

Tele-medicine application: self-serviced dental cleaning solution based on soft
robot and remote control

OUYANG Xiaotong

MASTER OF SCIENCE
City University of Hong Kong
July 2022

CITY UNIVERSITY OF HONG KONG
香港城市大學

**Tele-medicine application: self-serviced dental cleaning solution
based on soft robot and remote control**

**远程医疗应用：基于软体机器人和远程控制的自助式
牙齿清洁解决方案**

Submitted to
Department of Biomedical Engineering
生物醫學工程學系

in Partial Fulfillment of the Requirements
for the Degree of Master of Science
理學碩士學位

By
OUYANG Xiaotong
欧阳晓彤

July 2022
2022年7月

CITY UNIVERSITY OF HONG KONG
Department of Biomedical Engineering
Master of Science in Biomedical Engineering

Statement on the Extent of Research Collaboration

If any part of your research has been carried out in collaboration with other parties, please indicate the extent of collaboration, and to identify any parts of the dissertation thesis which are not the result of your own work.

☒ No.

☐ Yes. If yes, please specify.

Signed by

OUYANG Xiaotong
OUYANG Xiaotong
Dated : 25/07/2022

CITY UNIVERSITY OF HONG KONG

Department of Biomedical Engineering
Master of Science in Biomedical Engineering

Dissertation Acceptance

Student Name	:	OUYANG Xiaotong
		Tele-medicine application: self-serviced dental cleaning
Thesis Title (English)	:	solution based on soft robot and remote control
		远程医疗应用：基于软体机器人和远程控制
(Chinese)	:	的自助式牙齿清洁解决方案

This is to certify that the student's thesis (as presented above) has been examined by the following examiners and has received full approval for acceptance in partial fulfillment of the requirements for the Degree of Master of Science in **Biomedical Engineering**.

Dr. ZHANG Jiachen	Department of Biomedical Engineering City University of Hong Kong
-------------------	--

Assessment Panel	Department of Biomedical Engineering
Prof. Lixin Dong	City University of Hong Kong
Dr. Wai Yan Kannie Chan	
Dr. Ting-hsuan Chen	
Dr. Yajing Shen	

Signed by

Dr. Yajing Shen
Dissertation Coordinator
BME, CityU
Dated :

ACKNOWLEDGMENTS

The past year has been a very intense and fulfilling one. I feel very grateful for all my memories in CityU.

First of all, I would like to express my gratitude to my supervisor, Dr. Zhang Jiachen. Without his help, I cannot complete this dissertation. Every discussion of us is enriching for me.

Secondly, I want to thank the other lab members, especially Mr. WEI Zihan. He has always been very clear about my progress in the process of completing this dissertation and has put forward many valuable suggestions for completing my experiment and revising my writing.

Then, I want to thank the friends I met in CityU and Hong Kong. We study together, have fun together, and talk to each other. And they helped me a lot whenever I encountered difficulties in my studies and life. Even including the tech problems of this thesis, when I faced problems completing this dissertation, they always tried their best to help me.

Finally, I want to thank my family. Thank them for providing me with a loving and relaxing family environment, as if all the shortcomings and deficiencies can be easily accommodated in this environment.

In short, I'm fortunate to have all the experience in CityU. Thank you very much for meeting all of you.

SUMMARY

The traditional and mainstream way of dental cleaning is to go to the dentist for dental cleaning. However, this method of teeth washing has some disadvantages. It requires close contact between doctors and patients. Due to the shortage of doctors, the surge of patients in many areas, and the repeated use of equipment, it is easy to produce cross-infection in various situations, especially in the COVID-19 era, which is undoubtedly causing a tremendous challenge to block the spread of the virus.

Here, a remote dental cleaning solution is proposed to change the challenging situation caused by the traditional dental cleaning method. In this paper, the software-related work of the system is mainly done, including the connection between the upper computer and the lower computer, the real-time detection of the projection faces of the teeth model, the estimation of the three-dimensional rotation angles of the teeth, and the development of the GUI.

The computer is used as the upper computer at the doctor's site, and the Raspberry Pi is used as the lower computer to receive the signal. They communicate wirelessly through the SSH protocol, and the control of the upper computer to the lower computer is verified by the experiment of the LEDs connected to the GPIO pins. The deep learning algorithm is used to recognize the three projection views of the teeth model (Left, Front, Right). The experimental process includes image collection, data preprocessing, and model training, as well as verification in the test set and real-time video. The accuracy of this detection model is 0.88. The pitch, yaw, and roll angles of the teeth model are estimated by using the ArUco marker and PnP principle. First, the camera is calibrated to get the internal parameter matrix, and then when the program detects the marker taped on the teeth model, the rotation vector of the teeth at this time can be obtained according to the PnP principle, and then the rotation vector can transform to the Euler angle, realizing the real-time estimation of the teeth angles. The

average error of the angle estimation is 0.82088245° . A GUI for doctors is developed by using MATLAB APP Designer. This app contains functions such as communication with the lower computer, connecting with the camera, importing the 3D teeth model, detecting teeth projection directions, etc.

In conclusion, the solution proposed in this project can achieve zero contact between doctors and patients based on the completion of teeth cleaning and solve the problem of cross-contamination of traditional dental methods. It provides a promising method to solve the problem of cross-infection of bacteria and viruses caused by teeth washing, especially during the COVID-19 pandemic.

Keywords: Dental cleaning; Posture estimation; YOLOv2; Telemedicine

TABLE OF CONTENTS

ACKNOWLEDGMENTS	I
SUMMARY.....	II
TABLE OF CONTENTS	IV
LIST OF TABLES.....	VII
LIST OF FIGURES.....	VIII
CHAPTER 1	1
INTRODUCTION	1
1.1 Background	1
1.2 Other related methods of dental cleaning or avoiding exposure	2
1.3 Research protocol and purpose	4
<i>1.3.1 Objectives.....</i>	<i>4</i>
<i>1.3.2 Solution and system description.....</i>	<i>4</i>
1.4 Dissertation overview.....	6
CHAPTER 2	9
LITERATURE REVIEW.....	9
2.1 Soft robotics	9
2.2 Review of teleoperation systems in real-time	9
<i>2.2.1 Real-time operation</i>	<i>10</i>
<i>2.2.2 Applications of visual feedback</i>	<i>11</i>
<i>2.2.3 Different techniques applied to teleoperation.....</i>	<i>13</i>

2.3 Vision and image	14
2.3.1 <i>Image preprocessing</i>	15
2.3.2 <i>Camera-based measurement</i>	16
2.3.3 <i>Object detection and classification</i>	18
2.3.4 <i>3D estimation - depth, posture, and angle</i>	20
CHAPTER 3	23
METHODOLOGY	23
3.1 3D modelling and importing	26
3.2 Telecommunication between the upper and lower computer	27
3.2.1 <i>Wireless connection between Raspberry Pi and computer</i>	28
3.2.2 <i>Control of GPIO pins by the upper computer</i>	29
3.3 Detection and classification of the projection images of the physical model	30
3.3.1 <i>Image preprocessing</i>	31
3.3.2 <i>YOLOv2 model training</i>	33
3.4 Solving the 3D rotation angle problem of the teeth from 2D video	38
3.4.1 <i>Camera calibration</i>	38
3.4.2 <i>ArUco marker generation</i>	42
3.4.3 <i>Detection and identification principle of the ArUco marker</i>	43
3.4.4 <i>Estimation of the camera's pose using the Perspective-n-Point principle</i>	44
3.4.5 <i>From Rotation vector to Euler angles</i>	46
3.5 GUI design	49
CHAPTER 4	51
RESULTS AND ANALYSIS	51
4.1 The connection between the laptop and Raspberry Pi	51
4.2 Detection of the different views of the real teeth model	53
4.2.1 <i>Detection on the test set</i>	53

4.2.2 Testing in live video.....	53
4.3 The 3D rotation angles of the teeth model	54
4.3.1 The detection of the marker in the real-time video	54
4.3.2 The calculation of Euler angles	55
4.4 MATLAB APP	56
4.5 Result analysis of this project	60
4.5.1 The Intersection over Union analysis of the 2D detection mission	60
4.5.2 Quantitative characterization of the estimation of the 3D angle.....	63
CHAPTER 5	66
CONCLUSION AND FUTURE WORK	66
5.1 Conclusion	66
5.2 Future work.....	68
5.2.1 Improvement.....	68
5.2.2 Next step.....	69
BIBLIOGRAPHY	70
APPENDICES	75

LIST OF TABLES

Table 1.	The hardware equipment and their information.....	24
Table 2.	The operation system and program development environment.....	24
Table 3.	The operation system and program development environment.....	25
Table 4.	MATLAB add-ons	25
Table 5.	Anchor boxes sizes (Unit: pixel).....	37
Table 6.	The training parameters	38
Table 7.	Relevant definitions	61
Table 8.	The mean values of the three classes and total test data set.....	63
Table 9.	Parameters of the camera's internal reference matrix	63
Table 10.	Detection range of different rotation axes (Unit: °)	64
Table 11.	Error condition of different rotation axes (Unit: °).....	65

LIST OF FIGURES

Figure 1.	Visualization of the aerosol problem during scaling [2].....	1
Figure 2.	An example of a remotely controlled medical system [8]	4
Figure 3.	The schematic overview of the system concept that proposed for this project.	5
Figure 4.	The telecommunication structure of the two sides	6
Figure 5.	A simplified and classical master-slave remote operation system model [10]....	10
Figure 6.	Examples of Time Series Prediction [13]	11
Figure 7.	An example of the control process of a teleoperation system [17].....	12
Figure 8.	The robotic teleoperation [18]	13
Figure 9.	The VR-based teleoperation system [20].....	14
Figure 10.	The preprocessing process for mammograms [28]	16
Figure 11.	(a) The procedure of image acquisition; (b) From world coordinates to camera coordinates geometry [30]	17
Figure 12.	Schematics of different detection algorithm [34]	20
Figure 13.	Pose estimation and optimization for teleoperated robots [37]	21
Figure 14.	The icon of MAYA software [40]	26
Figure 15.	Modules and interfaces used in this project of Raspberry Pi 4 Model B.....	28
Figure 16.	The Raspberry Pi and laptop wirelessly connected within the same Wi-Fi	29
Figure 17.	Information and SSH connection of the Raspberry Pi in this project.....	29
Figure 18.	The pins used in this project and the parts represented	30

Figure 19.	Examples of the acquired images. (a)(d) Images taken from left view with different distances. (b)(e) Images taken from front view with different distances. (a)(d) Images taken from right view with different distances.....	31
Figure 20.	The comparison of the image before and after adding noises. (a) Before adding noises; (b) After adding noises	32
Figure 21.	Examples of the image with label. (a) Image labeled with “Left”. (b) Image labeled with “Front”. (c) Image labeled with “Right”	33
Figure 22.	The YOLO processing schematic [41]	34
Figure 23.	Area vs. Bounding box area.....	35
Figure 24.	The bounding boxes distribution after clustering	36
Figure 25.	The model of the three coordinate systems.....	40
Figure 26.	The chessboard used in this project	41
Figure 27.	The example of the process of calibration. (a) The image acquired; (b) Corner points detected during calibration; (c) Image after distortion removal.....	42
Figure 28.	The marker used in this project (ID: 25).....	43
Figure 29.	The pinhole camera imaging model and PnP schematic. From top to bottom are the camera coordinate, the pixel coordinate, and the world coordinate systems	45
Figure 30.	Coordinate representation of Euler angles	48
Figure 31.	Description of pitch, yaw and roll angles of the teeth model	48
Figure 32.	Information of the Raspberry Pi seen from the computer side	52
Figure 33.	(a) one LED is turned on, (b) All the LED lights are turned on	52
Figure 34.	Instance detection results from the test set	53
Figure 35.	(a) The APP for detecting teeth in live video before pushing the “Detect” button. After pushing “Detect” button, according to the view of the camara, the blank axis area will show: (b) The detection result of the left side. (c) The detection result of the front	

side. (d) The detection result of the right side	54
Figure 36. The detection result of the ArUco marker	55
Figure 37. The testing result of the Euler angles. Colors of axes: X axis – Red; Y axis – Green; Z axis – Blue	56
Figure 38. 3D dental model made by Maya	57
Figure 39. The start page of this APP	57
Figure 40. The 3D model importing and view changing functions of this APP. (a) The front view at the beginning of finishing importing the model. (b) (c) (d) View changes via the operation of the operator within the APP	58
Figure 41. The projection of the teeth model in different directions – Left, Front, Right	58
Figure 42. Gauges vary with spinners	58
Figure 43. The changes in the APP page and the LEDs on the Raspberry Pi. (a) "power supply" is turned on, (b) "power supply" and "video system" are turned on, (c) "power supply" and "video system" and "water system" are turned on, (d) "power supply" and "video system" and "water system" and "ultrasound system" are turned on.....	59
Figure 44. An example of the recognition of the front view of the teeth.....	60
Figure 45. The confusion matrix	62
Figure 46. (a)(b)(c) Detected angles and error condition of X-axis, Y-axis, Z-axis, respectively. (d) comparison of errors about different axes.	64

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dental cleaning removes plaque, tartar, pigment, etc., from our teeth using specialized devices and equipment. Many studies over the years have shown that dental cleanings once or twice a year are effective in preventing tooth decay and periodontal disease. However, the health risks associated with the cleaning process should not be ignored.

During the dental cleaning in a hospital or clinic, the dentist and the patient are usually in the same confined space being close to each other for a long time. As the scaling process proceeds, because of water coolants for cleaning and ultrasonic instruments for removing calculus, large amounts of aerosols with saliva and blood containing dangerous pathogens (viruses, bacteria, and fungi) are created between the patient and the dentist [1]. As shown in **Figure 1**, using a fluorescein model to simulate aerosol and splatter, we can visualize the severity of the aerosol problem during scaling [2]. And they can spread to other parts of the confined space. Such a situation would not only pose a health risk to the patients and dentists in the room at the same time. Still, it could also lead to environmental contamination and cross-contamination between doctors and other patients, posing a great challenge for people to contain the spread of infectious diseases.

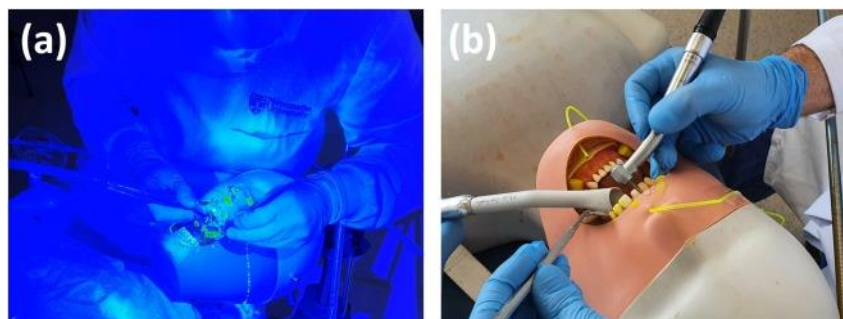


Figure 1. Visualization of the aerosol problem during scaling [2].

What made matters worse is that the global new coronavirus epidemic, which has continued

from 2020 to the present, has created even more uncertainty and difficulty in dental cleaning. Many coronavirus cases and related studies have shown that aerosol transmission is one possible way to transmit the coronavirus when exposed to high aerosol concentrations in a relatively closed environment. Aerosols from routine dental procedures, including dental scaling, would pose a potentially high risk of infection to dental caregivers and patients [3]. This kind of risk, at different levels and perspectives, already hinders routine dental care. For dental clinics and dentists, continuing to operate oral cleanings or other oral care and procedures in such a situation requires significant safety and hygiene measures, which can notably reduce the number of patients a doctor can handle per day and place a burden on hospital finances. For some dental clinics under lockdown, the situation makes their operations unsustainable, private clinics may face bankruptcy, and dentists may have difficulty securing their income [4]. The current situation is also aggravating for patients who need dental scaling, especially those with underlying dental diseases. And a survey from Madrid showed that as many as 24.5% of the people surveyed do not go to the dentist because of the fear of SARS-CoV-2 [5], which indicates that the fear of the new coronavirus pandemic and the resulting avoidance of the infection have contributed psychologically to the resistance to dental cleaning.

Therefore, in such a difficult time, we need a solution to solve this problem that can guarantee regular dental cleaning for those who need it and, at the meantime, reduce or even avoid, to some degree, the possibility of cross-infection caused by dental cleaning, especially the possibility of contracting the coronavirus due to dental cleaning.

1.2 OTHER RELATED METHODS OF DENTAL CLEANING OR AVOIDING EXPOSURE

Telehealth systems based on digital imaging are one way to think about solving this problem. Rapidly evolving digital imaging technology offers dental healthcare service providers an alternative to traditional methods and is further becoming an essential element of the daily dental routine. Many studies have found that remote diagnosis of dental diseases that are supported by intra-oral photographic images may provide an effective and the reliable approach to replace the classical oral examinations [6]. By capturing intraoral pictures of the patient and saving and viewing them on the cloud server, a dentist from another area can

know and diagnose the patient's condition accordingly and give expert advice based on the pictures. Such an approach avoids contact between the patient and the dentist and naturally avoids cross-contamination due to contact. However, this method is only effective for some patients who do not have emergencies. Meeting and contacting doctors are still unavoidable for some people who must have their teeth cleaned.

Ultrasonic toothbrushes and home ultrasonic scaling devices that can also avoid the contact caused by dental cleaning in some way are coming to market. One of the main physical processes involved in the ultrasonic cleaning process is the cavitation effect, which consists of the generation, growth, oscillation, and collapse of bubbles, generating significant mechanical forces that result in permanent chemical and physical changes on the surface of the object in contact [7]. Ultrasonic toothbrushes not only remove more bacteria from the oral cavity than regular toothbrushes but also reduce plaque in patients and have a positive effect on slowing down the development of gum inflammation. Such an invention allows for the exclusive use of the instrument for the individual, blocking cross-contamination. However, incorrect use of such devices often causes more damage to the user's teeth, such as gum bleeding and damage to the enamel caused by improper handling. It is difficult for patients with more serious conditions to achieve results only using those devices. Therefore, there is no substitute for professional operation from the medical professional.

During the coronavirus pandemic, the sampling of oropharyngeal swabs is broadly applied to viral nucleic acid testing for the diagnosis of COVID-19. Healthcare workers who collect nucleic acids are often exposed to a high-level risk of getting infected due to patient aerosols. For avoiding the close exposure between the medical staff and the health care patient and therefore reducing risks of COVID-19 infections due to sample collection, nucleic acid sampling systems that are automated by robots performing work or remotely operated by healthcare workers in combination with vision systems have been invented [8][9]. These systems hold the possibility of minimizing the infection risk for healthcare professionals by avoiding any close exposure between medical professionals and patients while ensuring detection accuracy. This approach to infection avoidance has good potential for other situations that require close contact between the operator and the patient, such as dental cleaning.

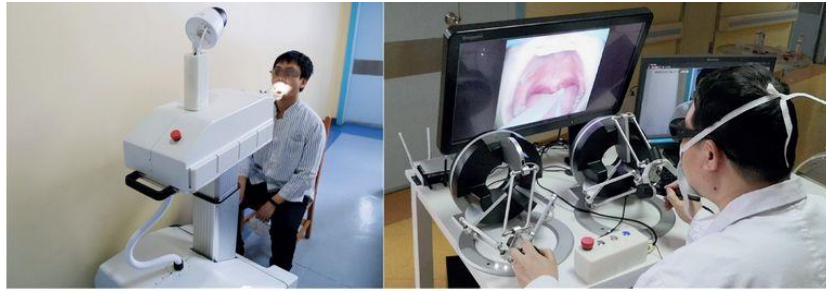


Figure 2. An example of a remotely controlled medical system [8].

1.3 RESEARCH PROTOCOL AND PURPOSE

Based on the above problems and the current situation, this subsection will give the solution of this project for solving the problem of cross-contamination during the dental cleaning process, especially in the Covid-19 era. Firstly, the objectives of this solution will be given. Next, the system architecture, teleoperation structure of this system will be described in detail.

1.3.1 Objectives

The project aims to propose a remote tooth cleaning solution to restrict the spread of COVID-19 via tooth washing. This solution mainly aims to achieve the following goals:

First, patients can avoid contacting doctors or other patients to reduce personnel aggregation during teeth washing.

Second, the doctor can receive the video about the equipment's condition and the patient's oral condition in real-time. The doctor can remotely control the robot to do tooth washing for the patient in real-time, including the physical displacement of ultrasonic equipment.

Third, due to the high precision of the robot in the operation process and the feedback from the camera. The safety of this remote dental washing method will be guaranteed.

1.3.2 Solution and system description

In order to achieve these objectives and to overcome this challenge, this thesis proposes a possible solution and presents the corresponding system diagram for this solution. The concept of this project is shown in **Figure 3**. The entire system consists of a master side and a slave side. On the Master side, the doctor operates in a particular GUI on the computer. At the

same time, on the Slave side, the lower unit receives the signals and operates on the patient's teeth. The lower computer is connected to the power supply, the water supply, the camera, the ultrasound control circuit, and the motion controller of the soft robots. The Master and Slave communicate in real-time via the wireless network.

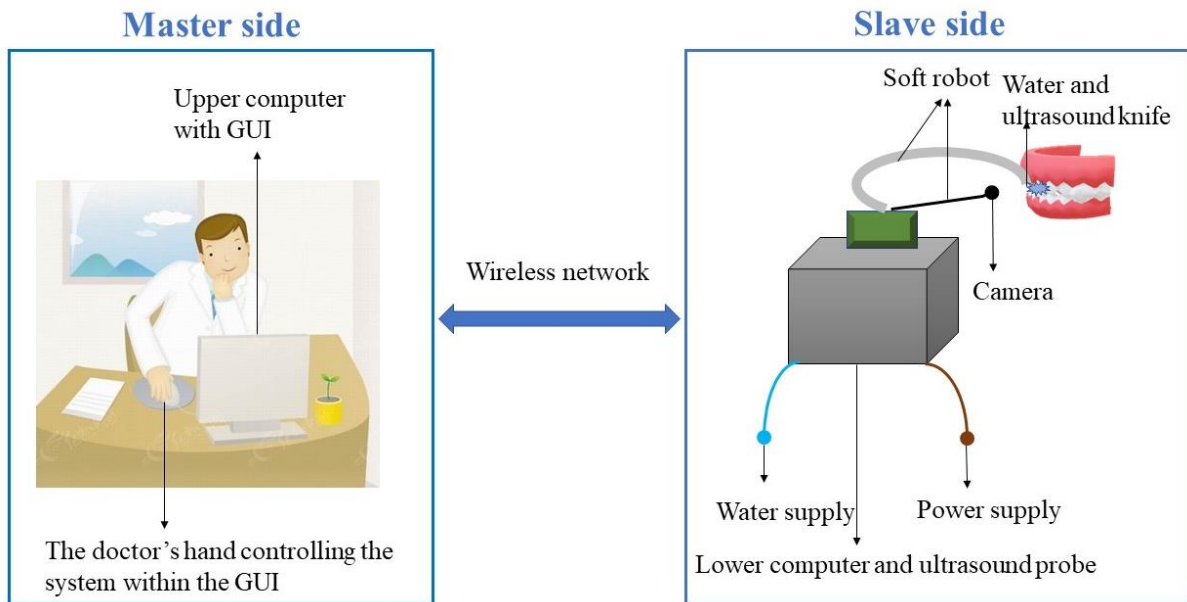


Figure 3. The schematic overview of the system concept that proposed for this project.

The structure of the remote-control system in this project is shown in **Figure 4**. The system consists of client and server. The client is in the upper computer, receives various control commands from the operator through the DirectInput packaging module, and provides a human-machine operation interface. The client is also used to complete the functions, including the translation of control commands, robot, ultrasound control, and data exchange with the server. The server is in the lower computer and connected with the robot and other operation tools. It is mainly responsible for forwarding the translated operation instructions from the client to the robot and transmitting the robot status and video data back to the client through the network.

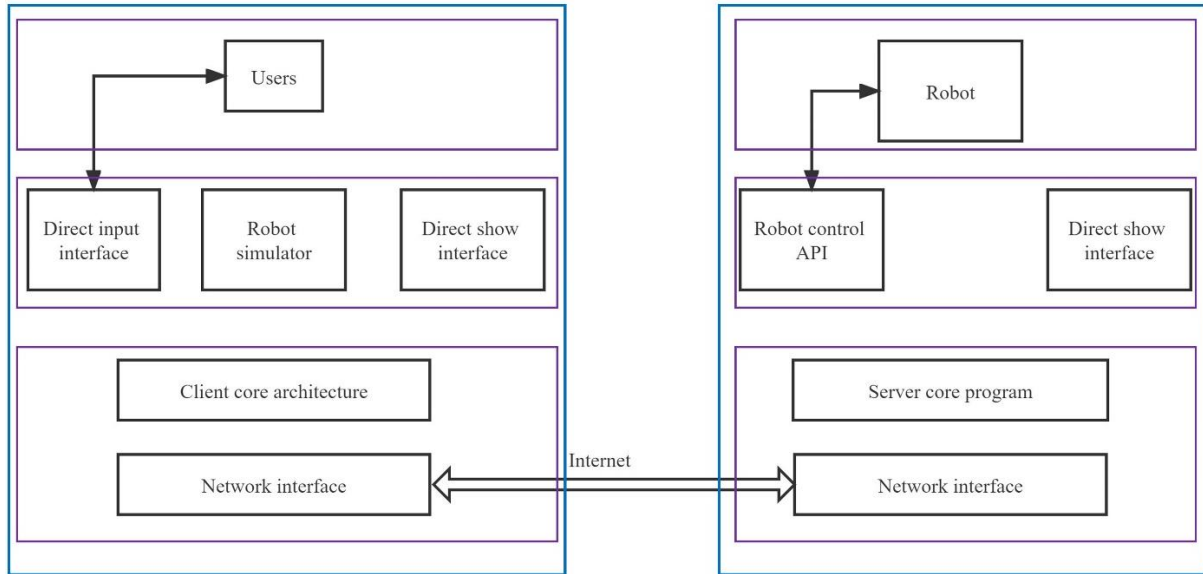


Figure 4. The telecommunication structure of the two sides.

1.4 DISSERTATION OVERVIEW

Dental-related care and procedures are often highly risky and very unfriendly for the blocking of infectious diseases. In order to address the spread of infectious diseases due to the dental cleaning process, and in particular, to lessen the impact of the COVID-19 pandemic due to dental cleaning, this paper proposes a system for dental cleaning based on a remotely operated robotic system. The software part of the system, especially the remote control and the visual feedback, has been developed, the experimental results have been tested and analyzed and a MATLAB APP has been developed with the corresponding functions. The detailed contents and the organization of the chapters of this thesis are shown below.

Chapter 1 provides a general introduction to this project, including the background of this project, the relevant solutions, the purpose of this study, and the design of the overall system. The dangerous situation of the spread of infectious diseases in the dental environment is described in detail, especially in the COVID-19 era, and the impact of the lack of suitable dental cleaning on various aspects of people's lives. The shortcomings of the solutions are then presented as well as relevant methods that can provide inspiration. Finally, the solution to this project and a schematic statement of this system are given.

Chapter 2 collates research that is relevant to the technology required for this project. Firstly,

by describing the research related to soft robotics, it is shown that the soft robots that can be applied to minimally invasive surgery are of interest to this project. Secondly, the literature on real-time teleoperation systems is collated, including three different topics: real-time operation, the use of visual feedback, and the use of different technologies in teleoperation systems. Finally, a summary of work in vision and imaging is presented, including image processing, camera-based (especially monocular cameras) measurements, object detection, and 3D estimation. The above literature survey is of great reference value for the work of this project.

Chapter 3 provides a detailed description of the methodology used in this paper and can be divided into five subsections. First of all, a detailed summary of the software and hardware equipment used in this paper is given. After that, in the first subsection, the process and methodology for building the 3D model of the teeth are described. In the second subsection, the process of communication between the upper and lower computer is explained as well as how to show the control of the upper computer over the circuit connected to the lower computer by means of LED lights. In the third subsection, the method of using the YOLOv2 model for the detection of the different faces of the tooth is given, containing the pre-processing of the image and the process of training the model. In the fourth subsection, experimental procedures and principles are given for solving the rotation angle of the teeth model, including the camera calibration, the production and detection of the ArUco marker, and the estimation of the camera pose using the PnP principle and the Eulerian angle translation method. In section 5, the method of making the GUI on the upper computer and the functions to be implemented are explained.

Chapter 4 is devoted to the representation of the experimental results and the analysis of results of the experiments. In the first subsection, the wireless communication between the computer and the Raspberry Pi is shown and the wireless control of the upper computer to the lower computer is indicated by the LEDs lighting up and down. In the second subsection, the task of detecting different profiles of teeth is validated in the test set and in the live video, respectively. In the third subsection, the real-time 3D rotation angles of the teeth are obtained, including the detection results of the marker and the calculation of the Euler angles are shown. In the fourth subsection, the MATLAB APP developed in this paper is shown. The panel interface and the implementation of the various functions of the APP are demonstrated in detail. The fifth subsection is the analysis of the results, starting with the analysis of the effect

of the 2D identification task on the training set, using six commonly used metrics to make a comprehensive judgment of the model effect. Then there is the analysis of the results for the 3D angle estimation. The camera's internal reference matrix and the distortion coefficients, as well as the inverse projection error, are given first, followed by an explanation of the angle recognition ranges and errors.

Chapter 5 summarizes the work done in this paper and discusses and conceptualizes possible ways of improving the results of this work and the next steps.

CHAPTER 2

LITERATURE REVIEW

2.1 SOFT ROBOTICS

Biologically inspired soft robots are unique in their strong adaptability and dealing with complicated environments. It can be used for interacting with humans [51]. A new type of "layer jamming" mechanism is proposed in 2013, which can realize mutable stiffness [52]. And in 2019, an independent, untethered, programmable soft actuator inspired by the sunflower stem bending was proposed on the basis of liquid-vapor phase change [51]. A normally closed robot of long-form having a retractable function with selectable extension direction and with a "non-sealed" drive system of liquid was also presented in 2021 [53]. Due to those soft robots' compact size and lightweight are ideal for a variety of minimally invasive surgical procedures and suitable for this project.

2.2 REVIEW OF TELEOPERATION SYSTEMS IN REAL-TIME

Remote operating systems have undergone decades of development and a lot of research related to them. The master-slave remote operating system is one of the typical ways to implement the remote operation. Such a mode of operation remotely controls an object by providing the operator with the required conditions, such as remote location [10]. Generally, this system can be divided into a master and slave sides. The operator operates on the master side to achieve the object's movement, executing the task on the slave side. And these two ends can send and receive data through specific communication protocols to make the whole system work well. **Figure 5** provides a simple and classical model of a master-slave remote operating system.

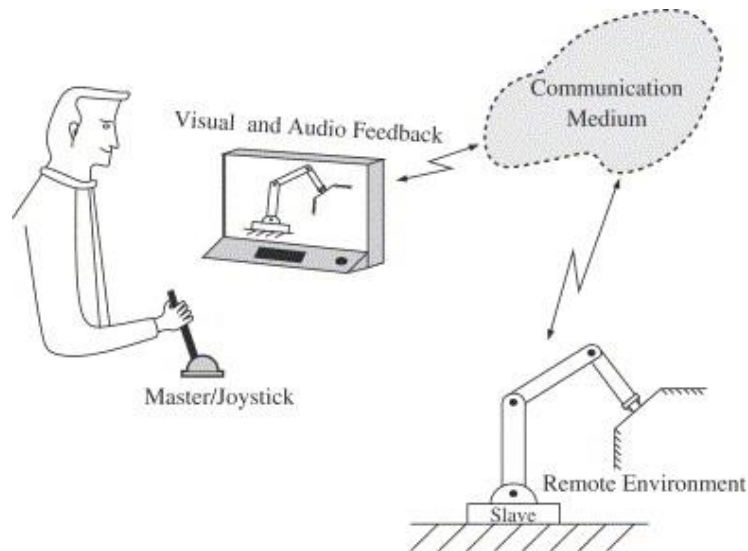


Figure 5. A simplified and classical master-slave remote operation system model [10].

Nowadays, teleoperated robots have been used in several fields, including but not limited to unmanned underwater vehicles, space robots, remote surgical robots, and mobile robots for teleoperation [11]. There are also many types of research to improve teleoperation based on the remote control, human-robot interaction, software-based teleoperation, latency reduction, and visual feedback. Real-time is an essential consideration for the application of teleoperation systems in medicine. Delays in communication media can lead to unstable systems. Visual feedback is also of significant use as one of the essential types of feedback in remote operating systems. In recent years, some new technologies have been applied to give new forms of performance to remote operating systems. This part will describe some studies for the real-time operation of remote operating systems, visual feedback, and different techniques applied in remote operating systems.

2.2.1 Real-time operation

It is essential to research the problem of delay in master-slave communication networks for remotely operated robots, especially for surgical robots. It has been noted that a delay of 400 *ms* is acceptable, that delays between 100 and 250 *ms* are not significantly different from no delay, and that delays beyond 400 *ms* affect surgeon performance [12]. And, if the delay fluctuates too much, the surgery will be difficult to perform smoothly.

Nowadays, many methods to solve network delays are proposed and applied, including direct control, supervisory control, passivity theory, event-based, sliding mode control, predictive control, and other methods.

Time series prediction methods are an excellent approach to reducing the time delay of a system [13]. By predicting and simulating the actions to be performed by the robot, the time series prediction may be used to compensate for timing delays, and based on the predicted data, the reliability of system operation can be better ensured, and the impact of time delay on the system can be solved to some extent [14].

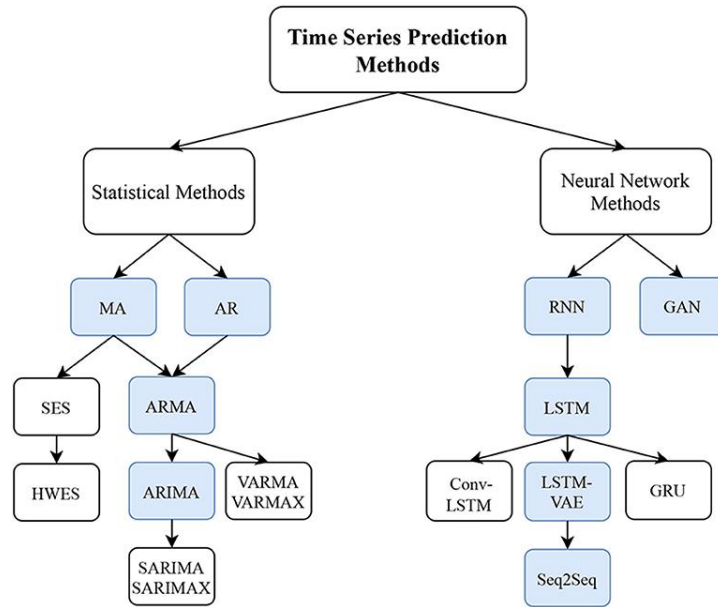


Figure 6. Examples of Time Series Prediction [13].

2.2.2 Applications of visual feedback

In general, visual feedback is the most critical feedback in remote control systems [11]. However, how to process the video information we get and make it more useful is still a problem that deserves to be studied. Many researchers have used the visual feedback obtained in teleoperation systems to try to solve some long-standing issues or to give some new applications to teleoperation systems.

An underactuated bionic hand teleoperated control system based on visual tracking is an application that utilizes visual feedback from a teleoperated system. Currently, the bionic hand is already used in a wide range of applications, like grasping tasks, industrial applications, human-computer interaction, and fine manipulation in hazardous situations [15][16]. J.S. Artal-Sevil et al. proposed a small bionic hand control solution for real-time gesture identification with vision-based feedback [17]. In this work, the bionic hand has five degrees of freedom, and each finger can be driven independently with the servo motor. After the motion of the forearm, hand and fingers is recognized by the sensors and application

software, and the computer processes these parameters and variables. The parameters and linked data then are transmitted to the Arduino platform that acts as the controller of the robotic hand prosthesis. Commands are sent and received via serial communication between the Arduino platform and the computer. After this, the controller passes the necessary information to each actuator involved in the different finger movements, implementing a teleoperation system based on visual feedback. **Figure 7** shows the control process.

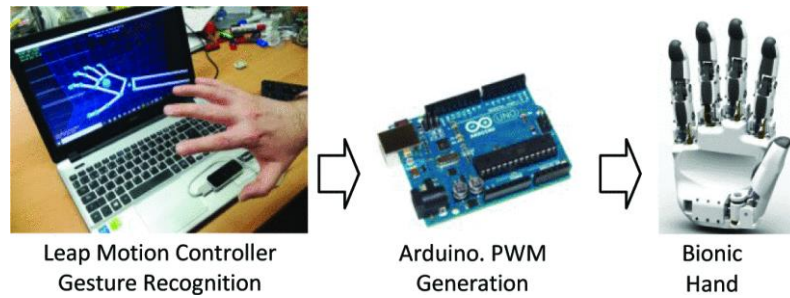


Figure 7. An example of the control process of a teleoperation system [17].

Visual marker-based hand tracking is also an application that enables teleoperation. Yongseok Lee et al. report a framework for visual-inertial skeleton tracking (VIST) that offers robust and reliable hand tracking for real-world teleoperation of equipment [18]. In this work, the framework consists of a sensor glove and a head-mounted stereo camera. The VIST sensor glove comprises an inner glove layer with seven inertial measurement units (IMUs) and an outer glove layer with 37 visual markers. The VIST algorithm includes visual information extraction (3D position observation of anonymous visible markers obtained by fusing the stereo camera and TC with IMUs information) and visual-inertial hand motion estimation (hand motion is estimated by combining data of IMUs with extracted visual information and hand anatomical constraints). **Figure 8** illustrates the robotic teleoperation of this work.

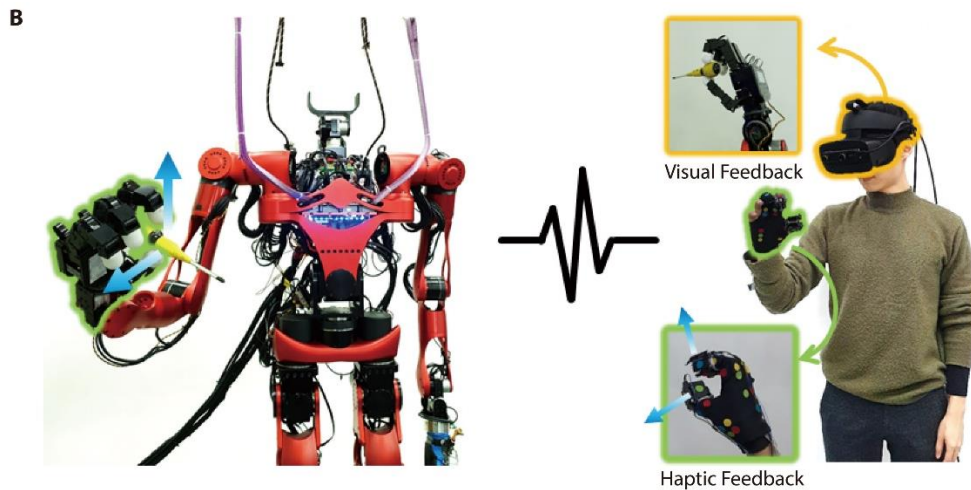


Figure 8. The robotic teleoperation [18].

These visual feedback-based applications, such as visual tracking methods, make good use of visual information in teleoperation. They have the potential to enrich the user experience not only in human-computer interface applications but also in some virtual reality and augmented reality scenarios, allowing users to have more accessible and more convenient modes of operation.

2.2.3 Different techniques applied to teleoperation

Remote robotic-assisted surgery is one of the application scenarios of the teleoperation system in the medical field. And the Internet of Things (IoT) usage has enriched the network and operation of this scenario. The work of Mohamad Khairi Ishak et al. reports an IoT-based controllable robotic arm system [19]. The robotic arm system is composed of both the software and hardware components. The hardware part uses a phone to remotely control the Raspberry Pi and uses the Raspberry Pi as the robotic arm's central controller. For the software part, the Raspbian Jessie operating system been adopted for the Raspberry Pi's operating system. During teleoperation, the data flow process can be divided into two sections. In the transmitter portion, the accelerometer and gyroscope in the Android phone have been utilized to catch the postures and gestures from the phone. Signals coming from both the accelerometer and gyroscope are then captured and processed in the Android App. The data is then sent to the Raspberry Pi. in the receiver portion, the Raspberry Pi gets the data from the Android phone through the Message Queue Telemetry Transfer (MQTT) protocol. Calculating the inverse kinematics of the robot arm to determine the joint angles. Finally, the calculated joint angles from the inverse kinematics are then mapped to the servo

angles. With this system, the operators can operate the robotic arm remotely by phone, aid the surgeon in surgery, and use the robotic arm to care for patients. This work's latency is 1 *ms* for access from the local network and 50-69 *ms* for access from the public network with the server.

The use of VR technology in teleoperated robotic systems allows humans and robots to share the same visual and action space, eliminating the need for the operator to make any decisions based on information unavailable to the robot. Tianhao Zhang et al. presented a teleoperation robot system on the basis of virtual reality and hand tracking hardware [20]. This teleoperation system includes a VR system, a cell phone, two hand controllers, and a PR2 robot. A 3D camera is connected to the head of the robot to provide visual information to the headset. This system also uses an RGB-D camera to snap images in color with depth values for each pixel to match the DoFs of the robot and the controller. The 3D point cloud of corresponding colors then is processed and rendered to eliminate the gaps between the points and serve as the actual physical objects of the virtual environment. During operation, observers view the environment from a virtual camera that automatically updates its pose in order to represent the operator's head motions. As for the control interface, both the operator and the robot can be projected into a unique coordinate frame within the virtual environment. Overall, this VR teleoperation system achieves minor kinematics differences and enables real-time environment feedback. **Figure 9** shows this operating VR teleoperation system.



Figure 9. The VR-based teleoperation system [20].

2.3 VISION AND IMAGE

When remotely controlling a robot using video captured by a remote camera, it is difficult for the operator to measure depth and orientation from a single viewpoint due to the incomplete access to information [21]. The machine vision-based teleoperation assistance provides depth

and relative orientation information etc. to remote operators for use with remote operation systems better to accomplish the scenario requirements.

2.3.1 Image preprocessing

It has been proven that medical image processing is already a core area of innovation in modern healthcare [22]. And it has been pointed out that, more than model algorithms, the final accuracy of the diagnostic and classification results is highly dependent upon the image quality [23][24]. Therefore, for the vision field, effective image preprocessing is essential.

For CT and MRI images, the article by Samira Masoudi et al. describes detailed common preprocessing steps [25]. For CT images, the first step may be to denoise in the spatial or frequency domain. Then interpolation may be applied either on the XY plane or in the Z direction according to the required physical interval to avoid center-specific or reconstruction-dependent results. Thirdly, image registration may be performed based on methods ranging from simple thresholding to segmentation. Next, colors may be added to the clinical image, or image windows may be used for increasing its contrast in the region of interest and thus better. Then, color can be added to the clinical image, or image windows can be used to improve the contrast of the region of interest to analyze complex images faster and easier. Finally, the normalization operation is used to normalize the input image. For MRI images, the possible preprocessing steps are denoising, bias field correction, alignment, and normalization, respectively. M. Vijaya Maheswari et al. also point out that pre-processing is a fundamental and essential step to improve the contrast of images, detailing methods such as histogram equalization, mask filtering, and color space-based preprocessing [22].

Image enhancement methods that can improve visual quality are highlighted in [26]. And [27] illustrates the effective image processing in remote sensing and machine vision in agriculture. Then [28] emphasizes the preprocessing steps of image binarization, active contour, and contrast enhancement and gives the preprocessing process for mammograms as shown in **Figure 10**.

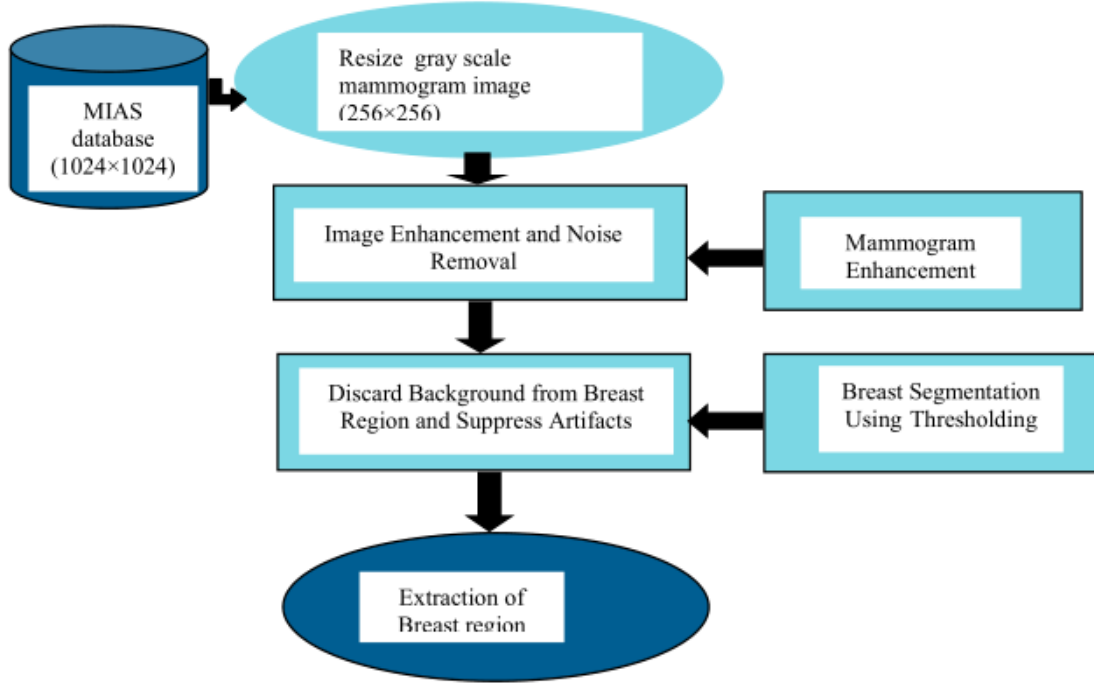


Figure 10. The preprocessing process for mammograms [28].

2.3.2 Camera-based measurement

Vision-based single-camera robot calibration systems can improve robot localization precision as well as serve as a diagnostic instrument for robotic manufacturing and upkeep [29]. This subsection will review some work that has been done related to the camera parameters obtained from a single monocular camera and its contribution to vision-based robotic systems.

In 2001, Motta, J. M. S. et al. explored the theoretical aspects of a robotic calibration method as well as a system for developing a viable vision-based, low-cost measuring system [29]. In their work, they used a monocular camera that is positioned onto the tool flange of the robot and constructs a model prototype of a robot calibration system. It is demonstrated experimentally that the proposed calibration system has improved the accuracy of the robot to well below 1 mm, but the model is not robust enough, and it may be improved by using a larger calibration plate or a camera with higher resolution.

In 2007, Li Mei Song et al. explained in detail the parameters used in the camera calibration and proposed a new high precision calibration method in their study [30]. This study has been performed to evaluate the calibration accuracy by comparing it with the measurement result

of a 3D coordinate machine with high accuracy. The accuracy of the calibrated vision system is up to 0.01 mm. **Figure 11** illustrates the process of image acquisition as well as from world to camera coordinates geometry.

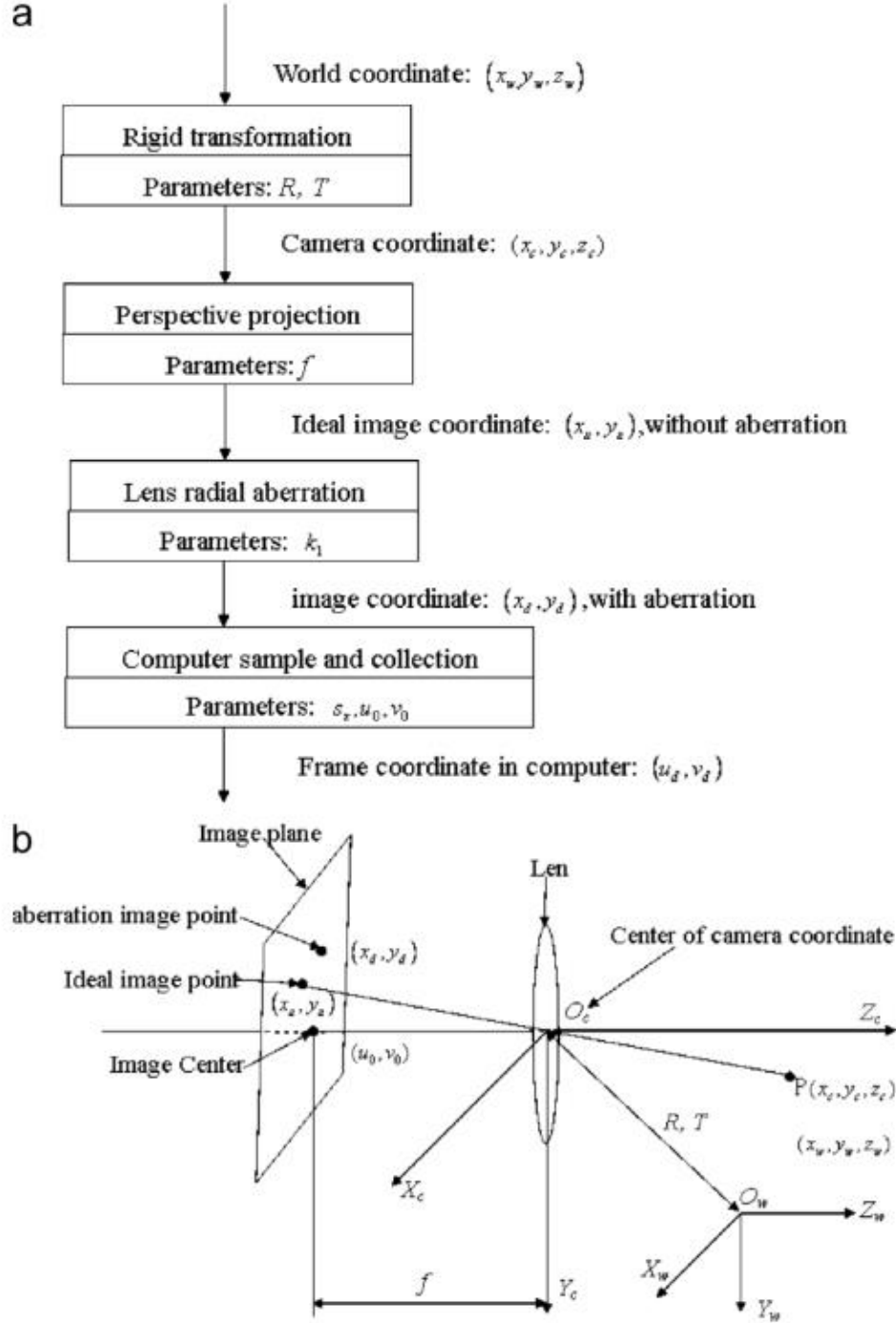


Figure 11. (a) The procedure of image acquisition; (b) From world coordinates to camera coordinates geometry [30].

In 2012, Dali Wang et al. presented a neural network-based algorithm for robot localization error estimation [31]. In this article, they have explained in detail two methods of robot

calibration (modelless manipulator calibration and bilinear manipulator calibration). During the experiment, the position error of each grid point in the space of calibration is acquired initially from the camera mounted to the end-effector on the robot that is being calibrated. And the input and output pairs for the training dataset are formed using the window composed of several cells around their interpolated positions. The local features of the error surface are then extracted using a neural network model. Finally, the object pose has been then compensated from the error in position acquired from this neural network model. After validation, it has been found that this algorithm improves the error estimation accuracy compared to previous analysis methods.

2.3.3 Object detection and classification

In the past several years, object detection systems have been rapidly developing for various applications, including face recognition, fault detection, medical image detection, etc. [32][33]. The rise of convolutional neural networks (CNNs) has led to a great development in object detection. Basically, there are two main technical development lines for CNNs-based target detection algorithms: methods based on anchors and anchor-free methods, while methods with anchors include both one-stage and two-stage detection architectures. Generally, the detection methods of two-stage object detection algorithms are usually more accurate than one-stage ones, however, the one-stage detection algorithms are much faster. Typical two-stage algorithms consist of Faster R-CNN, SPPNet, Fast R-CNN, R-CNN, etc. And algorithms based on one stage stand for the You Only Look Once (YOLO) series. Many studies have been conducted recently to compare and improve different algorithms on different application scenarios.

Prinsi Patel et al. have presented a YOLO algorithm on the basis of the convolutional network layer for object detection [33]. In the article, they use common COCO datasets and give a system work diagram that includes the input of data, standardization of image size, detection algorithm, and the result of object detection. The proposed convolutional neural network, the loss function of the YOLO algorithm is also explained. By comparing the performance with other algorithms in terms of mean average precision (mAP), Frames per second (FPS), etc., it is shown that object detection using the YOLO library has shorter object detection time, higher accuracy, and faster label conversion to text-to-speech (TTS) conversion and has a higher prediction of the box boundary coordinates and class

probabilities. A comparison of OpenCV-YOLO and Tensorflow-YOLO shows that the system proposed in this paper outperforms existing systems.

Samet Akcay et al. then thoroughly explore the task of using CNN in x-ray baggage image classification and detection [34]. They compare CNNs with the traditional manual feature based BoVW approach. Besides that, there is an SVM classifier trained in the final layer in the network for making a consistent comparison among the CNN and handcrafted features. Different CNN algorithms are also discussed to explore the effect of the complexity of the network with respect to the total performance. Experiments show the CNN features obtain a better performance than the hand-made features. For this issue, the fine-tuning of the whole network results in 0.996% true positives, respectively 0.011 false positives (FP) and 0.994 accuracies, a notable performance increase over that of the top performing handmade feature detector. The ResNet-50 reached 0.986 accuracies for the multiple X-ray baggage objects classification. This research has been conducted in this context and investigated target recognition algorithms to further enhance performance on clustered data sets. The results indicate that RCNN, R-FCN, and YOLOv2 methods, which are faster, exceed SW-CNN in terms of both accuracy and speed. YOLOv2 shows mAPs of 0.885 and 0.974 for the 6-class object detection and 2-class firearm detection issues, respectively.

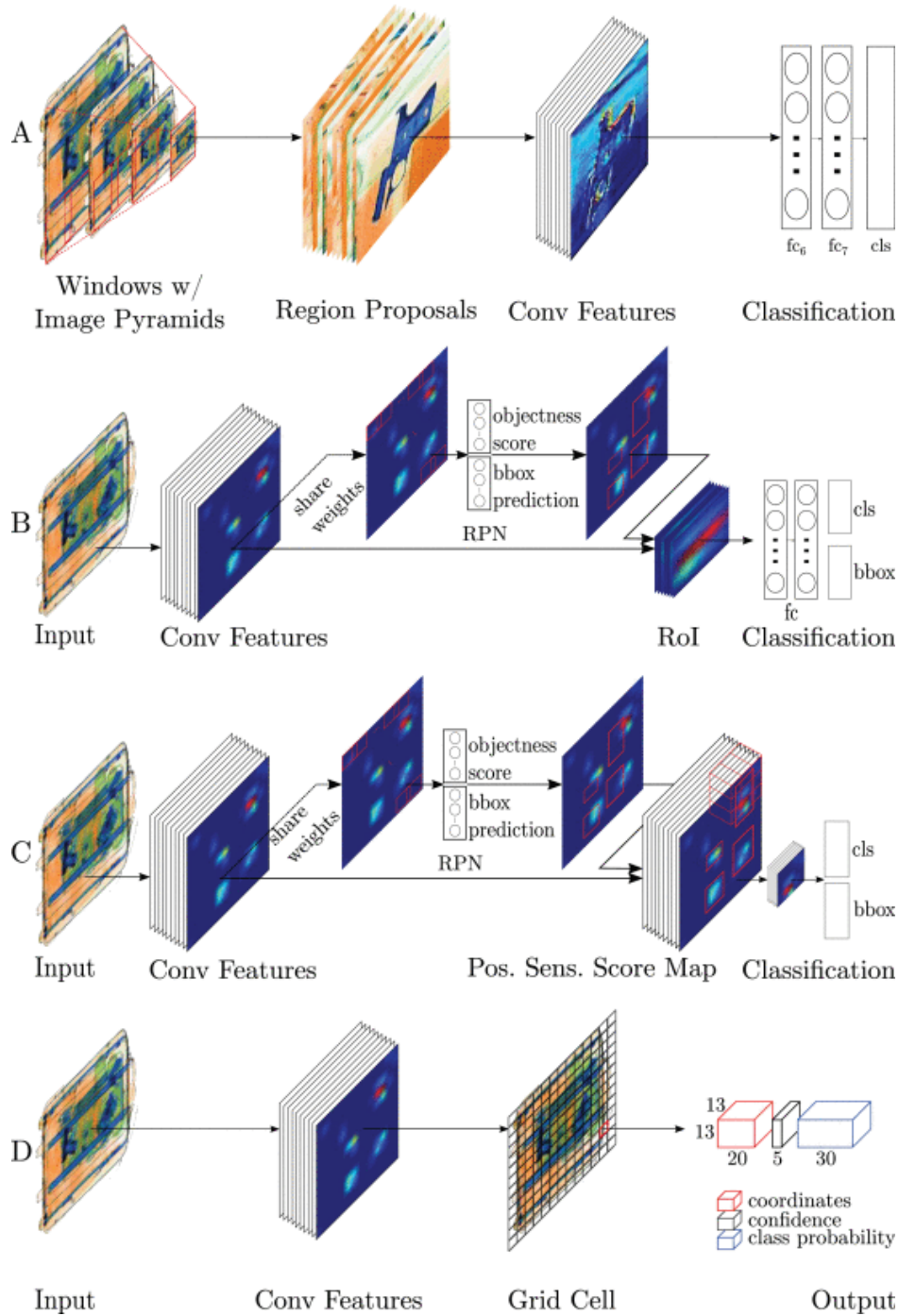


Figure 12. Schematics of different detection algorithm [34].

2.3.4 3D estimation - depth, posture, and angle

In recent years, most of the object detection frameworks proposed in current research have been two-dimensional. However, two-dimensional object detection cannot understand three-dimensional space, which leads to its uselessness for solving real-world problems [35]. When facing the localization, pose estimation and detection of real-world objects, we still need

some research on the detection and estimation of 3D objects.

The estimation of face pose has been an essential field of research in the human-machine interface, and in 2002, Qiang Ji described a novel method to estimate and track the three-dimensional pose from one single, one-sided perspective of a human face [36]. In this study, it is assumed with the assumption that a three-dimensional face shape can be approximated by an ellipse and the aspect ratio of the three-dimensional face ellipse is fixed. When a single-eye image of a human face is given. Ellipse detection is first performed to localize the image face, then the three-dimensional orientation and position are estimated based on detecting the face in the image. This study also explores the physical characteristics as well as some geometric limitations of the eyes under specific infrared illumination that promotes the development of human face detection. The detected original face ellipse has been tracked in the following frames, thus enabling three-dimensional face pose tracking from one frame to another. According to the experimental results, the algorithm works well. The estimation error is less than 0.5° from the synthesized data with no noise. And the estimation error is only 2.5° when adding Gaussian noise having 2.0 pixels of standard deviation.

In 2021, Amir Yazdani et al. developed a framework for pose estimation and optimization for tele-operated robots [37]. The framework does not require any additional sensors, estimates the operator's pose based only on the trajectory of the lead robot, as well as offers both online pose corrections and initial offline pose corrections depending on the type of teleoperation mission. The framework of their proposed intelligent teleoperation system is shown in **Figure 13**.

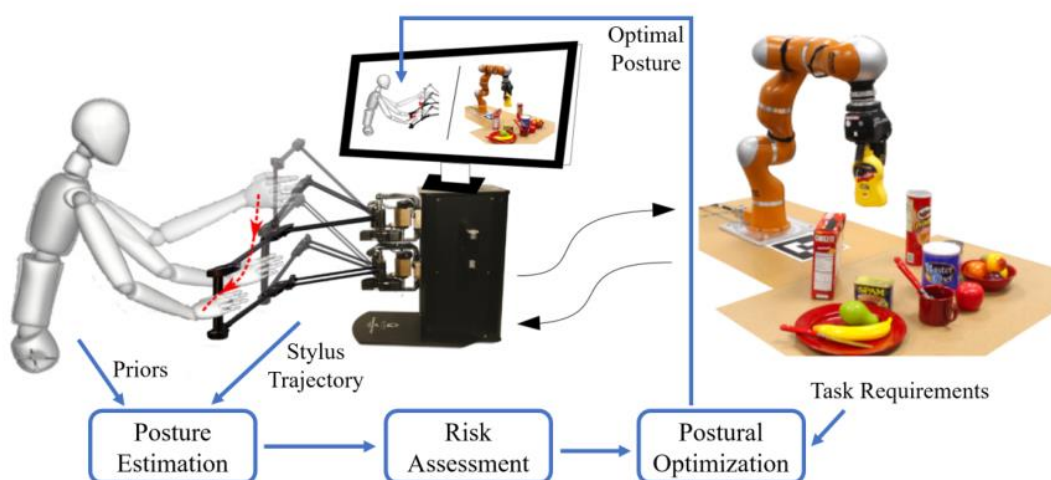


Figure 13. Pose estimation and optimization for teleoperated robots [37].

In 2022, depth estimation and Perspective-n-Points methods for monocular cameras were newly developed.

A robust monocular 3D detector was proposed by Zhuoling Li et al. [38]. In their paper, different methods for depth estimation and 3D object detection using the monocular camera are first reviewed and compared, including direct depth, depth based on height, depth from keypoint and their proposed method for estimating depth. Then, their own framework is given, which can produce different depth estimates for each object and combine reliable estimates to a single depth. In addition, they also propose a new method for establishing conditional 3D confidence. Their work has established a new SOTA for monocular 3D object detection.

The work of Hansheng Chen et al. then improves on the Perspective-n-Points monocular object pose estimation [39]. In their study, EPro-PnP is proposed to transform the non-minimizable deterministic PnP operations into a minimizable probabilistic layer, which enables end-to-end 2D-3D correspondence learning with unprecedented flexibility. In their paper, they fully discuss and compare the existing work on PnP and give the network of their research.

CHAPTER 3

METHODOLOGY

In this chapter, the methodology of this project is mainly described in detail.

In the first subsection, it is explained in detail how to use Maya software to create the corresponding 3D dental model on the computer according to the physical model and import it into MATLAB as an .obj file.

In the second subsection, the principle of communication between the upper and lower computer in the teleoperation system is described in detail, especially for the Raspberry Pi of this project, the information related to the communication between it and the computer, which is the upper computer of this project (including wireless communication information, communication protocol, etc.), and according to the remote control system of tooth cleaning proposed in this project, the implementation of the master side to indicate the control of the slave side according to the light on and off. The slave side is controlled according to the light on and off.

In the third subsection, the method of detecting and identifying the physical model of the tooth from different viewpoints is given. In this section, the project first captures and preprocesses the images of different viewpoints, and then trains them using the YOLOv2 model.

In the fourth subsection, the method to obtain the 3D rotation angles of realistic tooth from the real-time video is illustrated. This step includes mainly the calibration of the camera, the application of the ArUco code, the detection, the description and application about the Perspective-n-Point principle, and the transformation from the rotation vector to the rotation matrix to the Euler angles.

In the last section, which is the fifth subsection, the process of GUI design through MATLAB APP Designer is given, including the design of panels, the use of callback functions and the acquisition of camera images.

The following is the description of the hardware and software used in this project, as shown

in **Table 1**, **Table 2**, **Table 3**, and **Table 4**.

Table 1. The hardware equipment and their information.

Hardware and equipment	Version and other information	
Camera	Sensor Type	1 MP CMOS
	Highest Resolution	1280 (horizontal) \times 720 (vertical)
	Video frame rate	30/25 fps
	Lens	Fixed focus: 3.6 mm
	Video Output	USB 2.0
Calibration plate	Material	Aluminum oxide
	Model number	GC100-9*9
	Dimension	100 mm \times 100 mm
	Square side length	8 mm
	Pattern Array	9 \times 9
	Pattern size	72 mm \times 72 mm
	Precision	± 0.01 mm
LED lamps	Specification	5 mm
	Voltage	1.8 V – 2.4 V

Table 2. The operation system and program development environment.

Program development and operation environment	Version
MATLAB	R2022a
PyCharm	2022.1 (Professional Edition)
Visual Studio	2017
Maya	AUTODESK MAYA2023

Table 3. The operation system and program development environment.

Program development and operation environment	Version and other information	
The upper computer (Main operating system & Laptop)	CPU	11 th Gen Intel(R) Core (TM) i7-11800H_@_2.30GHz
	GPU	NVIDIA GeForce RTX 3060 Laptop GPU
	RAM	16.0 GB (15.7 GB available)
	System Type	Windows 11 64-bit OS x64-based CPUs
The lower computer (Raspberry Pi)	Model number	Raspberry Pi 4 Model B
OpenCV	Version	3.4
OpenCV-contrib	Version	3.4
Python	Version	3.8

Table 4. MATLAB add-ons.

Add-ons	Version
MATLAB Support Package for USB Webcams	22.1.0
MATLAB Support Package for Raspberry Pi Hardware	22.1.0
Image Processing Toolbox	11.5
Computer Vision Toolbox	10.2
Deep Learning Toolbox	14.4
Parallel Computing Toolbox	7.6
Image Acquisition Toolbox	6.6
Image Acquisition Toolbox Support Package for OS Generic Video Interface	22.1.0
Computer Vision Toolbox Interface for OpenCV in MATLAB	22.1.0
Deep Learning Toolbox Model for AlexNet Network	22.1.0

3.1 3D MODELLING AND IMPORTING

Maya is the world-class three-dimensional software for animation by Autodesk, Inc that contains many kinds of content. It is widely used in film and visual effects, animation production, and the game industry. It is also used in medicine, the military, and construction [40].



Figure 14. The icon of MAYA software [40].

This project uses Maya to build a 3D model corresponding to the solid tooth model in the computer and applies it to the app. This model includes upper and lower teeth and gums, and the shape of teeth in various positions is modified according to the solid model. This can ensure that the teeth in each place have a corresponding feature, convenient for subsequent recognition using the algorithm. The process of building this model is described in detail as follows:

- (1) Create the upper gum by taking a "pipe" from "Poly Modeling". Then the proper Subdivisions Axis value (28 for this project, since we generally have 14 upper teeth), Height value (0.77 for this project), and Thickness value (0.13 for this project, in order to make the relative thickness of the gum model roughly the same as a real one) are selected for this pipe. Finally, the rough prototype of the upper gum is done by deleting half of this pipe to keep the 14 Subdivisions Axis.
- (2) By changing the downward facing side of the remaining pipe, a corresponding space is provided for the creation of the upper teeth section. First, select all faces that are facing down. Close "Keep Faces Together". In the next step, set the "Offset" to a suitable value (0.03 for this project). Then open "Keep Faces Together" again and move the newly obtained face upwards by an appropriate distance, and those faces will be used in the next procedure to create the teeth. In order to make the junction between the gum and the tooth more fitting, the "Edge Loop Tool" is used here to get more changeable faces and move them to get a more matching model.

- (3) Rough tooth creation. Select all the new surfaces mentioned in the previous procedure, pull them down to the appropriate distance and move each new surface thus formed left and right towards the middle to form each tooth.
- (4) Tooth shape optimization. Use the "Edge Loop Tool" to form more tiny faces on the teeth and make the model shape more like the real teeth by panning these points, lines and faces. Finally, the teeth in different positions are fine-tuned according to the physical model, so that the teeth in different positions look different.
- (5) Form the upper and lower teeth and gum. First, use "Combine" in "Mesh" on the upper teeth and the already made teeth. Then use "Lattice" in "Deform" to adjust the shape of the whole model. Finally, adjust the "Scale Y" to -1 to get the corresponding lower teeth and lower gum, and then adjust them.
- (6) Finally, select the entire model to make sure all faces are connected, and select "Smooth" in "Mesh" to make more lines to improve the smoothness of the model, and then select "Triangulate" to make all the faces triangular, so that the model can be imported in MATLAB later. Then save and export the model as an .obj format.

After building the model, it needs to be imported into MATLAB for use in the app. And this method will show in 3.5.

3.2 TELECOMMUNICATION BETWEEN THE UPPER AND LOWER COMPUTER

As a card computer, Raspberry Pi has many powerful functions. In this project, Raspberry Pi, as the lower computer, communicates with the upper computer under the same WLAN through Secure Shell (SSH) communication protocol to send the doctor's instructions from the app of the upper computer to Raspberry Pi and control the level change of specific General Purpose Input Output (GPIO) pins on Raspberry Pi, to control the circuit connected to GPIO pins. This process can be displayed by the light on and off in some way.

Figure 15 illustrates the structure and interfaces of the Raspberry Pi used in this project. In this picture, the important interfaces and modules used in this project are marked. The **first** one marked is the SD card slot, which is located on the back of the Raspberry Pi board. The

SD/MicroSD card is an essential storage component for the Raspberry Pi, used to install the operating system and store data. In this project, the MALAB version of the Raspberry Pi operating system was chosen to be installed as the operating system used. The **second** one marked is the Wi-Fi and Bluetooth module. the Wi-Fi function allows the Raspberry Pi to wirelessly access the wireless network for interconnection, and the Bluetooth function allows the Raspberry Pi to connect with Bluetooth enabled devices (such as mouse, keyboard, joystick, etc.). This project uses the Raspberry Pi's Wi-Fi function to support its communication with the computer. The **third** one marked is the SOC chip. Raspberry Pi uses Broadcom BCM2711 chip as SOC chip, which integrates CPU, GPU, DSP and SDRAM memory, among which CPU and GPU share memory. The **fourth** one marked is GPIO pins. There are a total of 40 GPIO pins on the Raspberry Pi board and their functions are not identical. GPIOs can be used to connect various peripheral electronics and sensors by inputting/outputting level signals in order to control or monitor these devices. In the concept of this project, these pins are used to transmit signals to the different remote operation equipment that needs to be controlled by the slave side in this system. In this article, LEDs are used instead for these devices. The **fifth** is the Micro USB power connector, which is one of the main power supply methods for the Raspberry Pi and is rated at 5V.

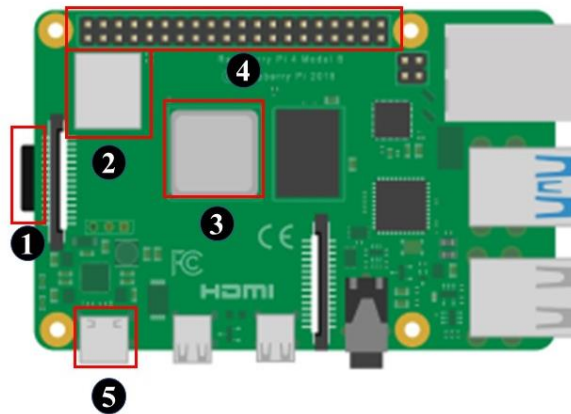


Figure 15. Modules and interfaces used in this project of Raspberry Pi 4 Model B.

3.2.1 Wireless connection between Raspberry Pi and computer

To enable the communication between Raspberry Pi and the computer in MATLAB application, the corresponding support package in MATLAB is required. After downloading the support package, first select the version of Raspberry Pi, download the appropriate mirror system for Raspberry Pi, then select "Connect to wireless network" in the network settings,

and then enter the SSID and Password of Wi-Fi to automatically assign an IP address for Raspberry Pi. Then install the system on the Raspberry Pi and check if the connection is successful. The communication schematic is shown in **Figure 16**.

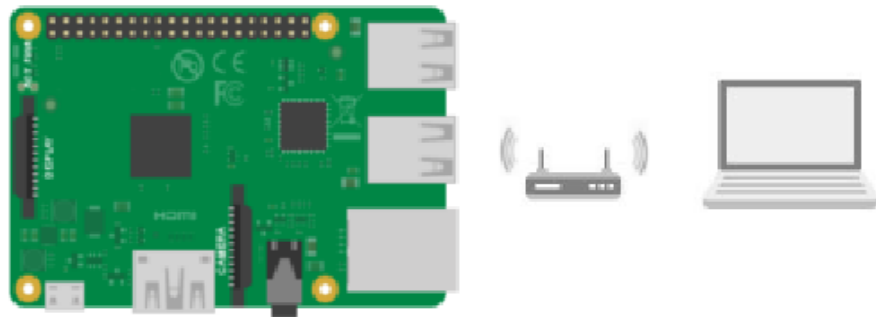


Figure 16. The Raspberry Pi and laptop wirelessly connected within the same Wi-Fi.

In this form of connection, the Raspberry Pi as a client will have a special IP address to connect with the server, and the communication protocol used is SSH protocol, as shown in **Figure 17**.

The screenshot shows the 'Hardware Setup' window with the 'Confirm Hardware Configuration' tab selected. It displays the following configuration details:

IP address	192.168.41.9
Host name	raspberrypi-3BmKrOfKqW
User name	pi
Password	raspberry

Below the table is a 'Test Connection' button. Underneath it is a checkbox labeled 'Connect via SSH, pinging 1 of 3', which is currently unchecked. To the right of the configuration fields, there is a 'What to Consider' section with instructions on how to proceed based on the test result, and a 'Note' section with two points about IP address output and mDNS.

At the bottom of the window are three buttons: '< Back', 'Cancel', and 'Next >'.

Figure 17. Information and SSH connection of the Raspberry Pi in this project.

3.2.2 Control of GPIO pins by the upper computer

In this subsection, we use four common GPIO interfaces for level output, use small lights to

indicate shim high and low, and use four grounded interfaces to form loops. In the upper MATLAB program, we first use the `raspi()` function to create the Raspberry Pi object. Then according to the situation, when the high-level output of the pin is needed, use the `writeDigitalPin()` function and change the parameter which represents the high or low level of the pin to 1, and when the low-level output of the pin is needed, change the corresponding parameter to 0. The pins used in this paper and the parts of the system they represent are shown in the **Figure 18**.

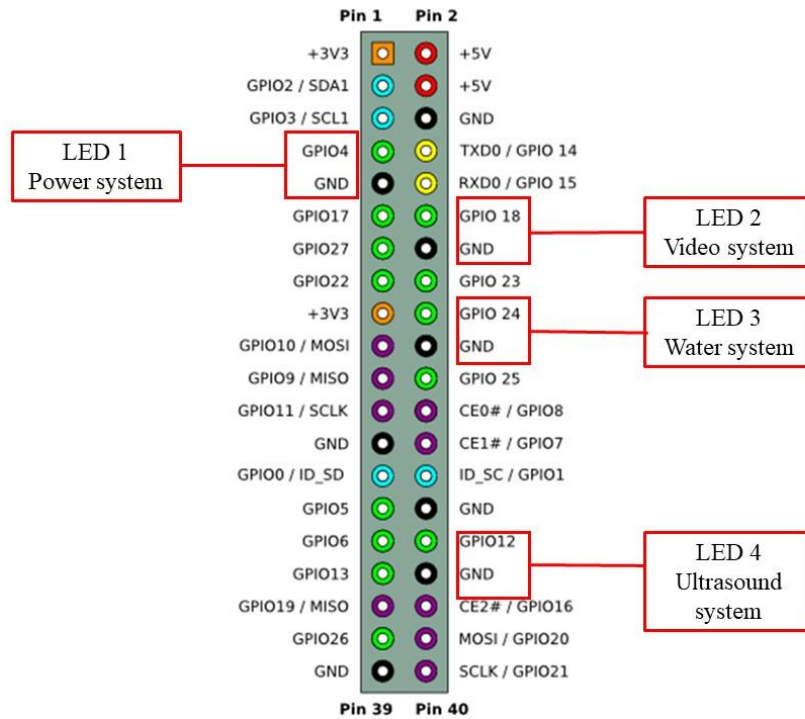


Figure 18. The pins used in this project and the parts represented.

3.3 DETECTION AND CLASSIFICATION OF THE PROJECTION IMAGES OF THE PHYSICAL MODEL

Prior to the dental cleaning, the doctor usually needs to observe the overall tooth of the patient. Therefore, the identification of the overall condition of the teeth is required in our teleoperation system. In this one part, this paper will identify the three aspects of the tooth model (front, left, and right) by the deep learning algorithm, mark which aspect is which while boxing the identified teeth in the picture or video, and output the coordinates of the

four corners of this box in order to intercept, save, and view the part of the tooth in the picture or video.

This work focuses on data acquisition, image pre-processing, model training, and testing and running in the App. The result of the tests and the application in the MATLAB APP will be given in 4.4.

3.3.1 Image preprocessing

The first step is the acquisition of images. Using Image Acquisition Explorer in MATLAB, images of the teeth model from three viewpoints at different distances from the camera were acquired using the USB camera. The resolution of each image is 1280×720 . **Figure 19** shows examples of the acquired images.

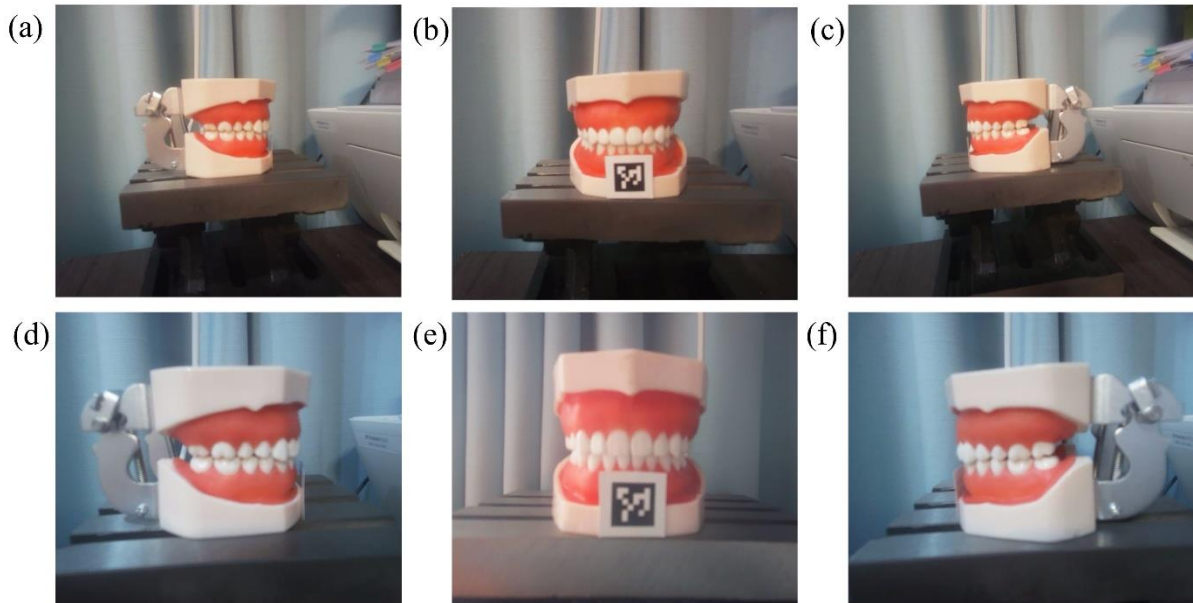


Figure 19. Examples of the acquired images. (a)(d) Images taken from left view with different distances. (b)(e) Images taken from front view with different distances. (a)(d) Images taken from right view with different distances.

The second step is the enhancement of the data. Since this scenario is special and there is no publicly accessible dataset, the quantity of the acquired images is low, and the teeth model is single, so for reducing the overfitting possibility of the deep learning model and to increase the effective sample quantity and the robustness of the learning model, randomly selected

some the acquired images in this project are processed by noise addition. Here, the `imRefiensalt()` function is used in MATLAB to add the pepper noise to the images, and Parallel Pool is started to perform the parallel operation to speed up the image processing. The comparison of the image before and after adding noise is shown in **Figure 20**.

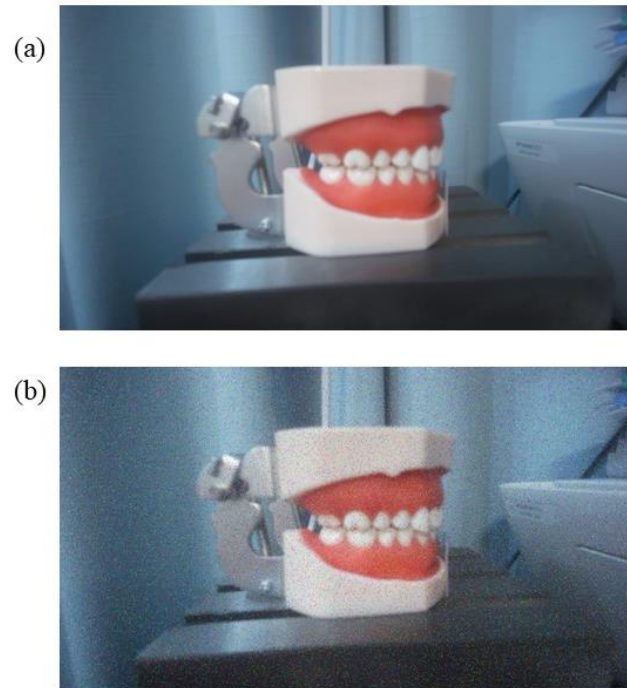


Figure 20. The comparison of the image before and after adding noises. (a) Before adding noises; (b) After adding noises.

The third step requires labeling the different aspects of the teeth model for use in the deep learning model. Here the different aspects are labeled using Image Labeler in MATLAB, and a file containing the labeling information is exported. An example of the image with label is shown in **Figure 21**.

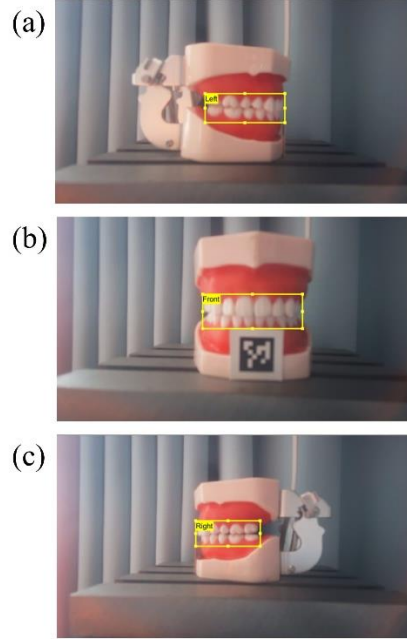


Figure 21. Examples of the image with label. (a) Image labeled with “Left”. (b) Image labeled with “Front”. (c) Image labeled with “Right”.

3.3.2 YOLOv2 model training

3.3.2.1 YOLOv2 algorithm

Traditional image processing algorithms usually have high accuracy, but they are generally hard to be used in real-time systems because of their slow computing speed. And for solving the problem of slow speed, the YOLO algorithm is proposed trendily, which combines target detection and classification and unifies joint training, allowing regression detection and classification in the output layer, solving detection and classification by regression, and significantly improving detection speed. The YOLO detection process can be simply divided into three steps. (1) Resize to 448×448 for the input image, (2) Run an image on the single convolutional network, and (3) Threshold detection results through the confidence level of the model [41]. The YOLO processing schematic is shown in **Figure 22**.

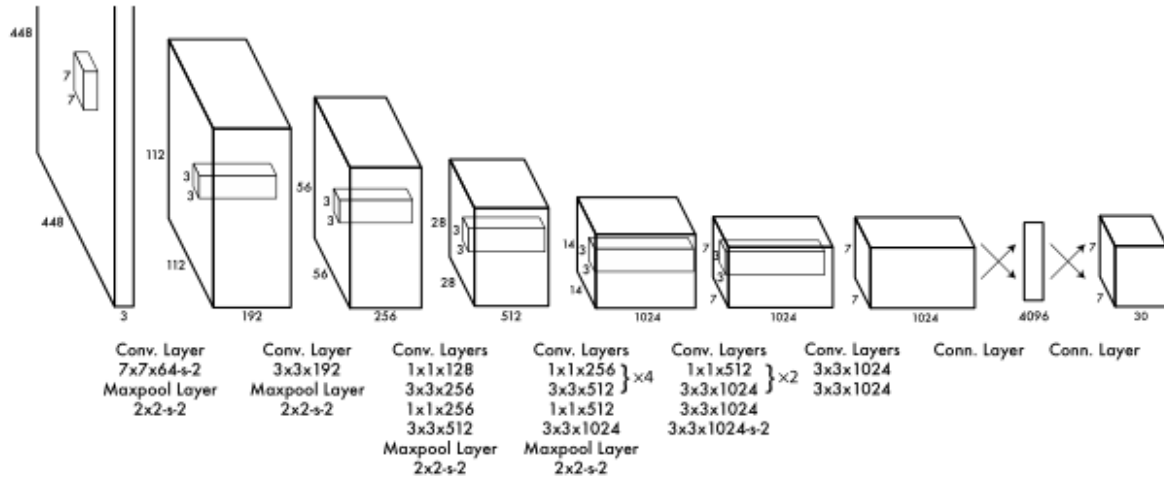


Figure 22. The YOLO processing schematic [41].

YOLOv2 introduces a series of improvements based on YOLO. YOLOv2 has added a "batch normalization" layer after every convolutional layer, which improves the model's convergence speed, reduces the model's overfitting, and thus improves the training speed of the network, increasing the mAP by 2.4%. YOLOv2 uses a high-resolution classifier to increase the resolution of the pre-trained network to 448×448 and fine-tunes the detection data using high-resolution, leading to a 4% increase in mAP. YOLOv2 uses convolution and anchor boxes to predict the bounding boxes so that YOLOv2 can predict thousands of bounding boxes, enabling a significant increase in recall and clustering analysis of bounding boxes using K-means clustering [42].

3.3.2.2 Network training process

Visualize the bounding box after importing the label information and image position data. First, using the `table2array()` function, `reshape()` function, and `cell2mat()` function, the label information of all the images (Coordinates of the upper left corner of the bounding box, width and height of the bounding box) is converted into a matrix format, and then the length to width ratio "Aspect ratio" and the area of the bounding box "Area" are calculated. The result is shown in **Figure 23**.

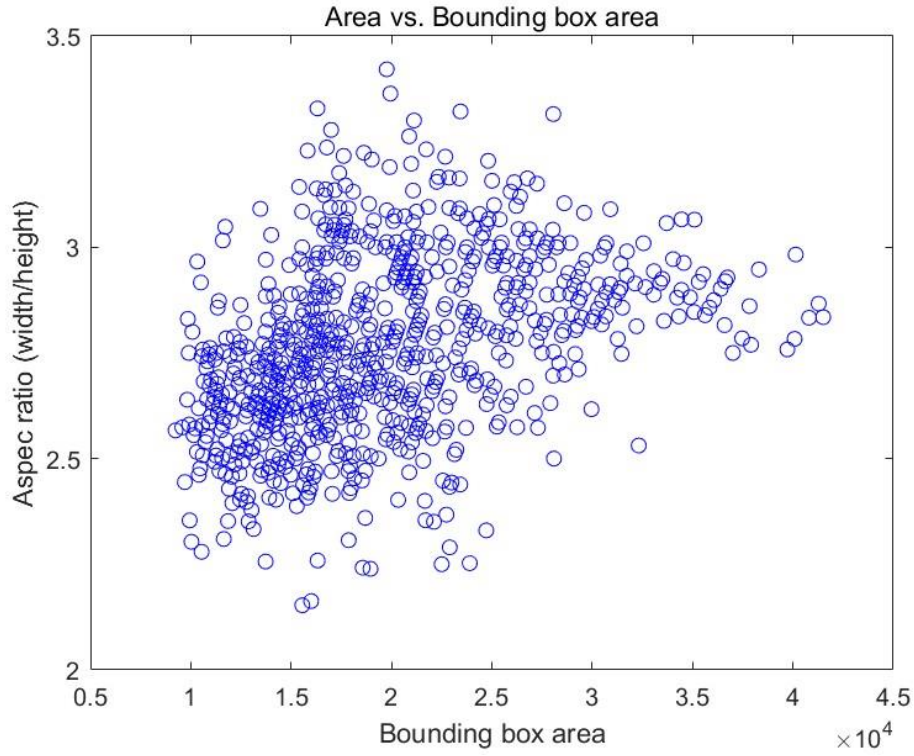


Figure 23. Area vs. Bounding box area.

Confirm the quantity and sizes of the Anchor boxes and the mean Intersection over Union (IoU). The convolutional network has translation invariance, so only the width and height of Anchor boxes need to be calculated. The K-means clustering method can be used for manually clustering the training set in advance, and forty clustering centers are chosen in this network to obtain forty Anchor boxes. The result of clustering is indicated in **Figure 24**.

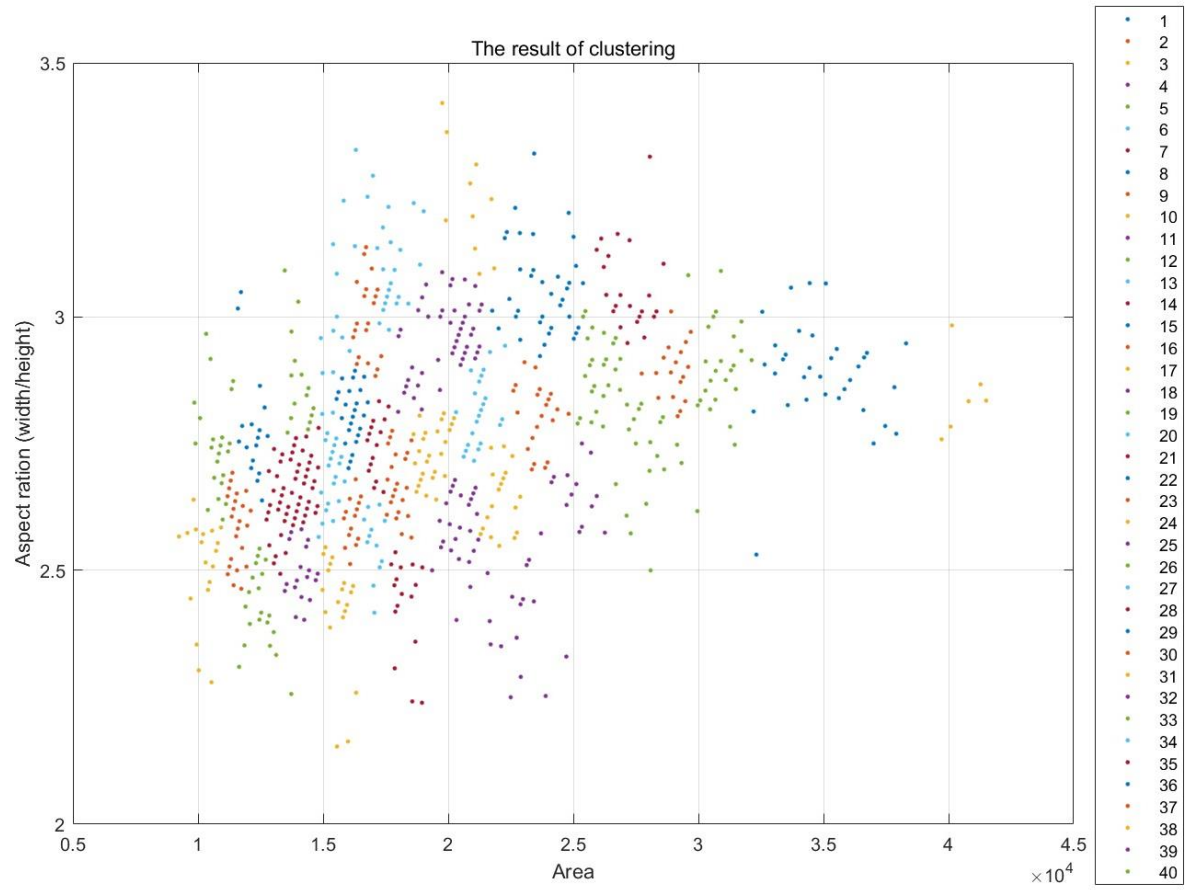


Figure 24. The bounding boxes distribution after clustering.

The sizes of the Anchor boxes are shown in **Table 5**.

Table 5. Anchor boxes sizes (Unit: pixel).

Width	Height
273	90
219	82
340	120
247	83
175	71
209	82
287	95
313	107
204	79
229	84
230	89
278	101
233	76
195	73
184	67
258	92
195	80
236	97
276	95
246	87
215	79
213	76
289	100
161	65
232	80
300	103
203	76
211	86
322	112

The training process is mainly to adjust the learning rate, decay coefficient, the number of samples selected for one training and the impulse and other parameters. The training

parameters set in this paper are shown in **Table 6**. The detailed training process can be found in 1.

Table 6. The training parameters.

Parameter	Value
Batch size	32
Classes	3
Learning rate	0.0001
Max epochs	1000
Shuffle	every-epoch
Verbose frequency	30

3.4 SOLVING THE 3D ROTATION ANGLE PROBLEM OF THE TEETH FROM 2D VIDEO

The dental scaling process is often combined with the change of the patient's pose, thus creating a certain angle between the camera and the teeth. In this part, the ArUco marker from the OpenCV library is used to estimate the pose of the teeth model and to calculate the Euler angles of its rotation. This section includes camera calibration, marker generation, marker detection, and solving the 3D rotation angle of the tooth using the Perspective n-Point principle and the rotation matrix to Euler angles transformation.

3.4.1 Camera calibration

3.4.1.1 Basic principles of camera calibration

During measurement of images as well as machine-vision applications, for determining the 3D position of a particular point on a spatial object surface along with the corresponding point within the image, it is necessary to establish the geometric model relationships for imaging by the camera, and the parameters of these geometric models refer to the camera parameters. These parameters have to be acquired by experimentation as well as by

calculation under the majority of conditions, and solving the parameters is a process known as camera calibration, which is the process of the world coordination system to the image coordination system and the final projection matrix. The camera parameters calibration is one of the most important parts, where calibration results in accuracy and the stability of the algorithm of calibration have a direct impact on the accuracy of the camera work results. Therefore, it is a prerequisite for the camera calibration to do a good job in the subsequent work, and the improvement of the accuracy of calibration refers to the key of the research work. The purpose of the calibration is for the camera's internal reference, external reference, aberration parameters.

Three basic coordinate systems are involved in camera calibration. (1) World coordinate system: a user-defined three-dimensional world coordinate system introduced for the purpose of describing the object's position from reality; (2) Camera coordinate system: established for the camera, defined to represent the object's position from the camera's point of view, and serves as an intermediate bridge from the world coordinate system to the image/pixel coordinate system; (3) Image coordinate system: Presentation describes object projection during imaging on camera from camera coordinate system to image coordinate system to facilitate the acquisition of further coordinates of the pixel coordinate system. Generally speaking, there are two parts to the calibration process: In the first step, the conversion of the world to the camera coordinate system, which is a step from three-dimensional points to three-dimensional points, and includes parameters such as R and t (external camera reference); the second step is the conversion of the camera to the image coordinate system, which is a step from three-dimensional points to two-dimensional points, and includes parameters such as K (internal camera reference). **Figure 25** shows the model of the three coordinate systems.

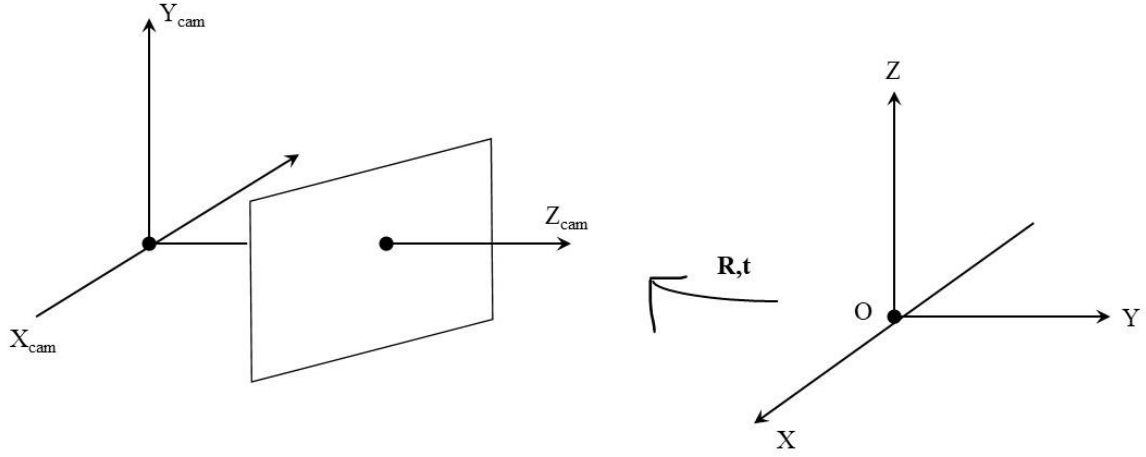


Figure 25. The model of the three coordinate systems.

3.4.1.2 Calibration using checkerboard grid

The method of calibration utilizes a checkerboard calibration board for getting pixel coordinates (u, v) of each corner point using the corresponding corner point extraction algorithm (e.g., Harris corner point) [43]. This calibration method fixes the world coordinate system on the checkerboard grid, then the physical coordinates of any point on the checkerboard grid $W = 0$. In view of the fact that the calibration plate world coordinate system is determined artificially beforehand and that the dimensions for each grid on the calibration plate are already known, we can calculate each corner point's physical coordinates $(U, V, W) = 0$ in the world coordinate system.

Using this information: the pixel coordinates (u, v) of each corner point, each corner in the world coordinate system, and the physical coordinates $(U, V, W) = 0$, we can perform the camera calibration and get the internal as well as external reference matrix, and aberration of the camera parameters.

The 2D image point is

$$\mathbf{m} = [u, v]^T \quad (3.1)$$

The 3D space point is

$$\mathbf{M} = [X, Y, Z]^T \quad (3.2)$$

The homogeneous coordinates are

$$\tilde{\mathbf{m}} = [u, v, 1]^T, \tilde{\mathbf{M}} = [X, Y, Z, 1]^T \quad (3.3)$$

Describe the mapping of spatial coordinates to image coordinates

$$s\tilde{\mathbf{m}} = \mathbf{K}[\mathbf{R}\mathbf{t}]\tilde{\mathbf{M}} \quad (3.4)$$

$$\mathbf{K} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Where s is the scale factor from the world coordinate system to the image coordinate system; \mathbf{K} is the camera internal reference matrix; (u_0, v_0) is the image principal point coordinate; α , β are the fusion of focal length and pixel aspect ratio; γ is the radial distortion parameter. The chessboard grid used in this paper is shown in **Figure 26**. And the board is photographed from different angles and distances using the camera, the example of the process of calibration is shown in **Figure 27**.

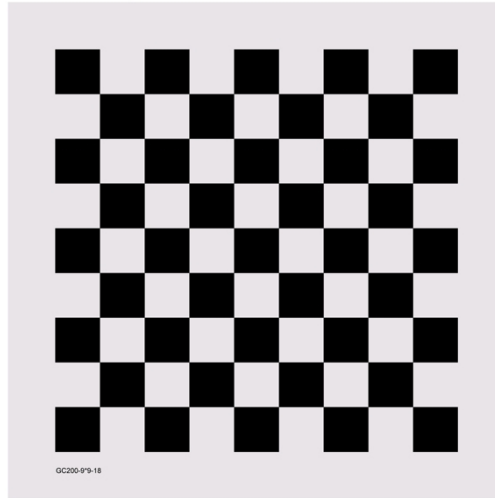


Figure 26. The chessboard used in this project.

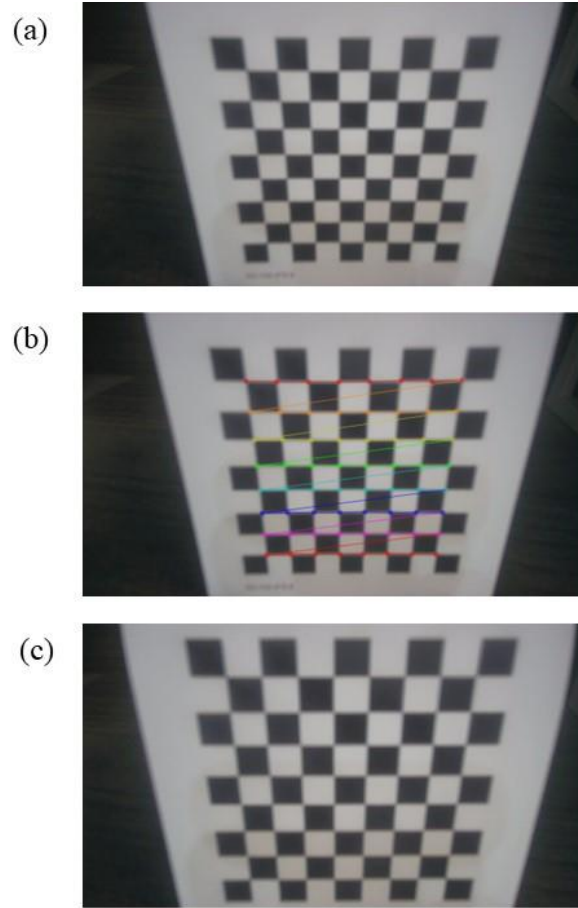


Figure 27. The example of the process of calibration. (a) The image acquired; (b) Corner points detected during calibration; (c) Image after distortion removal.

3.4.2 ArUco marker generation

An open-source augmented reality library, ArUco features a range of augmented reality (AR) applications, from visual calibration to artificial marker generation, camera pose estimation, and more [44][45][46]. ArUco markers are produced from ArUco, which consists of a square artificial marker with a surrounding black edge and an internal Hamming code matrix. It is widely used in the AR field to estimate the camera pose because of its advantages in high-level robustness and detection speed. Each marker has a unique internal code and belongs to different marker dictionaries depending on its internal code bits and encoding method.

The marker with the ID of 25 in the "DICT_6X6_250" dictionary is chosen as the marker for the detection and pose estimation in this paper, as shown in **Figure 28**.

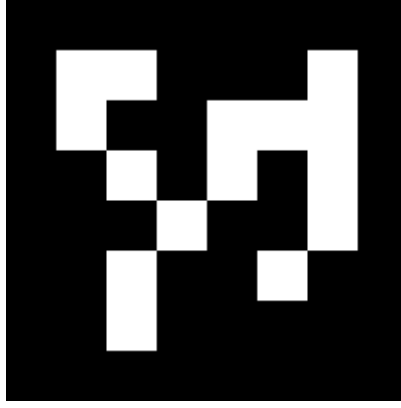


Figure 28. The marker used in this project (ID: 25).

3.4.3 Detection and identification principle of the ArUco marker

The detection and recognition of the ArUco marker is a process of image processing. Given an image containing an ArUco marker, by processing, detecting, and recognizing the picture, we can finally get the ID of the marker and the pixel coordinates of the corresponding four corner points in the image. The following steps accomplish the process of marker detection and recognition [47][48][49]:

- (1) Threshold segmentation. The original image should be converted into a grayscale image and segmented by the locally adaptive thresholding method.
- (2) Extraction of contours. The contour extraction is performed using the contour tracking algorithm proposed by Suzuki for the thresholded image, and then the polygon approximation is performed using the Douglas-Peucker algorithm.
- (3) Contour filtering. Since the actual marker is enclosed in the rectangular contour, the contours that do not approximate the quadrilateral will be discarded. Then the corner points of the quadrilateral are sorted counterclockwise, the overly close rectangles will be deleted, and the outermost boundary will be kept.
- (4) Obtain the encoding. First, a perspective transformation is performed to obtain the front view of the marker outline. Then the front view is binarized using the Otsu algorithm to separate the white and black bits. The front view is decomposed into a black and white lattice according to the size and boundary of the marker used, and the number of black

and white pixels falling in each lattice is counted to determine whether the bit is 0 or 1.

- (5) Encoding recognition. Based on the obtained binary encoding information, we search in the given dictionary. If the match is successful, we can get the ID, nRotations of the current marker, and the dictionary name to which the current marker belongs.
- (6) Corner point adjustment. The identified marker is refined by sub-pixel interpolation, and then the order of the corner points of the marker is adjusted to the original order.

3.4.4 Estimation of the camera's pose using the Perspective-n-Point principle

Perspective-n-Point (PnP) has been developed as an approach to solving three-dimensional to two-dimensional point pairs movement. It characterizes the estimation of the camera pose with known coordinates of three-dimensional spatial points as well as their two-dimensional projection positions. In the PnP problem, in an image, we only must know the 3D spatial coordinates of at least three of the points to be used to estimate the camera motion as well as the camera pose (at least one additional point is needed to verify the result). The PnP problem is the problem of obtaining the camera or object pose by computing the projection relationship between n feature points in the world and the corresponding n image points in the image. The pinhole camera imaging model and PnP schematic shows in **Figure 29**.

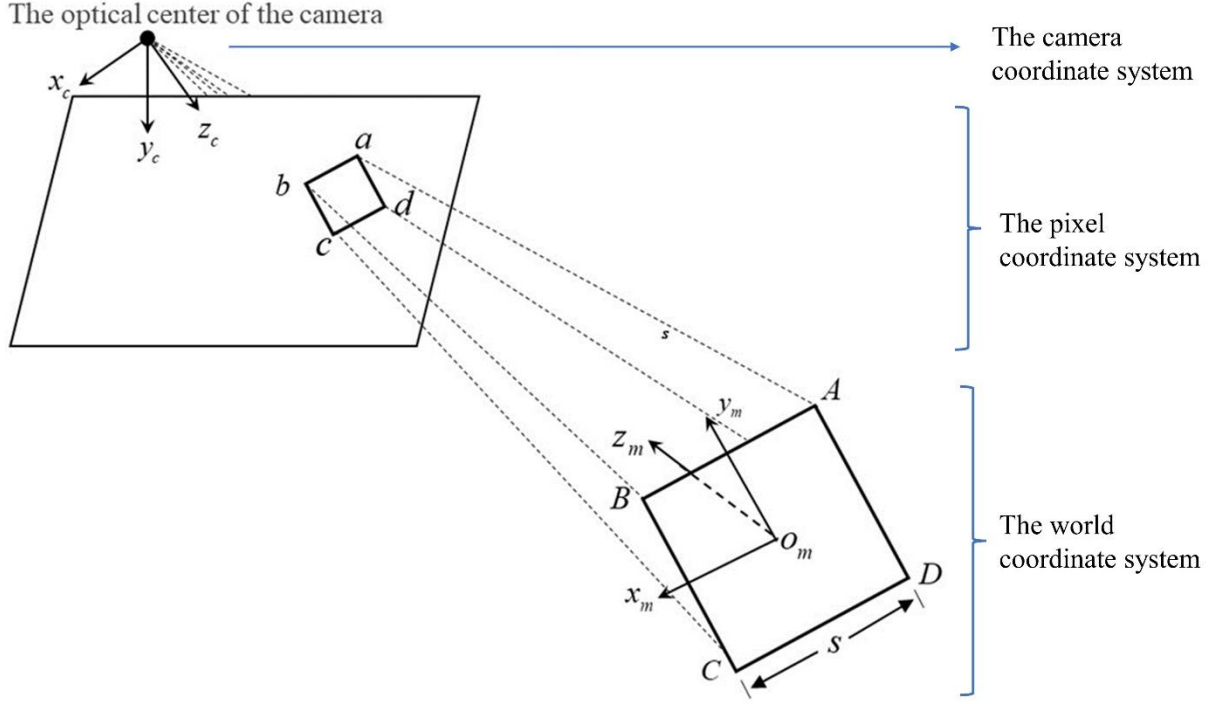


Figure 29. The pinhole camera imaging model and PnP schematic. From top to bottom are the camera coordinate, the pixel coordinate, and the world coordinate systems.

In **Figure 29**, the coordinate system of the ArUco marker is the world coordinate system, then the three-dimensional spatial coordinates of the four corner points of the marker are $A = (-s/2, s/2, 0)^T$, $B = (s/2, s/2, 0)^T$, $C = (s/2, -s/2, 0)^T$, $D = (-s/2, -s/2, 0)^T$. Assume that the pixel coordinates of their corresponding projection points in the image are $a = (u_a, v_a)^T$, $b = (u_b, v_b)^T$, $c = (u_c, v_c)^T$, $d = (u_d, v_d)^T$. According to (3.5), The internal reference matrix of the camera is now known. Then, taking **Figure 29** as an example, the conversion relationship between 3D spatial coordinates and pixel coordinates is

$$s \begin{bmatrix} a \\ 1 \end{bmatrix} = s \begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = \mathbf{K} [\mathbf{R} | \mathbf{t}] \begin{bmatrix} A \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} -s/2 \\ s/2 \\ 0 \\ 1 \end{bmatrix} \quad (3.6)$$

Where $\mathbf{R} = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, $\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$ are the rotation matrix and the parallel

vector, which is the camera external parameter for point A. The camera's external reference

converts points in the world coordinate system to points in the camera coordinate system, while the camera's internal reference converts points in the camera coordinate system to points in the pixel coordinate system. To simplify the representation, the internal reference

(which is a fixed value exclusive to the camera) is ignored here, and let $\mathbf{P}_A = \begin{bmatrix} A \\ 1 \end{bmatrix}$,

$$\mathbf{T} = [\mathbf{R} | \mathbf{t}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}, \text{ then (3.6) can be presented as}$$

$$s \begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{bmatrix} \begin{bmatrix} A \\ 1 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \mathbf{P}_A \quad (3.7)$$

Then use the last line to eliminate the s ($s = \tau_3 \mathbf{P}_A$), two constraints will be obtained:

$$u_a = \frac{\tau_1 \mathbf{P}_A}{\tau_3 \mathbf{P}_A}, v_a = \frac{\tau_2 \mathbf{P}_A}{\tau_3 \mathbf{P}_A} \quad (3.8)$$

So, there is

$$\begin{cases} \tau_1 \mathbf{P}_A - \tau_3 \mathbf{P}_A u_a = 0, \\ \tau_2 \mathbf{P}_A - \tau_3 \mathbf{P}_A v_a = 0. \end{cases} \quad (3.9)$$

It can be seen, each eigen point provides two linear constraints on \mathbf{T} , and with twelve unknowns in \mathbf{T} , six pairs of matching points are theoretically required to solve the matrix \mathbf{T} .

And the rotation matrix is a unitary orthogonal array with six constraints of its own:

$$\begin{cases} \mathbf{r}_1^T \cdot \mathbf{r}_2 = \mathbf{r}_1^T \cdot \mathbf{r}_3 = \mathbf{r}_2^T \cdot \mathbf{r}_3 = 0 \\ \|\mathbf{r}_1\| = \|\mathbf{r}_2\| = \|\mathbf{r}_3\| = |\mathbf{R}| = 1 \end{cases} \quad (3.10)$$

In this case, the three corner points of the marker along with the six constraints of the rotation matrix are used to calculate \mathbf{R}, \mathbf{t} .

The above explains the basic principle of the PnP problem theoretically and discusses the feasibility of estimating camera poses with a single ArUco marker. [50] has a more detailed explanation of the PnP principle for the QR code marker.

3.4.5 From Rotation vector to Euler angles

In **3.4.4**, the principle of object pose estimation by ArUco marker is explained based on the

PnP principle. After this step, the rotation vector and translation vector of the object can be obtained, but to get the rotation angle of the object in 3-dimensional space, the rotation vector needs to be converted into the corresponding Euler angles. The following is the procedure for resolving the Euler angles.

When dealing with three-dimensional rotation problems, they are usually represented by rotation matrices. The multiplication of a vector with a rotation matrix corresponds to that vector being rotated somehow. In addition to using a rotation matrix description, rotations can also be described by rotation vectors. Rotation vectors and rotation matrices can be converted by the Rodrigues's Formula:

$$\theta = \sqrt{r_x^2 + r_y^2 + r_z^2} \quad (3.11)$$

$$\mathbf{n} = \mathbf{r} / \theta \quad (3.12)$$

$$\mathbf{R} = \cos(\theta)\mathbf{I} + (1 - \cos(\theta))\mathbf{nn}^T + \sin(\theta) \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} \quad (3.13)$$

Where \mathbf{r} is the rotation vector, and $\mathbf{r} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$, $\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$, $\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. So far, the rotation matrix is obtained.

Next, the rotation matrix needs to be used to solve for the Euler angles. In the representation of Euler angles, the order of yaw, pitch, and roll influences the rotation result. That is, given a set of Euler angle values, the final orientation of the rigid body is different when rotated in the order of yaw-pitch-roll and in the order of yaw-roll-pitch. In this paper, the rotation order is pitch \rightarrow yaw \rightarrow roll order, and the corresponding coordinate system and the corresponding angle change of the tooth model are shown in **Figure 30** and **Figure 31**.

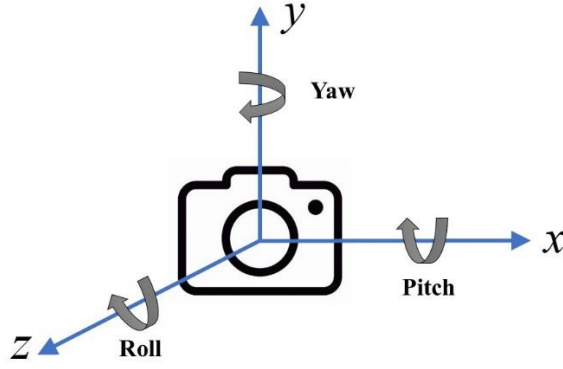


Figure 30. Coordinate representation of Euler angles.

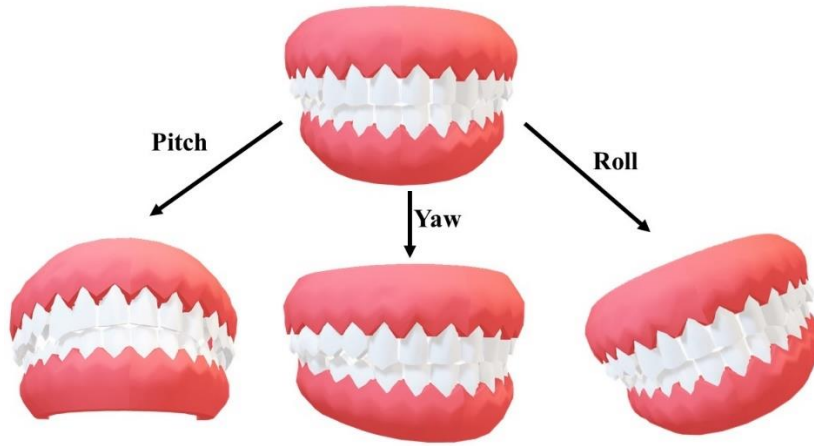


Figure 31. Description of pitch, yaw and roll angles of the teeth model.

For the case of rotation around the x-axis only, find only the pitch angle, this angle is represented as γ , and the relationship between the rotation matrix and the Euler angles is

$$\mathbf{R}_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix} \quad (3.14)$$

For the case of rotation around the y-axis only, find only the yaw angle, this angle is represented as β , and the relationship between the rotation matrix and the Euler angles is

$$\mathbf{R}_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \quad (3.15)$$

For the case of rotation around the z-axis only, find only the roll angle, this angle is represented as α , and the relationship between the rotation matrix and the Euler angles is

$$\mathbf{R}_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.16)$$

When the angles of these three rotations are sequentially multiplied, the expressions for the rotation matrix and Euler angles with these three angles can be obtained

$$\begin{aligned} \mathbf{R}(\alpha, \beta, \gamma) &= \mathbf{R}_z(\alpha) \mathbf{R}_y(\beta) \mathbf{R}_x(\gamma) \\ &= \begin{pmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{pmatrix} \end{aligned} \quad (3.17)$$

Combining equations (3.13) and (3.17), the pitch, yaw, and roll angles of the teeth can be calculated now.

3.5 GUI DESIGN

To develop a simple platform suitable for doctors to operate robots to wash teeth remotely, MATLAB APP designer is chosen to create a convenient environment in the computer in this project. The APP created by MATLAB APP Designer can get an effect that doctors can control the operation of the entire system by tapping a few specific keys on the keyboard, including the power supply system, ultrasonic system, water system, video system, and the movement of the robot in different directions.

In this project, the functions to be realized by MATLAB APP and the methods of achieving the requirements are as follows:

- (1) To be able to display the video taken by the camera connected to the computer in real time, and based on that, to implement the required functions, such as the recognition of different views or angles, for each frame of the video through algorithms. In order to achieve this function, the UIAxes component should be added in the design section of APP Designer, and the webcam() function should be used in the code section to read out the live video from the camera, and the snapshot() function should be used to extract individual frames, and the image() function, axis() function and preview() function are used to display them on the corresponding UIAxis.
- (2) Representing different parts of the system with different components and adding the corresponding callback functions to enable them to be changed on the GUI with the

operator's operation. To achieve this, a signal lamp component is added to the design window as a power lamp to represent the power supply, and single button group components are used to represent the Video system, Water system and Ultrasound system respectively. The trimmer and linear meter components are also used to represent the robot's motion in 3D space (X, Y, Z axis translation). All these components are programmed by callback functions for the use of the operator to control.

- (3) Implement the connection to the Raspberry Pi and change the level of the corresponding GPIO pins on the Raspberry Pi through the operation on the GUI. After completing the connection to Raspberry Pi, it is necessary to be able to operate the Raspberry Pi as the lower computer directly in the GUI. At this time, the information of Raspberry Pi is stored in the MATLAB workspace, and custom functions are needed to operate on the GPIO pins of Raspberry Pi and import the functions into the code section of the GUI to ensure that when the operator presses certain buttons in the GUI or presses certain keys on the keyboard, it can make the level value of the corresponding GPIO pins on the Raspberry Pi change. Here, according to the operator's desired state of the Power lamp, Video system, Water system, and Ultrasound system, the level of the GPIO pins can be controlled by pressing the corresponding keys on the keyboard and indicated by LEDs.
- (4) Implement the import and view of the 3D model and enable the operator to control the perspective of the 3D model in real time, such as dragging with the mouse and changing the perspective of the 3D model according to the angle of the teeth in the video. Firstly, a coordinate area needs to be created in the design view of APP Designer to display the 3D model. Then the faces and vectors of the .obj model are read in the code view. After that, the model color and other information are edited and represented in the axis area. Finally, the view is edited using the view() function and rotate3d is on.

CHAPTER 4

RESULTS AND ANALYSIS

In this chapter, the results will be presented and analyzed.

First, the connection and control of the upper and lower computers are shown.

Then the detection of the different faces of the teeth model and the automatic drawing of the teeth regions by the deep learning algorithm is shown, this detection includes the identification on the test set images and in the live video.

Then the calculation of the rotation angles of the mouth is shown. This demonstration includes the identification of the marker on the tooth in the real-time video, the estimation of the oral posture, the rotation matrix, and Euler angles are given.

After that, the MATLAB APP is demonstrated. It includes the display of the live video by keyboard, the control of the Raspberry Pi, and the change of the corresponding controls, as well as the display of the 3D model and the change of the view angles.

Finally, quantitative characterization results will be obtained using the results obtained in this project. The performance of the deep learning model on the test set and the range and error of the angle estimation are included. The resulting data will also be analyzed.

4.1 THE CONNECTION BETWEEN THE LAPTOP AND RASPBERRY PI

The connection between the upper computer and the lower computer is an essential part of communication. Here, the computer is used as the upper computer, and the Raspberry Pi is used as the lower computer, and they are connected to each other via the wireless network.

The status of the connection can be viewed both on the computer and by the level change of

the GPIO pins on the Raspberry Pi.

The connection of the Raspberry Pi displayed from the computer side is shown in **Figure 32**.

The information about the Raspberry Pi included in the figure includes the IP address of the Raspberry Pi, the name, the GPIO pins available, and other information.

```
DeviceAddress: '192.168.2.17'  
Port: 18734  
BoardName: 'Raspberry Pi 4 Model B'  
AvailableLEDs: {'led0'}  
AvailableDigitalPins: [4, 5, 6, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]  
AvailableSPIChecks: {'CE0', 'CE1'}  
AvailableI2CBuses: {'i2c-1'}  
AvailableWebcams: {}  
I2CBusSpeed: 100000  
AvailableCANInterfaces: {}
```

Figure 32. Information of the Raspberry Pi seen from the computer side.

And four different LED lamps are connected to different GPIO pins to show this change physically from the Raspberry Pi side. They represent power supply, video, water, and ultrasonic systems from top to bottom. The light on indicates that the system is on, and the light off suggests that the system is off. **Figure 33** shows that the situation of one LED is lightened, and all the four LEDs are lightened to indicate the level value changes of the GPIO pins so that we can see the connection and control of the upper and lower computers directly.

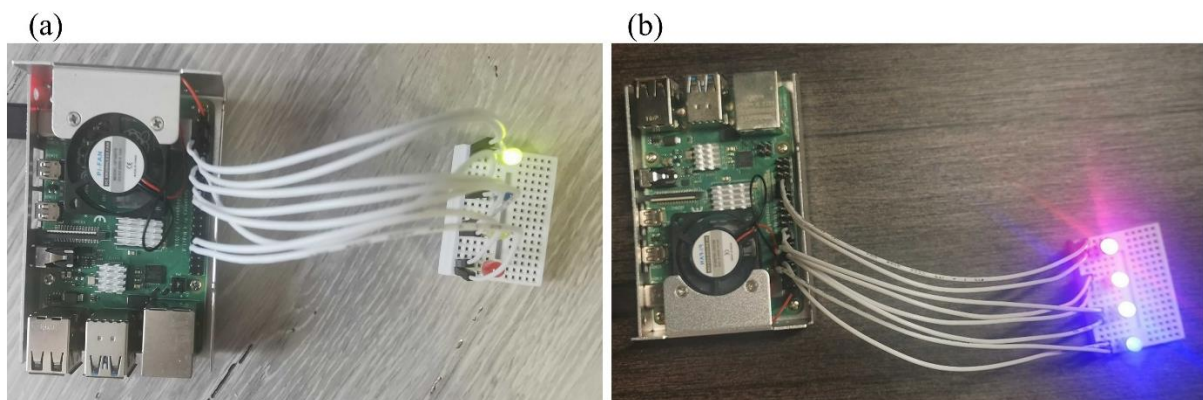


Figure 33. (a) one LED is turned on, (b) All the LED lights are turned on.

4.2 DETECTION OF THE DIFFERENT VIEWS OF THE REAL TEETH MODEL

4.2.1 Detection on the test set

Here, after training the training set, the model is tested using a test set of 300 images from the three views: left, front and right. The examples of the respective results are shown in **Figure 34**. In this image, the teeth model is identified in different perspectives from left to the right, and the area of the teeth and the label which this perspective attributes to (Right, Left, Front) are framed.

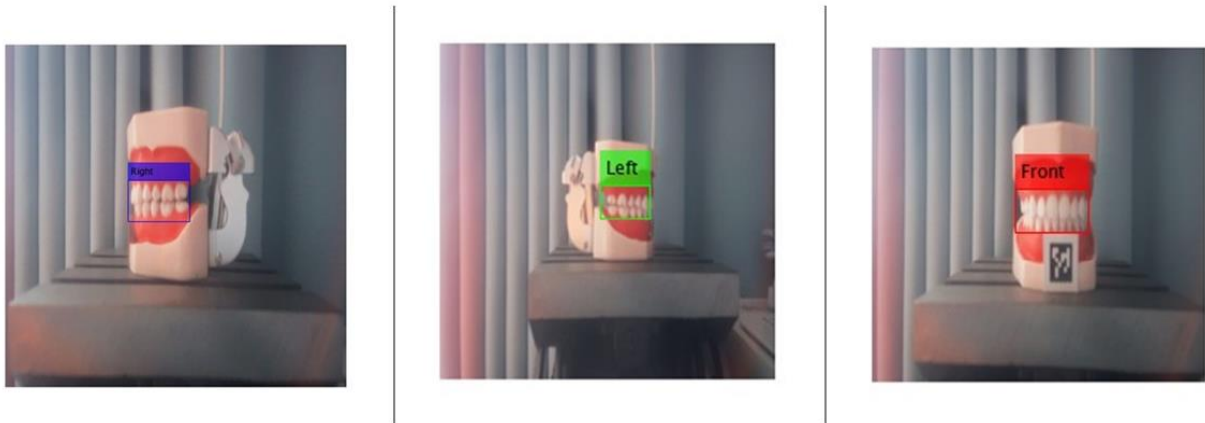


Figure 34. Instance detection results from the test set.

4.2.2 Testing in live video

A MATLAB APP is developed to validate the recognition of the model in real-time video in this project. As shown in **Figure 35**, By pressing the button "Button", it is able to import live video in the axis area of the APP and detect the teeth model and its different profiles. This demonstrates the effectiveness of the model in real-time video.

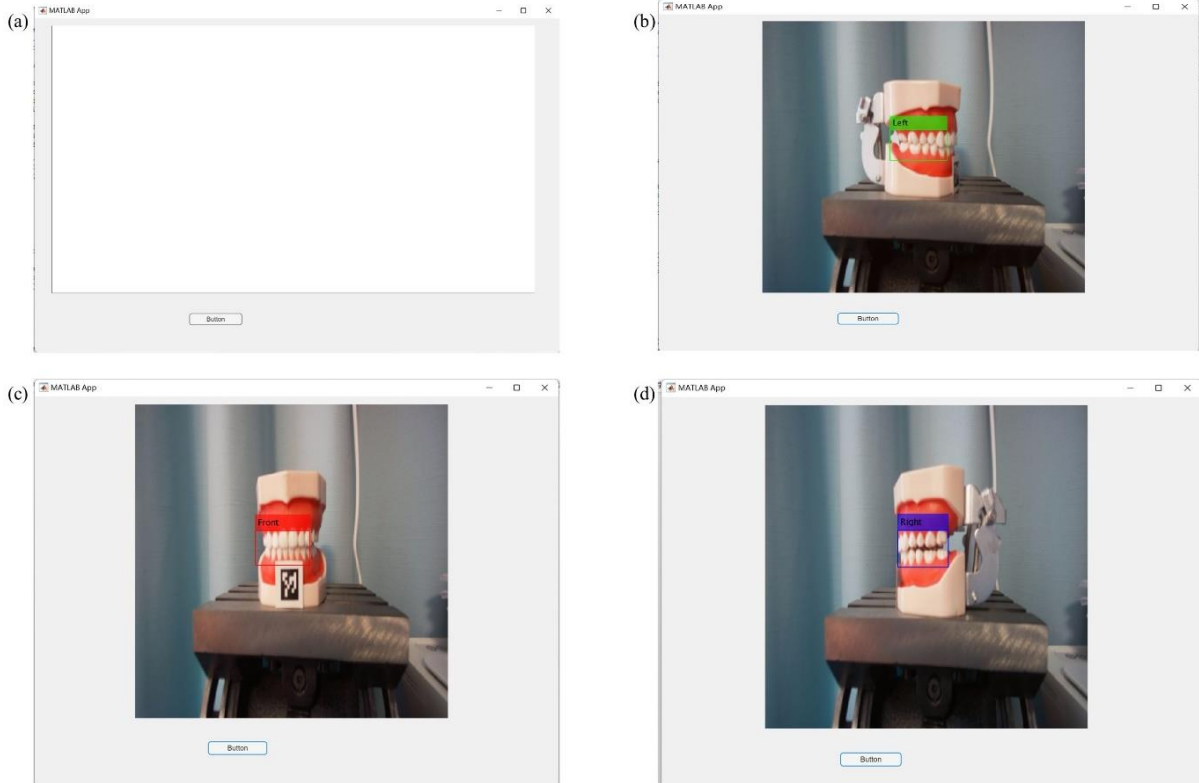


Figure 35. (a) The APP for detecting teeth in live video before pushing the “Detect” button. After pushing “Detect” button, according to the view of the camera, the blank axis area will show: (b) The detection result of the left side. (c) The detection result of the front side. (d) The detection result of the right side.

4.3 THE 3D ROTATION ANGLES OF THE TEETH MODEL

4.3.1 The detection of the marker in the real-time video

Since this project requires real-time feedback, real-time detection tasks must be completed. Here, the tooth model, also referred to as the ArUco marker in this project, is detected in the real-time video acquired from the camera. The results are shown in **Figure 36**. This figure is a screenshot of the detection process, which detects both the marker's ID and its rotation and translation vectors, providing a prerequisite for the calculation of the angles later.

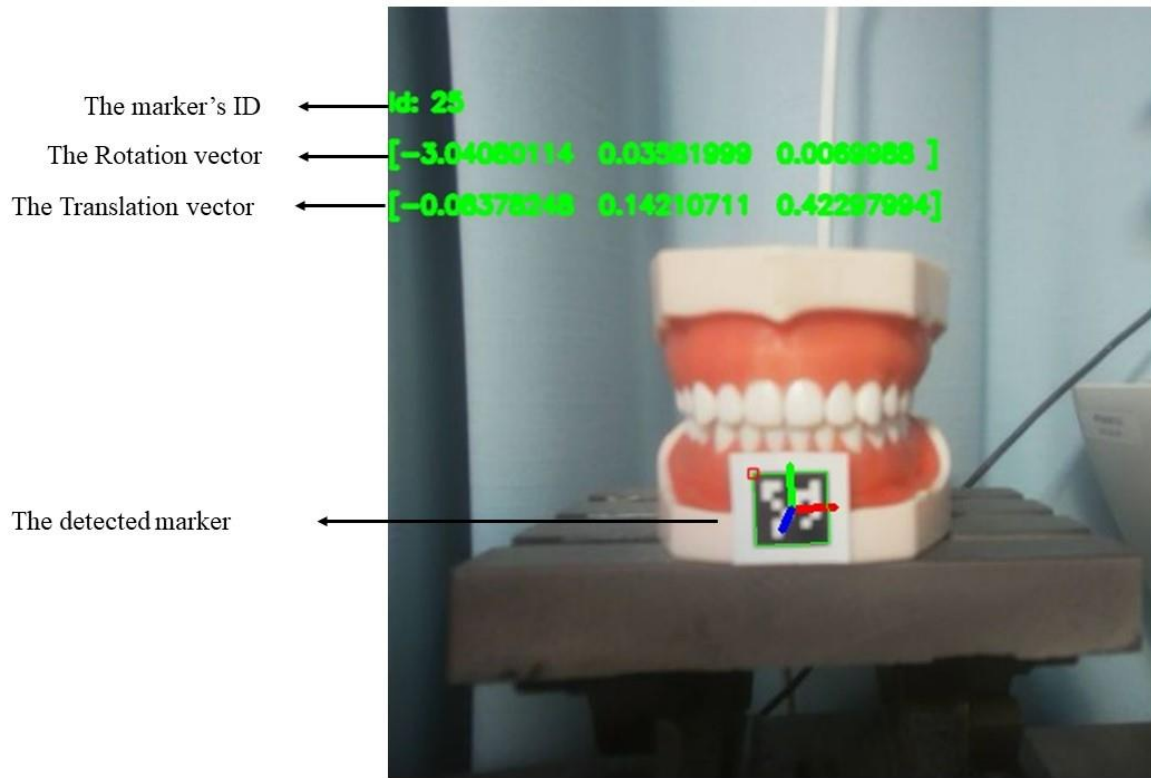


Figure 36. The detection result of the ArUco marker.

4.3.2 The calculation of Euler angles

After the marker is detected, the Euler angles can be obtained from the rotation vectors. Here, three cases are selected as examples in this paper to show the detection results. As shown in **Figure 37**, in addition to the rotation vector and translation vector mentioned above, the rotation matrix and Euler angles are newly solved here. The figure at the top is an example of testing the Pitch angle, where the tooth model is rotated upward by 20° . In the figure in the middle, there is an example testing the Yaw angle, where the model is rotated by 20° to the right. In the bottom figure, which is an example of testing the Roll angle, the model is rotated by 20° in the plane parallel to the camera.

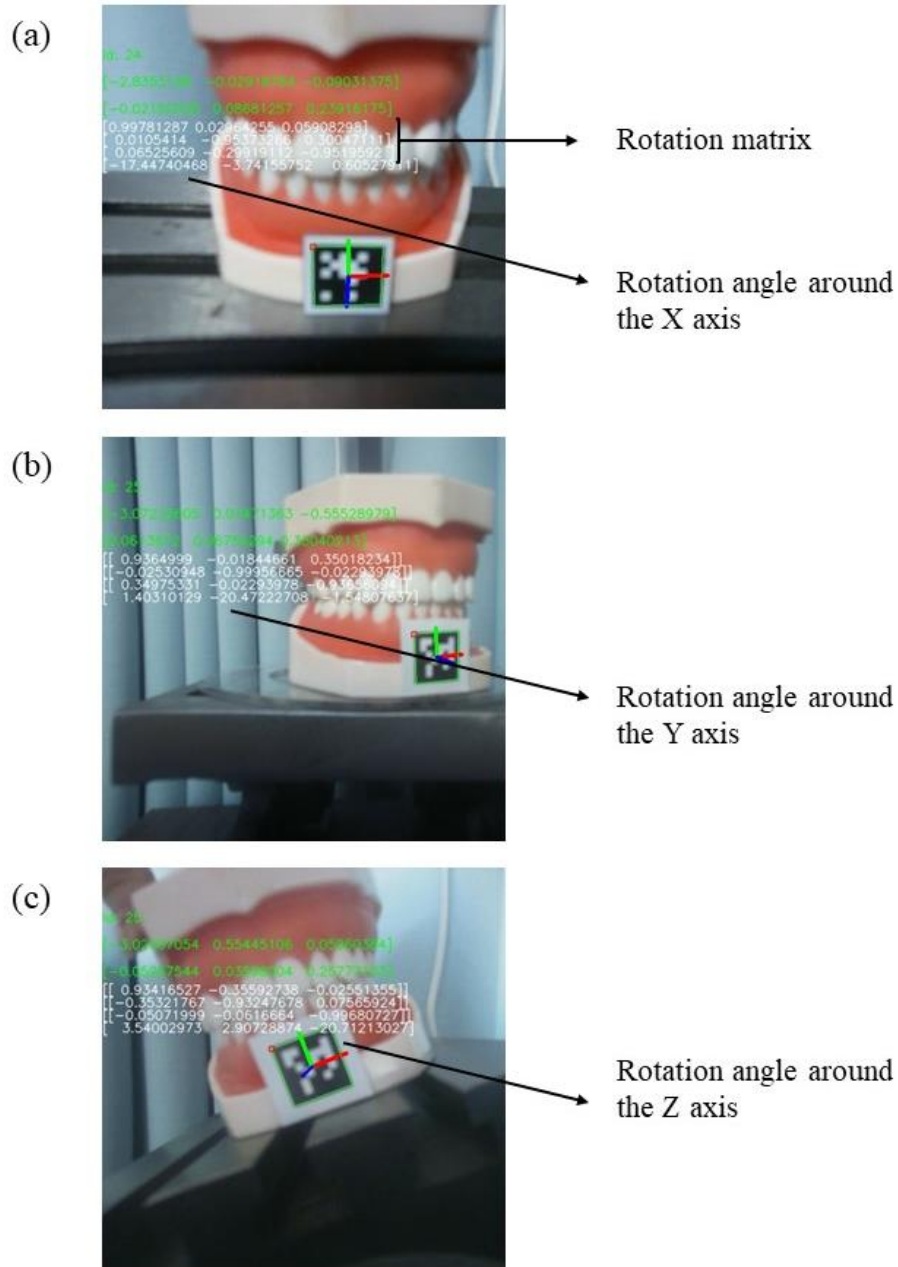


Figure 37. The testing result of the Euler angles. Colors of axes: X axis – Red; Y axis – Green; Z axis – Blue.

4.4 MATLAB APP

This app page includes a power system, ultrasonic system, water system, video system, feedback on robot action, and projection of a 3D model at three angles, Projection of three views of the dental model and the 3D dental model that can change viewing perspectives.

The 3D model made by Maya is shown in **Figure 38**.

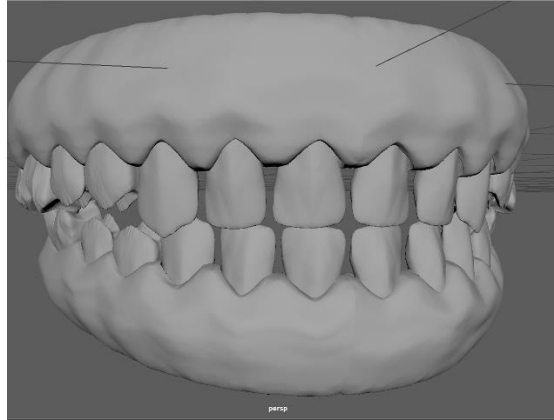


Figure 38. 3D dental model made by Maya.

The start page of the APP is shown in **Figure 39**.

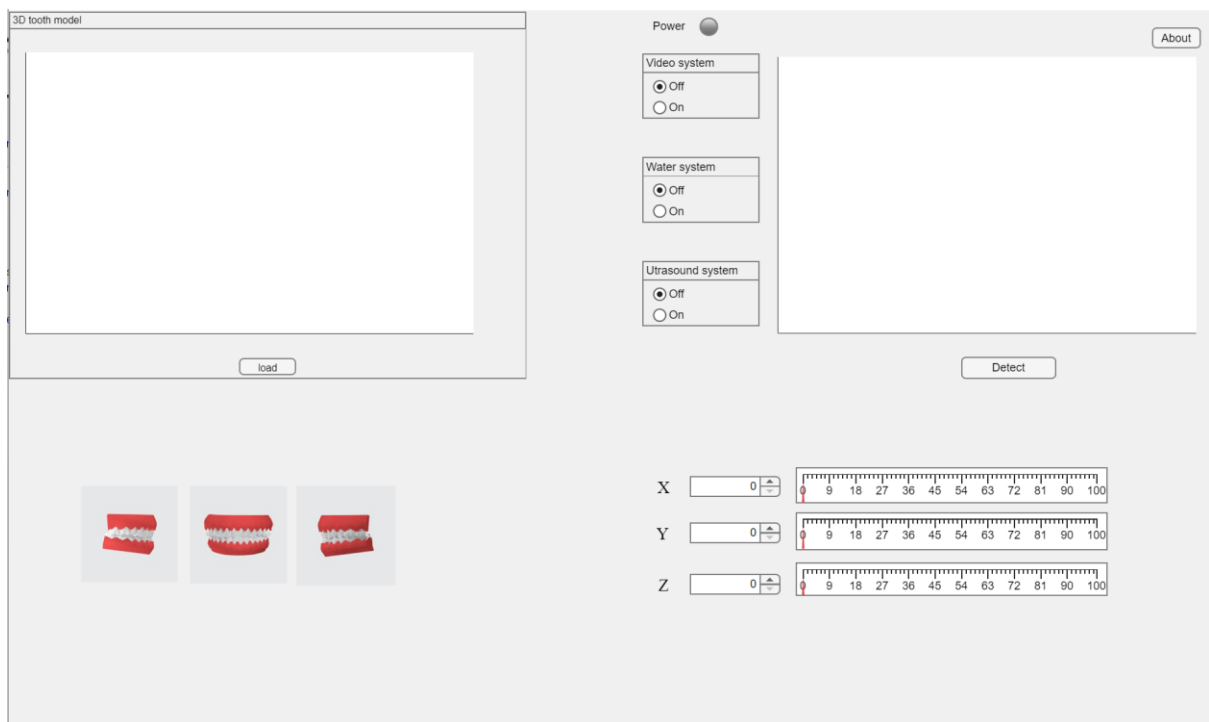


Figure 39. The start page of this APP.

Following is a demonstration of the various functions of this APP.

Figure 40 shows the function of importing a 3D model within the app and changing the view of the 3D model in real-time. By simply pressing the "load" button, the corresponding 3D model of the patient's teeth can be found on the computer and imported into the APP.

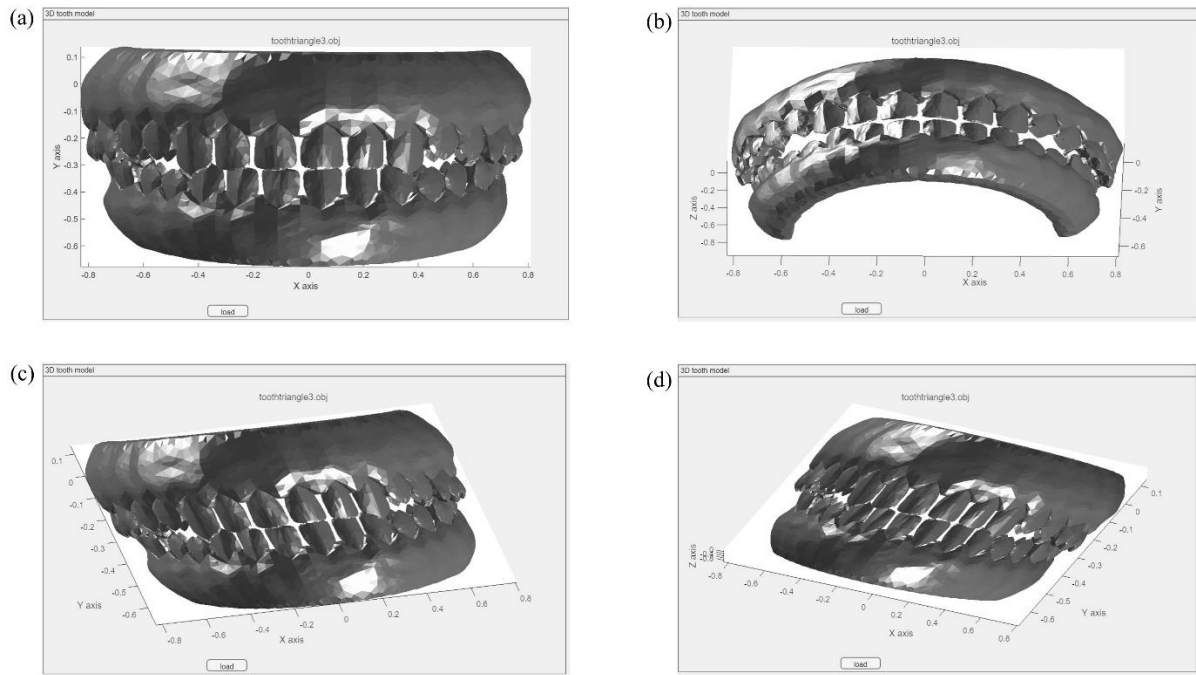


Figure 40. The 3D model importing and view changing functions of this APP. (a) The front view at the beginning of finishing importing the model. (b) (c) (d) View changes via the operation of the operator within the APP.

Figure 41 shows the projection of the tooth model in different directions.

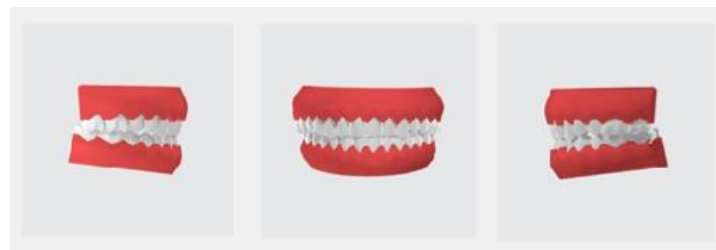


Figure 41. The projection of the teeth model in different directions – Left, Front, Right.

And when we adjust numbers in different directions (X, Y, Z), gauges will show this change. This function needs to be cooperated with the robot's movement. **Figure 42** illustrates this change process.

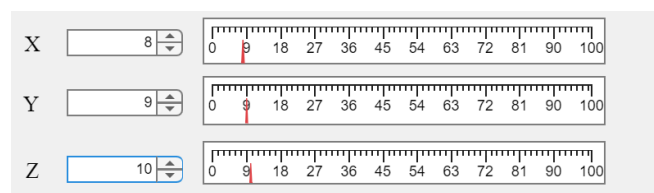


Figure 42. Gauges vary with spinners.

When pressing ‘p’, the power system will be turned on. When pressing ‘v’, the video system will be turned on. When pressing ‘w’, the water system will be turned on. When pressing ‘u’, the ultrasound system will be turned on. When pressing ‘1’, the power system will be turned off. When pressing ‘2’, the video system will be turned off. When pressing ‘3’, the water system will be turned off. When pressing ‘4’, the ultrasound system will be turned off. And pressing ‘o’ will turn off the entire system and close this APP. **Figure 43** shows the changes in the APP page and the LEDs on the Raspberry Pi when (a) "power supply" is turned on, (b) "power supply" and "video system" are turned on, (c) "power supply" and "video system" and "water system" are turned on, (d) "power supply" and "video system" and "water system" and "ultrasound system" are turned on, respectively.

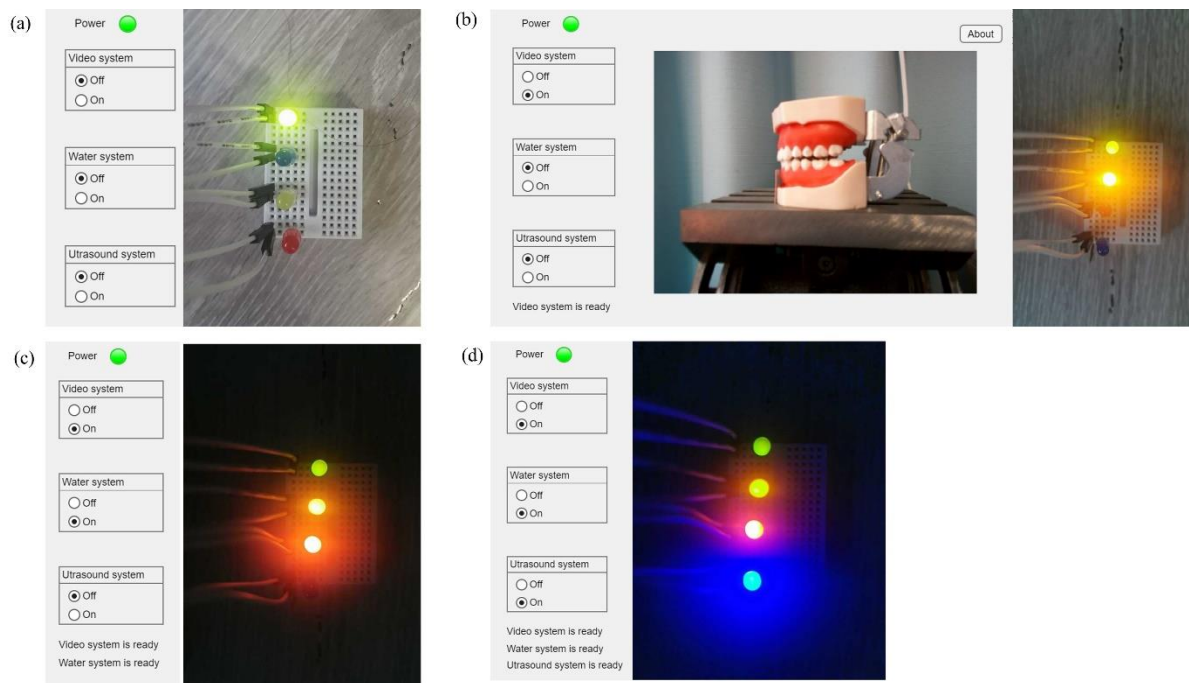


Figure 43. The changes in the APP page and the LEDs on the Raspberry Pi. (a) "power supply" is turned on, (b) "power supply" and "video system" are turned on, (c) "power supply" and "video system" and "water system" are turned on, (d) "power supply" and "video system" and "water system" and "ultrasound system" are turned on.

Finally, the APP uses the YOLOv2 model mentioned above to identify the different sides of the teeth and for detecting the teeth areas in real-time. The detection is performed when the "Detect" button is pressed. **Figure 44** shows an example of the recognition of the front view.

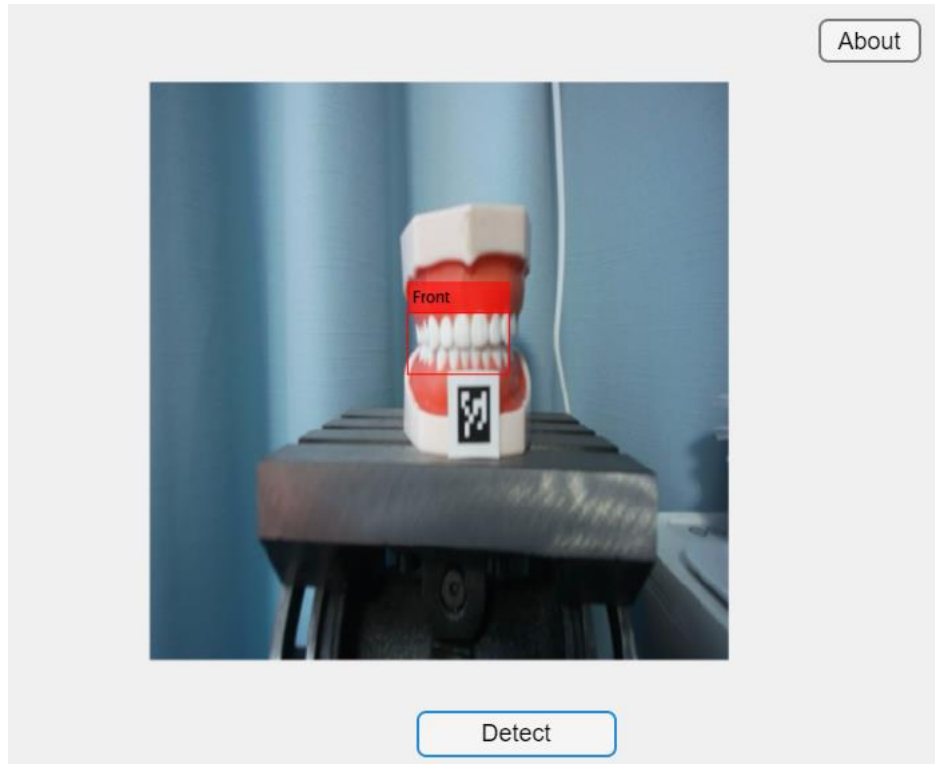


Figure 44. An example of the recognition of the front view of the teeth.

4.5 RESULT ANALYSIS OF THIS PROJECT

In this subsection, the above results will be quantitatively characterized and analyzed. The focus is on further discussion of the effectiveness of the learning model on the test set and the detection of the Euler angles of the teeth using the PnP method.

4.5.1 The Intersection over Union analysis of the 2D detection mission

Object detection evaluation metrics commonly include accuracy, confusion matrix, IoU, recall, etc. First, a few relevant definitions need to be given in **Table 7**.

Table 7. Relevant definitions.

Definition	Description
True Positive (TP)	Predict positive samples as positive
True Negative (TN)	Predict negative samples as negative
False Positive (FP)	Predict negative samples as positive
False Negative (FN)	Predict positive samples as negative
TP+TN	All the samples that predicted correctly
TP+FP	All samples with positive prediction
TP+FN	All samples that are actually positive (all positive samples, i.e., Ground Truth)
TP+TN+FP+FN	All samples

With the basic definition, calculations need to be performed for the different indicators.

- (1) Accuracy: the percentage of samples with correct predictions to the total sample.

Although Accuracy can determine the total correct rate, the evaluation index of Accuracy has a great drawback in the case of unbalanced positive and negative samples. For example, in a classification task, if the number of samples in each category is unbalanced, especially in binary classification, If the number of positive samples is too different from that of negative samples, in this case, it is only necessary that all samples are predicted as positive or negative for achieving high accuracy. Therefore, there are some other evaluation indicators to be considered besides accuracy. The Accuracy is calculated as

$$A(accuracy) = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

- (2) Precision: the percentage of positive samples correctly predicted out of all detected (positive predicted) samples is an example of the prediction result. The Precision is calculated as

$$P(precision) = \frac{TP}{TP + FP} \quad (4.2)$$

- (3) Recall: the proportion of correctly predicted positive samples to all positive samples (Ground Truth), for the original sample. The Recall is calculated as

$$R(recall) = \frac{TP}{TP + FN} \quad (4.3)$$

(4) F-measure: weighted summed average of precision and recall. When $\beta=1$, it is called F1-Score. The F-measure is calculated as

$$F - measure = (1 + \beta^2) \frac{P * R}{\beta^2 * P + R} \quad (4.4)$$

(5) Specificity: equivalent to the definition of the Recall for Negative samples. The Specificity is calculated as

$$Specificity = \frac{TN}{TN + FP} \quad (4.5)$$

(6) Intersection over union (IoU): an evaluation metric that is calculated by the area contained in the ground truth bounding box and the predicted bounding box. The IoU is calculated as

$$IoU = \frac{\text{Area of overlap}}{\text{Area of union}} \quad (4.6)$$

In this project, there are 100 samples labeled "Left", 100 samples labeled "Front", and 100 samples labeled "Right". samples. The above values will be calculated for these 300 samples.

The results are shown in **Figure 45** and **Table 8**.

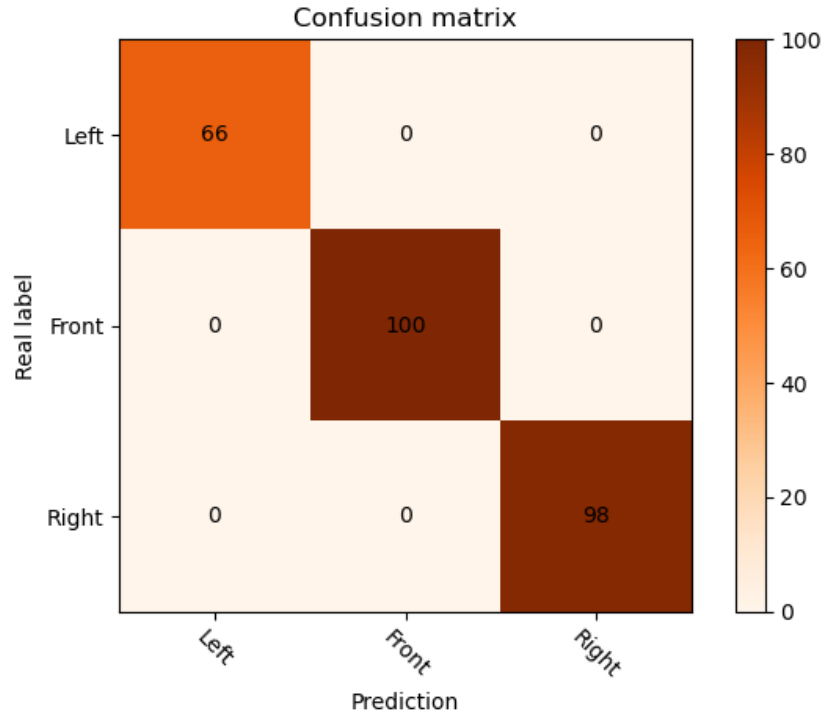


Figure 45. The confusion matrix.

Table 8. The mean values of the three classes and total test data set.

	Left	Front	Right	Total
Mean IoU	0.561751301	0.832452446	0.841476596	0.745226781
Accuracy	0.66	1	0.98	0.88
Precision	1	1	1	1
Recall	0.66	1	0.98	0.88
F1-Score	0.795180723	1	0.989898990	0.936170213

4.5.2 Quantitative characterization of the estimation of the 3D angle

After camera calibration, the internal parameter matrix, distortion coefficient and reprojection error of the camera are described. The parameters of the internal parameter matrix are shown in **Table 9**. The distortion coefficient is

[2.25291918e-01, -6.47644437e-01, 1.34044426e-03, 3.47655364e-04, 7.59953114e-01].

The reprojection error is 0.09487756648028772.

Table 9. Parameters of the camera's internal reference matrix.

Parameter	Value
α	742.67520754
γ	0
u_0	654.74668031
β	742.62949976
v_0	371.91979465

After the camera internal parameters are obtained, the position of the dental model can be obtained by combining the real-time camera external parameters obtained from the marker and the PnP principle. The rotation vectors in the external parameters are then converted to Euler angles (Pitch - Rotation angle around X axis, Yaw - Rotation angle around Y axis, Roll - Rotation angle around Z axis). The experimental measurements show that the angular range of the teeth model obtained with a single marker is shown in **Table 10**.

Table 10. Detection range of different rotation axes (Unit: °).

	X	Y	Z
Detection range	-40 - 45	-40 - 40	-180 - 180

To show the effect of angle estimation, the detected angles are compared with those measured with a precision instrument. In this comparative experiment, Pitch, Yaw and Roll rotation angles were measured and detected once every 5 degrees, ranging from -40 degrees to 40 degrees. **Figure 46** shows how the detected rotation angle of the teeth model changes with the measured angle. **Table 11** illustrates the maximum, minimum and average errors between angles detected and measured around different axes, as well as the maximum, minimum and average errors of all data.

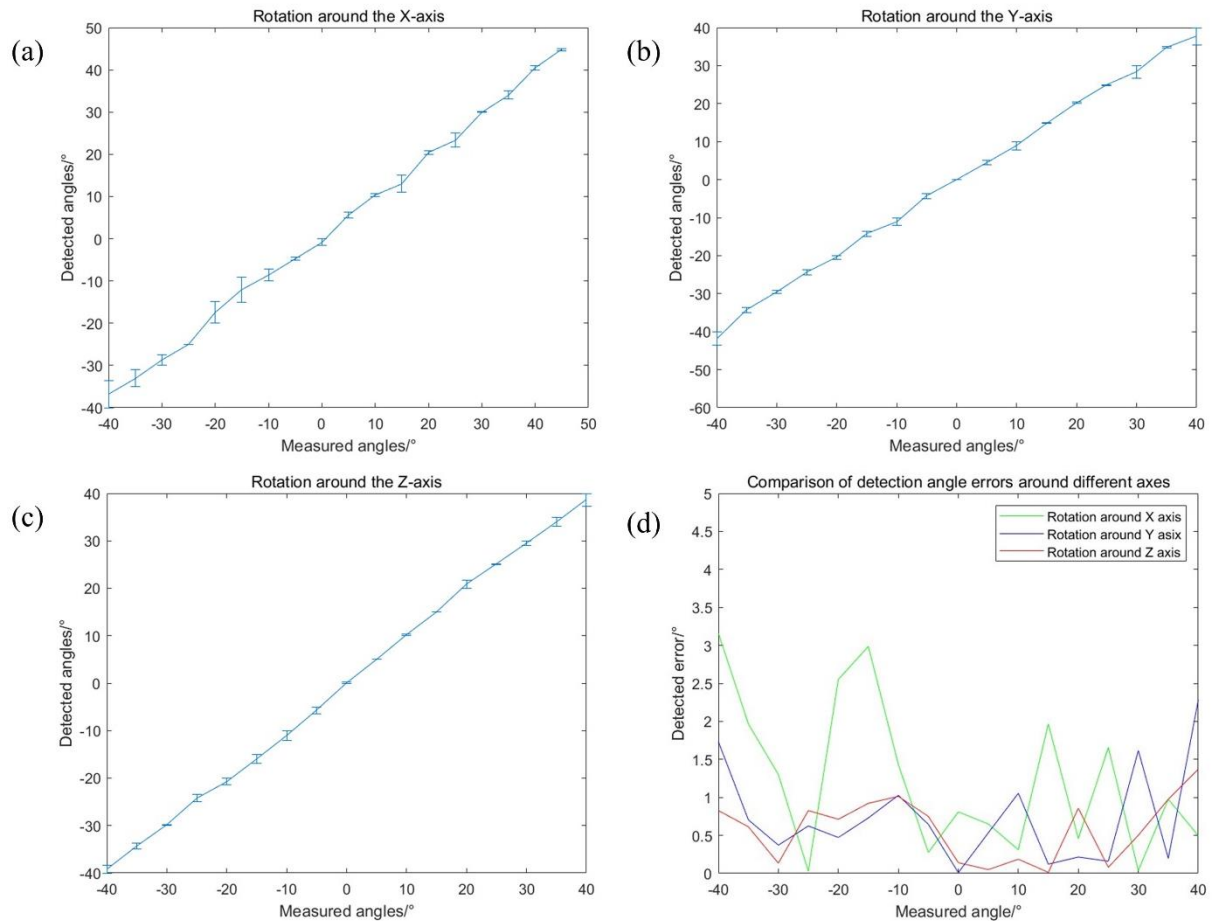


Figure 46. (a)(b)(c) Detected angles and error condition of X-axis, Y-axis, Z-axis, respectively. (d) comparison of errors about different axes.

Table 11. Error condition of different rotation axes (Unit: °).

	X	Y	Z	Total
Max error	3.16297116	2.28390297	1.37091516	3.16297116
Min error	0.02860975	0.01014531	0.01338388	0.01014531
Mean error	1.183691423	0.670888559	0.586725651	0.82088245

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This paper presents a system model for remote scaling of teeth which allows that the doctor and the patient are in different spaces. In the concept of this system, the doctor performs the control of the modules on the patient's side via the GUI on the computer and views the patient's situation in real-time through visual feedback.

Apart from the proposal of the system, in the experimental part, this paper mainly focuses on the implementation of the software part of the system. Experiments on communication and control between the PC and the Raspberry Pi and LED lights have been carried out, as well as some preliminary experiments on a semi-automated teleoperated scaling system for the real-time visual feedback provided by the camera.

For the communication between the upper and lower computers, the computer and the Raspberry Pi are used respectively. Connecting them within the same Wi-Fi, on the upper computer, communication with the lower computer can be achieved through the SSH communication protocol. In order to react to the control of the upper computer to the lower computer, LED lights are connected to the GPIO pins of the Raspberry Pi. By controlling the GPIO pins' level on the upper computer, the lights can be controlled in real time.

For the visual feedback, this paper addresses the problem of recognition of both planar graphics and 3D angles.

In order to classify the projections of the teeth model in three different viewing perspectives (Left, Front, Right), a dataset containing images of the three classes (both training and test sets) was created and a deep learning model was used to identify the different projections and

draw the teeth regions. In order to achieve this, these images first need to be pre-processed. The pre-processing process included data enhancement and image size uniformity. After pre-processing, we obtained 900 images of size 224×224 that could be used for training. The images were then labeled, and the images and label information were fed into the YOLOv2 model, the model parameters were adjusted, and the model was trained. The trained model was then tested using the test set data (300 images). The test results show that the accuracy of this model is 0.88.

In order to identify the 3D rotation angle of the tooth model rotated, this paper uses the ArUco marker as the marker to be attached underneath the teeth and uses the PnP method for calculating the 3D angles of the teeth model rotation in real-time from the 2D image obtained by the camera. The solution to this issue may consist of three steps. First is the calibration of the camera. In this paper, a checkerboard grid of 9×9 has been applied for calibrating the camera, which yields the internal reference matrix as well as the distortion coefficient of the camera, and a de-distortion operation is performed, resulting in a reprojection error of 0.09487756648028772. The second step is the detection of the teeth, where the ArUco marker (ID:25) is detected by the algorithm used for its ID, corner points, and the real-time rotation and translation vectors. The third step is the calculation of the Euler angles, where the rotation vectors are transformed into rotation matrices, and the rotation matrices into Euler angles (here the three degrees of freedom Pitch, Yaw, and Roll are used to split the 3D angles to make them easier to understand). After verification, the range of angle detection around the X, Y, and Z axes is obtained as -40° to 45° , -40° to 40° , and -180° to 180° respectively. The largest average error is in the case of rotation around the X axis, with an average error of 1.183691423° and a maximum error of 3.16297116° ; the smallest average error is in the case of rotation around the Z axis, with an average error of 0.586725651° .

Finally, the GUI was developed. This GUI has the following functions: (1) real-time view of the images taken by the camera; (2) the operator can operate the controls in this APP and change the state of the controls via the mouse and keyboard; (3) the Raspberry Pi can be controlled via this GUI, especially the GPIO pins; (4) the importing, viewing and changing direction of the 3D model is implemented; (5) the GUI can be accessed via the “Detect”

button on the GUI to identify the different projection surfaces of the teeth.

In conclusion, this paper presents a teleoperated dental scaling system with a functional implementation based on the communication and visual feedback between the upper and lower computers of this system, providing a viable solution for the scenario of dental scaling, a high-risk environment with susceptible infectious diseases, especially during the COVID-19 pandemic.

5.2 FUTURE WORK

5.2.1 Improvement

For dental care, compared with other scenarios using teleoperation systems, the accuracy required by the system needs to be very high. At the same time, the dentist's treatment should be timely enough to remove dental stones, dental plaque, etc. without damaging the patient's teeth. Therefore, the accuracy and time delay are the most important aspects of this paper.

For the communication between the upper computer and the lower computer, the existing wireless connection method using MATLAB through SSH protocol is likely to not achieve the time delay range required for remote real-time operation. Methods such as time series prediction may be used to reduce the time delay of system operation.

For the detection of the projection surface, after obtaining the projection surface area, we may further realize the automatic screenshot of the area in the APP and generate panoramic images in the future, so that doctors can observe the overall situation of the actual teeth. But this requires the detection algorithm to be accurate enough to cover all areas of the teeth. To solve this problem, we can use other algorithms to compare and select a better algorithm to use. Another important problem is the problem of data sets. The data set used in this paper only comes from the teeth model. If we want to make the system more robust, we need to use more real data. However, due to the particularity of the scene, it is difficult to find the corresponding usable data set at present.

For the estimation of three-dimensional angles, the current detection error is still not small enough, and due to the problem of multiple solutions of PnP algorithm, the value obtained is very unstable. Therefore, how to reduce the error and make the estimation more stable is a problem. In order to reduce the error, different markers can be pasted at multiple positions to reduce the error caused by angle changes. Cameras are also a source of error. Using a camera with a higher resolution and a calibration plate with a higher precision is a good way to reduce errors. In addition, larger marker, closer distance, and more suitable light source can reduce the error to a certain extent. For the problem of estimation instability, we can combine the machine learning model to make the obtained values more stable.

5.2.2 Next step

The work of this paper focuses on the operation of the doctor side and does not involve much in the work of the patient side. Therefore, the next step should focus on the realization of the slave site, including the collection of ultrasonic system, the movement of robots and the control of water used to wash teeth.

BIBLIOGRAPHY

- [1] Han, P., Li, H., Walsh, L. J., & Ivanovski, S. (2021). Splatters and aerosols contamination in dental aerosol generating procedures. *Applied Sciences*, 11(4), 1914.
- [2] Holliday, R., Allison, J. R., Currie, C. C., Edwards, D. C., Bowes, C., Pickering, K., ... & Jakubovics, N. (2021). Evaluating contaminated dental aerosol and splatter in an open plan clinic environment: Implications for the COVID-19 pandemic. *Journal of dentistry*, 105, 103565.
- [3] Ge, Z. Y., Yang, L. M., Xia, J. J., Fu, X. H., & Zhang, Y. Z. (2020). Possible aerosol transmission of COVID-19 and special precautions in dentistry. *Journal of Zhejiang University-SCIENCE B*, 21(5), 361-368.
- [4] Cousins, M., Patel, K., Araujo, M., Beaton, L., Scott, C., Stirling, D., ... & Knights, J. (2022). A qualitative analysis of dental professionals' beliefs and concerns about providing aerosol generating procedures early in the COVID-19 pandemic. *BDJ open*, 8(1), 1-6.
- [5] González-Olmo, M. J., Delgado-Ramos, B., Ortega-Martínez, A. R., Romero-Maroto, M., & Carrillo-Díaz, M. (2022). Fear of COVID-19 in Madrid. Will patients avoid dental care?. *International dental journal*, 72(1), 76-82.
- [6] Estai, M., Winters, J., Kanagasingam, Y., Shiikha, J., Checker, H., Kruger, E., & Tennant, M. (2016). Remote dental screening by dental therapists. *BDJ Team*, 3(10), 16169.
- [7] Digel, I., Kern, I., Geenen, E. M., & Akimbekov, N. (2020). Dental plaque removal by ultrasonic toothbrushes. *Dentistry journal*, 8(1), 28.
- [8] Li, S. Q., Guo, W. L., Liu, H., Wang, T., Zhou, Y. Y., Yu, T., ... & Li, S. Y. (2020). Clinical application of an intelligent oropharyngeal swab robot: Implication for the COVID-19 pandemic. *European Respiratory Journal*, 56(2).
- [9] Xing, W., Wang, J., Zhao, C., Wang, H., Bai, L., Pan, L., ... & Cheng, J. (2021). A highly automated mobile laboratory for on-site molecular diagnostics in the COVID-19 pandemic. *Clinical chemistry*, 67(4), 672-683.
- [10] Hokayem, P. F., & Spong, M. W. (2006). Bilateral teleoperation: An historical survey. *Automatica*, 42(12), 2035-2057.
- [11] Opiyo, S., Zhou, J., Mwangi, E., Kai, W., & Sunusi, I. (2021). A review on teleoperation of mobile ground robots: Architecture and situation awareness. *International Journal of*

- Control, Automation and Systems*, 19(3), 1384-1407.
- [12]Madder, R. D., VanOosterhout, S., Mulder, A., Bush, J., Martin, S., Rash, A. J., ... & Nowak, B. (2020). Network latency and long-distance robotic telestenting: exploring the potential impact of network delays on telestenting performance. *Catheterization and Cardiovascular Interventions*, 95(5), 914-919.
- [13]Farajiparvar, P., Ying, H., & Pandya, A. (2020). A brief survey of telerobotic time delay mitigation. *Frontiers in Robotics and AI*, 7, 578805.
- [14]Hauser, K. (2013). Recognition, prediction, and planning for assisted teleoperation of freeform tasks. *Autonomous Robots*, 35(4), 241-254.
- [15]Nikafrooz, N., & Leonessa, A. (2021). A single-actuated, cable-driven, and self-contained robotic hand designed for adaptive grasps. *Robotics*, 10(4), 109.
- [16]Said, S., Boulkaibet, I., Sheikh, M., Karar, A. S., Alkork, S., & Naït-Ali, A. (2020). Machine-learning-based muscle control of a 3d-printed bionic arm. *Sensors*, 20(11), 3144.
- [17]Artal-Sevil, J. S., Montañés, J. L., Acón, A., & Domínguez, J. A. (2018, June). Control of a bionic hand using real-time gesture recognition techniques through leap motion controller. In *2018 XIII Technologies Applied to Electronics Teaching Conference (TAEE)* (pp. 1-7). IEEE.
- [18]Lee, Y., Do, W., Yoon, H., Heo, J., Lee, W., & Lee, D. (2021). Visual-inertial hand motion tracking with robustness against occlusion, interference, and contact. *Science Robotics*, 6(58), eabe1315.
- [19]Ishak, M. K., & Kit, N. M. (2017, November). Design and implementation of robot assisted surgery based on Internet of Things (IoT). In *2017 International conference on advanced computing and applications (ACOMP)* (pp. 65-70). IEEE.
- [20]Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K., & Abbeel, P. (2018, May). Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5628-5635). IEEE.
- [21]Hoff, W. A., Gatrell, L. B., & Spofford, J. R. (1991). Machine-vision-based teleoperation aid. *Telematics and informatics*, 8(4), 403-423.
- [22]Maheswari, M. V., & Murugeswari, G. (2020, February). A survey on computer algorithms for retinal image preprocessing and vessel segmentation. In *2020 International Conference on Inventive Computation Technologies (ICICT)* (pp. 403-

- 408). IEEE.
- [23] Schwier, M., van Griethuysen, J., Vangel, M. G., Pieper, S., Peled, S., Tempany, C., ... & Fedorov, A. (2019). Repeatability of multiparametric prostate MRI radiomics features. *Scientific reports*, 9(1), 1-16.
- [24] Sarki, R., Ahmed, K., Wang, H., Zhang, Y., Ma, J., & Wang, K. (2021). Image preprocessing in classification and identification of diabetic eye diseases. *Data Science and Engineering*, 6(4), 455-471.
- [25] Masoudi, S., Harmon, S. A., Mehralivand, S., Walker, S. M., Raviprakash, H., Bagci, U., ... & Turkbey, B. (2021). Quick guide on radiology image pre-processing for deep learning applications in prostate cancer research. *Journal of Medical Imaging*, 8(1), 010901.
- [26] Vasuki, P., Kanimozhi, J., & Devi, M. B. (2017, April). A survey on image preprocessing techniques for diverse fields of medical imagery. In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)* (pp. 1-6). IEEE.
- [27] Vibhute, A., & Bodhe, S. K. (2012). Applications of image processing in agriculture: a survey. *International Journal of Computer Applications*, 52(2).
- [28] Tripathy, S., & Swarnkar, T. (2020). Unified preprocessing and enhancement technique for mammogram images. *Procedia Computer Science*, 167, 285-292.
- [29] Motta, J. M. S., De Carvalho, G. C., & McMaster, R. S. (2001). Robot calibration using a 3D vision-based measurement system with a single camera. *Robotics and Computer-Integrated Manufacturing*, 17(6), 487-497.
- [30] Song, L. M., Wang, M. P., Lu, L., & Huan, H. J. (2007). High precision camera calibration in vision measurement. *Optics & Laser Technology*, 39(7), 1413-1420.
- [31] Wang, D., Bai, Y., & Zhao, J. (2012). Robot manipulator calibration using neural network and a camera-based measurement system. *Transactions of the Institute of Measurement and Control*, 34(1), 105-121.
- [32] Liu, Y., Sun, P., Wergeles, N., & Shang, Y. (2021). A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172, 114602.
- [33] Patel, P., & Bhavsar, B. (2021). Object Detection and Identification. *International Journal*, 10(3).
- [34] Akcay, S., Kundegorski, M. E., Willcocks, C. G., & Breckon, T. P. (2018). Using deep convolutional neural network architectures for object classification and detection within

- x-ray baggage security imagery. *IEEE transactions on information forensics and security*, 13(9), 2203-2215.
- [35]Liang, W., Xu, P., Guo, L., Bai, H., Zhou, Y., & Chen, F. (2021). A survey of 3D object detection. *Multimedia Tools and Applications*, 80(19), 29617-29641.
- [36]Ji, Q. (2002). 3D face pose estimation and tracking from a monocular camera. *Image and vision computing*, 20(7), 499-511.
- [37]Yazdani, A., & Sabbagh Novin, R. (2021, March). Posture estimation and optimization in ergonomically intelligent teleoperation systems. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction* (pp. 604-606).
- [38]Li, Z., Qu, Z., Zhou, Y., Liu, J., Wang, H., & Jiang, L. (2022). Diversity Matters: Fully Exploiting Depth Clues for Reliable Monocular 3D Object Detection. In Proceedings of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2791-2800).
- [39]Chen, H., Wang, P., Wang, F., Tian, W., Xiong, L., & Li, H. (2022). EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2781-2790).
- [40][Maya Software | Get Prices and Buy Maya 2023 | Autodesk](#)
- [41]Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- [42]Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).
- [43]Zhang, Z. (1999, September). Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the seventh ieee international conference on computer vision* (Vol. 1, pp. 666-673). Ieee.
- [44]Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., & Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6), 2280-2292.
- [45]Kalaitzakis, M., Cain, B., Carroll, S., Ambrosi, A., Whitehead, C., & Vitzilaios, N. (2021). Fiducial markers for pose estimation. *Journal of Intelligent & Robotic Systems*, 101(4), 1-26.
- [46]Babinec, A., Jurišica, L., Hubinský, P., & Duchoň, F. (2014). Visual localization of mobile robot using artificial markers. *Procedia Engineering*, 96, 1-9.

- [47] Suzuki, S. (1985). Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1), 32-46.
- [48] Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2), 112-122.
- [49] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
- [50] Collins, T., & Bartoli, A. (2014). Infinitesimal plane-based pose estimation. *International journal of computer vision*, 109(3), 252-286.
- [51] Han, J., Jiang, W., Niu, D., Li, Y., Zhang, Y., Lei, B., ... & Lu, B. (2019). Untethered soft actuators by liquid–vapor phase transition: remote and programmable actuation. *Advanced Intelligent Systems*, 1(8), 1900109.
- [52] Kim, Y. J., Cheng, S., Kim, S., & Iagnemma, K. (2013). A novel layer jamming mechanism with tunable stiffness capability for minimally invasive surgery. *IEEE Transactions on Robotics*, 29(4), 1031-1042.
- [53] Takahashi, T., Tadakuma, K., Watanabe, M., Takane, E., Hookabe, N., Kajihara, H., ... & Tadokoro, S. (2021). Eversion Robotic Mechanism With Hydraulic Skeleton to Realize Steering Function. *IEEE Robotics and Automation Letters*, 6(3), 5413-5420.

APPENDICES

1. Model training process: [training process.pdf - Google](#)
2. Trained model of this project: [trainedYOLOv2Detector.mat - Google](#)
3. Data set: [Data set - Google](#)