compute $C_i, f_\phi(x_j)$ using Eq.1, where $i \in (1, n), j \in (1, q)$;
    Compute $R_p^c, R_q^x$ using Eq.2 and Eq.3, where $p \in (1, m), q \in (1, m)$;
    Compute $a_p^c, a_q^x$ using Eq.4 and Eq.5, where $p \in (1, m), q \in (1, m)$;
    Compute $\bar{C}_i, \bar{f}_\phi(x_j)$ using $a_p^c, a_q^x$;
    Connect $\bar{C}_i, \bar{f}_\phi(x_j)$ with its labels $\rightarrow V = \{v_1, ..., v_z\}$, where $z = N + q$;
    For all layers $= 1, ..., l$ do:
        Compute $A^{(l)}$ using Eq.6;
        Compute $V^{(l+1)}$ using $A^{(l)}, V^{(l)}$;
    End
    Predict the label $y_j$ of $x_j$ based on $A$;
    Compute the cross-entropy loss function $L$ using Eq.7;
    Output the label $y_j$, where $j \in (1, q)$, of $x_j$ based on the minimum value of $L$;
End
```

the input context is mapped into a target unbiased feature vector, and then particularise the related architecture. Also, our proposed model is divided into three modules: feature extraction module, interactive attention extraction module, and graph classification module. The visualisation of the operation process is shown in **Algorithm 1 (Figures 4.2 and 4.3)**.

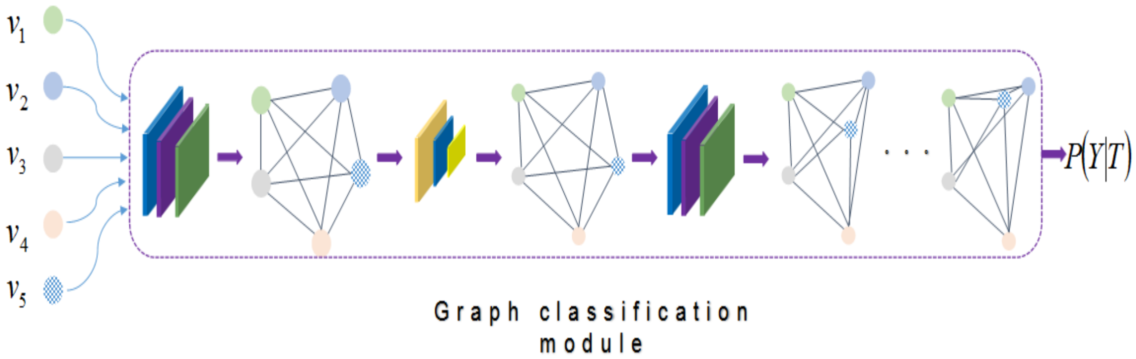


Figure 4.3: **Overview of unbiased feature vectors classification process in a meta-learning task.**The graph classification module has been used to update feature vectors of nodes and adjacency matrix, ultimately, obtaining the similarity scores between nodes.

Feature extraction module: We use the traditional prototype network to extract the original feature vectors. In like manner, the entire support set $S = \{(X_1, Y_1), ..., (X_n, Y_n)\}$, where $n = N \times K$, and query set samples $Q = \{x_1, ..., x_q\}$ are used as the inputs of the embedded function f_ϕ. The class feature mapping (prototype) C_i(Eq.4.4.1), where $i = 1, ..., N$, of the support set S and feature mapping $f_\phi(x_j)$ of query set samples x_j, where $j \in (1, q)$, are outputs:

$$C_i = \frac{1}{\mid K \mid} \sum_{k=1}^{K} f_\phi(X_i^k) \tag{4.4.1}$$

where $i = 1, ..., N$, C_i represents the class centre of the $i-th$ class in support set, and $f_\phi(X_i^k)$ indicates the feature mapping of the $k-th$ sample of the $i-th$ class in support set.

The network structure that realises the above functions consists of four convolutional blocks, and each convolutional block includes a convolutional layer, a batch normalisation layer and a ReLU activation function layer. Its intention is to utilize a simple four-level network structure to map the input information to a high-dimensional feature space, that is $R^D \rightarrow R^M$, where $M = c \times h \times w$, and c, h, w represents the number of channels, the height and width of the feature map, respectively. And the mean value of its high dimensional feature vectors is taken as the prototype C_i. The ReLU activation function is made use of nonlinear mapping. The visualisation is shown in **Figure 4.4**.

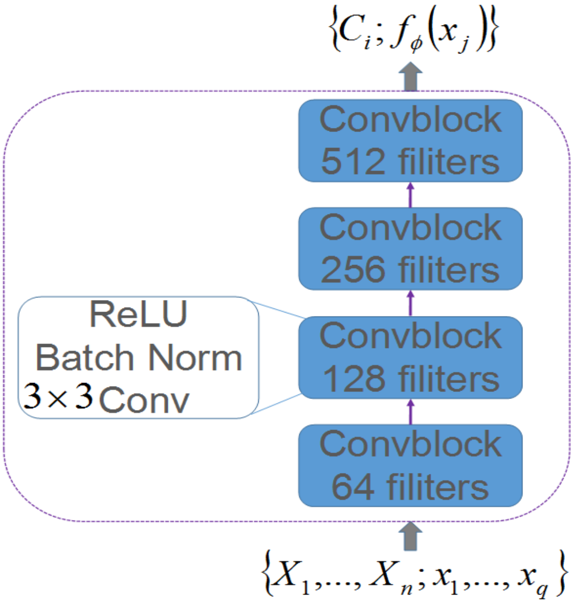


Figure 4.4: **Overview of the network structure** of feature extraction module.

Interactive attention extraction module: In order to further process the original feature mappings and obtain the unbiased target feature vectors, the interactive attention extraction module is introduced. First, the prototypes $C_i \in R^{c \times h \times w}$ of support set and the feature mappings $f_\phi(x_j) \in R^{c \times h \times w}$ of query set samples are mapped to $R^{c \times m}$ where

$m = h \times w$, using a multilayer neural network. At this time, m represents the number of spatial positions on each feature mapping. The feature mappings extracted by the prototype network now become: $C_i \to (c_1, ..., c_m) \in R^{c \times m}, f_\phi(x_j) \to (x'_1, ..., x'_m) \in R^{c \times m}$ (where $c_p|_{p=1}^{m}$ and $x'_q|_{q=1}^{m}$ represents the class feature vectors and the query sample feature vectors of the $p-th$ and $q-th$ spatial position, respectively), and then use Eq.4.4.2 and Eq.4.4.3 to calculate the interaction relationship mapping $R^c \in R^{m \times m}$ and $R^x \in R^{m \times m}$:

$$R_p^c = (\frac{c_p}{\|c_p\|_2})^T(\frac{x'_q}{\|x'_q\|_2}) \in (r_1^c, ..., r_m^c) \tag{4.4.2}$$

where $p \in (1, ..., m), q = 1, ..., m$.

$$R_q^x = (\frac{x'_q}{\|x'_q\|_2})^T(\frac{c_p}{\|c_p\|_2}) \in (r_1^x, ..., r_m^x) \tag{4.4.3}$$

where $p = 1, ..., m, q \in (1, ..., m)$.

Now the interactive attention mappings can be obtained, where $R^c = (r_1^c, ..., r_m^c) \in R^{m \times m}$ represent the relationship between the prototype C_i and the query set sample feature mapping $f_\phi(x_j)$, and the attention mappings $R^x = (r_1^x, ..., r_m^x) \in R^{m \times m}$ represent the relationship between the query set sample feature mapping $f_\phi(x_j)$ and the prototype C_i. Furthermore, from the spatial position, $r_p^c \in R^m$ represents the interaction between the $p-th$ spatial position of class feature mapping c_p and all spatial position of query sample feature mappings $x'_q|_{q=1}^{m}$, similarity, $r_q^x \in R^m$ represents the interaction between the $q-th$ spatial position of query sample feature mapping x'_q and all spatial position of class feature mappings $c_p|_{p=1}^{m}$. At this point, the local interaction between class prototypes and query set sample feature vectors are defined by $R^c \in R^{m \times m}, R^x \in R^{m \times m}$ respectively.

Next, convolution blocks is used to convolve $R^c \in R^{m \times m}, R^x \in R^{m \times m}$ (convolution kernel function $\omega \in R^m$ with $m \times 1$ size can be generated adaptively by a meta-learner, which will be elaborated in Figure 4.5) to change them from vectors to scalars, and a Softmax function limits them keep numbers a_p^c(the interactive attention coefficient, which represents the interaction between the $p-th$ spatial position of class feature mapping c_p and all spatial position of query sample feature mappings $x'_q|_{q=1}^{m}$),a_q^x (the interactive attention coefficient, which represents the interaction between the $q-th$ spatial position of query sample feature mapping x'_q and all spatial position of class feature mappings $c_p|_{p=1}^{m}$) belonging to $(0, 1)$ (shown as Eq.4.4.4 and Eq.4.4.5). The feature vector of the $i-th$(where $i \in (1, N)$) support set class with interactive attention coefficient in the $p-th$(where $p \in (1, m)$) spatial poaition is recorded as $a_p^c C_i$, and the sample feature vector of the $j-th$(where $j \in (1, q)$) query set sample in the $q-th$(where $q \in (1, m)$) with interactive attention coefficient is recorded as $a_q^x f_\phi(x_j)$.

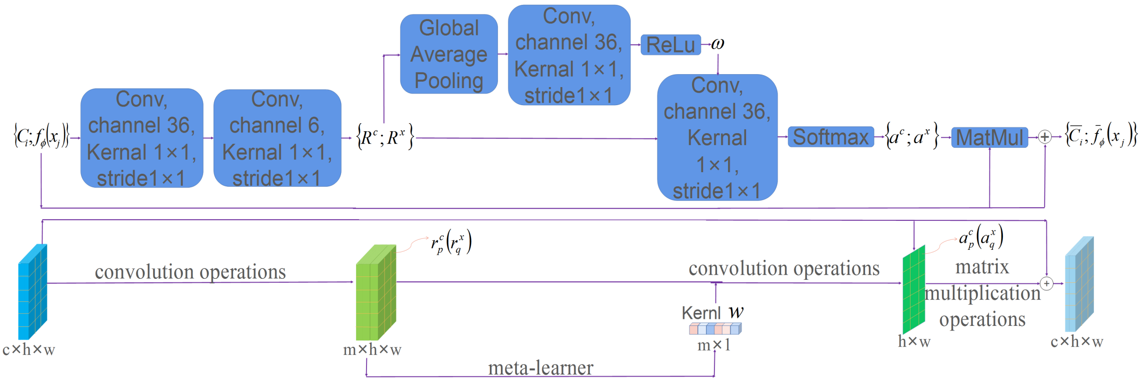


Figure 4.5: **Overview of the network structure** of the interactive attention extraction module.

$$a_p^c \propto exp(\frac{\omega^T r_p^c}{t}) \tag{4.4.4}$$

where $p \in (1, ..., m)$ and t represent a temperature hyperparameter which be set to 0.025 in this research.

$$a_q^x \propto exp(\frac{\omega^T r_q^x}{t}) \tag{4.4.5}$$

where $q \in (1, ..., m)$ and t represent a temperature hyperparameter which be set to 0.025 in this research.

We add the vectors C_i and $f_\phi(x_j)$ which containing historical information and the vectors $a_p^c C_i$ and $a_q^x f_\phi(x_j)$ which containing interactive attention, respectively, to generate the target unbiased support set class feature vectors $(1 + a_p^c)C_i$ and the target unbiased query set sample feature vectors $(1 + a_q^x)f_\phi(x_j)$. In brief, the former of $(1 + a_p^c)C_i$ and $(1+a_q^x)f_\phi(x_j)$ represent the feature vectors containing prior knowledge, and the latter of $(1+a_p^c)C_i$ and $(1 + a_q^x)f_\phi(x_j)$ represent the feature vectors related to the current classification target. We denote class prototype C_i weighted by $(1 + a_p^c)$ and query set sample feature map $f_\phi(x_j)$ weighted by $(1+a_q^x)$ as $\bar{C}_i$ and $\bar{f}_\phi(x_j)$, respectively. Up to now, target unbiased feature mappings $\bar{C}_i$ and $\bar{f}_\phi(x_j)$ are acquired, which are more suitable for classifying unseen targets.

Two convolutional blocks have been used to obtain the interactive attention mappings R^c and R^x (the internal structure of the convolution block is shown in Figure 4.5), and in order to acquire the interactive attention coefficients a_p^c and a_q^x, first a convolutional block has been utilized to map R^c and R^x to a low-dimensional feature space, and then use a softmax function to limit them among (0,1). For the sake of generating convolution kernel adaptively according to the interactive attention mappings R^c and R^x, we first average pool them globally, and map them to the low-dimensional space through the convolution block, and then output the convolution kernel $\omega \in R^m$ by the activation function ReLu. Furthermore, so as to gain the target unbiased feature mappings $\bar{C}_i$ and $\bar{f}_\phi(x_j)$, matrix multiplication operations are used to multiply matrix C_i and matrix a_p^c as well as matrix $f_\phi(x_j)$ and matrix a_q^x, and then add operations are let them combine with original feature vectors C_i and $f_\phi(x_j)$, respectively. The network structure of the interactive attention extraction module is shown in Figure 4.5.

Graph classification module: In order to improve the accuracy of a classification and finally complete the classification of unknown label samples in the query set at the graph level, graph neural networks are used to fully mine the relationship between support set categories and query set samples. The graph neural network is a graph model composed of nodes and edges. We make the target unbiased category prototype $\bar{C}_i$ and the target unbiased query set sample feature vector $\bar{f}_\phi(x_j)$ in the query set concatenate with the one-hot encoding of the label, and regard it as the node of the graph structure and express it as $V = (v_1, ..., v_z)$, where $z = N + q$, and N is the number of classes of support set, and q is the number of samples in the query set. The weight of each edge represents the similarity between the query set sample and the query set category. The process of obtaining adjacency matrix A from target unbiased feature vector v_i and v_j, where $i, j = 1, ..., z$, is shown in Eq.4.4.6, where each element $A_{i,j}$ of the adjacency matrix is processed by the absolute difference between the class center feature vector and query set sample feature vector by multilayer neural network.

$$A_{i,j}^{(l)} = MLP(abs(v_i^{(l)} - v_j^{(l)})) \tag{4.4.6}$$

where l represents the $l-th$ layer network.

The graph classification module is composed of multiple graph networks. A graph neural network module is made up of an adjacency matrix (weight) update module that changes the graph model structure and a node feature update (graph convolution) module that does not change the graph model structure. Where the weight update module consists of 5 fully connected layers, their function is to make similar nodes closer, and in the last graph neural network module, its output represents the similarity between the query set sample and the support set category.

The graph convolution module consists of graph convolution layer, batch normalisation layer, Leaky ReLU activation function layer and a cascade operation layer. Its function is to utilise the adjacency matrix and node feature vectors of the last layer network to update the node feature vectors of this layer so that it contains global information, which is beneficial to the classification operation. Its visualisation diagram is shown in Figure 4.6.

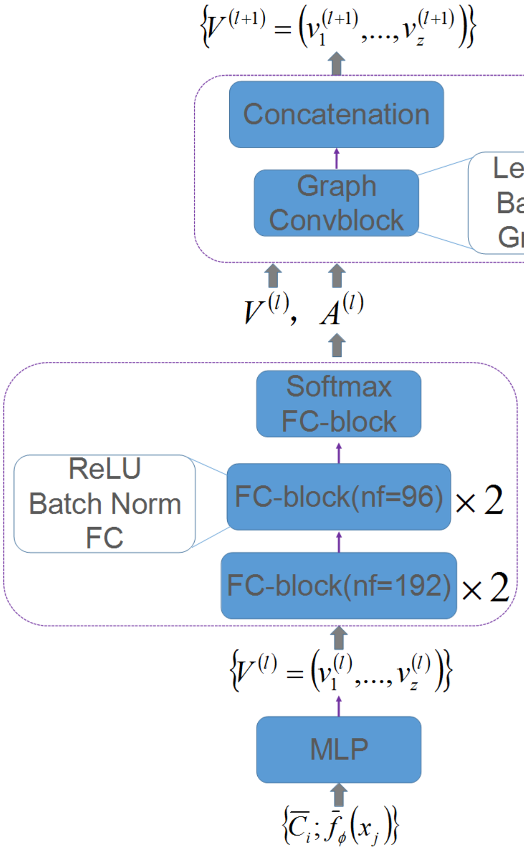


Figure 4.6: **Overview of the network structure** of the graph classification module.

Training method: When performing few-shot classification, the model is optimised by minimizing the cross-entropy loss (Eq.4.4.7). Under semi-supervised learning and transductive inference settings, the training process is similar to the few-shot learning process except that the sample label is represented by its K-simplex uniform distribution.

$$L(\bar{f}_\phi(x_j), Y) = -\sum_k Y_k log P(y_j = Y_k|T) \tag{4.4.7}$$

where y_j is the prediction label of the query set sample x_j.

4.5 Experiments

To validate the efficacy and advancement of our meta-learning method, comprehensive experiments were conducted across three benchmark datasets: Omniglot, MiniImageNet, and TieredImageNet. Evaluation encompassed multiple few-shot learning paradigms, including supervised classification, semi-supervised classification, and transductive inference. Empirical results demonstrate that our approach consistently achieves statistically significant improvements over state-of-the-art baselines across all experimental settings.

4.5.1 Datasets

Omniglot contains 1623 different handwritten characters from 50 different letters. Each character is drawn online by 20 different people. In other words, there are 1623 categories, but only 20 images per category in Omniglot. The dataset has been split into 1200 classes for training and the remaining 423 for testing[126].

MiniImageNet is taken from ImageNet, including 100 categories, each category contains 600 sample data. Among them, 64 category data are used as the training set, 16 category data as verification set and 20 category data as the testing set[119].

TieredImageNet with MiniImagenet, is a subset of ILSVRC-12, yet TieredImageNet represents a larger subset of ILSVRC-12 with 608 classes. As Omniglot groups characters into letters, TieredImageNet classifies categories into broader classes corresponding to higher-level nodes in the ImageNet hierarchy. There are 34 large classes, each containing 10 to 30 categories. These are divided into 20 training sessions, 6 validation sessions and 8 testing sessions[130].

4.5.2 Baselines

To situate our approach within the current research landscape and establish a rigorous comparative framework, we evaluate against several representative meta-learning paradigms:

Matching Net : it is based on cosine distance and proposed by Vinyals O et al., which is innovatively implemented in the form of matching, where few-shot classification task maps the feature information of the sample to a higher-dimensional and more abstract feature space[131].

Proto Net: it is based on squared Euclidean distance and proposed by Snell, J. et al., which uses squared Euclidean distance as the distance measurement method, instead of the commonly used cosine distance, and utilizes the episode mini-batch gradient descent training method to make the samples with more categories can be used during training[115].

Relation Net : it is based on the neural network and proposed by Sung, F. et al., which learned a learnable non-linear similarity measurement method for few-shot or even one-shot learning tasks[112].

GNN[132] and EGNN[133]: in order to prove the advanced nature of our feature extraction module, we compared the GNN based on the non-Euclidean domain proposed by Garcia V et al. and the EGNN proposed by Jongmin Kim et al.; they provided an algorithm that uses GNN to solve few-shot learning, and utilises graph model to calculate the relationship between images.

TPN: it proposes a few-shot learning algorithm with transductive propagation network(TPN), which motivation is that our common supervised learning methods consider separately the training set with a sample label and the test set without sample label, which causes poor generalisation. While transductive learning is to input the labelled training set and the unlabeled test set into the network for training, and then predict the results of this part of the test set[134].

4.5.3 Parameter settings

Data preprocessing: aiming to verify the feasibility of the implementation on three datasets, we set the number of novel categories as five($N = 5$), and the number of training examples per novel category as one($K = 1$) or five($K = 5$), that is to say, there has only one or five labelled samples of each class in the support set of meta-training and meta-testing tasks. At the same time, we set the number of testing examples for per novel category when training as six, and the number of testing examples for per novel category when testing as fifteen.

Implementation details: to verify the accuracy of the implementation on three datasets, 5-way 1-shot and 5-way 5-shot experiments will be conducted. We fix the values of episodes per epoch when training as 5000, 1200 and 13980, for Omniglot, MiniImageNet and TieredImageNet respectively. And we set the number of episodes per epoch when testing as 2000 for the three datasets. A Stochastic Gradient Descent is used to optimize the model, where the initial learning rate is set as 0.1 and weight decay is set as 5×10^{-4}; Meanwhile, the momentum of the optimization algorithm is set as 0.9; Batch size for training and testing is set as 4 and 8 respectively. All experiments in this research are accelerated training on two NVIDIA GTX 1080TiGPUs, using Pytorch programming language to train and test in Ubuntu 16.04 environment and the other settings of models are the same as the implementation of their original papers.

4.5.4 Experimental Results

Table 4.1: Few-shot classification results of 5-way(%)

Datesets			MinilmageNet		TieredlmageNet	
Methods	Taxonomy	Trans.	1-shot	5-shot	1-shot	5-shot
Matching Net	Image-level	N	48.72±0.26	55.33±1.85	63.41±0.78	66.47±0.18
Matching Net	Image-level	Y	49.21±0.32	62.74±0.37	66.85±1.84	69.57±0.59
Proto Net	Image-level	N	55.71±0.78	65.67±0.58	71.37±0.96	70.98±0.63
Proto Net	Image-level	Y	65.77±0.85	66.41±0.70	72.64±0.77	73.30±0.71
Relation Net	Image-level	N	60.29±0.86	63.11±0.66	73.21±0.99	74.03±0.36
Relation Net	Image-level	Y	63.14±0.21	63.85±0.69	74.19±0.28	75.31±0.88
GNN	Node-level	N	64.02±0.52	65.99±0.67	74.82±0.36	75.83±0.42
GNN	Node-level	Y	65.88±0.27	67.07±0.75	75.32±0.70	76.58±0.86
Meta-GNN	Node-level	N	66.41±0.28	66.19±0.76	77.02±0.92	78.15±0.63
Meta-GNN	Node-level	Y	67.97±1.81	69.43±0.74	77.33±0.71	80.23±0.01
TPN	Node-level	Y	64.53±0.93	67.86±1.19	74.85±0.92	78.41±0.21
Our method	Node-level	N	71.94±0.94	73.16±0.61	78.09±0.51	82.93±0.63
Our method	Node-level	Y	**73.72±0.95**	**76.37±0.62**	**78.63±0.63**	**84.23±0.66**

Few-shot learning: With respect to MiniImageNet and TieredImageNet datasets, we conduct 5-way classification tasks. In addition, we set the number of labelled samples per class between 1 and 5 ($K = 1$ and $K = 5$). Table 1 indicates that our target unbiased meta-learning method immensely outperforms the contrastive methods. Furthermore, the preponderance of transductive inference is capable of being predicted by labelled samples, and able to find clusters by using the information of unlabelled test samples, further classifying them more effectively. To achieve the best generalisation ability on these data, we add transductive settings (Y) and compare its results with inductive settings (N). The experimental results show that assuming unlabelled data being equivalent to the data to be tested (transductive) has an advantage over simply training labelled samples (inductive), with accuracy improving by at least 0.54% (78.09% VS 78.63%) in the 5-way 1-shot setting and 1.30% (82.93% VS 84.23%) in the 5-way 5-shot setting. Our method belongs to the category of metric learning. In order to prove the advancement of the classification module, we first compare the matching network based on the cosine distance proposed by Vinyals O et al.. And the accuracy of our method outperforms it under both 5-way 1-shot settings and 5-way 5-shot settings. Secondly, we compare the prototype network based on the squared Euclidean distance proposed by Snell, J. et al.. From Table 4.1 we can observe that our method outperforms it by at least 4.96% (68.20% VS 73.16%) under the 5-way 1-shot and the 5-way 5-shot settings. Then, we compare our method to the neural network-based relational network proposed by Sung, F. et al., and find that our work improves accuracy significantly. Furthermore, in order to prove the advanced nature of our feature extraction module, we compare our method with the GNN based on the non-Euclidean domain proposed by Garcia V et al. and the EGNN proposed by Jongmin Kim et al., and the accuracy of our method improved by at least 7.40% (63.54% VS 71.94%) under the 5-way 1-shot settings, and it is no worse than EGNN, in addition to improving at most 4.08% (80.15% VS 84.23%) under the 5-way 5-shot settings with the help of the interactive attention extraction module. Moreover, in the case of the same setting of transductive inference, since our method introduces the interactive attention extraction module, the accuracy rate is improved by at least 6.94% (69.43% VS 76.37%) under 5-way 1-shot and 5-way 5-shot settings.

Semi-supervised learning: It is introduced to utilise a mass of label-free samples, to obtain valuable information from the data, and to deal with the problem of the poor generalisation ability of supervised learning as well as inaccurate of unsupervised learning. We performed 5-way 5-shot semi-supervised learning experiments on two datasets, namely MiniImageNet and TieredImageNet. As shown in Table 4.2, diverse results are acquired, merely with labelled samples(Supervised, which be defined in the table as S) and with labelled as well as unlabeled samples(Semi-supervised, which be defined in the table as SS), by setting the proportion of labelled data to 20%, 40%, 60%, 100%. Results attest that our method can also achieve good performance with fewer labelled samples, and the accuracy is improved by at least 1.74% (61.88% VS 63.62%), 1.8% (62.52% VS 64.32%) and 4.32% (65.53% VS 69.85%) in the 20%, 40% and 60% settings in MiniImagenet. Simultaneously, at least 4.04% (65.76% VS 69.80%), 2.61% (69.32% VS 71.93%) and 2.98% (69.36% VS 72.34%) in the 20%, 40% and 60% settings in TieredImageNet. In addition, with the increase of labelled samples, the classification accuracy of various methods remains gradually improved.

Table 4.2: Semi-supervised results of 5-way 5-shot setting(%)

Datesets		MinilmageNet			
Methods	Class	20%	40%	60%	100%
GNN	S	50.33±0.81	56.91±0.64	61.02±0.28	66.41±0.53
GNN	SS	52.45±0.45	58.76±0.52	64.92±0.57	68.30±0.54
Meta-GNN	S	52.86±0.14	59.34±0.29	65.14±0.88	66.85±0.85
Meta-GNN	SS	61.88±0.25	62.52±0.18	65.53±0.01	68.93±0.74
TPN	SS	57.62±0.68	60.32±0.84	64.36±1.12	68.29±0.62
Our method	S	59.18±0.21	62.98±1.01	**69.85±0.46**	74.37±0.25
Our method	SS	**63.62±0.71**	**64.32±0.24**	68.47±0.59	**76.54±0.89**
		TieredlmageNet			
GNN	S	51.36±0.25	55.52±0.85	66.29±0.26	71.83±0.99
GNN	SS	54.45±0.81	56.68±0.26	64.18±0.36	73.58±0.46
Meta-GNN	S	59.91±0.42	68.31±0.68	69.36±0.57	80.95±0.23
Meta-GNN	SS	65.76±0.62	69.32±0.49	70.80±0.94	82.29±0.74
TPN	SS	61.52±0.87	64.32±0.53	69.52±0.57	77.45±0.62
Our method	S	66.41±0.14	71.12±0.72	72.34±0.64	82.53±0.41
Our method	SS	**69.80±0.31**	**71.93±0.56**	**74.56±0.33**	**84.36±0.54**

4.5.5 Ablation Experiments

The primary objective of ablation experiments is causal attribution analysis. In this study, we conduct multi-dimensional ablation studies to rigorously isolate the contributions of our proposed components. The analysis of the interactive attention extraction module is essential, as this core innovation directly addresses the critical challenge of ”target bias” in meta-learning. Evaluating the impact of intra-class sample quantity assesses the model's robustness under extreme data scarcity inherent to few-shot learning paradigms. Investigating the effect of inter-class task volume tests the scalability of meta-knowledge transfer across diverse task distributions. Finally, varying the network architecture decouples the contribution of the backbone network from the novel module, ensuring performance gains are correctly attributed to our methodological innovation rather than architectural advantages.

Impact of the interactive attention extraction module: to demonstrate the effectiveness of the interactive attention extraction module, we visualise the feature mappings of our target unbiased meta-learning method(TUML), squared Euclidean distance-based prototype network(Proto), graph models-based graph neural network(GNN), and TPN using transductive inference instead of inductive inference when training graph neural networks, and the results are shown in Figure 4.7. What remains obvious is that when the targets

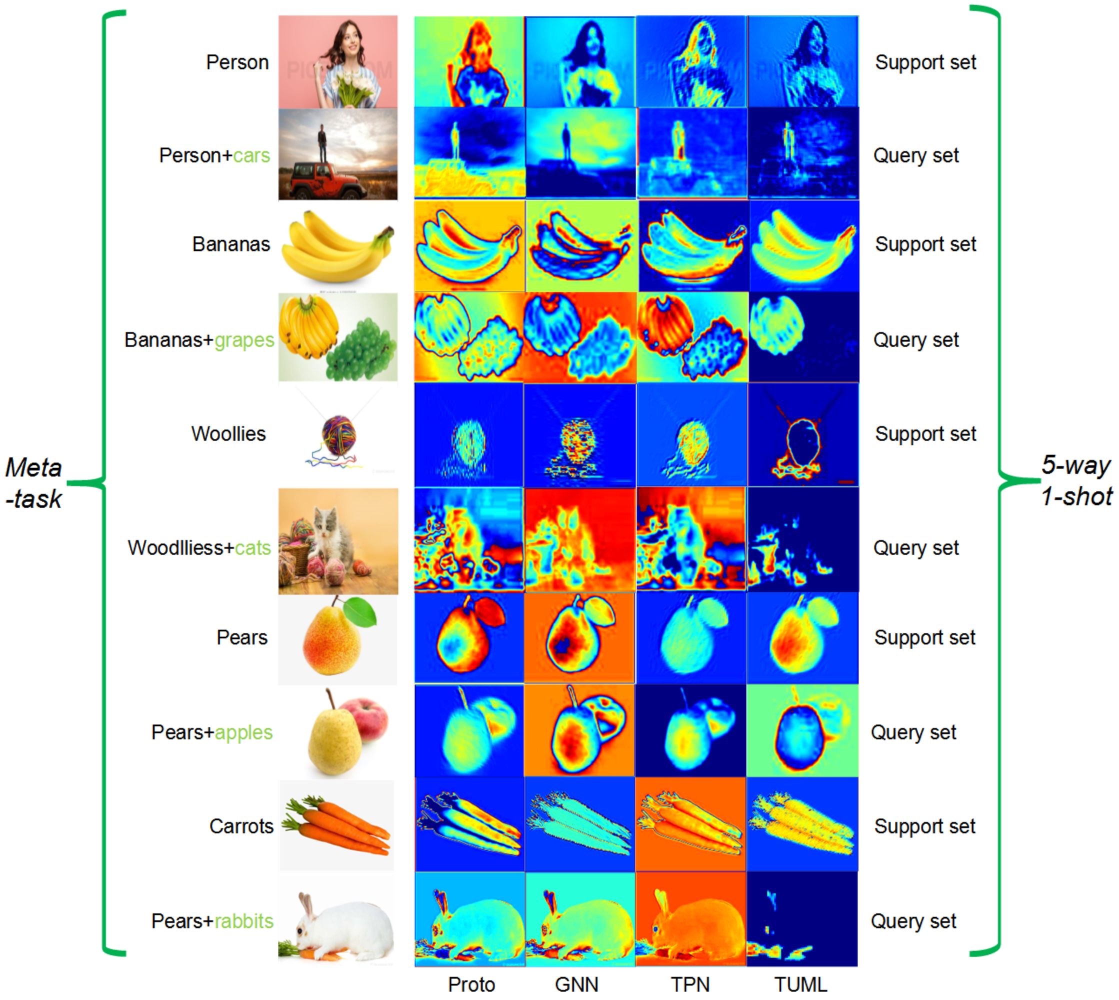


Figure 4.7: **The feature mapping visualization** of different methods

in the sample are chaotic, for example, the current classification targets woollies are mixed with the target cat already seen during training, the baseline methods cannot accurately identify the target domain, and our method can roughly locate the location of woollies. When the target distribution in the sample is relatively clear, the traditional meta-learning methods can identify the target domain, but they are susceptible to interference from the targets that have been seen, and cannot accurately locate the current classification target. For example, when there are existing current classification target bananas as well as target grapes that have been seen in the training process, due to the different classification targets of tasks are not considered, the feature extraction models of baselines are easy to be interfered with by grapes, and our method can accurately locate the target domain by introducing the feature extraction module, which is capable to eliminate the grapes interference and achieve target unbiased.

Impact of the number of samples per class(K): we quantify how influential it is

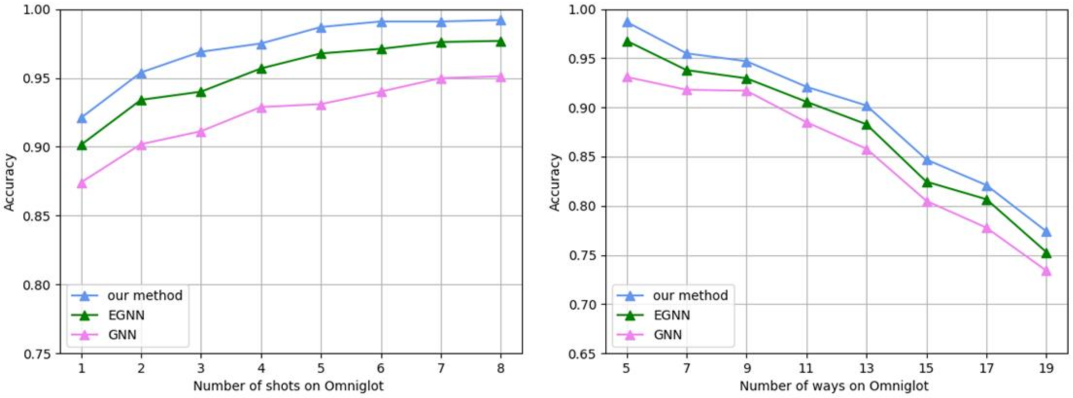


Figure 4.8: Plots of testing accuracy for different **number of ways**

to add the number of labelled samples per class on the Omniglot. We make K a number between (0, 10), and fix N to be 5. As expected, this change is often beneficial, both for our method and for baselines. From Figure 4.8, it can be seen that the difficulty degrades as the shot increases, and more examples per class indeed make it easier to correctly classify that class.

Impact of the number of classes per task(N): to examine the effect of N on test accuracy, we alter the value of N on the Omniglot by taking the integer between $(0, 20)$, and fix K to be 5. What proves evident is that our method is always superior to baseline methods. As illustrated in Figure 4.8, with the increase of N, the prediction accuracy of various methods has decreased, which is due to the incremental number of categories, increasing the difficulty of classification, and thus the accuracy of classification decreases.

Impact of the network structure: For further demonstration of the superiority of our approach, we performed ablation experiments of the network structure on the dataset TieredImageNet, the result as shown in Figure 4.9. where P, I and G represent the prototype network, the interactive attention extraction module, the graph classification module, respectively. First of all, we use ResNet12 to extract features and then classify them by square Euclidean distance. The accuracy of the prototype network is compared with the method of adding an interactive attention extraction module, the results certify that the addition of interactive attention extraction module improves the performance of the model by 4.31% (66.20% VS 70.51%) and 5.27% (69.32% VS 74.59%) corresponding to 1-shot and 5-shot settings. Last but not least, employing graph neural network as a classification module, the accuracy is able to be improved from 70.51% and 74.59% to 79.61% and 81.33%, under 1-shot and 5-shot settings, respectively.

Complexity analysis: we measure the number of parameters and time complexities for several comparisons on MiniImageNet, and the results are shown in Table 4.3. We

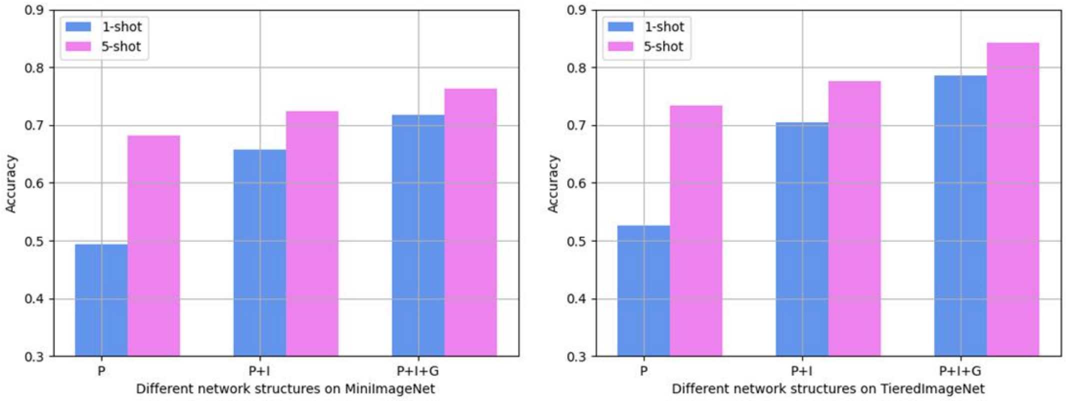


Figure 4.9: Histogram of testing accuracy for different **network structures**

evaluate the average time for a task with 15 query samples per class under 5-way 1-shot and 5-way 5-shot settings on the environment of the NVIDIA GTX 1080Ti GPU. Table 4.3 indicates that there are almost no additional parameters and the time consumption of our method is quite comparable. Specifically speaking, the parameters of the target unbiased feature extraction module are mainly on meta-learner, which aims to train kernel function w and the average time consumption of this module is T(h^2w^2c), where h, w, c represents the height of the feature map, the width of the feature map and the number of channels, respectively. From the above discussions, we have known that time consumption is related to the size of the feature map, so we inserted the attention extraction module after the last convolutional layer. Our method shows comparable time consumption against baselines, where the computational overhead is tiny under both inductive and transductive settings.

Table 4.3: Comparisons of the number of parameters and time complexity

			Time(sec.)	
Methods	Trans.	Parameters	1-shot	5-shot
Proto Net	N	8232.96K	0.96	1.25
Proto Net	Y	8232.96K	0.97	1.26
Our method	N	8232.96K	1.72	2.02
Our method	Y	8232.96K	1.93	2.25

4.6 Conclusion

The interactive attention extraction module based on the traditional feature extraction method has been developed in this research, which reduces the effect of the bias of the trained model to the target, therefore solving the problem of few-shot classification through meta-learning. Then the graph neural network has been utilised as a classification module to fully excavate the relationship between feature vectors and complete the classification at the node level while improving the accuracy of classification. Experimental results show that the proposed method has a more outstanding performance than most traditional methods of few-shot classification. In future work, the proposed model will be ulteriorly extended to the tasks of zero-shot node classification and few-shot graph-level classification problems, as well as the field of text and video classification tasks.

Chapter 5

BAS Optimised ELM for KUKA iiwa Robot Learning

This chapter is based on our published paper "BAS Optimised ELM for KUKA iiwa Robot Learning".

5.1 Introduction

The field of brief with respect to machine learning attracted great attention during the past decade. Machine learning has been used in many fields, such as autopilot filtering system of email, the recommendation system of e-shopping, handwriting recognition in the post office and the automatic driving system of the vehicle. The brief value of machine learning has attracted a growing number of institutions, companies as well as researchers devoting themselves into the field. In 2010, Google announced its driverless car project and released a video featuring Steve Mahan, an employee who has lost 95% of his vision, safely navigating a 12-mile route with the car's assistance[135]. An essential method applied in numerous domains is the Extreme Learning Machine (ELM), which is derived from single-hidden-layer feedforward neural networks and contributes to advancements in autonomous driving technology[136]. Hence, a growing number of researchers focus on Extreme Learning Machines in-depth and extensive brief. Ding et al. proposed the optimization Extreme Learning Machine (OM-ELM) method of the SMO, however, owing to the large data set optimal parameters C, the algorithm requires more iterations to converge to the optimal solution of the optimisation problem[137]. Huang and Chen, describe the enhanced incremental Extreme Learning Machine (EI-ELM) and convex incremental Extreme Learning Machine (Cl- ELM)[138]. Rong et al. describe learning machine algorithms based on fuzzy limits[139]. Li et al. proposed a comprehensive complex Extreme Learning Machine (FC-ELM)[140]. A brief team also describes a wind power pre-side model of wavelet decomposition and an Extreme Learning Machine to verify that this method has higher prediction accuracy[141]. A scientist added a class classification of neurons to form a new network structure in the fermentation process of Nosiheptide, in order to realise a better line estimation [142]. However, using the traditional ELM model alone to make predictions often has limitations in terms of prediction accuracy.

In recent years, intelligent algorithms such as neural networks, genetic algorithms (GA) and particle swarm optimisation were applied to solve the aforementioned optimisation problems. neural networks have the capabilities of adaptive learning, which can deal with parallel distributed processing. GA is often combined with polynomial interpolation for robot motion planning tasks, with a low risk of falling into local optimal solutions[143]. However, GA is with complex proceedings which need to go through the encoding, decoding, selection, crossover and mutation processes. This causes its operation time to be difficult to guarantee at all times. The particle swarm optimisation algorithm has a simple structure and its parameters are easy to adjust. Compared to GAs, particle swarm optimisation does not need to be performed for convex problems and the optimal solution can be found with faster convergence[144]. However, the particle swarm optimisation needs to randomly generate plenty of particles when the next populations are generated, which decreases the convergence rate. To break through the above-mentioned limits, BAS was selected to perform the optimization of ELM in this brief, which is a new type of computational intelligence algorithm developed in[145]. The main contribution of this research is investigating an ELM model to output angle values while inputting the endpoint position in Cartesian space for robot control, which is much faster while learning than traditional artificial neural network models and can be used to replace the complicated kinematics calculation; developing a strategy to create home-made dataset of joint values-endpoint positions; employing BAS algorithm to optimize the initial parameters of ELM to maximum the learning accuracy.

5.2 Dataset Acquisition

In this brief, the endpoint position was directly collected using Kinect SDK 2.0. Kinect v2 produced by Microsoft, is an RGB-D device, which can capture depth, colour, and IR images (also sound). Using the SDK, captured colour and depth information can be consolidated (transformed) into real-world coordinates, called Camera Space; these coordinates are referenced to the centre of the depth sensor[146]. Skeletal tracking can also be achieved with the use of the Kinect 2.0 SDK. Both the colour image, and depth image are transformed into a common frame, the origin of which is located at the centre of the depth camera, called Camera space; the coordinate system of this space follows a right-handed convention.

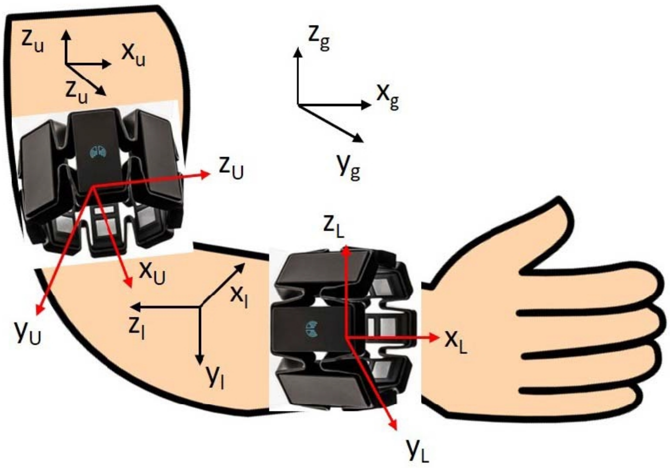


Figure 5.1: Schematic diagram of global coordinate system, local coordinate system and **MYO Armband coordinate system.**

By calculating the joint angles of the human arm joints to capture the operator's arm posture, the control of the remote mechanical arm is realized. The human arm includes 7 degrees of freedom (DoFs), among which, the shoulder joint has 3 DoFs, the elbow joint has 2 DoFs, and the wrist joint has 2 DoFs. It is assumed that the shoulder and elbow joints of the human arm are 5 DoFs chain manipulators. The modelling method of the upper limb using the standard DH parameter can be found from our previous brief[146]. In order to calculate the joint angles of the operator's elbow and shoulder joints (shown in Figure 5.1), the operator will wear a pair of MYO Armbands on the upper arm and lower arm, and define the global coordinate system (x_g, y_g, z_g), where: Z axis is up, Y-axis is forward, X is perpendicular to Z-axis and Y-axis, pointing to the human body.

On the upper and lower arms of the human body, separate coordinate systems are established to describe the spatial posture of the arms. The coordinate system (x_u, y_u, z_u) is established on the upper arm coordinate system, and (x_l, y_l, z_l) is the lower arm coordinate system. We assume that initially, the upper frame and the lower arm frame coincide with the global frame. Define the initial posture of the upper arm and lower arm as[147]:

$$R^i_{gl} = [x^i_{gl}\, y^i_{gl}\, z^i_{gl}], \quad R^i_{gu} = [x^i_{gu}\, y^i_{gu}\, z^i_{gu}] \tag{5.2.1}$$

where R represents the rotation matrix, the superscript i represents the initial position, the subscript g represents the global coordinate system, and u and l represent the coordinate system of the upper arm and the lower arm, respectively. The rotation matrix

$R_{AB} = [x_{AB}, y_{AB}, z_{AB}]$ represents the spatial posture of the coordinate system A in the coordinate system B. The vectors x_{AB}, y_{AB} and z_{AB} represent the position description of the coordinate axis of the coordinate system B in the coordinate system A. The posture of the operator's upper arm in the coordinate system of the first MYO Armband (worn on the upper arm) and the lower arm in the coordinate system of the second MYO Armband (worn on the lower arm) can be expressed as:

$$R^i_{Uu} = (R^i_{gU})^T R^i_{gu}, \quad R^i_{Ll} = (R^i_{gL})^T R^i_{gl} \tag{5.2.2}$$

where the subscript U represents the coordinate system of the MYO Armband worn on the upper arm, and the subscript L represents the coordinate system of the MYO Armband worn on the lower arm. When the operator moves the arm to a new position, the posture of the operator's upper arm and lower arm relative to the global coordinate system can be expressed as[147]:

$$R^n_{gu} = (R^n_{gU})^T R^i_{Uu}, \quad R^n_{gl} = (R^n_{gL})^T R^i_{Ll} \tag{5.2.3}$$

where the superscript n represents the new posture reached by the operator's arm.

If we first rotate a coordinate system around the X-axis of a second known coincident coordinate system by an angle of γ, then rotate around the Y-axis by an angle of β, and finally rotate the coordinate system by an angle of α around the Z-axis, the rotation angles γ, β, and α are defined as the roll, pitch, and yaw angles, respectively. Therefore, the rotation matrix can be written as(c for cos, s for sin):

$$R^n_{gu} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \tag{5.2.4}$$

Using the MYO's gyroscope, we can obtain a quaternion ($q = xi + yj + zk + w$, where the (x, y, z) is a vector and w is a scalar quantity) at the configuration with respect to the corresponding posture, which can be used to calculate $R_{n_{gu}}$. From the rotation matrix of the upper arm MYO Armband coordinate system, the three joint angles of the shoulder joint can be calculated as:

$$\alpha = \operatorname{atan2}(2(yz - wx)/s\beta, 2(wy + xz)/s\beta) \tag{5.2.5}$$

$$\beta = \operatorname{atan2}\left(\sqrt{4(xz - wy)^2 + 4(wx + yz)^2}, 1 - 2(x^2 + y^2)\right) \tag{5.2.6}$$

$$\gamma = \operatorname{atan2}(2(xz - wy)/s\beta, -2(wx + yz)/s\beta) \tag{5.2.7}$$

Let us define r_{ij} as one element of $R_{n_{gu}}$, and a_{ij} as one element of $R_{n_{gl}}$, which can be obtained by the MYO's gyroscope. The two joint angles of the elbow joint can be calculated through the MYO Armband worn on the lower arm:

$$\delta = \arccos(a_{12}r_{13} + a_{22}r_{23} + a_{32}r_{33}) \tag{5.2.8}$$

$$\epsilon = \arccos(a_{11}r_{11} + a_{21}r_{21} + a_{31}r_{31}) \tag{5.2.9}$$

5.3 ELM Learning Model Design Methodology

In 2006, Professor Huang at Nanyang Technological University proposed limits of learning feedforward neural networks and introduced the basic principles of ELM[137]. Compared with BP Algorithm and SVM, ELM has several salient features: no parameters need to be manually tuned; most training can be completed in milliseconds; it could obtain better generalisation performance than BP in most cases[148].

ELM is a neural network with only one hidden layer, which can be obtained through offline training, wherein the weights inputted from the input layer to the hidden layer and the bias system can be randomly initialized, and the value between the hidden layer and output layer is required in order to adjust the output of the system to minimize the target error. If the endpoint position data obtained from the Kinect sensor is inputted as x, then the L number of hidden nodes with output functions can be expressed as:

$$f_L(x) = \sum_{i=1}^{L} \beta_i g_i(x) = \sum_{i=1}^{L} \beta_i G(x, a_i, b_i) \tag{5.3.1}$$

where $a_i \in \mathbb{C}^d$, $x_i \in \mathbb{C}^d$, $b_i \in \mathbb{C}$, $\beta_i \in \mathbb{C}$, and $g_i(x)$ is the activation function, β_i is the output weights, a_i is the input weights, and b_i is the input bias.

Given N training samples, the output with L hidden nodes for the learning machine is:

$$f(x) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x) = \beta \cdot h(x) \tag{5.3.2}$$

where $h(x)$ is the output vector of the hidden layer, and the hidden layer node parameters (a_i, b_i) $(i = 1, 2, \ldots, L)$ are initialized by the Beetle Antennae Search (BAS) algorithm, which is able to connect the i-th hidden node with the output nodes via learning weights β_i.

By analysing, the matrix representation of the linear system is as follows:

$$H \cdot \beta = T \tag{5.3.3}$$

where H is the hidden layer output matrix, and T represents the target output.

Then we have:

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}^T = \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix} \tag{5.3.4}$$

According to the input x_i, the hidden layer output matrix H, where the i-th row represents the hidden layer output vector. Based on the values of all the inputs $(x_1, \ldots, x_N)$, the output values of the i-th column represent the i-th hidden node. Hence, we can obtain the least squares solution of the linear systems:

$$|H \cdot \hat{\beta} - T| = \min_{\beta} |H \cdot \beta - T| \tag{5.3.5}$$

where

$$H(a_1, \ldots, a_L; b_1, \ldots, b_L; x_1, \ldots, x_N) = \begin{bmatrix} G(a_1, b_1, x_1) & \ldots & G(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_N) & \ldots & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L} \tag{5.3.6}$$

is the output of the hidden layer nodes. Then, we can calculate the output weights as:

$$\hat{\beta} = H' \cdot T \tag{5.3.7}$$

where H' is the generalized pseudo-inverse Jacobian matrix of H, and the least squares solution of minimum norm β is unique, designed to minimize the training error.

Considering that ELM is based on supervised learning, in this brief, the endpoint positions collected from Kinect are used as the input of the ELM, while the angular joint values measured from the MYO Armband are used as output to train the ELM model. Once the training has been completed with a satisfactory accuracy rate, the model will be able to directly load and estimate a series of joint angles by giving the endpoint position data.

5.4 BAS Optimisation of the Initial Learning Weights and Bias

In this brief, BAS algorithm is employed to optimise the ELM model by finding a series of the most optimised parameter sets of learning weights and bias. BAS is an intelligent optimisation algorithm developed in 2017, inspired by the principle of beetle foraging[149]. Since the process of network learning and training is to adjust the learning weights and bias of each layer, the essence of network training is to continuously adjust the weights and bias.

The initial weights and bias are randomly generated, which have a great impact on network performance. Therefore, the optimisation of ELM mainly focuses on using optimisation algorithms to adjust the weights and bias. Using the optimised weight threshold for network training can greatly improve the fit of the network and reduce the error. The specific method is as follows:

1. Establish a random vector of the heading of the beetle and normalise it:

   $$\vec{b} = \frac{\text{rands}(k, 1)}{\|\text{rands}(k, 1)\|} \tag{5.4.1}$$

   where rands() is a random function, and k represents the spatial dimension. Assuming that the model structure is $N - L - l$, N is the number of nodes in the input layer, L is the number of nodes in the hidden layer, and l is the number of nodes in the output layer, the dimensional formula of the search space of the beetle antennae search algorithm is: $k = N \times L + L \times l + L + l$. In this brief, k is 255.

2. Create the spatial coordinates of the left and right whiskers of the beetle:

   $$x_{rt} = x_t + d_0 \times \frac{\vec{b}}{2}, \quad x_{lt} = x_t - d_0 \times \frac{\vec{b}}{2} \tag{5.4.2}$$

   where x_{rt} is the position coordinate of the beetle at the t-th iteration of the right whisker, x_{lt} is the position coordinate of the beetle at the t-th iteration of the left whisker, x_t is the centroid coordinate of the beetle at the t-th iteration, and d_0 is the distance between the two whiskers.

3. Judging the odour intensity of the left and right whiskers according to the fitness function, that is, the intensity of $f(x_{lt})$ and $f(x_{rt})$, where f is the fitness function. This brief uses the mean square error (MSE) of the test data as the fitness function.

4. Iteratively update the location of the beetle:

   $$x_{t+1} = x_t - \delta_t \times \vec{b} \times \text{sign}(f(x_{lt}) - f(x_{rt})) \tag{5.4.3}$$

   where δ_t is the step factor of the t-th iteration, and sign() is the sign function. The step factor in BAS is set to decrease linearly over iterations. First, determine the initial step size $\delta_0 = 1$, the number of iterations $n = 100$, and the attenuation coefficient η is a number close to 1, between [0, 1].

   The formula is:

   $$\delta_t = \delta_{t-1} \times \eta, \quad \eta = 1 - \frac{5}{n} \tag{5.4.4}$$

5. Initialisation of the beetle's position. Select a random number between $[-1, 1]$ as the initial position of the beetle, and save this initial position as Variable1. Solve the fitness function value corresponding to the left and right sides of the beetle in the initial position, and save it in Variable2.

6. After calculating the positions of the left and right whiskers according to (2), the fitness function values corresponding to the left and right whiskers are calculated respectively. After comparing the values, update the positions of the beetles according to (4), i.e., adjust the weight and bias, and obtain the fitness function value at the new position after training ELM. If the function value at this time is smaller, update Variable1 and Variable2.
7. When the MSE function value is lower than 0.1 and no longer decreases, or when the iteration reaches the maximum number of times (100 iterations), the iteration stops, and the optimal solution is output. If the condition is not met, return to step 6. The whole optimisation process is shown in Figure 5.2.

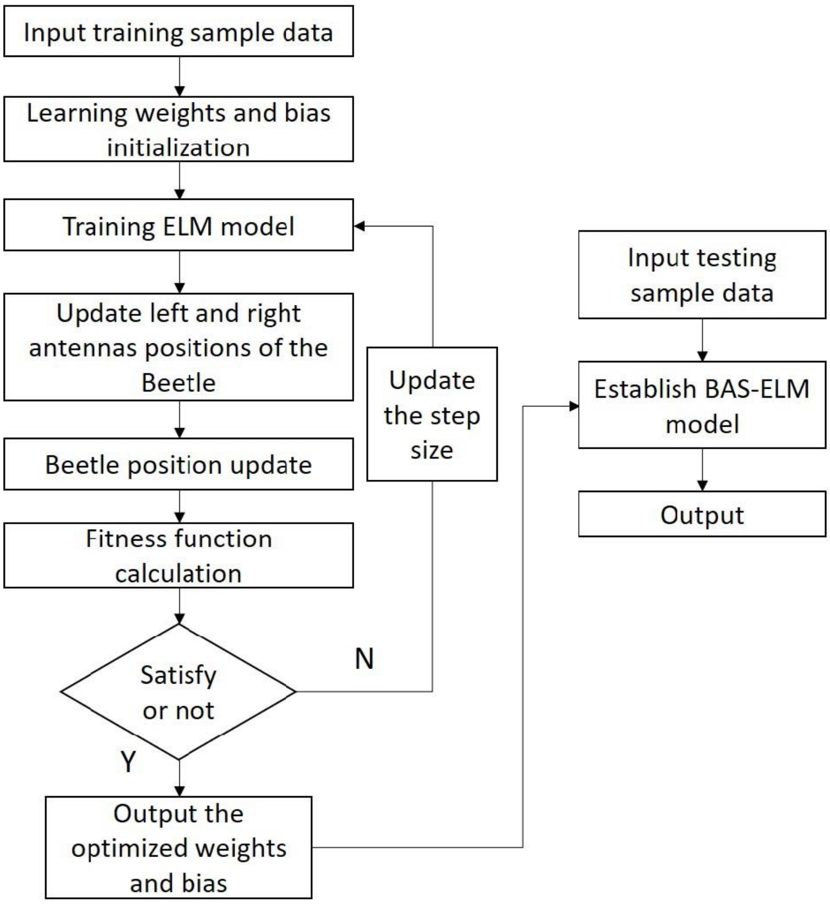


Figure 5.2: Flowchart of angle joints data generation

5.5 Experimental Studies

In this section, we conducted a three-step experimental brief to test the effectiveness of our proposed methodologies. Kinect SDK 2.0 for Windows, Visual Studio 2019, and the OpenCV library were used for interaction design. MYO Armband with its SDK was used for the calculation of arm joints. The experiment environment was an indoor environment with adequate illumination. Five joints were selected for the experimental studies: ShoulderPitch, ShoulderRoll, ShoulderYaw, ElbowPitch, and ElbowRoll.

5.5.1 Dataset Generation

The first step of the experiment was generating our own dataset samples, which include the endpoint positions of the upper limb and the aforementioned five joint angles. To do this, a human operator wore a pair of MYO armbands on his arm and stood in front of a table, where a Kinect v2 sensor was placed. The operator smoothly and slowly drew circle patterns with different radii and positions, and sine curves with different configurations on the table, respectively. It was significant to synchronise all the devices by setting the "scissor" posture as the starting signal. When MYO and Kinect detected this posture, the system began to collect data. The collection frequency of Kinect and MYO was set to 30 Fps. Each time the operator drew a graph of different sizes, it lasted for 10 seconds, corresponding to 300 sets of data, a total of 10 circles of different sizes and positions, and 10 sine curves of different lengths and small heights, totalling 6000 sets of data. The dataset was diversified to ensure generalisation for neural networks.

After the dataset generation was completed, a second dataset was required, where only the endpoint positions were collected. The collection frequency of the Kinect sensor was set to 15 Fps, and the operator slowly and smoothly drew a circle lasting for 10 seconds. The output data from this action would be used as the test sample dataset for the ELM model.

5.5.2 Parameter Optimisation by BAS

The second step was optimising the learning weights and bias of the ELM using the BAS algorithm. The dataset generated in the previous step was used to train the ELM model, where the endpoint positions were input samples, and the joint angles were output samples. In this brief, the number of hidden layer nodes was set to 60. The initial step size of BAS was set as 30, with 100 iterations, and the attenuation coefficient η was set as 0.8. The second dataset collected from step 1 was used to compare with the estimated joint values. The Mean Squared Error(MSE) was employed as the performance index to illustrate the efficiency of the ELM model, with and without BAS optimisation, respectively. The overall principle flowchart of joint angle data generation using BAS-ELM is shown in Figure 5.2.

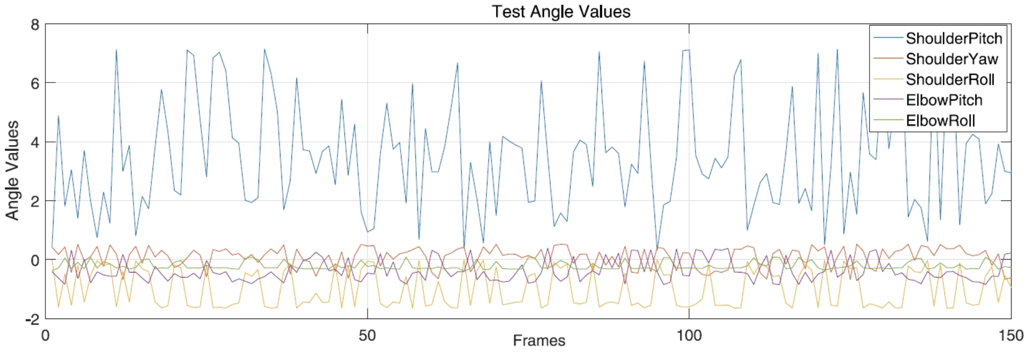


Figure 5.3: Joint angular values of the test sample.

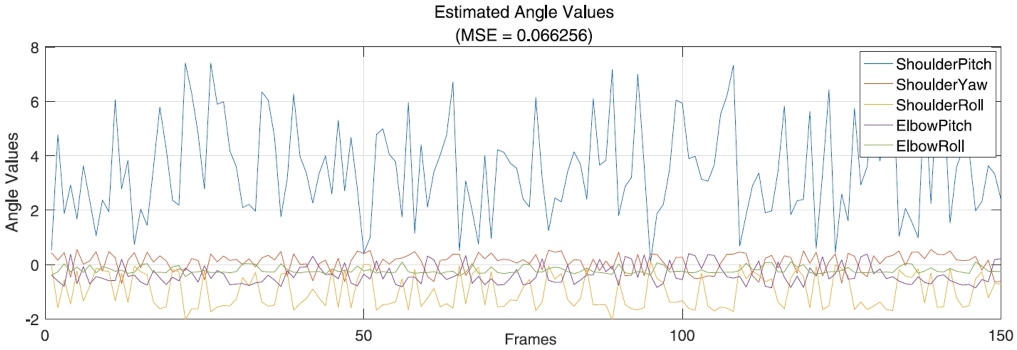


Figure 5.4: Joint angular values of the predicted sample outputted by BAS-ELM.

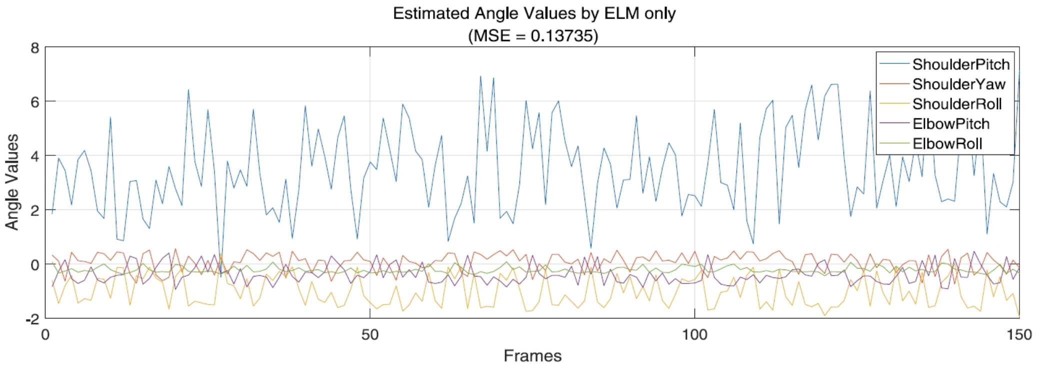


Figure 5.5: Joint angular values of the predicted sample outputted by ELM only.

The fitting result of the BAS-ELM network prediction model is shown in Fig. 5.3, 5.4

and 5.5. Comparing the unoptimized network fitting effect in Fig. 5.3 and 5.5, it can be seen that the curve of the simulated values predicted by the BAS-optimised model is closer to the true value of the predicted values, as shown in the comparison of Figs. 5.3 and 5.4. The MSE function was used to measure the performance of the ELM model. From the relative MSE results after optimisation, the average relative error of BAS-ELM was 0.060256, while the average relative error of the ELM network before optimisation was 0.13735. The average relative error of prediction was reduced by 56.1%, indicating that after adding the BAS optimisation, the ELM model can handle multi-dimensional variable problems more accurately. Fig. 5.6 shows the best fitness curve of the BAS-ELM model, that is, the curve of the best fitness function value for each update. From the figure, it can be seen that the fitness function value reaches 0.1 after 60 generations and does not decrease anymore, indicating that the optimal solution was found and that the convergence speed was relatively fast.

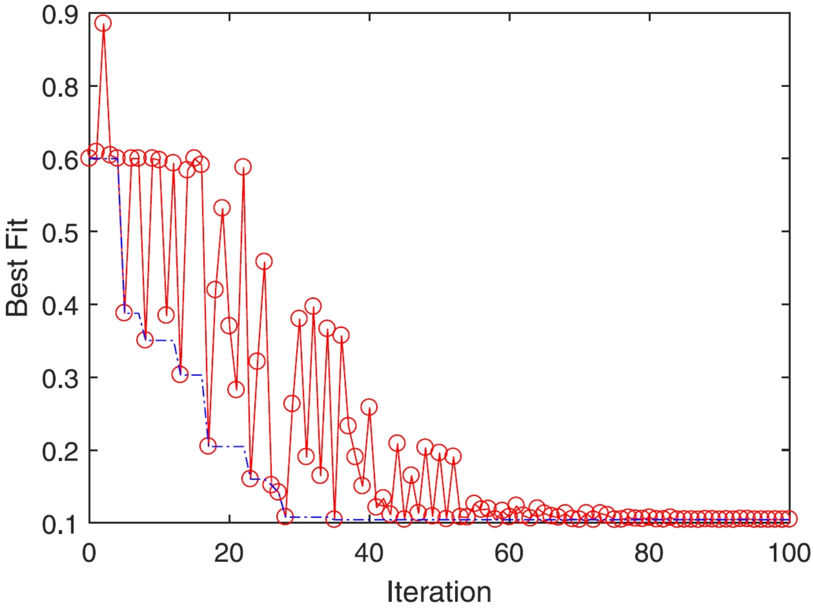


Figure 5.6: Fitting result of the BAS-ELM network prediction model.

5.5.3 Estimated Trajectory Regeneration

The third step was regenerating the trajectory outputted from ELM by a KUKA iiwa robot manipulator. All the five joint angles outputted by BAS-ELM with respect to ShoulderPitch, ShoulderRoll, ShoulderYaw, ElbowPitch, and ElbowRoll were sent to the robot controller for motion execution. The angles for WristPitch and WristYaw were set as the robot's initial constants.

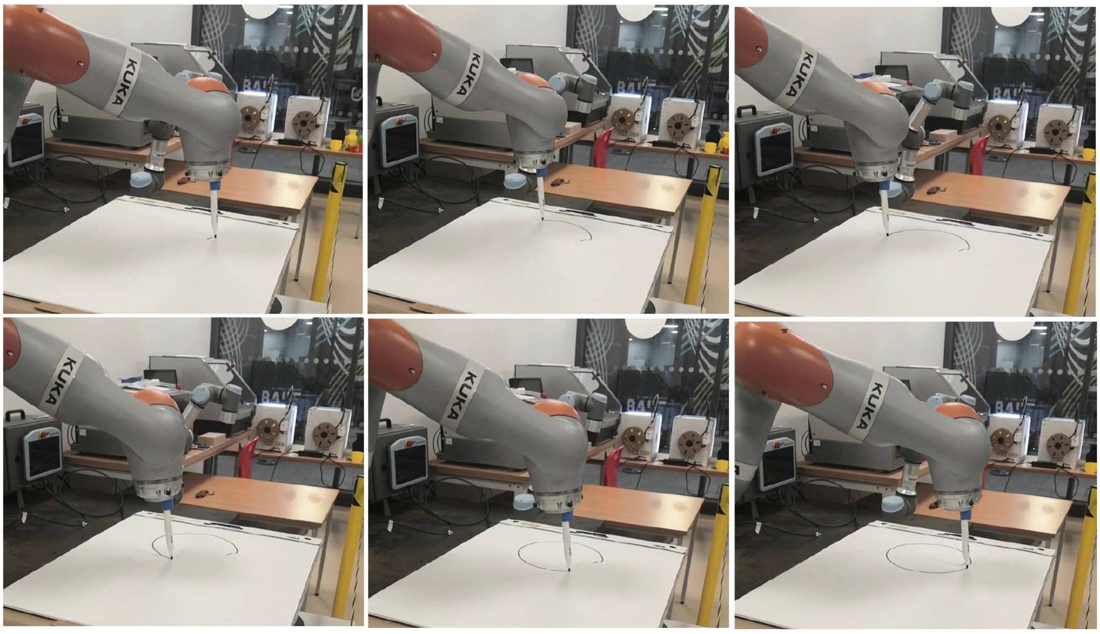


Figure 5.7: Snapshots of the KUKA iiwa robot.

Given the performance comparison results, as illustrated in the previous step (56.1% in error reduction), BAS has shown sufficient superiority over the non-optimized version in predicting accuracy. Thus, the BAS-ELM strategy was directly used for experimental validation in the KUKA iiwa robot's controller. The resolved joint angle data from the BAS method were sent into the KUKA manipulator's operating system to perform path tracking control. The snapshots of the experimental session on the KUKA manipulator are shown in Fig. 5.7, and its tracked path in Cartesian space is shown by Fig. 5.8. From the results of trajectory regeneration, we can conclude that the proposed method performs well on a real robotic platform.

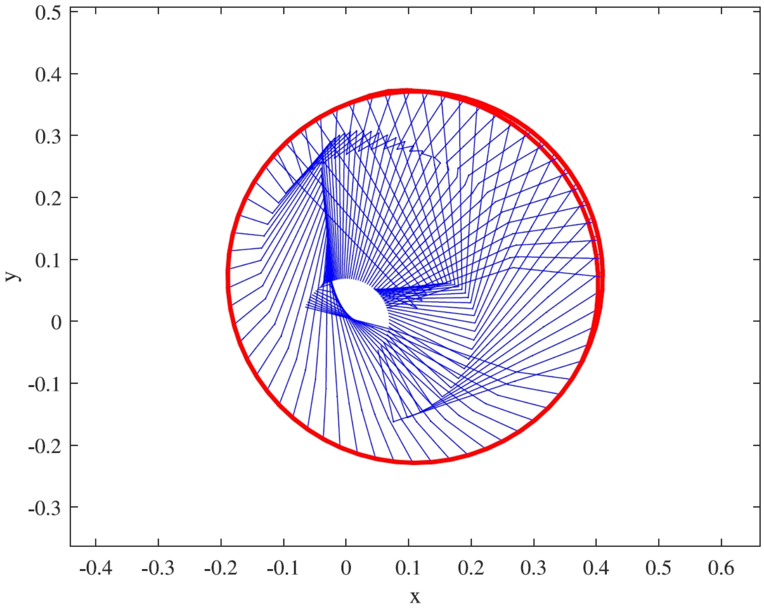


Figure 5.8: The tracked path in Cartesian for drawing the circle under the BAS-ELM space.

5.6 Conclusion

This brief developed an enhanced learning interface for robotics using ELM and BAS. The method of calculating the data of each joint angle through the MYO Armband has been studied, and the collection of arm endpoint's coordinate data with Kinect v2 sensor has been completed, thereby establishing a mini dataset including joint angles and endpoint positions. A BAS-ELM prediction model was proposed, which overcame the difficulty of selecting the initial learning weights and bias for ELM and greatly improved the prediction accuracy. Several experiments were conducted and validated the proposed method by increasing the prediction accuracy with a relative error reduction of 56.1% compared to the standard ELM without parameter optimisation. The successful running of the trajectory generation on a real robot platform indicated that when the training sample size is big enough, the proposed BAS-ELM strategy can be used to replace the kinematics calculations to map the relationship between the endpoint positions and the relevant joint angles.

Meanwhile, there are still some further tasks needing to be done in future work. One of the key challenges will be improving the selection of BAS initial step size and solving the problem of inappropriate convergence. Additionally, further exploration of how the BAS-ELM model performs in other complex robotic systems will also be crucial.

Chapter 6

A Robot Drawing System Based on Image Segmentation of Fault Tolerant Algorithm

This chapter is based on our paper “A Robot Drawing System Based on Image Segmentation of Fault Tolerant Algorithm”(under review).

6.1 Introduction

Drawing robots hold significant value in industrial production due to their ability to improve production efficiency, ensure product quality, reduce costs, enhance the working environment, and increase the flexibility and adaptability of production lines[150]. These robots can operate continuously, minimising downtime associated with manual labour, which significantly boosts production efficiency. They also utilise precision control systems to ensure the consistency and uniformity of each spray, thereby improving the product's appearance and durability[151]. While the initial investment in drawing robots can be high, they effectively reduce long-term production costs by minimising material waste and lowering labour expenses[152]. Additionally, drawing robots decrease the exposure of workers to harmful environments, improving workplace safety and employee health. Modern drawing robots are highly flexible, quickly adapting to various production tasks and enhancing the adaptability of production lines[153].

Common spraying technologies employed by drawing robots include air spraying, ultrasonic spraying, and electrostatic spraying, each with distinct advantages and disadvantages depending on the application scenario and coating characteristics[154]. In the automotive industry, drawing robots are used for car body spraying to ensure uniform and consistent coating while increasing production speed and efficiency. In the aerospace and defence sectors, they are employed for drawing aircraft and military equipment, providing high-quality surface treatment and corrosion protection[151]. Drawing robots are also extensively used in the production of high-quality consumer electronics, building materials, and household products[153].

To enhance the practical value of drawing robots and better integrate them with other industrial applications, this research explores the use of edge detection algorithms to obtain the contours of images to be drawn. Edge detection technology is widely used in image processing and is effective for image contour extraction, which is fundamental for drawing robots[155]. The quality of the image contours obtained directly affects the quality of the drawn image. At the same time, the combination of robot fault-tolerant algorithm and robot drawing system can greatly improve its stability and enable it to be more widely used in the industrial field.

Unlike the real-time droplet edge extraction introduced in Chapter 3, the image edge extraction of the painting robot has significant differences in core goals and technical requirements, mainly due to the different physical characteristics, time constraints and output purposes of the application scenarios. The movement or collision process of droplets often requires high-speed video recording of hundreds to thousands of frames per second. Edge extraction must be completed within 1 to 10 ms, otherwise the dynamic features will be lost, so the algorithm needs to be lightweight to achieve faster speeds; the static image processing delay requirements of the painting robot edge extraction are low (0.5 to 2 seconds), and more attention is paid to edge continuity and detail retention. Droplet edge extraction focuses on the precise coordinates of edge pixels, allowing discontinuities or noise, and then completing them through fitting curves; the edge extraction of the painting robot pays more attention to structural integrity and aesthetics. The edges need to be closed and smooth, and breaks will cause interruptions in the painting lines.

The Canny edge detection operator, a classic tool in image processing, has been extensively utilised and refined by numerous researchers. The traditional Canny edge detection algorithm consists of four main steps: Gaussian filtering, gradient calculation, non-maximum suppression, and double threshold processing[156]. Most improvements to the algorithm focus on these four areas. Zhang proposed a noise-robust edge detector based on an automatic anisotropic Gaussian kernel, which addresses the issue of the Canny edge detector potentially missing some cross-edge details[157]. Thirumavalavan et al. introduced an improved teaching-based robust noise image edge detection algorithm that effectively reduces broken and jagged edges, resulting in continuous and clear edges[158]. Adamik et al. took a different approach, using a genetic algorithm to decompose the image into line segments for vectorisation. This method allows for the creation of detailed and aesthetically pleasing pencil drawings using simple line segments[159].

In addition to these image-processing techniques, the integration of drawing robots with robotic arm fault-tolerant algorithms has been a significant area of research in industrial automation. Robotic arm fault-tolerant algorithms play a crucial role in maintaining system stability, especially in complex and high-risk environments. These systems are extensively utilized in production lines, medical surgeries, space exploration, and other fields. The development of AI, machine learning, deep learning, and digital twin technologies has significantly enhanced fault-tolerant control techniques[160]. For instance, AI and machine learning enable more precise fault prediction and diagnosis, thereby improving the system's adaptability and fault tolerance[161].

Advanced control methods, such as sliding mode control and neural network control, are widely implemented in fault-tolerant systems to strengthen the robustness and stability of robots in uncertain environments[162]. Research indicates that integrating the extended state observer with computed torque control can effectively mitigate the impact of unknown inputs on the robot system without relying on speed measurements, thereby achieving high-precision trajectory tracking[163]. For actuator failures, a fault-tolerant control method based on visual serving enables precise target tracking through a monocular vision system and automatically adjusts the control strategy to ensure stable system operation when a fault occurs[72]. AI technologies, such as deep neural networks and fuzzy logic control, are employed in active fault-tolerant control for multi-source navigation systems, significantly enhancing system autonomy and fault-handling capabilities in complex environments[164].

These advancements in edge detection and fault-tolerant algorithms are crucial for the continued development and application of drawing robots in industrial settings, offering both improved reliability and efficiency.

6.2 Edge Detection in Optical Imaging

The first part of this section discusses the improved canny edge detection algorithm for a drawing robot based on a fault-tolerant algorithm, which can obtain clear edges in the image for the robot to draw. The detection is not stable enough when the traditional edge detection algorithm is applied to high-speed cameras[97]. Therefore, the second part of this section proposes a double emulsion droplet detection method based on colour space and Hough circle transform. It can reach the point directly and has a strong purpose. At the same time, the detection speed and accuracy can also be guaranteed.

In order to enhance the practical value of the drawing robot and better integrate with the application of robots in other industries, this research discusses the use of edge detection algorithms to obtain the image contours that need to be drawn.

The Canny operator uses a Gaussian filter to perform convolution filtering on the image, mainly to eliminate noise and reduce the impact on gradient calculation[156]. Gaussian filter can be expressed as:

$$G\left(x, y, \sigma\right) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{6.2.1}$$

The parameter standard deviation σ directly determines the width of the filter and the smoothness of the image. The σ is positively related to the smoothness of the image and inversely related to the accuracy of the extracted edges.

The standard deviation of the traditional Canny edge detection algorithm is fixed when processing an image, which may cause the signal-to-noise ratio of the edge detection results to be small and the extracted edges to be inaccurate[156]. Gómez proposed a method for adaptively improving the standard deviation. The core idea is to determine the value range of the Gaussian standard deviation based on the image information and take a series of standard deviations of different sizes within this range. The information description structure is used to describe the process of Gaussian smoothing, and the pixels in the image are encoded based on the description length under different standard deviations. Finally, the standard deviation corresponding to the information optimisation scheme of the description length is selected as the adaptive standard deviation[165]. The equation can be expressed as:

$$\sigma_{best} = \left[argmin\frac{c}{\sigma^2} + \varepsilon^2\right] \tag{6.2.2}$$

c is a constant term. ε^2 is the local variance after Gaussian smoothing, this value is smaller in smooth areas of the image and larger at the edges of the image. This allows Gaussian filtering to be performed on values close to the maximum standard deviation in flat areas to obtain a smoother image; Gaussian filtering is performed on values close to the minimum standard deviation in edge areas to obtain more accurate edges. Through experiments, it can be found that this kind of optimisation can achieve a better signal-to-noise ratio[166].

The classic Canny operator uses the finite difference of adjacent regions of size 2×2 to calculate the value and direction of the image gradient. As shown in equations 6.2.3 and 6.2.4 :

$$P_x\left(i, j\right) = \left(I\left(i, j+1\right) - I\left(i, j\right) + I\left(i+1, j+1\right) - I\left(i+1, j\right)\right)/2 \tag{6.2.3}$$

$$P_y\left(i, j\right) = \left(I\left(i, j\right) - I\left(i+1, j\right) + I\left(i, j+1\right) - I\left(i+1, j+1\right)\right)/2 \tag{6.2.4}$$

The gradient magnitude $M\left(i, j\right)$ represents the edge strength at a certain point $\left(i, j\right)$ on the image:

$$M\left(i, j\right) = \sqrt{P_x\left(i, j\right)^2 + P_y\left(i, j\right)^2} \tag{6.2.5}$$

The gradient direction $\theta\left(i,j\right)$ is perpendicular to the edge direction and is the normal vector of the point $\left(i,j\right)$ on the image:

$$\theta\left(i,j\right)=arctan\left(\frac{P_y\left(i,j\right)}{P_x\left(i,j\right)}\right) \tag{6.2.6}$$

Liu extends the regions of size to 3×3 to calculate the image gradient, and uses the Prewitt operator to perform convolution[166]. Wang et al. also chose a 3×3 region of size to calculate the image gradient, but used the Sobel operator[167]. They all chose larger templates to calculate the gradient of the image for higher accuracy results.

The edge of the image calculated by the gradient is very blurred, and there are many edge gradient responses. Images are often wide edges composed of multiple pixels, so the image needs to be further refined to obtain a more accurate image.

Due to the symmetry properties, it can be divided into four categories according to the non-maximum suppression area. First, determine the direction of the gradient and classify it. After determining the area to which it belongs, the gradient amplitude of this point is compared with two adjacent points in the gradient direction. Only when the point is larger than two adjacent points, it can be determined as an edge point[156].

The classical double threshold processing algorithm theory is to artificially set high and low thresholds, and then retain all pixels bigger than the high threshold as edge points, and all points smaller than the low threshold as non-edge points. Points between high and low thresholds are used as candidate edges. If these pixel points are connected to the determined edge, these points are also considered edge points; otherwise, they are considered non-edge points[156].

This method requires the artificial setting of thresholds, which is not only inefficient but also very difficult to set thresholds with appropriate values for different images, which greatly affects the performance of the algorithm[168]. Japanese scholar Otsu divided the image into two parts: background and foreground according to the distribution of gray values on the image. The segmentation of background and foreground requires a threshold, and the key point is how to obtain this value. Traverse different thresholds and calculate the inter-class variance between the corresponding background and foreground under different thresholds, when it reaches the maximum value, the corresponding threshold at this time is the required threshold[169].

If the pixel in the image is $A\times B$, $f\left(x,y\right)$ is the pixel value of the pixel $\left(x,y\right)$, L is the gray level of the image, then the probability that the pixel value is i is :

$$p\left(i\right)=\frac{1}{AB}\sum_{f(x,y)=i}f\left(x,y\right),i\in\left[0,L-1\right] \tag{6.2.7}$$

After selecting a segmentation threshold T to divide all pixels in the image into target areas and background areas, the probability and number of pixels in each area can be calculated:

$$p_0(H) = \sum_{0 \le i \le H} p(i)\ ,\ p_1(H) = \sum_{H \le i \le L-1} p(i) \tag{6.2.8}$$

$$n_0(H) = AB \sum_{0 \le i \le H} p(i)\ ,\ n_1(H) = AB \sum_{H \le i \le L-1} p(i) \tag{6.2.9}$$

The average pixel value of each area image is:

$$u_0(H) = \sum_{0 \le i \le H} \frac{i \cdot p(i)}{p_0(H)}\ ,\ u_1(H) = \sum_{H \le i \le L-1} \frac{i \cdot p(i)}{p_1(H)} \tag{6.2.10}$$

Then the average pixel value of the image is:

$$u = p_0(H)\, u_0(H) + p_1(H)\, u_1(H) \tag{6.2.11}$$

Then the optimal threshold for the edge is:

$$T = ArgMax \left[p_0(H)\left(u_0(H) - u\right)^2 + p_1(H)\left(u_1(H) - u\right)^2 \right] \tag{6.2.12}$$

6.3 A Fault-Tolerant Algorithm for Motion Planning

Fault-tolerant algorithms are designed to maintain normal operation or quickly recover even in the presence of errors, failures, or anomalies in a system or computational process. The primary objective of these algorithms is to minimise the impact of errors on the overall functionality of the system and to ensure its reliability and stability under various unforeseen conditions, such as hardware failures, network issues, and data errors[162]. This research uses the algorithm to optimise the working process of the drawing robot.

6.3.1 Failure Causes and Their Determination Methods

After long periods of repetitive operation, redundant manipulators may not operate as expected due to the sudden failure of joint actuators or unexpected joint locking. These failures may be caused by mechanical wear, actuator failure, or external impact. In addition to mechanical failures, some joints may suddenly lose velocity (velocity becomes zero) or become accidentally locked, resulting in the end effector's path-tracking task being unable to continue or accurately complete. In the case of failures caused by some joint-free cases, their velocity may become arbitrary. Therefore, we can express the mathematical model of the failure of the i-th joint as $\dot{\theta}_i(t > t_f)$ for time instant $t > t_f$, Where $\dot{\theta}$ represents the velocity of the joint angle and t_f represents the time when failure occurs. So the joint failure can be expressed as follows:

$$joint\ failure \begin{cases} \dot{\theta}_{j1}\left(t > t_{f_1}\right) = 0 \\ \dot{\theta}_{j2}\left(t > t_{f_2}\right) = 0 \\ \quad\quad ... \\ \dot{\theta}_{jm}\left(t > t_{f_m}\right) = 0 \end{cases} \tag{6.3.1}$$

Where $t_{f_1}, t_{f_2}...t_{f_m}$ represents the time point when the $j_1, j_2...j_m$th joint fails.

6.3.2 Solution Algorithm

In this section, Li et al. proposed a robust motion planning strategy and provided its optimal solution and convergence analysis. When some joints lose their normal motion ability, the normal motion of the end effector still needs to be guaranteed. In this case, joint failure can be regarded as a constraint to solve the redundancy resolution. Therefore, a fault-tolerant redundancy elimination optimisation criterion is proposed:

$$minimize\ \left\|\dot{\theta}\right\|,\quad subject\ to\ J\dot{\theta} = v_d,\quad \dot{\theta} \in \Omega \tag{6.3.2}$$

where v_d donates the desired velocity vector of the end effector, J represents Jacobian matrix, and Ω represents the joint velocity fault tolerance set.This formula can minimise the joint velocity of the robot arm while satisfying specific kinematic constraints and fault tolerance conditions, thereby reducing kinetic energy and wear.

The primal-dual neural network is well suited to solving this constrained optimisation problem. It gradually approaches the optimal solution by iteratively updating the joint velocities and Lagrange multipliers. In each iteration, it adjusts the joint velocities to satisfy the kinematic constraints and error tolerance conditions while minimising the objective function[170]. It can be expressed as:

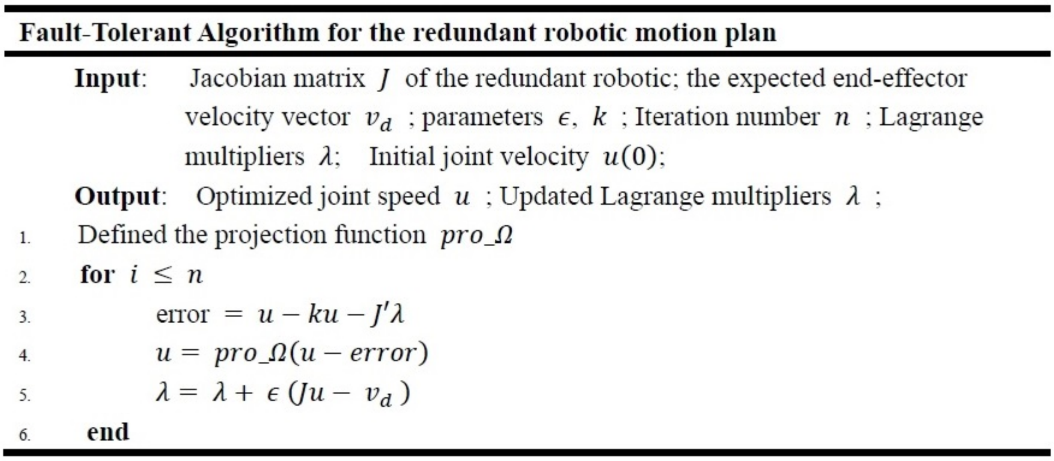
$$\begin{cases} \epsilon\dot{u} = -u + P_\Omega\left(u - ku - J^t\lambda\right) \\ \epsilon\dot{\lambda} = Ju - v_d \end{cases} \tag{6.3.3}$$

where $u = \dot{\theta}$ means the joint velocity vector; λ is the Lagrange multiplier vector used to handle equality constraints in optimization problems; ϵ is used to adjust the convergence speed of the dynamic equation, and k is a proportional factor used to control the update step size of the algorithm. $P_\Omega(\cdot)$ can be specifically expressed as:

$$P_\Omega(u) = \begin{cases} u^+,\ u > u^+ \\ u,\ u^- \le u \le u^+ \\ u^-,\ u < u^- \end{cases} \tag{6.3.4}$$

where the projection operation of $P_\Omega(\cdot)$ ensures that the joint velocity u satisfies the constraints of the fault-tolerant set Ω, that is, the case of joint failure is taken into account.

The convergence of the optimisation solver has also been proven in our previous work[171]. Figure 5.1 shows the algorithm's brief process.



Fault-Tolerant Algorithm for the redundant robotic motion plan

```
Input:  Jacobian matrix $J$ of the redundant robotic; the expected end-effector
        velocity vector $v_d$ ; parameters $\epsilon$, $k$ ; Iteration number $n$ ; Lagrange
        multipliers $\lambda$;  Initial joint velocity $u(0)$;
Output: Optimized joint speed $u$ ; Updated Lagrange multipliers $\lambda$ ;
Defined the projection function $pro\_\Omega$
for $i \le n$
    error $= u - ku - J'\lambda$
    $u = pro\_\Omega(u - error)$
    $\lambda = \lambda + \epsilon\,(Ju - v_d)$
end
```

Figure 6.1: Fault-Tolerant Algorithm for the redundant robotic.

6.4 Simulation and Experiment

In this part, the author initially used MATLAB to simulate the robot's motion trajectory and obtain data for each joint angle under multiple joint failure scenarios, thereby verifying the effectiveness of the fault-tolerant algorithm. Subsequently, MATLAB was employed to facilitate communication with the Kuka LBR iiwa robot, enabling a real-time online process for image recognition, trajectory planning, and robot drawing.

6.4.1 Simulation verification of Fault-Tolerant Algorithm

The Denavit-Hartenberg parameters of the KUKA LBR iiwa's joints and links are shown in Table 6.1. It expresses the correlation between the joints and links of the robot through the geometric structure of the system. The kinematic model of the KUKA LBR iiwa can be represented by a transformation matrix that describes the relationship between the base coordinate system and the end effector:

$$T(\theta) = T_1(\theta_1)\,T_2(\theta_2)\,T_3(\theta_3)\,T_4(\theta_4)\,T_5(\theta_5)\,T_6(\theta_6)\,T_7(\theta_7) \tag{6.4.1}$$

Equation 6.4.1 can be used to obtain the position and posture of the end effector relative to the robot base frame.

The robot is programmed to draw a regular octagonal trajectory in the xy-plane. The initial joint angle of the robot is set to $\theta_0 = 0.8\times\left(\frac{\pi}{5}\ \frac{\pi}{3}\ \frac{\pi}{6}\ \frac{\pi}{3}\ \frac{-\pi}{3}\ \frac{\pi}{6}\ \frac{-\pi}{3}\right)^T$, and the convergence parameter ϵ for the optimization solver is configured to 0.0001. During the simulation, joint

Table 6.1: The Denavit-Hartenberg (DH) parameters of the Kuka LBR iiwa robot

Link	$a_i\,(m)$	$\alpha_i\,(rad)$	$di\,(m)$
1	0	$-\frac{\pi}{2}$	0.34
2	0	$\frac{\pi}{2}$	0
3	0	$\frac{\pi}{2}$	0.4
4	0	$-\frac{\pi}{2}$	0
5	0	$-\frac{\pi}{2}$	0.4
6	0	$\frac{\pi}{2}$	0
7	0	0	0.126

2 experiences a failure after 6 seconds, and joint 6 fails after 4 seconds. Joint 7 is locked throughout the process to maintain the stability of the end effector brush.

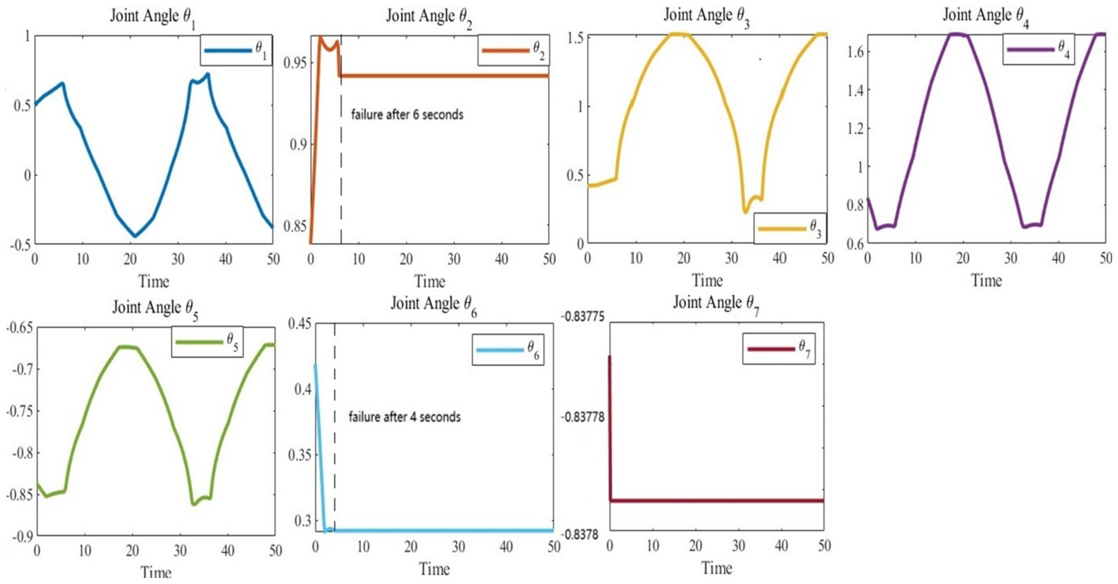


Figure 6.2: The joint angle data returned by the MATLAB simulation. Joint 2 fails at 6 seconds, joint 6 fails at 4 seconds, and joint 7 is always locked. All other joints are in normal working condition.

Figure 6.2 shows the change of the angle of each joint overtime during the robot's trajectory under the above constraints. It can be seen that joint 2 fails at 6 seconds, joint 6 fails at 4 seconds, and joint 7 is always locked without any angle change. All other joints are in normal working condition. Figure 6.3 illustrates the trajectory of the end effector with a red line, while the blue line represents the projection of the robot onto the xy-plane. The left image depicts the trajectory generation under normal joint conditions, whereas the middle image shows the trajectory generation with joint failure. Comparing these trajectories demonstrates that the scheduled work can still be completed effectively even when the joint experiences failure. According to the rightmost image depicting error variation over time, the deviation between the expected path and the path planned by the fault-tolerant algorithm remains within 0.17 mm throughout the entire process.

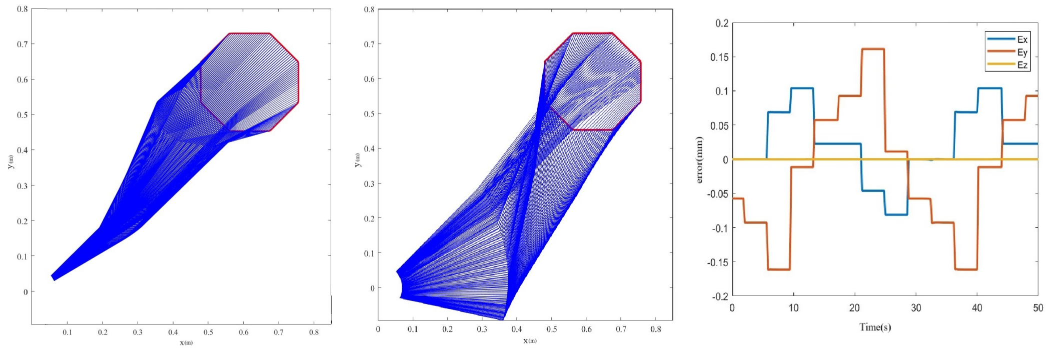


Figure 6.3: The left image shows the motion trajectory when the joint is working properly, the middle one shows the motion trajectory when the joint fails, and the line graph on the right shows the error between the expected trajectory and the trajectory obtained by the fault-tolerant algorithm.

6.4.2 Realisation of Robot Drawing

First, the image edges are extracted using an improved Canny edge detection algorithm. The extracted coordinates are then used to define the ideal trajectory. In the event of a joint failure, the fault-tolerant algorithm optimizes the trajectory fitting based on the ideal path, enabling the robot to follow the trajectory planned by the algorithm.

The snapshot in Figure 6.4 shows that the robot can complete the work well according to the planned path, and the content drawn is shown in Figure 6.5. Figure 6.5(a) is the original image to be recognised, Figure 6.5(b) is the path image planned by the fault-tolerant algorithm, and Figure 6.5(c) is the image drawn by the robot. It can be seen that the KUKA LBR iiwa robot performed the drawing task very well.

Figure 6.6 illustrates the robot's ability to draw a regular octagon despite joint failures. Joint 3 fails at the 60th sample, joint 4 fails at the 80th sample, and joint 7 remains locked throughout. With ten samples taken per second, the data shows that the robot can still complete its task, even when unexpected failures occur.

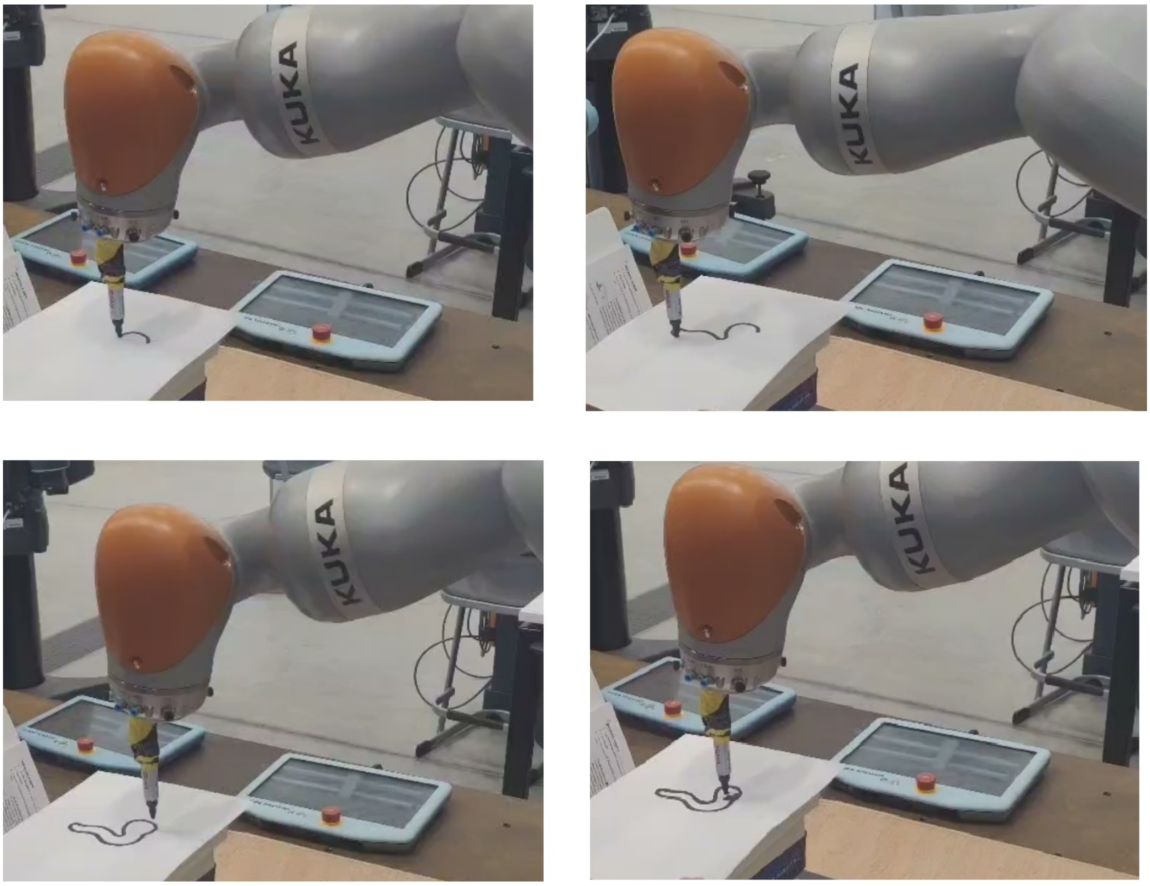
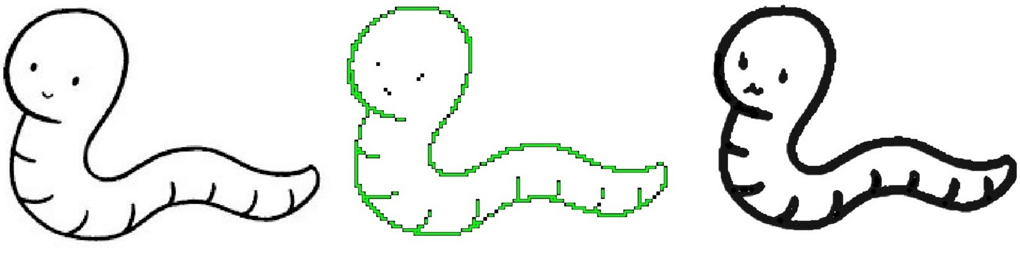


Figure 6.4: Snapshots of the KUKA LBR iiwa robot drawing a trajectory planned by the fault-tolerant algorithm.



(a) (b) (c)

Figure 6.5: (a) is the original image, (b) is the path image planned by the fault-tolerant algorithm, and (c) is the image drawn by the robot.

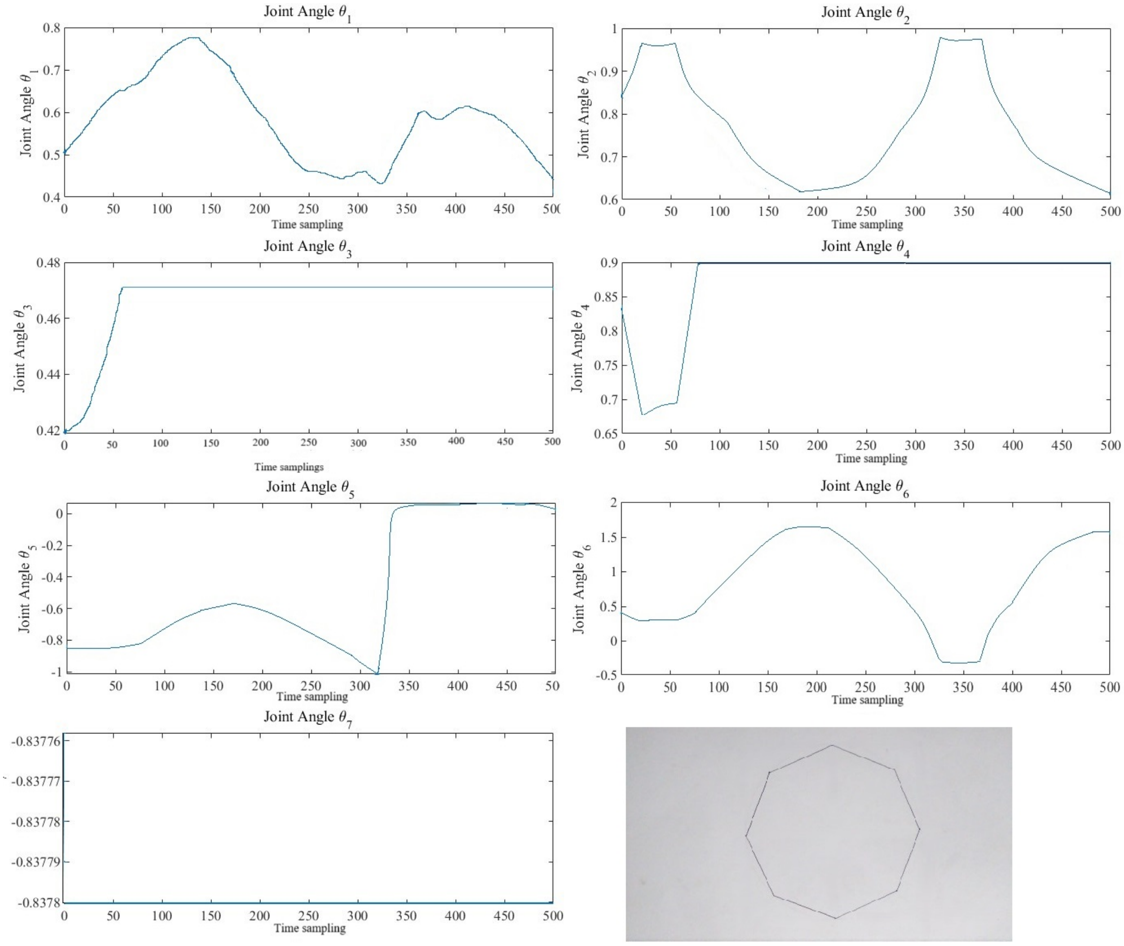


Figure 6.6: The real joint angle changes of the seven joints collected by the robot when drawing a regular octagon. The lower right corner is the drawn regular octagon.

6.5 Results and Discussion

This chapter validated the effectiveness of combining fault-tolerant algorithms with drawing robots through experiments conducted on the KUKA LBR iiwa robot platform. The experiments simulated mechanical failures typical of industrial environments to assess the robot's performance under multi-joint failure conditions. By optimising the redundancy resolution strategy, the normal movement of the end effector was maintained despite failures, thereby improving the reliability of the drawing robot. This approach not only enhances the robot's autonomy and adaptability but also expands its potential applications in industrial automation systems, particularly in scenarios requiring high precision and continuous operation. The findings offer valuable insights for other robotic applications that demand high reliability. Future research could focus on developing drawing robots capable of operating on uneven surfaces, necessitating continuous adjustment of the end effector's angle. Additionally, to ensure smooth painting lines, the robot's force control system must be engaged, which presents further challenges for fault-tolerant algorithms.

Chapter 7

Development of a Data Glove Using Flex Sensors for the Robot Hand Teleoperation

This chapter is based on our published paper “Development of a low-cost data glove using flex sensors for the robot hand teleoperation”.

7.1 Introduction

Converting sensor input to program output is the basis of many fields of robotics, from computer vision to virtual and augmented reality, both commercially and in research. This project was originally planned to use a real anthropomorphic robotic hand. However, current circumstances have led to the project being simulation-based.

A data glove is a piece of wearable technology that enables human-machine interaction, converting real-world movement of the hand into usable sensory data for software. Many approaches to capturing physical data have been developed over the past 50 years, starting even before the conception of the Sayre glove in 1977, transitioning from optical and magnetic-based motion capture to a more hands-on wearable form[172]. However, the data captured is often still used in conjunction with gyroscopic and inertial sensors for tracking the hand in 3D space.

Its applications can vary from biomedical research, the teleoperation of a system, to virtual and augmented reality, with different, focuses on precise finger control to more gesture-based means, allowing for different techniques to be applicable in different scenarios. This gives more creativity and engineering freedom over the way in which the glove is designed to suit its functionality. The core principle of operation between gloves remains very similar, in that flex sensors are still often placed on the fingers to measure finger flexion, whilst positional sensors track the whole hand.

This project will explore a variety of different sensors before choosing the most viable for the task of simple teleoperation. A simulated anthropomorphic hand will be controlled, with one degree of freedom per finger and each finger containing 3 coupled joints. The goal is to allow for operation over said hand with both a commercially available glove and a personally designed glove. By achieving this, the data gloves will prove themselves as an effective tool for real-world teleoperation, as well as any other suitable task. The hope is to build on this in the future, so that positional and haptic feedback could be implemented within the glove, along with more degrees of freedom to allow for increased tracking precision.

Most data gloves available on the market can cost thousands of pounds, effectively limiting their accessibility to much of their potential target audience. Despite this, completing this project will demonstrate that the key elements of these systems can be mirrored and even improved upon for a far cheaper price.

7.2 Requirements and Aims

The project was based around both a robotic hand and a data glove, with more focus on designing and building a real-world hand. The initial requirements were to create both glove and hand with one degree of freedom per finger, and then expand on that to give more degrees of freedom and control for both. These requirements form the basis of the aims of the project, with each aim representing a milestone towards assembling and testing the data gloves:

1. An anthropomorphic robotic hand will be modelled in CAD software, then exported and assembled as a robot within simulation software. Mapping the joint angles and distances would allow for forward and inverse kinematics to be used to calculate end effector location from known joint angles, or vice versa.

2. The joint angle data from the previously bought data glove will be used to control the simulated robotic hand. The data can be transferred over serial or writing to an array file to allow the simulation to read the joint angles and move to the target position.

3. A data glove will be made to be used in conjunction with the simulation, having already completed the software design for the operation of the hand. Both gloves can use a recalibrated version of the same code.

7.3 Proposed Methodologies

7.3.1 Motion tracking

The initial approach to capturing physical data from the hand was motion tracking in an enclosed environment. There were predominantly 4 methods employed to track the hands' position, which later developed into what is used today. Markers placed on the hand in specific points would be tracked using a set of cameras around the hand, using their relative positions in space to build a graphic model that can calculate joint angles between each point. The Massachusetts Institute of Technology(MIT) Light Emitting Diode(LED) glove used LEDs to capture the motion of just the hand, which was transposed onto the whole body afterwards for mapping[173]. This formed the foundations of CGI and motion capture technology used today in industries such as film and research[172].

A low-frequency magnetic field emitted from the hand would communicate with the receiver where the hand is in space relative to the sensor. This had the advantage of a large range with a high degree of accuracy but was prone to interference from any magnetic objects decreasing accuracy. This was largely used for positional tracking of the whole hand for a lot of early data gloves[174].

High-frequency ultrasound was used to 'ping' the location of the component on the hand to several microphones, which would use triangulation to calculate the relative position and orientation of the hand. This method was far cheaper and lightweight, but was easily interfered with by external sound, and the accuracy was dependant on the acoustic properties of the room. This technique was used in the Mattel (Nintendo) Power Glove[172].

Finally, the position and orientation data can be relayed to a receiver from gyroscopes and accelerometers. This method uses units that contain multiple tracking components but is susceptible to inaccuracies over time and gyroscopic drift, but this is correctable with filtering techniques such as the Kalman filter, which is widely used today for all positional tracking techniques[174].

The most relevant to the project would be the aforementioned first methods. The first being represented by the Leap Motion Controller, an optical hand tracking module commonly used for joint angle and positional data acquisition[175]. Features such as fingertips are extracted from the camera's shot, and vectors are drawn in to represent position and direction. Angles can then be calculated using the intersection between vectors[176].

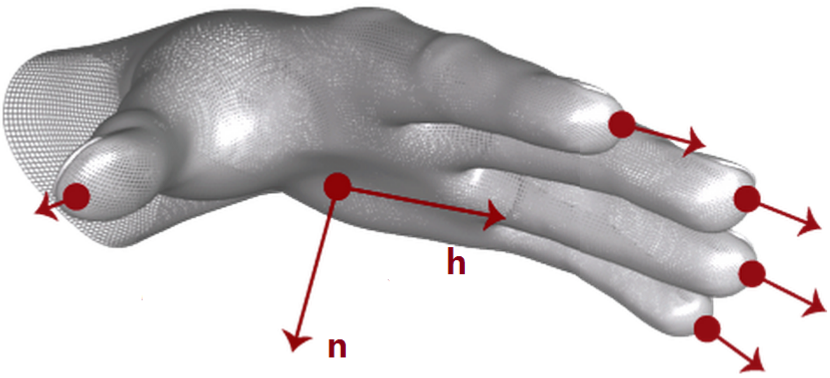


Figure 7.1: Features are extracted to draw vectors on each point, which can be used in finding the joint angles and relative position of the rest of the hand. Adapted from reference 176.

7.3.2 Potentiometers and String Potentiometers

A potentiometer is a three-terminal adjustable voltage divider that comes in many forms, widely used for a plethora of different applications[177]. They typically consist of a sliding contact that moves along a resistive element, so that the position of the slider dictates the value of the output resistance. Used in conjunction with an attached mechanism, such as a lever or joystick, they become a very versatile tool in measuring position in a system[178].

The voltage across the component can be found using the simple voltage divider rule[177]:

$$V_L = \frac{R_2 R_L}{R_1 R_L + R_2 R_L + R_1 R_2} \cdot V_S \tag{7.3.1}$$

Dividing through by R_L and cancelling gives the simplified and well-known form of the equation, as the load is large in comparison to other resistances:

$$V_L = \frac{R_2}{R_1 + R_2} \cdot V_S \tag{7.3.2}$$

This can be used coupled with a mechanism to convert joint angles to a resistance value, which could then be reinterpreted back to a joint angle by the data glove controller. The only problem with this approach is designing a method in which desired movements can be translated into rotation of the potentiometer handle. This also limits the potentiometer to one degree of freedom, but this can be circumvented by using a dual-axis potentiometer commonly used in joysticks.

An extension of the potentiometer, the string potentiometer attaches a pullable string mechanism to the adjustable handle of the potentiometer, accompanied by a gearing and

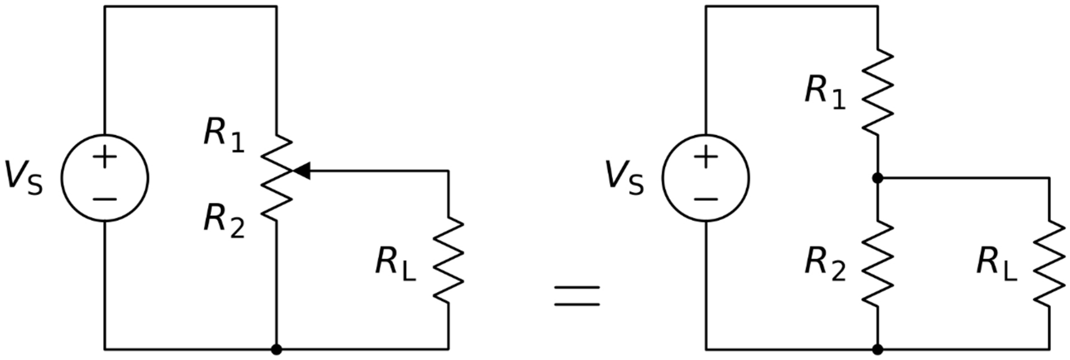


Figure 7.2: A further simplified version of the voltage divider circuit within the potentiometer, adapted from reference 178.

spring mechanism for multiple turns and returning the string potentiometer to a neutral position. The rotational sensor can be in the form of a potentiometer or a rotary encoder[179]. An electrical signal proportional to the cable's linear extension is produced, which can be differentiated as a function of position to give velocity.

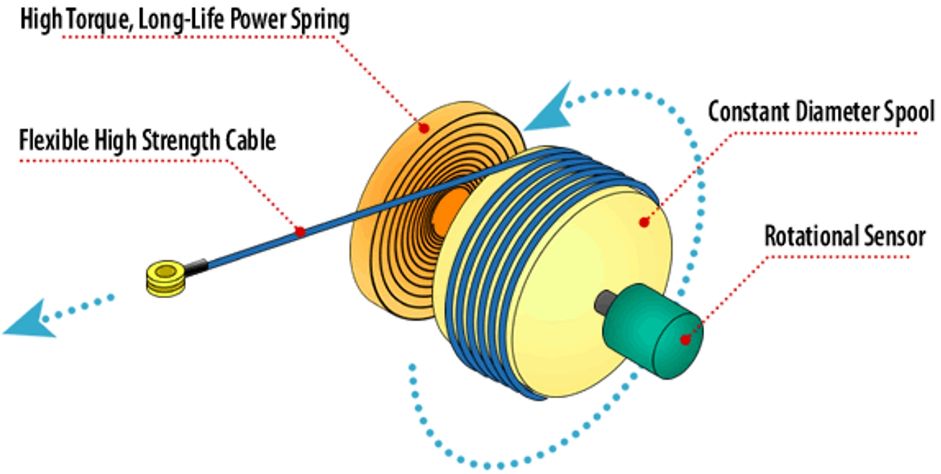


Figure 7.3: The inner workings of a string pot. The cable is pulled and extends from around the spool, which is measured by the rotational sensor. The spring returns the string pot back to a resting position. Adapted from reference 179.

This method could be useful in measuring joint angles by attaching the ends of the cables to each fingertip so that flexion of the fingers pulls the cables and increases the resistance. The advantage of this would be that the housing of each potentiometer doesn't have to be fully aligned, and the string will easily stay where it is lined up despite contortion.

An advanced way of implementing this would be in the same fashion that tendons operate within the forearm and fingers, except in the opposite direction. However, string pots are very expensive to buy and would have to be homemade. Furthermore, this method is too similar to lever-action with potentiometers, in that it would only give one degree of freedom per finger for a higher price.

7.3.3 Flex Sensors

Flex sensors come in different varieties and have been used since the first data glove[172]. The Sayre glove used optical flex sensors consisting of a flexible tube housed with a light source at one end and a photovoltaic cell at the other. Bending the tube would evenly decrease the amount of light reaching the photocell[180]. This principle has been taken forward and adapted over time to make different kinds of flex sensors.

The fibre optic flex sensor is similar to the optical flex sensor, but instead uses optical fibres. Applying stress to the fibres will change the optical properties of the material, affecting the properties of the light passing through them. Different properties such as intensity, phase, polarisation, wavelength and transmit time can be analysed with a spectrum analyser to provide sensing over a large distance[181]. Fibre optic sensors are unaffected by electromagnetic fields and are less likely to be interfered with by external forces. They are also lightweight and can be applied with minimal intrusiveness[182]. However, more advanced hardware and software such as a spectrum analyser is needed, making it more expensive and unnecessarily technologically advanced for the scope of the project.

Capacitive flex sensors work by detecting changes in the electric field when the sensor undergoes stress. The sensor will consist of single or multiple conductive layers consisting of flexible microstructures, separated by a dielectric[183]. Applying a negative strain perpendicular to the capacitive plates (a compression) changes the distance between them, causing a positive change in capacitance. To avert three-dimensional strains, the top layer is made larger than the bottom, so that the capacitive area of the plate does not change as much, keeping the sensor selective towards loading normal to the sensor plane[184].

These sensors are more geared towards being used as a pressure sensor, easily being placed in situ close to areas of interest. A flex sensor can be made by applying this principle across a longer distance, where changes in the alignment between the two capacitive plates change the capacitance of the component and output[185].

Finally, conductive ink flex sensors work in much the same way as the capacitive flex sensor. Set up in a similar fashion to the capacitive flex sensor, a strip of flexible resin has a conductive ink deposited thereon with a segmented conductor on top, forming a flexible potentiometer[186]. Bending the sensor decreases the cross-sectional area and an increase

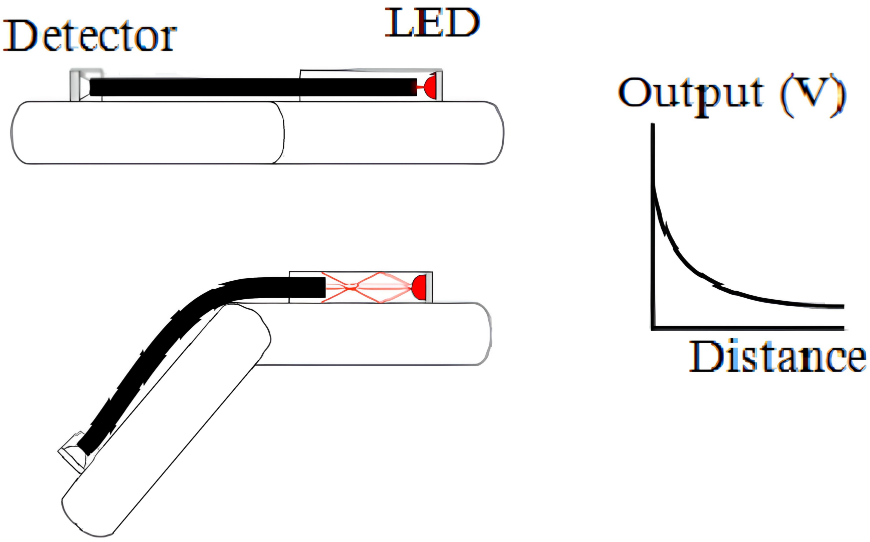


Figure 7.4: Principle of operation for a photocell flex sensor. Flexion results in a decrease in light intensity, as shown by the graph. Adapted from reference 180.

in length, resulting in a predictable change in resistance[187].

These have recently been adapted to use a new conductive carbon material, commercially known as Velostat or Linqstat. It is used as a packaging material made of a polymeric foil impregnated with carbon black to make it electrically conductive, making it useful for protection against electrostatic shock[188]. It can be used in place of conductive ink, and is commonly used for most hobby flex sensors today.

7.3.4 Strain Gauges

A more unorthodox method would be to use strain gauges, or load cells for each finger to measure flexion. A load cell is a transducer that converts an applied force into a desired and measurable output. This can be in the form of detecting different properties, such as bending, compression, changes in the magnetic field as input, and electrical, hydraulic or pneumatic for output[189]. The most popular and relevant type of which is the strain gauge.

The strain gauge consists of a resistive foil coated onto a gauge backing material, which converts small changes in the material's cross-sectional area caused by stress into a measurable electrical signal[190]. Using 4 strain gauges in a Wheatstone bridge allows for minute changes in stress to be detected in multiple axes[191].

These could be used at the knuckle of a data glove so that flexion of the fingers causes an

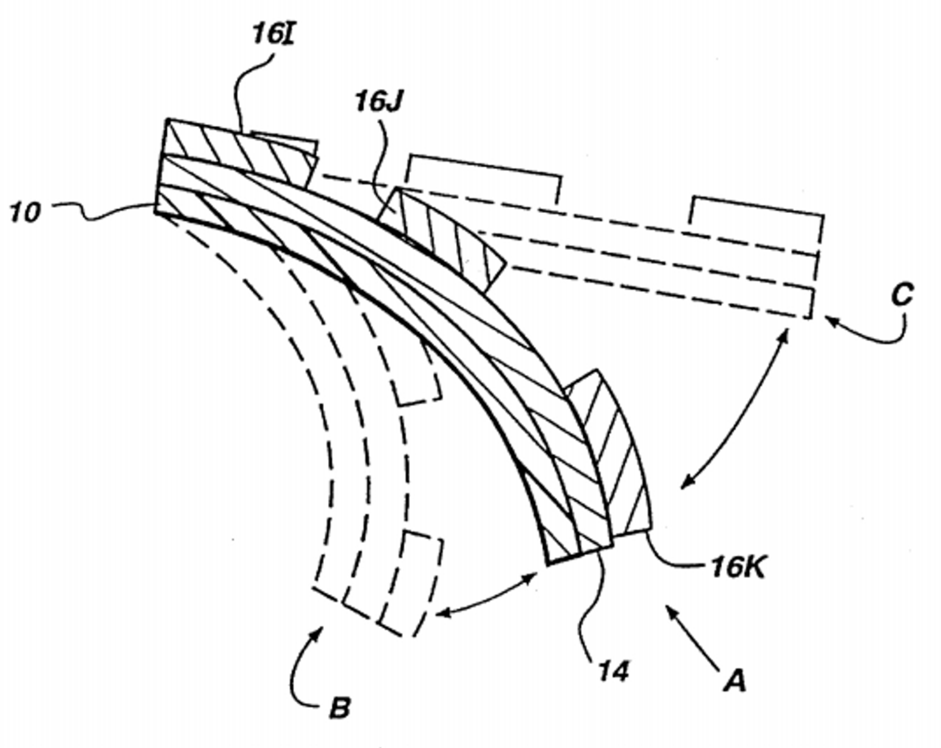


Figure 7.5: Operation of a conductive ink flex sensor. Different levels of bend result in different resistive properties of the interaction between the ink and conductive pads, which can be predictably measured. Adapted from reference 183.

increase in strain proportional to the angle of bending. This would allow for not only joint angle calculation, but for precise control of how much force is applied across the robotic hand. It would also interact well with haptic feedback and play more into the illusion of a robotic hand being an extension of one's own body.

This principle has been applied but in reverse with the Vive Knuckles controllers for their virtual reality system. The controller is fixed to the hand and wrist and is held more like a handle than worn like a glove. Joint angles are calculated by varying the grip on the controller, with a trigger finger for gripping and haptic feedback.

7.3.5 Choice of Microcontroller and Simulation Platform

The Arduino Uno R3 microcontroller has the same functionality and capabilities as the Nucleo. It has flexible power supply options such as ST-LINK, USB V bus or external sources, as well as connectivity to other boards. The Arduino's 16MHz clock speed and 14 I/O pins were enough, as only 5 analogue pins out of 6 were needed. It can provide 3.3V or 5V, with an input limit of 20V, making it more than suitable.

CoppeliaSim (formerly known as V-REP) is a robot simulation framework with its own Integrated Development Environment, based on a distributed control architecture in which each object can be controlled using an embedded script, plugin, node, API or custom solutions. It can be used to model a single robot with multiple links, or a whole environment containing multiple robots, and can be programmed in C/C++, Python, Java, Lua, Matlab or Octave. This makes it an extremely versatile tool for many applications, from robotic hand simulation to a Mars Rover simulation[192][193].

CoppeliaSim is an incredible fit for the project, as its advanced simulation platform will allow STL files to be imported in as a mesh and converted into many linked robots. Child scripts can then be written that move the hand to the target position based on data from the data gloves. External files can also be accessed by CoppeliaSim, meaning an external file containing joint angle data could be referenced during the simulation. Plugins can be installed that allow for this as well as many other useful functions.

MATLAB is a programming and numeric computing platform for the analysis of data, development of algorithms, and creation of models. It can be used for research and mathematics as well as the simulation of robots. It uses *C* style programming, Python, Java, Fortran and many other languages, with matrix and array mathematics and an iterative analysis and design process environment[194]. The simulation software ran slower than CoppeliaSim, as it is not as well suited as CoppeliaSim for more advanced simulations. It could however be used in conjunction with CoppeliaSim, for data transfer or calculation of joint angles from resistances.

7.4 Hardware Design

7.4.1 Hiwonder Data Glove

The glove uses lever-actioned potentiometers and does not involve the glove's or one's own nervous system. It has a lever coupled to a potentiometer for each finger and the thumb, which attach via a red adjustable ring. The rings wrap around the finger just below the proximal interphalangeal joint, also known as the second knuckle. It uses an Arduino Uno R3, housing a Bluetooth module and gyroscopic/inertial sensors for gesture control. These components allow it to communicate wirelessly with other products from Hiwonder, as well as other Bluetooth devices that are configured correctly[195]. The inertial and position sensors read the orientation and position of the glove relative to its starting point when running the demonstrational code that came with the product. Each potentiometer gives a reading of 500-2500 arbitrary units when running the demo program, but when purely reading from the analogue pins the values go from approximately 290-800 arbitrary units. The resistor values and range are unknown, but the website states it runs on a 7.4V LiPo battery and it is known to operate at 5V from plugged in. As the ADC 10-bit, its maximum count value is 1024 (2^{10}) and the I/O pins draw 40mA. Therefore, the resistance at rest and max flexion of the whole finger can be calculated. This does not solve the resistance of just the potentiometer, but the theoretical resistance of the finger, as there is probably a potential divider circuit for each finger.

$$ADCoutput = \frac{V_{in}}{V_{ref}} \cdot ADCmaxcount \tag{7.4.1}$$

7.4.2 Prototype Data Glove

Flex sensors were chosen as the optimal sensor for each finger, as they were easily available and relatively inexpensive compared to other options, given the lack of facilities and resources to 3D print or create a more advanced solution. Each flex sensor in its own potential divider circuit with a 10k resistor. Each flex sensor has a resting resistance of 30k and a fully bent resistance of 90k. Using these values, the voltage for each state and range of voltages can be calculated.

In order for each sensor to have the full range of motion, they have to be able to bend and relax while staying seated on top of the finger. They are usually sewn into the fabric or held underneath in some other fashion. For the project, small holes were cut into the sides of each finger so that zip ties could be used to create a track for the flex sensors to slide down. Each one is fixed at the top of the finger so that it will always bend from the top, allowing the rest of the sensor to slide through the zip ties up to the clinch connector. The wires coming from the sensors are fed through a small loop to avoid tangling, and each pair of wires leads to its respective voltage divider.

7.5 Software Design

7.5.1 Data Gloves

Both data gloves use an Arduino as their microprocessor. The Arduino Analogue and Serial classes were used to transmit data from the glove to the simulation in the form of float values representing resistance. The simulation receives this data and converts it into a joint angle in radians, which each joint in the finger will move to.

The loop runs at the clock speed of the Arduino, which is 16MHz. The simulation runs at a slower rate than the Arduino's clock speed, meaning it will never have to wait for new values to come up. As the values are not stored in an array or external file, a buffer is not needed to ensure proper data transfer. Storing these values would be helpful for replaying the movements of the hand, which could be used to teach an AI how to carry out a task.

7.5.2 Simulation

A model of the hand is established, and the measurement specification of the model allows the calculation of a homogeneous transformation matrix[196]. This would have allowed for a kinematic model to be produced that could use forward and inverse kinematics to calculate end effector locations and joint angles given values of one or the other.

The hand was imported into Fusion360 as a single body mesh. Each link and separate body in the hand had to be manually separated to build the linked robot. To place the joints, a cylinder had to be extracted from the housing of each joint. During the meshing and decimation process, the model's joints lost their cylindrical properties and became warped, which placed each joint off-centre and made the collisions between each body a lot slower to calculate within the physics engine. In order to find the homogeneous transformation matrices, the Denavit-Hartenburg parameters need to be established. However, the dimensions of the hand are not specified. This can still be approximated for each finger to find its homogeneous transformation matrix, as each finger is essentially a 3-linked robot with an end effector, and the measurements can be taken directly from the model in CoppeliaSim.

Each finger will be approximated to follow the Denavit-Hartenberg parameters[197]:

-The z-axis must be the axis of rotation or direction of motion.

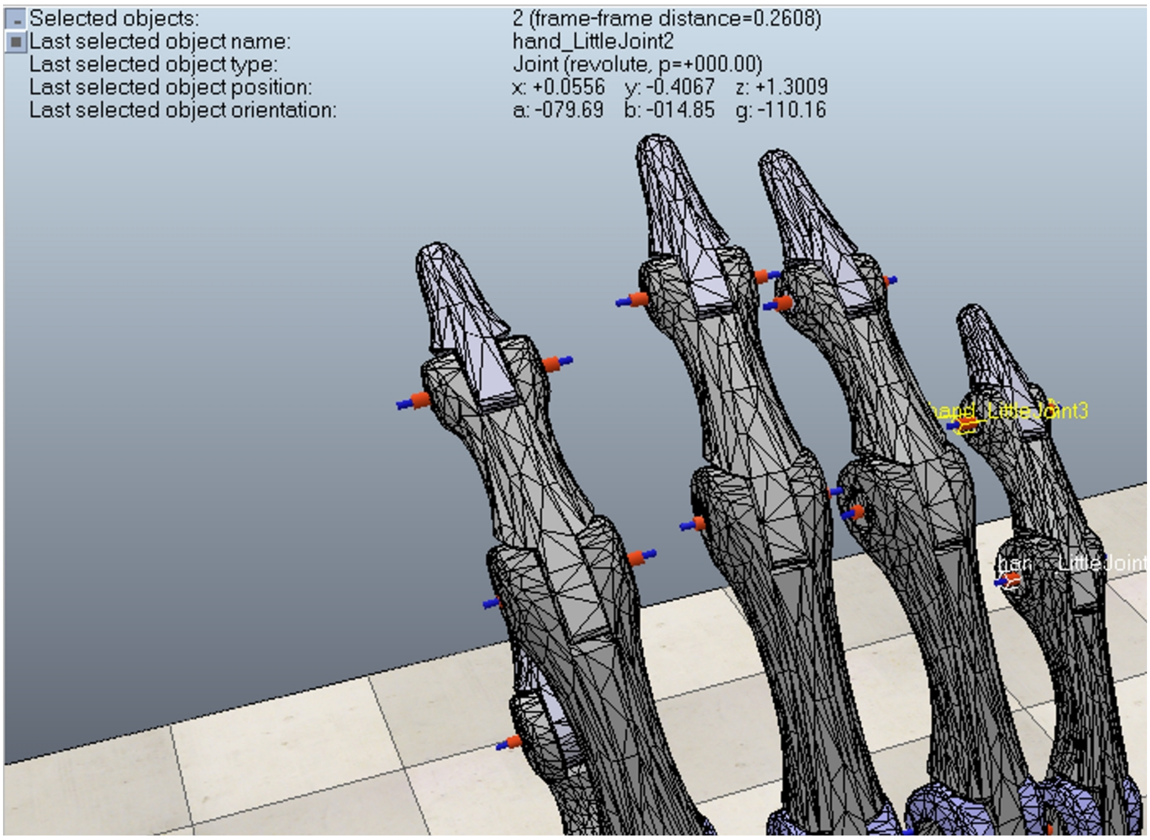
-The x_n-axis must be perpendicular to both the z_n and z_{n-1} axes.

-The x_n-axis must intersect both the z_n and z_{n-1} axes.

-The y_n axis' coordinate frame must follow the right-hand rule.

Each joint in their respective fingers was aligned with each other using CoppeliaSim, the z-axis of each coordinate frame is already aligned and therefore alpha and z-axis displacement 'd' will always be zero. Theta denotes the angle of rotation through the z axis, and displacement 'a' represents the distance between links. The thumb is the only negative value for theta as the thumb needs to rotate in the opposite direction (clockwise) to the other fingers. Each coordinate frame is linked to the next by a rotation through z between 0-63 degrees. A rotation in the z axis is represented by the matrix:

$$R_z = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, where\theta = (-)63^{\circ} \tag{7.5.1}$$



```
Selected objects:                  2 (frame-frame distance=0.2608)
Last selected object name:         hand_LittleJoint2
Last selected object type:         Joint (revolute, p=+000.00)
Last selected object position:     x: +0.0556   y: -0.4067   z: +1.3009
Last selected object orientation:  a: -079.69   b: -014.85   g: -110.16
```

Figure 7.6: The information text in the top left shows the distance between two joints, as well as the difference in orientation and position. This can be used to fill out an approximate Denavit-Hartenberg table.

To approximate further, each translation occurs along the y-axis, when in the simulation it is actually a combination of displacements in the x and y axes. The transformation matrix is given by:

$$T = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, where\ y = displacement \tag{7.5.2}$$

Therefore, the homogeneous transformation matrix between one coordinate frame and the next would be represented by:

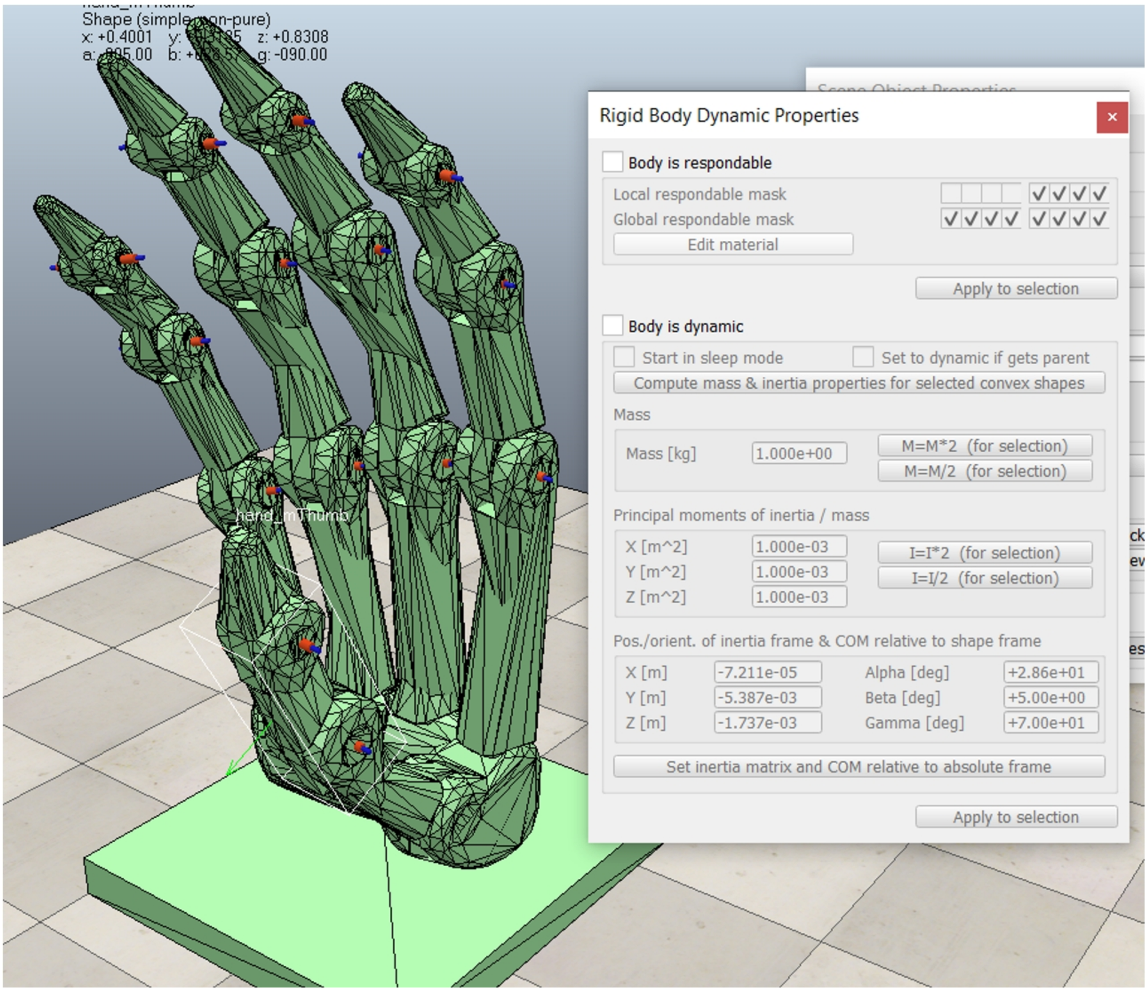
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{7.5.3}$$

$$H_n^{n-1} = \begin{bmatrix} cos\theta & -sin\theta & 0 & x \\ sin\theta & cos\theta & 0 & y \\ 0 & 0 & 0 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.5.4}$$

Filling the values in from the table for each finger would give the total homogeneous transformation matrix, used in forward and inverse kinematics.

7.6 Testing and Analysis

It was found in the test that once one finger was functioning correctly, the same code was used to read the joint angles of each finger, so that the fingers can operate together. The main problem was the speed at which the simulation was occurring, coupled with a constantly vibrating ring finger. To speed up the simulation, a load must be taken off the physics engine by decreasing the number of calculations it has to make. To do this, alternate links in the robot can be set to different mask levels within the simulation so that they do not collide. Also, an invisible and simplified version of the model was created and attached to the visible model, so that the simplified version could run faster, whilst the original can visually copy it.



Rigid Body Dynamic Properties

☐ Body is respondable

Local respondable mask ☐☐☐☐ ☑☑☑☑

Global respondable mask ☑☑☑☑ ☑☑☑☑

☐ Body is dynamic

☐ Start in sleep mode ☐ Set to dynamic if gets parent

Mass

Mass [kg] 1.000e+00

Principal moments of inertia / mass

X [m^2] 1.000e-03

Y [m^2] 1.000e-03

Z [m^2] 1.000e-03

Pos./orient. of inertia frame & COM relative to shape frame

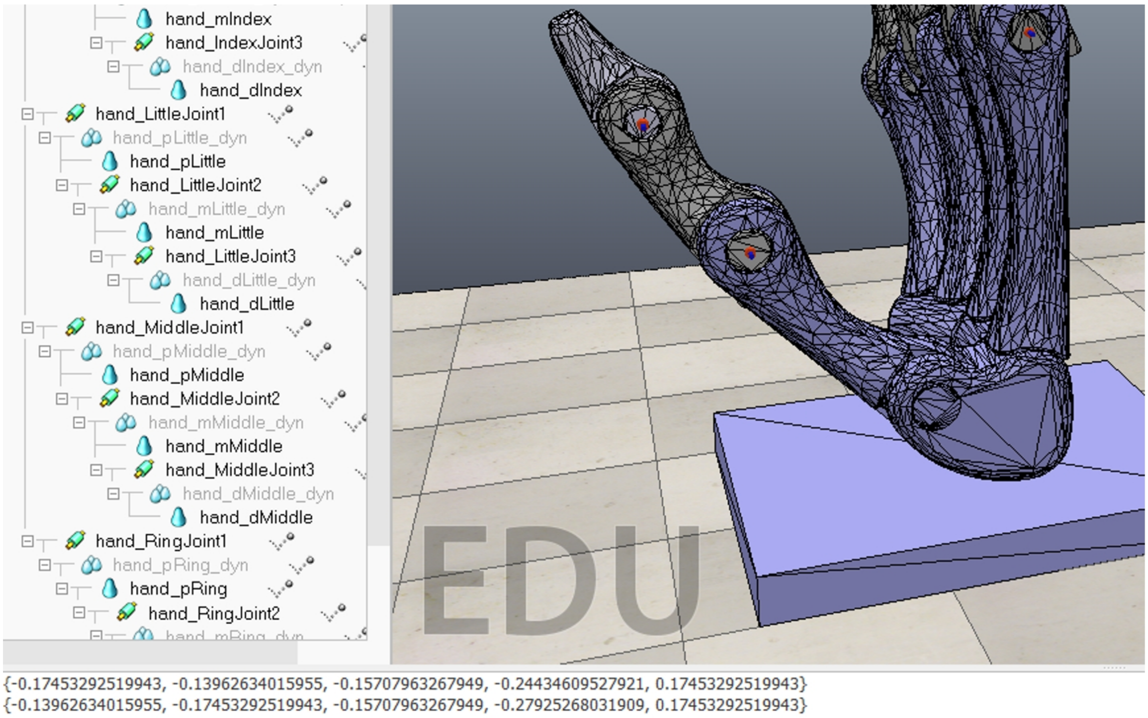
X [m]	-7.211e-05	Alpha [deg]	+2.86e+01
Y [m]	-5.387e-03	Beta [deg]	+5.00e+00
Z [m]	-1.737e-03	Gamma [deg]	+7.00e+01

Figure 7.7: Simplified version of model to speed up the physics engine. The mask layers indicate which objects the individual body will collide with.

The green hand shown in Figure 7.7 is a simplified mesh version of the original hand. The local respondable layers show that the base of the thumb does not interact with anything on the first 4 mask layer boxes, i.e., the tip of the thumb and the base. The segments of the fingers alternate in masks so that adjacent parts do not collide but anything past that will, which is an effective way of both speeding up the simulation and fixing the issue of parts colliding with themselves.

With the simulation working smoothly with the Hiwonder hand, the prototype hand could be constructed. The index finger was completed first as a test for how fixing the sensors in place would work. It was found that fixing them at the tip and letting the back slide freely was the optimal solution for bending each sensor fully whilst not obstructing the clinch connector.

Putting a voltmeter in parallel with the output showed the voltage fluctuating around



```
{-0.17453292519943, -0.13962634015955, -0.15707963267949, -0.24434609527921, 0.17453292519943}
{-0.13962634015955, -0.17453292519943, -0.15707963267949, -0.27925268031909, 0.17453292519943}
```

Figure 7.8: The joint angles being printed to the CoppeliaSim output. They have been converted to radians by the simulation.

1.2V when relaxed, slightly under the number predicted earlier. This dropped to around 400mV when flexed. The glove can now be plugged into the simulation with the same code as with the Hiwonder glove. The values used within the code that convert and calibrate the float values into joint angles must be changed as well. A flat value of the minimum reading was subtracted from the resistance read from a finger, to give a number in degrees between 0 and approximately 70. This was then converted to radians to set the joint angle targets for each motor in the joints.

The project has proven successful in controlling a simulated anthropomorphic hand using two separate data gloves. The model used for the hand was very complex with a large mesh, which caused the simulation to run slower, and made the importation process more difficult. Investing more time into making a new and simpler model in CAD would have decreased the number of simulation-based issues later on. This also would have fixed the misalignment of the joints and parts of the hand interacting with each other in unexpected ways. The jitter experienced in the hand is also partially due to the rate of change of joint angles. Currently, the joint angles are sent at a rate of 16Mhz and are subject to slight changes, which causes the hand to vibrate slightly at times. Taking a running average of the joint angles coming in would have decreased their variation and made the operation smoother.

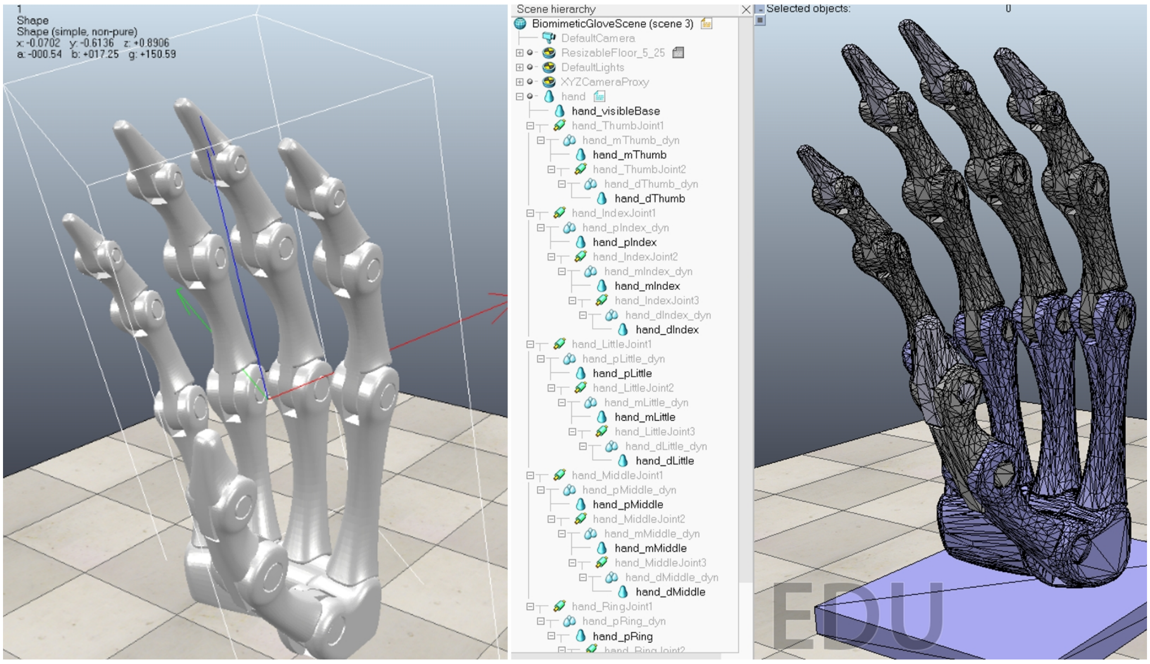


Figure 7.9: Side by side comparison of hand before and after mesh decimation. This is so the simulation can run faster with a simpler physics model. The right side shows the hand after having been split into separate links and bodies.

The data gloves and simulation are used together to find the joint angle data and convert it to a usable form. This would also allow the gloves to be more compatible with other systems without much modification. Moreover, storing the joint angles in an array allows for a recreation of the simulation, which would be helpful for teaching an AI to copy a task or replaying the users' actions.

Furthermore, the function used to move the joints to the target angle works in such a way that does not allow for feedback and optimisation, by simply telling the simulation to move the joint. It is less accurate compared to what would be used in a real-world system, where a target position would be set and a Proportional-Integral-Derivative controller would be used to adjust the response to a movement. The torque of the motors would also be a function of the motors' current, instead of moving to the target angle arbitrarily like the function being used does.

Figure 7.10: The completed Data Glove for Simulation.

7.7 Conclusion

The project's goal was met and demonstrated the key features of what forms a data glove. Different components and methods of data acquisition were explored before choosing suitable ones for a prototype glove that was made. The model was used for the simulation of an anthropomorphic robotic hand and set up for simulated teleoperation. This was then interfaced with a premade Hiwonder data glove, which proved effective in its operation. Using the knowledge gained from setting up and testing both the Hiwonder data glove and simulation, the prototype glove was produced and used in conjunction with the same simulation.

Additionally, the project was an opportunity to learn more about current industry applications and conventions. It is estimated that the addition of features such as position and orientation data, battery life and wireless connectivity, and haptic feedback could be implemented for a low cost, allowing the glove to compete on the market.

The prototype as of this moment is a functional demonstration of the basis of data gloves, with much room for enhancements and improvement. For example, increase the sensor in the glove, which can provide more degrees of freedom to increase the range of motion that the glove can capture; Rumble motors or contacts that provide a minute electric shock could be installed on the inner layer or the front of the glove that stimulate the fingers and trigger a haptic response, which is useful in situations where tactile feedback is required.

Chapter 8

Conclusion and Prospect

8.1 Summary of Thesis and Contributions

This thesis is dedicated to developing and implementing an intuitive HMI system to improve the interaction efficiency and experience between users and machines by combining the latest neural network algorithms, ELM technology, and fault-tolerant algorithms. The following is a detailed overview of the main contents of this study and its contributions:

Application of neural networks in graphic classification: Through an in-depth analysis of the relevant literature on “target-unbiased meta-learning for graphic classification”, we studied the performance of neural networks when processing graphic optical imaging data and proposed an improvement strategy to optimise its performance. This includes the introduction of an interactive attention extraction module to reduce the model’s preference for seen targets, thereby improving classification accuracy.

Application of extreme learning machines in robot trajectory prediction: Based on the study of “BAS-optimised ELM for KUKA iiwa robot learning”, we explored how to use extreme learning machines for robot trajectory prediction and improved the prediction accuracy through optimisation algorithms. Experimental results show that the effectiveness of the proposed method on the KUKA iiwa robot has been verified.

Research on fault-tolerant algorithms: The theoretical basis and practical application of sparsity-based fault-tolerant algorithms in redundant robot operations are explored. The study found that in the case of multiple joint failures, the normal movement of the end effector can be effectively maintained by properly designing a fault-tolerant mechanism.

Design and implementation of HMI system: In order to address the limitations of existing interactive systems, this thesis also focuses on building a more natural and intuitive operation interface. For example, for the design of smart gloves, not only cost-effectiveness is considered, but also sufficient flexibility and accuracy are ensured to support gesture recognition with more degrees of freedom and tactile feedback in the future.

These research results not only promote the academic development of this field, but also provide new possibilities for practical applications, especially in scenarios such as medical surgery assistance and industrial automation.

8.2 Proposals for Future Works

Building on the existing work, the next step is to integrate the robotic arm and hand and implement real-time control using the MYO armband and data gloves. This approach aims to enhance real-time control capabilities and optimise the system's functionality, leveraging prior advancements effectively. At the same time, the robot manipulator can also provide real-time haptic feedback.

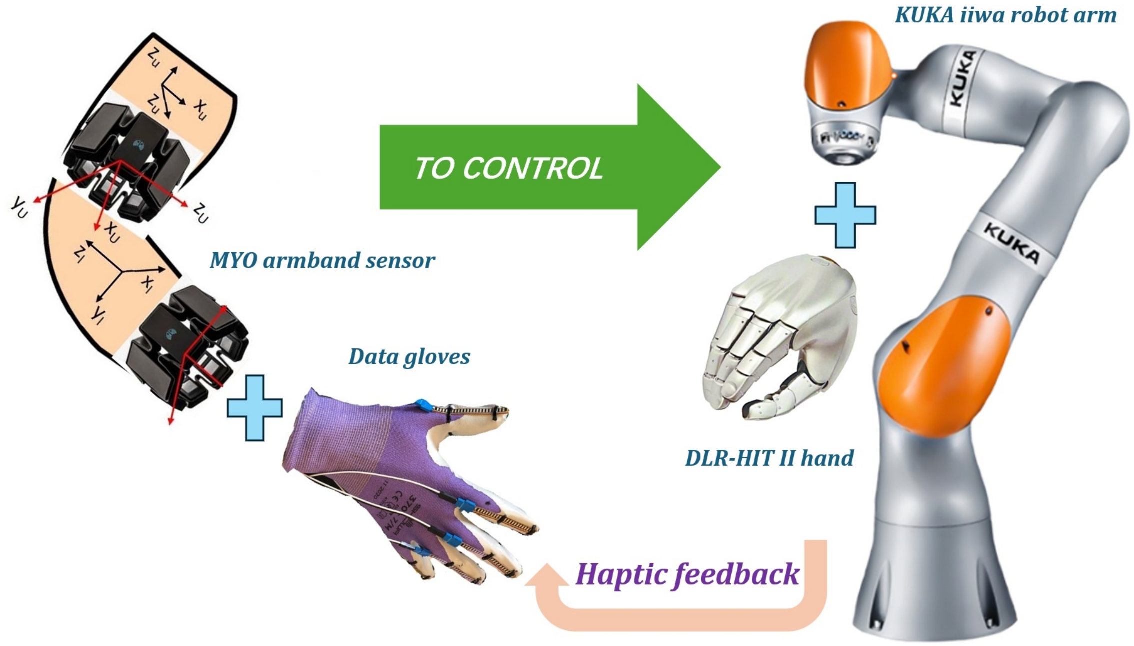


Figure 8.1: Use the MYO armband and data gloves to control KUKA iiwa robot arm and DLR-HIT II hand.

In order to further improve and develop the various technologies proposed in this study, the following are some specific and detailed suggestions for future work:

Enhance image processing capabilities: develop a more robust droplet detection algorithm that can operate stably under different lighting conditions. The environmental adaptability of the algorithm can be improved by increasing the diversity of training samples or using adaptive threshold adjustment. Explore the use of deep learning frameworks (such as CNNs) to automatically extract features to further improve the accuracy and speed of droplet detection.

Deepen the application of extreme learning machines: Combine extreme learning machines with other advanced learning algorithms, such as ensemble learning methods, to achieve better prediction results. Expand to other types of robot platforms to test and verify the versatility and practicality of the proposed method in different application scenarios.

Optimise fault-tolerant mechanisms: Design more complex fault scenario simulation experiments to evaluate the performance of current fault-tolerant algorithms under extreme conditions and make corresponding improvements accordingly. Consider integrating artificial intelligence technology into fault-tolerant strategies so that the system can dynamically adjust its configuration to cope with emergencies.

Expand the functions of HMI interfaces: Add additional sensor types, such as electromyography sensors, to the existing ones to capture more subtle hand movements. Achieve true two-way communication, which allows users to not only control the device, but also receive information from the device and provide real-time feedback to the user. Research how to integrate multiple input modes (such as voice, gestures, eye tracking, etc.) to create a multimodal interaction system to provide users with a richer and more natural experience.

Interdisciplinary cooperation: Strengthen exchanges and cooperation with fields such as psychology and cognitive science to better understand human behaviour patterns and guide interface design principles. Work with material science experts to develop new sensing materials to create more possibilities for future wearable devices.

In short, with the advancement of technology, the field of HMI will continue to evolve. Through continuous technological innovation and interdisciplinary cooperation, we can expect to see a new generation of interactive systems that are more efficient, friendly and widely applicable.

Bibliography

[1] M. Pantic and L. J. Rothkrantz, *Toward an affect-sensitive multimodal human-computer interaction*, *Proceedings of the IEEE* **91** (2003), no. 9 1370–1390.

[2] D. Akoumianakis and C. Stephanidis, *Universal design in hci: A critical review of current research and practice*, *Engineering and Construction* **754** (1989).

[3] P. M. Mullins and S. Treu, *A task-based cognitive model for user-network interaction: defining a task taxonomy to guide the interface designer*, *Interacting with Computers* **5** (1993), no. 2 139–166.

[4] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-computer interaction.* Addison-Wesley Longman Ltd., 1994.

[5] B. Kernighan and D. Ritchie, *The C Programming Language: 2nd Edition.* How to Make Fast Money on Amazon. Prentice Hall, 1988.

[6] J. Nielsen, *Iterative user-interface design*, *Computer* (1993), no. 11 32–41.

[7] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum, *Reality-based interaction: a framework for post-wimp interfaces*, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 201–210, 2008.

[8] A. Malizia and A. Bellucci, *The artificiality of natural user interfaces*, *Communications of the ACM* (2012), no. 3 36–38.

[9] P. H. Lindsay and D. A. Norman, *Human information processing: An introduction to psychology.* Academic press, 2013.

[10] B. Reeves and C. Nass, *The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places.* Cambridge University Press, New York, 1996.

[11] J. Sweller, *Cognitive load during problem solving: Effects on learning*, *Cognitive science* (1988), no. 2 257–285.

[12] I. Kirsch, *Response expectancy theory and application: A decennial review*, *Applied & Preventive Psychology* **6** (1997), no. 2 69–79.

[13] J. M. Carroll and M. B. Rosson, *Paradox of the active user*, in *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction* (J. M. Carroll, ed.), pp. 80–111. The MIT Press, 1987.

[14] S. Alowais, S. Alghamdi, N. Alsuhebany, and et al., *Revolutionizing healthcare: the role of artificial intelligence in clinical practice*, *BMC Medical Education* **23** (2023), no. 689.

[15] J. Schreiter, F. Heinrich, B. Hatscher, and et al., *Multimodal human–computer interaction in interventional radiology and surgery: a systematic literature review*, *International Journal of Computer Assisted Radiology and Surgery* (2024).

[16] O. Ziaee and M. Hamedi, *Augmented reality applications in manufacturing and its future scope in industry 4.0*, 2021.

[17] L. Elhattab, J. Khairalla, R. Al-Attar, S. Albert, N. Shorim, and E. Eliwa, *Augmented reality applications in the automotive industry*, in *2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 357–364, 2023.

[18] K. Dautenhahn, *Socially intelligent robots: Dimensions of human-robot interaction*, in *Social intelligence: From brain to culture* (N. Emery, N. Clayton, and C. Frith, eds.), pp. 313–351. Oxford University Press, 2007.

[19] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, *An affective guide robot in a shopping mall*, in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 173–180, 2009.

[20] A. Graves, A.-r. Mohamed, and G. Hinton, *Speech recognition with deep recurrent neural networks*, in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee, 2013.

[21] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., *Deep speech 2: End-to-end speech recognition in english and mandarin*, in *International conference on machine learning*, pp. 173–182, PMLR, 2016.

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

[23] J. Redmon, *You only look once: Unified, real-time object detection*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[24] R. T. Azuma, *A survey of augmented reality*, *Presence: Teleoperators and Virtual Environments/MIT press* (1997).

[25] M. Slater, *Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments*, *Philosophical Transactions of the Royal Society B: Biological Sciences* **364** (December, 2009) 3549–3557.

[26] H. Culbertson, S. B. Schorr, and A. M. Okamura, *Haptics: The present and future of artificial touch sensation*, *Annual review of control, robotics, and autonomous systems* **1** (2018), no. 1 385–409.

[27] S. Choi and K. J. Kuchenbecker, *Vibrotactile display: Perception, technology, and applications*, *Proceedings of the IEEE* **101** (2012), no. 9 2093–2104.

[28] A. Graves, A.-r. Mohamed, and G. Hinton, *Speech recognition with deep recurrent neural networks*, in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee, 2013.

[29] N. Krahnstoever, S. Kettebekov, M. Yeasin, and R. Sharma, *A real-time framework for natural multimodal interaction with large screen displays*, in *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, pp. 349–354, 2002.

[30] S. Hurtado, *The next generation of diversity and intergroup relations research*, *Journal of Social Issues* **61** (2005), no. 3 595–610.

[31] R. Fiebrink and P. Cook, *The wekinator: A system for real-time, interactive machine learning in music*, *Proceedings of The Eleventh International Society for Music Information Retrieval Conference (ISMIR 2010)* (01, 2010).

[32] W. Xu, M. J. Dainoff, L. Ge, and Z. Gao, *Transitioning to human interaction with ai systems: New challenges and opportunities for hci professionals to enable human-centered ai*, 2023.

[33] Y. Shi and G. Shen, *Haptic sensing and feedback techniques toward virtual reality*, *Research* **7** (2024) 0333, [`https://spj.science.org/doi/pdf/10.34133/research.0333`].

[34] A. R. See, J. A. G. Choco, and K. Chandramohan, *Touch, texture and haptic feedback: A review on how we feel the world around us*, *Applied Sciences* **12** (2022), no. 9.

[35] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, *The future of the human–machine interface (hmi) in society 5.0*, *Future Internet* **15** (2023), no. 5.

[36] J. D. Azofeifa, J. Noguez, S. Ruiz, J. M. Molina-Espinosa, A. J. Magana, and B. Benes, *Systematic review of multimodal human–computer interaction*, *Informatics* **9** (2022), no. 1.

[37] M. Schellekens, *Human–machine interaction in self-driving vehicles: a perspective on product liability*, *International Journal of Law and Information Technology* **30** (05, 2022) 233–248.

[38] J. Aggarwal and M. Ryoo, *Human activity analysis: A review*, *ACM Comput. Surv.* **43** (Apr., 2011).

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[40] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, *Openpose: Realtime multi-person 2d pose estimation using part affinity fields*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43** (2021), no. 1 172–186.

[41] T. MAENO, K. KOBAYASHI, and N. YAMAZAKI, *Relationship between the structure of human finger tissue and the location of tactile receptors*, *JSME International Journal Series C* **41** (1998), no. 1 94–100.

[42] M. Bouzit, G. Burdea, G. Popescu, and R. Boian, *The rutgers master ii-new design force-feedback glove*, *IEEE/ASME Transactions on Mechatronics* **7** (2002), no. 2 256–263.

[43] A. Graves, A. rahman Mohamed, and G. Hinton, *Speech recognition with deep recurrent neural networks*, 2013.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, 2023.

[45] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S.-y. Chang, K. Rao, and A. Gruenstein, *Streaming end-to-end speech recognition for mobile devices*, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6381–6385, 2019.

[46] F. Cincotti, D. Mattia, C. Babiloni, F. Carducci, L. Bianchi, del R. Jose Millan, J. Mouriño, S. Salinari, M. G. Marciani, and F. Babiloni, *Classification of eeg mental patterns by using two scalp electrodes and mahalanobis distance-based classifiers*, *Methods of Information in Medicine* **41** (2002) 337 – 341.

[47] L. F. Nicolas-Alonso and J. Gomez-Gil, *Brain computer interfaces, a review*, *Sensors (Basel)* **12** (2012), no. 2 1211–1279.

[48] M. Duvinage, T. Castermans, M. Petieau, T. Hoellinger, G. Cheron, and T. Dutoit, *Performance of the emotiv epoc headset for p300-based applications*, *Biomedical engineering online* **12** (June, 2013) 56.

[49] A. Bora, *Exploring robot trajectory planning – a comparative analysis of algorithms and software implementations in dynamic environments*, 2024.

[50] Y. Dai, C. Xiang, Y. Zhang, Y. Jiang, W. Qu, and Q. Zhang, *A review of spatial robotic arm trajectory planning*, *Aerospace* **9** (2022), no. 7.

[51] G. Frediani, D. Mazzei, D. E. De Rossi, and F. Carpi, *Wearable wireless tactile display for virtual interactions with soft bodies*, *Frontiers in bioengineering and biotechnology* **2** (2014) 31.

[52] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, *Continual lifelong learning with neural networks: A review*, *Neural Networks* **113** (2019) 54–71.

[53] T. Talaei Khoei, H. Ould Slimane, and N. Kaabouch, *Deep learning: Systematic review, models, challenges, and research directions*, *Neural Computing and Applications* **35** (2023) 23103–23124.

[54] R. Wang, D. Zhao, D. Suh, Z. Yuan, G. Chen, and B.-C. Min, *Personalization in human-robot interaction through preference-based action representation learning*, 2024.

[55] R. C. Gonzalez, R. E. Woods, and B. R. Masters, *Digital image processing, third edition*, *Journal of Biomedical Optics* **14** (2009), no. 2 029901.

[56] A. Buades, B. Coll, and J.-M. Morel, *A non-local algorithm for image denoising*, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 60–65 vol. 2, 2005.

[57] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features*, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–I, 2001.

[58] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015.

[59] J. Canny, *A computational approach to edge detection*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1986), no. 6 679–698.

[60] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, *International Journal of Computer Vision* **60** (2004) 91–110.

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, *Commun. ACM* **60** (May, 2017) 84–90.

[62] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39** (2017), no. 6 1137–1149.

[63] J. Heaton, *Ian goodfellow, yoshua bengio, and aaron courville: Deep learning*, *Genetic Programming and Evolvable Machines* **19** (2018) 305–307.

[64] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*, *Nature* **323** (1986) 533–536.

[65] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, *The graph neural network model*, *IEEE Transactions on Neural Networks* **20** (2009), no. 1 61–80.

[66] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, 2017.

[67] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018.

[68] C. Finn, P. Abbeel, and S. Levine, *Model-agnostic meta-learning for fast adaptation of deep networks*, 2017.

[69] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, *Extreme learning machine: Theory and applications*, *Neurocomputing* **70** (2006), no. 1 489–501. Neural Networks.

[70] G.-B. Huang, D. Wang, and Y. Lan, *Extreme learning machines: A survey*, *International Journal of Machine Learning and Cybernetics* **2** (2011) 107–122.

[71] S. Levine, C. Finn, T. Darrell, and P. Abbeel, *End-to-end training of deep visuomotor policies*, 2016.

[72] A. T. Vo and H.-J. Kang, *A novel fault-tolerant control method for robot manipulators based on non-singular fast terminal sliding mode control and disturbance observer*, *IEEE Access* **8** (2020) 109388–109400.

[73] Z. Wang, L. Liu, and H. Zhang, *Neural network-based model-free adaptive fault-tolerant control for discrete-time nonlinear systems with sensor fault*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47** (2017), no. 8 2351–2362.

[74] Z. Li, C. Li, S. Li, and X. Cao, *A fault-tolerant method for motion planning of industrial redundant manipulator*, *IEEE transactions on industrial informatics* **16** (2019), no. 12 7469–7478.

[75] *Lbr iiwa— kuka ag*, 2024.

[76] Toni-SM, "libiiwa: Scalable cross-platform multi-control framework for kuka lbr iiwa cobots (python - ros - ros2)." `https://github.com/Toni-SM/libiiwa`, 2024.

[77] R. Diankov, *Automated construction of robotic manipulation programs.* PhD thesis, Carnegie Mellon University, USA, 2010. AAI3448143.

[78] W. Samek, T. Wiegand, and K.-R. Müller, *Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models*, 2017.

[79] N. Garti and C. Bisperink, *Double emulsions: progress and applications*, *Current opinion in colloid & interface science* **3** (1998), no. 6 657–667.

[80] D. Chong, X. Liu, H. Ma, G. Huang, Y. L. Han, X. Cui, J. Yan, and F. Xu, *Advances in fabricating double-emulsion droplets and their biomedical applications*, *Microfluidics and Nanofluidics* **19** (2015), no. 5 1071–1090.

[81] G. Xin, C. Ke, and H. Xiaoguang, *An improved canny edge detection algorithm for color image*, in *IEEE 10th International Conference on Industrial Informatics*, pp. 113–117, 2012.

[82] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2016.

[83] P. Bao, Lei Zhang, and Xiaolin Wu, *Canny edge detection enhancement by scale multiplication*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27** (2005), no. 9 1485–1490.

[84] C. Zhan, X. Duan, S. Xu, Z. Song, and M. Luo, *An improved moving object detection algorithm based on frame difference and edge detection*, in *Fourth International Conference on Image and Graphics (ICIG 2007)*, pp. 519–523, 2007.

[85] S. Mallick, *Blob Detection Using OpenCV (Python, C++) — Learn OpenCV*, 2015.

[86] W. Rong, Z. Li, W. Zhang, and L. Sun, *An improved canny edge detection algorithm*, in *2014 IEEE International Conference on Mechatronics and Automation*, pp. 577–582, 2014.

[87] Z. Xu, X. Baojie, and W. Guoxin, *Canny edge detection based on open cv*, in *2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, pp. 53–56, 2017.

[88] Z. Chen, *How to understand YUV*, 2019.

[89] Z. H. Al-Tairi, R. Rahmat, M. I. Saripan, and P. S. Sulaiman, *Skin segmentation using yuv and rgb color spaces*, *J. Inf. Process. Syst.* **10** (2014) 283–299.

[90] B. Gordon, N. Chaddha, and T.-Y. Meng, *A low-power multiplierless yuv to rgb converter based on human vision perception*, in *Proceedings of 1994 IEEE Workshop on VLSI Signal Processing*, pp. 408–417, 1994.

[91] “OpenCV: Introduction.”

[92] B. Smith, *Hough Circle Transform - ImageJ*, 2017.

[93] R. Grycuk, M. Gabryel, M. Korytkowski, R. Scherer, and S. Voloshynovskiy, *From single image to list of objects based on edge and blob detection*, in *Artificial Intelligence and Soft Computing* (L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, eds.), (Cham), pp. 605–615, Springer International Publishing, 2014.

[94] H. Su and X. Yuan, *An Improved Canny Edge Detection Algorithm*, *Computer Simulation* (2010).

[95] H. Xu, H. Qin, and Y. Chen, *An Improved Algorithm for Edge Detection Based on Canny*, *Infrared Technology* (2014).

[96] Wenshuo Gao, Xiaoguang Zhang, Lei Yang, and Huizhong Liu, *An improved sobel edge detection*, in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 5, pp. 67–71, 2010.

[97] K. Beant and A. Garg, *Comparative study of different edge detection techniques*, .

[98] E. Davies, *A modified hough scheme for general circle location*, *Pattern Recognition Letters* **7** (1988), no. 1 37–43.

[99] H. Ye, G. Shang, L. Wang, and M. Zheng, *A new method based on hough transform for quick line and circle detection*, in *2015 8th International Conference on Biomedical Engineering and Informatics (BMEI)*, pp. 52–56, 2015.

[100] C. Li, C. Yang, and C. Giannetti, *Segmentation and generalization for writing skills transfer from humans to robots*, *Network Computation in Neural Systems* **1** (01, 2019) 20–25.

[101] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, *Geometric deep learning: Going beyond euclidean data*, *IEEE Signal Processing Magazine* **34** (2017), no. 4 18–42.

[102] Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei, *Memory matching networks for one-shot image recognition*, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4080–4088, 2018.

[103] W. Deng, J. Xu, X.-Z. Gao, and H. Zhao, *An enhanced msiqde algorithm with novel multiple strategies for global optimization problems*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **52** (2022), no. 3 1578–1587.

[104] W. Deng, J. Xu, Y. Song, and H. Zhao, *Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem*, *Applied Soft Computing* **100** (2021), no. 3.

[105] B. Pfahringer, H. Bensusan, and C. G. Giraud-Carrier, *Meta-learning by landmarking various learning algorithms.*, in *ICML*, pp. 743–750, Citeseer, 2000.

[106] X. Zhu and A. B. Goldberg, *Introduction to semi-supervised learning.* Springer Nature, 2022.

[107] T. Joachims et al., *Transductive inference for text classification using support vector machines*, in *Icml*, vol. 99, pp. 200–209, 1999.

[108] S. L. Chelsea Finn, Pieter Abbeel, *Model-agnostic meta-learning for fast adaptation of deep networks*, in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, PMLR, 2017.

[109] A. D. Nichol, M. J. Holle, and R. An, *Glycemic impact of non-nutritive sweeteners: a systematic review and meta-analysis of randomized controlled trials*, *European journal of clinical nutrition* **72** (2018), no. 6 796–804.

[110] M. A. Jamal and G.-J. Qi, *Task agnostic meta-learning for few-shot learning*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11719–11727, 2019.

[111] C.-G. Guo, K. S. Cheung, F. Zhang, E. W. Chan, L. Chen, I. C. Wong, and W. K. Leung, *Delay in retreatment of helicobacter pylori infection increases risk of upper gastrointestinal bleeding*, *Clinical Gastroenterology and Hepatology* **19** (2021), no. 2 314–322.

[112] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, *Learning to compare: Relation network for few-shot learning*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

[113] A. J. Koch, S. D. D'Mello, and P. R. Sackett, *A meta-analysis of gender stereotypes and bias in experimental simulations of employment decision making.*, *Journal of applied psychology* **100** (2015), no. 1 128.

[114] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., *Matching networks for one shot learning*, *Advances in neural information processing systems* **29** (2016).

[115] J. Snell, K. Swersky, and R. Zemel, *Prototypical networks for few-shot learning*, *Advances in neural information processing systems* **30** (2017).

[116] T. Munkhdalai and H. Yu, *Meta networks*, in *International conference on machine learning*, pp. 2554–2563, PMLR, 2017.

[117] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, *Geometric deep learning: going beyond euclidean data*, *IEEE Signal Processing Magazine* **34** (2017), no. 4 18–42.

[118] V. Garcia and J. Bruna, *Few-shot learning with graph neural networks*, 2018.

[119] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018.

[120] Z. Zhang, Y. Zhao, J. Liu, S. Wang, R. Tao, R. Xin, and J. Zhang, *A general deep learning framework for network reconstruction and dynamics learning*, *Applied Network Science* **4** (2019) 1–17.

[121] C. Luo and A. Yuille, *Grouped spatial-temporal aggregation for efficient action recognition*, 2019.

[122] C. Luo and A. L. Yuille, *Grouped spatial-temporal aggregation for efficient action recognition*, in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5512–5521, 2019.

[123] G. Lin, Y. Yang, Y. Fan, X. Kang, K. Liao, and F. Zhao, *High-order structure preserving graph neural network for few-shot learning*, *arXiv preprint arXiv:2005.14415* (2020).

[124] K. Ding, J. Wang, J. Li, K. Shu, C. Liu, and H. Liu, *Graph prototypical networks for few-shot learning on attributed networks*, 2020.

[125] N. Ma, J. Bu, J. Yang, Z. Zhang, C. Yao, Z. Yu, S. Zhou, and X. Yan, *Adaptive-step graph meta-learner for few-shot graph classification*, 2020.

[126] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, *Learning to propagate labels: Transductive propagation network for few-shot learning*, *arXiv preprint arXiv:1805.10002* (2018).

[127] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, *Meta-learning with memory-augmented neural networks*, in *International conference on machine learning*, pp. 1842–1850, PMLR, 2016.

[128] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, *A simple neural attentive meta-learner*, *arXiv preprint arXiv:1707.03141* (2017).

[129] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, et al., *Meta-dataset: A dataset of datasets for learning to learn from few examples*, *arXiv preprint arXiv:1903.03096* (2019).

[130] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, *Meta-learning for semi-supervised few-shot classification*, *arXiv preprint arXiv:1803.00676* (2018).

[131] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., *Matching networks for one shot learning*, *Advances in neural information processing systems* **29** (2016).

[132] V. Garcia and J. Bruna, *Few-shot learning with graph neural networks*, 2018.

[133] J. Kim, T. Kim, S. Kim, and C. D. Yoo, *Edge-labeling graph neural network for few-shot learning*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11–20, 2019.

[134] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, *Learning to propagate labels: Transductive propagation network for few-shot learning*, 2019.

[135] C. Li, C. Yang, and C. Giannetti, *Segmentation and generalisation for writing skills transfer from humans to robots*, *Cogn. Comput. Syst.* (2019), no. 1 20–25.

[136] H. T. Huynh, Y. Won, and J.-J. Kim, *An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks*, *Int. J. Neural Syst.* (2008), no. 5 433–441.

[137] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, *Extreme learning machine: Theory and applications*, *Neurocomputing* (2006), no. 1 489–501.

[138] G.-B. Huang and L. Chen, *Enhanced random search based incremental extreme learning machine*, *Neurocomputing* (2008), no. 16 3460–3468.

[139] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, *Online sequential fuzzy extreme learning machine for function approximation and classification problems*, *IEEE Trans. Syst., Man, Cybern. B. Cybern.* (Mar., 2009) 1067–1072.

[140] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, *Fully complex extreme learning machine*, *Neurocomputing* (May, 2005) 306–314.

[141] B. P. Chacko, V. V. Krishnan, G. Raju, and P. B. Anto, *Handwritten character recognition using wavelet energy and extreme learning machine*, *Int. J. Mach. Learn. Cybern.* (2012), no. 2 149–161.

[142] Q. Yang, H. Gao, W. Zhang, Z. Chi, and Z. Yi, *A new data-driven modeling method for fermentation processes*, *Chemometrics Intell. Lab. Syst.* (Mar., 2016) 88–96.

[143] Y. Cheng, C. Li, S. Li, and Z. Li, *Motion planning of redundant manipulator with variable joint velocity limit based on beetle antennae search algorithm*, *IEEE Access* (2020) 138788–138799.

[144] K. Ishaque, Z. Salam, M. Amjad, and S. Mekhilef, *An improved particle swarm optimization (pso)-based mppt for pv with reduced steady-state oscillation*, *IEEE Trans. Power Electron.* (Jan., 2012) 3627–3638.

[145] X. Jiang and S. Li, *Beetle antennae search without parameter tuning (bas-wpt) for multi-objective optimization*, 2017. Available: arXiv:1711.02395.

[146] C. Li, C. Yang, Z. Ju, and A. S. Annamalai, *An enhanced teaching interface for a robot using dmp and gmr*, *Int. J. Intell. Robot. Appl.* (2018), no. 1 110–121.

[147] Y. Xu, C. Yang, P. Liang, L. Zhao, and Z. Li, *Development of a hybrid motion capture method using myo armband with application to teleoperation*, in *Proc. IEEE Int. Conf. Mechatron. Autom.*, pp. 1179–1184, 2016.

[148] G.-B. Huang, L. Chen, and C. K. Siew, *Universal approximation using incremental constructive feedforward networks with random hidden nodes*, *IEEE Trans. Neural Netw.* (Jan., 2006) 879–892.

[149] X. Jiang and S. Li, *Bas: Beetle antennae search algorithm for optimization problems*, 2017.

[150] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, *Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017*, *Advanced Robotics* **33** (2019), no. 15-16 764–799.

[151] M. Hägele, K. Nilsson, J. N. Pires, and R. Bischoff, *Industrial robotics*, *Springer handbook of robotics* (2016) 1385–1422.

[152] M. Ronzoni, R. Accorsi, L. Botti, and R. Manzini, *A support-design framework for cooperative robots systems in labor-intensive manufacturing processes*, *Journal of Manufacturing Systems* **61** (2021) 646–657.

[153] M. L. Grilli, D. Valerini, A. E. Slobozeanu, B. O. Postolnyi, S. Balos, A. Rizzo, and R. R. Piticescu, *Critical raw materials saving by protective coatings under extreme conditions: A review of last trends in alloys and coatings for aerospace engine applications*, *Materials* **14** (2021), no. 7 1656.

[154] C. Heyer, *Human-robot interaction and future industrial robotics applications*, in *2010 ieee/rsj international conference on intelligent robots and systems*, pp. 4749–4754, IEEE, 2010.

[155] P. R. G. Kurka and A. A. D. Salazar, *Applications of image processing in robotics and instrumentation*, *Mechanical Systems and Signal Processing* **124** (2019) 142–169.

[156] J. Canny, *A computational approach to edge detection*, *IEEE Transactions on pattern analysis and machine intelligence* (1986), no. 6 679–698.

[157] W. Zhang, Y. Zhao, T. P. Breckon, and L. Chen, *Noise robust image edge detection based upon the automatic anisotropic gaussian kernels*, *Pattern Recognition* **63** (2017) 193–205.

[158] S. Thirumavalavan and S. Jayaraman, *An improved teaching–learning based robust edge detection algorithm for noisy images*, *Journal of advanced research* **7** (2016), no. 6 979–989.

[159] M. Adamík, J. Goga, J. Pavlovicova, A. Babinec, and I. Sekaj, *Fast robotic pencil drawing based on image evolution by means of genetic algorithm*, *Robotics and Autonomous Systems* **148** (10, 2021) 103912.

[160] M. M. Quamar and A. Nasir, *Review on fault diagnosis and fault-tolerant control scheme for robotic manipulators: Recent advances in ai, machine learning, and digital twin*, 02, 2024.

[161] M. Krupáš, E. Kajati, C. Liu, and I. Zolotova, *Towards a human-centric digital twin for human–machine collaboration: A review on enabling technologies and methods*, *Sensors* **24** (Mar., 2024). Publisher Copyright: © 2024 by the authors.

[162] S. Zeghlache, D. Saigaa, and K. Kara, *Fault tolerant control based on neural network interval type-2 fuzzy sliding mode controller for octorotor uav*, *Frontiers of Computer Science* **10** (2016) 657–672.

[163] V.-C. Nguyen, A.-T. Vo, and H.-J. Kang, *A finite-time fault-tolerant control using non-singular fast terminal sliding mode control and third-order sliding mode observer for robotic manipulators*, *IEEE Access* **9** (2021) 31225–31235.

[164] S. Wen, M. Z. Chen, Z. Zeng, T. Huang, and C. Li, *Adaptive neural-fuzzy sliding-mode fault-tolerant control for uncertain nonlinear systems*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47** (2017), no. 8 2268–2278.

[165] G. Gómez, *Local smoothness in terms of variance: the adaptive gaussianfilter*, 01, 2000.

[166] J. Yuan, F. Xu, S. Hao, W. Liu, and L. Jiang, *Research on the trajectory planning method of custom pattern robot based on image processing*, pp. 748–754, 04, 2021.

[167] W. Rong, Z. Li, W. Zhang, and L. Sun, *An improved CANNY edge detection algorithm.* 2014.

[168] Y.-K. Huo, G. Wei, Y.-D. Zhang, and L. nan Wu, *An adaptive threshold for the canny operator of edge detection*, *2010 International Conference on Image Analysis and Signal Processing* (2010) 371–374.

[169] N. Otsu, *A threshold selection method from gray-level histograms*, *IEEE Transactions on Systems, Man, and Cybernetics* **9** (1979), no. 1 62–66.

[170] Y. S. Xia, G. Feng, and J. Wang, *A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control*, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **35** (2005), no. 1 54–64.

[171] Z. Li, C. Li, S. Li, and X. Cao, *A fault-tolerant method for motion planning of industrial redundant manipulator*, *IEEE transactions on industrial informatics* **16** (2019), no. 12 7469–7478.

[172] D. Sturman and D. Zeltzer, *A survey of glove-based input*, *IEEE Computer Graphics and Applications* **14** (1994), no. 1 30–39.

[173] J. Hong and X. Tan, *Calibrating a vpl dataglove for teleoperating the utah/mit hand*, in *Proceedings, 1989 International Conference on Robotics and Automation*, pp. 1752–1757 vol.3, 1989.

[174] J. LaViola, *A survey of hand posture and gesture recognition techniques and technology*, *Brown university, providence, ri* **29** (1999).

[175] *Ultraleap,World-leading Hand Tracking: Small. Fast. Accurate.* , 2021.

[176] Y. Du, S. Liu, L. Feng, M. Chen, and J. Wu, *Hand gesture recognition with leap motion*, 2017.

[177] HolyDumphy, *Potentiometer Adjustable Voltage Divider Introduction*, 2020.

[178] *Potentiometer, Preset Potentiometers and Rheostats*, 2021.

[179] *Te connectivitywhat is a string pot: String potentiometers*, 2020.

[180] L. Majeau, J. Borduas, S. Loranger, Y. El-Iraki, J. Lavoie, D. Banville, V. Latendresse, V. Béland, J. Daniel-Rivest, A. Thiaw, H.-D. Bambara, T. P. Beausoleil, W. Trottier-Lapointe, and J. Lapointe, *Dataglove for consumer applications*, in *2011 7th International Workshop on Fibre and Optical Passive Components*, pp. 1–4, 2011.

[181] "Fiber-optic sensor."

[182] K. S. Kuang, W. J. Cantwell, and P. J. Scully, *An evaluation of a novel plastic optical fibre sensor for axial strain and bend measurements*, *Measurement Science and Technology* **13** (2002), no. 10 1523.

[183] J. A. Dobrzynska and M. A. Gijs, *Capacitive flexible force sensor*, in *Procedia Engineering*, vol. 5, pp. 404–407, Elsevier Ltd, jan, 2010.

[184] E. Pritchard, M. Mahfouz, B. Evans, S. Eliza, and M. Haider, *Flexible capacitive sensors for high resolution pressure measurement*, in *SENSORS, 2008 IEEE*, pp. 1484–1487, 2008.

[185] J. S. Neely and P. J. Restle, *Capacitive bend sensor.* United States of America Patent, jun, 1995.

[186] G. B. Langford, *Flexible potentiometer in a horn control system.* United States of America Patent, may, 1994.

[187] *Flex Sensor Data Sheet*, tech. rep., Spectra Symbol, 2001.

[188] N. Drake, *Polymeric Materials for Electrostatic Applications: A Report from Rapra's Industry Analysis and Publishing Group*. 1996.

[189] "What is a Load Cell and how does it work?."

[190] *What is a Strain Gauge and How Does it Work?*, 2021.

[191] H. Akbari and A. Kazerooni, *Improving the coupling errors of a maltese cross-beams type six-axis force/moment sensor using numerical shape-optimization technique*, *Measurement* **126** (05, 2018) 342–355.

[192] "Robot simulator CoppeliaSim: create, compose, simulate, any robot."

[193] E. Rohmer, S. P. N. Singh, and M. Freese, *CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework*, *Coppelia Robotics*.

[194] "MATLAB & Simulink."

[195] "Hiwonder Home Page."

[196] 90srobot, "Biomimetic Hand Endoskeleton."

[197] M. W. Spong, S. Hutchinson, M. Vidyasagar, et al., *Robot modeling and control*, vol. 3. wiley New York, 2006.