APSC 486/COMM 466 – New Venture Design

April 10, 2025

Industry: Medical Device Development

ROBOT®

SURGICAL TOOL DISPENSING SYSTEMS

Venture: Semi-Automated Dispensing & Retrieving System to Streamline Scrubbing Procedures & Improve Operational Efficiency in Surgical Rooms

Team B4: Aly Khan Nuruddin, Connor Johst, Isaac Dee, Neil Thakur, Ying Min Chong

Industry Mentor: George Psiharis

THE UNIVERSITY OF BRITISH COLUMBIA

Report Type: Technical Report

Word Count: 7266

I. *Table of Contents*

1. Product Brief 4

2. Requirements Specification 5

2.1. Context 5

2.2. Stakeholders 5

2.3. Problem Statement 5

2.4. Requirements 6

2.4.1. Functional 6

2.4.2. Non-Functional 7

2.4.3. Constraints 7

3. Design 8

3.1. Product Architecture 8

3.2. Component Description 9

3.2.1. Voice Service 9

3.2.2. Control Service 10

3.2.3. Serial Service 12

3.2.4. Web Application 13

3.2.5. Camera Service 13

3.2.6. Tool Classifier 14

3.2.7. Rail & Gripper Control (Arduino) 15

3.2.8. Rail & Gripper Control (Circuit) 16

3.2.9. Rail & Gripper Control (Mechanics) 17

4. Verification & Validation 18

4.1. Introduction 18

4.2. Verification Tests 20

4.3. Validation Tests 23

5. Intellectual Property 24

5.1. Related Intellectual Property 24

5.2. Freedom to Operate 24

5.3. Patentability 25

6. Archive Contents 26

7. References 28

8. Appendices 30

8.1. Appendix #1: Marketing Video 30

8.2. Appendix #2: Demo Video 30

8.3. Appendix #3: Total Expenses 30

1. Product Brief

Surgical tool management is essential for patient safety and operational efficiency. Scrub nurses manage up to 150 instruments per procedure under high-pressure conditions that increase risk of delays, errors, and retained items, with significant clinical and financial consequences (1). Shortages of trained personnel further strain intraoperative workflows (2).

Hence, our unique device called 'ORobot' addresses these challenges by using voice recognition and image classification to dispense, retrieve, and track instruments in real-time. By automating tool handling and counting, the system reduces cognitive and physical load on scrub nurses, supports surgical teams, and enhances health outcomes for patients (3).

Designed for seamless integration into existing operating rooms, ORobot delivers tools within 10 seconds, maintains sterility, handles instruments up to 30g, and includes a manual override for reliability. Its compact system features a voice-command interface, dual-tool rail-guided gripper, and camera-based inventory tracking. Validation efforts will assess performance, accuracy, and compliance with sterile field requirements, ensuring that ORobot meets the stringent regulatory demands of modern surgical environments.

The following sketch depicts the working principles of the ORobot, our semi-automated prototype for real-time tracking of surgical instruments (*Figure 1*).

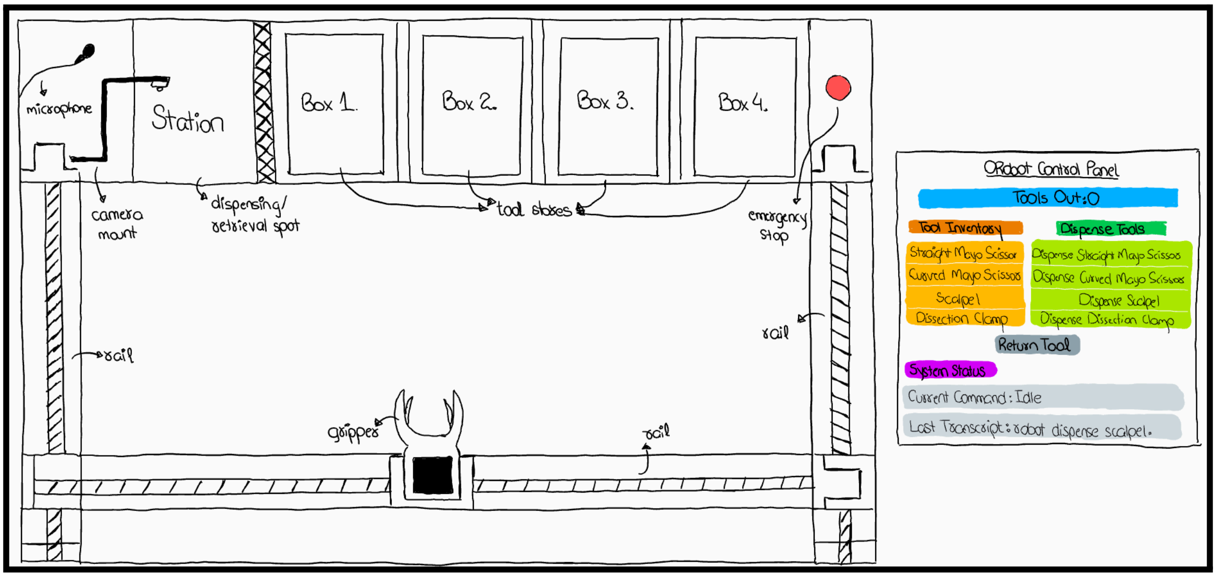

Figure #1: *Illustration of ORobot's Tool Exchange System & Control Panel (Top-Down View)*

During surgery, the ORobot is activated via voice commands from the scrub nurse. Upon recognizing the requested tool, the system locates it using image classification and signals the rail-guided gripper to retrieve it by transporting it along the X and Y-axis to the dispensing station. An emergency stop is added to abort operation in case of mechanical failure. Note that the commercially available version of the device would consist of another rail mechanism in the z-axis for vertical scalability to house over 100 tools. However, the system would adhere to any space constraints in the surgical suite. A marketing simulation and proof-of-concept demonstration for ORobot can be found as follows (*Appendix #1, #2*).

2. Requirements Specification

2.1 Context

ORobot addresses the growing need for semi-automated intraoperative support. Positioned in the surgical workflow, it integrates voice recognition, image classification, and robotic manipulation to automate tool handling and inventory tracking. This system aims to enhance efficiency, reduce cognitive burden on staff, and improve safety in operating rooms.

2.2 Stakeholders

Key stakeholders in the development, deployment and use of the ORobot include (4):

- **Patients:** Primary beneficiaries of improved surgical safety, efficiency, and reduced risk of retained instruments or procedural delays.
- **Surgeons:** Primary end users requiring fast, accurate, and sterile tool delivery.
- **Scrub Nurses:** Operational users who benefit from reduced tool-handling workload.
- **Surgical Assistants & Anesthesiologists:** Adjacent team members impacted by efficiency and physical workflows of the system.
- **Operating Room Managers & Hospital Administrators:** Oversee surgical throughput, staffing, and cost-effectiveness.
- **IT and Integration Teams:** Ensure seamless system setup, software updates, and interoperability with hospital infrastructure.
- **Biomedical Engineers & Designers:** Develop, test, and refine hardware and software components of the device.
- **Clinical Trainers & Educators:** Facilitate familiarity with system operations through effective training and adoption among surgical staff.
- **Quality Assurance & Regulatory Personnel:** Ensure compliance with sterilization, safety, and certification requirements.
- **Procurement Officers:** Evaluate system feasibility, budgeting, and acquisition.
- **Legal & Compliance Officers:** Validate adherence to regulatory, legal, and data handling standards for clinical utilization.

2.3 Problem Statement

In modern surgical environments, precision, efficiency, and safety are paramount. Operating rooms are high-stakes settings where even minor delays or errors can result in significant patient harm and legal risk (5). A central component of surgical success lies in the seamless coordination between surgeons and scrub nurses for handling, tracking, and retrieval of surgical tools. Scrub nurses manage numerous instruments during procedures, a responsibility that requires intense focus, multitasking, and decision-making under pressure.

However, operating rooms face increasing strain due to staff shortages, heightened surgical demands, and the growing complexity of operations. Hospitals are continually seeking solutions that enhance workflow, minimize human error, and ensure patient safety without adding burden to clinical staff. In this context, automation presents an opportunity to support surgical teams with smarter, more reliable tool-handling systems that maintain sterility, reduce errors, and integrate dynamically into existing operating room workflows (2). Hence, a solution like ORobot can significantly enhance intraoperative efficiency, ensure patient safety, and improve scalability of surgical support systems in modern healthcare.

2.4 Requirements

2.4.1 Functional Requirements

FR1. Perform Core Actions

The ORobot shall execute the following primary actions; tool dispensing and tool retrieval. Each action must be performed while adhering to the system's operational constraints, including weight limitations, spatial dimensions, and time sensitivity.

Tool Dispensing Workflow:

- When a scrub nurse issues a verbal command such as "robot, scalpel," the system shall process the request and initiate a coordinated action flow:
    - The control domain (Raspberry Pi) processes the command and performs an internal tool lookup to confirm that the requested tool is present in the system.
    - The control domain sends the tool location to the robot domain over serial.
    - The robot moves the gripper to the correct location, retrieves the tool-containing box, and delivers it to the designated tool I/O area.

Tool Retrieval Workflow:

- The tool buffer contains one empty box designated for returns.
- The scrub nurse places a used tool in the empty box and issues a command, "robot, return tool."
- The control domain captures an image of the tool, performs classification, and sends back the appropriate slot ID to the robot domain.
- The robot gripper moves to the buffer and grips the return box, moving it back to its correct location.

FR2. Tool Tracking and Display

The ORobot shall maintain a real-time inventory of all tools stored within the machine. The system must achieve the following tasks to ensure transparency and quick verification during procedures:

- Accurately monitor the current status of each tool (in-use vs. available).
- Track their location in storage.
- Display this data in an intuitive interface accessible to surgical staff.

FR3. Manual Tool Access

In the event of equipment malfunction or system failure, the ORobot shall provide a secure, manual override system. This system must allow human access to retrieve tools in a timely manner without compromising the safety or sterility of the procedure. Manual access should be simple, well-documented, and fast to ensure that surgical flow is not interrupted.

2.4.2 Non-Functional Requirements

NFR1. Efficiency and Speed

The ORobot must operate at a level of efficiency equal to or exceeding that of an experienced scrub nurse. Both tool retrieval and return should be optimized to minimize delays and preserve the natural rhythm of surgical procedures.

NFR2. Space and Integration

The ORobot must be designed to fit within the constrained environment of a standard operating room, adhering to the following considerations:

- The device must not interfere with staff movement or disrupt workflow.
- The system should be compact, lightweight, and ergonomically constructed.
- The mechanism must be compliant with applicable operating room safety and design standards to ensure seamless adoption and approval.

NFR3. Sterilizability and Patient Safety

To ensure safe use in clinical environments, the ORobot must be designed with infection control and patient safety in mind (6). All components must support hospital-grade sterilization and reduce contamination risks during surgical procedures:

- The system shall use materials compatible with standard sterilization methods such as autoclaving, UV, and chemical disinfection.
- All surfaces must be smooth, corrosion-resistant, and easy to clean to prevent bacterial accumulation.
- The design must minimize crevices, seams, and exposed electronics in areas interacting with tools or staff.
- The mechanism shall comply with clinical safety and sterility regulations such as FDA guidance and ISO 11737 standards.

2.4.3 Constraints

C1. Tool Weight Limit

The robot must be able to handle surgical instruments weighing up to 30 grams. This constraint ensures compatibility with standard surgical tools and guarantees smooth operation of the gripper and rail-guided transport mechanisms.

C2. Supported Tools

The system shall support recognition, retrieval, and delivery of the following tools (7):

- Scalpel
- Straight Dissection Clamp
- Straight Mayo Scissor
- Curved Mayo Scissor

Each tool must be reliably identified by the classification system and physically compatible with the box and gripper systems. The commercialized, scalable version of the device must be capable of handling a variety of tools with different shapes, sizes, and weights, ranging from commonplace grasping, clamping, dissecting and suturing instruments to more specialized tools utilized for specific surgical purposes. Adherement to this requirement is critical to demonstrate the use case of ORobot across varied clinical settings.

C3. Response Time

From the time a tool request is made by voice command to the time the tool is delivered to the I/O area, the system shall complete the transaction in under 10 seconds. This ensures no perceptible delay in the surgical process and meets the benchmark set by the industry standard, manual handoff currently performed by scrub nurses.

C4. Tool Classification

The ORobot will utilize a proprietary dataset to classify surgical tools using images captured during return events. This classification system must be accurate and robust, enabling proper tool storage and reuse. Although initial development uses readily-available data sources, the finalized commercial system will be trained on a comprehensive set of tools specific to each surgical suite in order to promote customizability for operations.

3. Design

3.1 Product Architecture

The following diagram provides a high-level overview of ORobot's components under the Raspberry Pi 5 and Arduino domains, which interface seamlessly with the mechanical sub-system to enable an efficient automated tool handling workflow (*Figure #2*). This distributed, layered architecture allows ORobot to maintain robust real-time performance and modular scalability, while ensuring compatibility with surgical environments.

At the heart of the system is the Raspberry Pi 5, which orchestrates the domain-level operations through modular services. The Voice Service enables hands-free interaction, translating verbal commands into actionable inputs that are processed by the Control Service, which serves as the central coordinator. This service interfaces with multiple subsystems including the Serial Service, which handles communication with the Arduino board, the Camera Service, responsible for visual input, and the Web App, which is used for remote control and monitoring via a graphical user interface. All captured visual data is routed through the Tool Classifier, which identifies surgical instruments in real-time using computer vision and machine learning. The Arduino subsystem handles the lower-level actuation, where an Arduino Controller receives movement commands via the Serial Service and directs the Rail + Gripper Control Circuit. This circuit manipulates the Rail + Gripper Mechanics, ensuring precise physical handling of tools for dispensing or retrieval. The total expenditure for prototyping of the proof-of-concept is found as follows (*Appendix #3*).

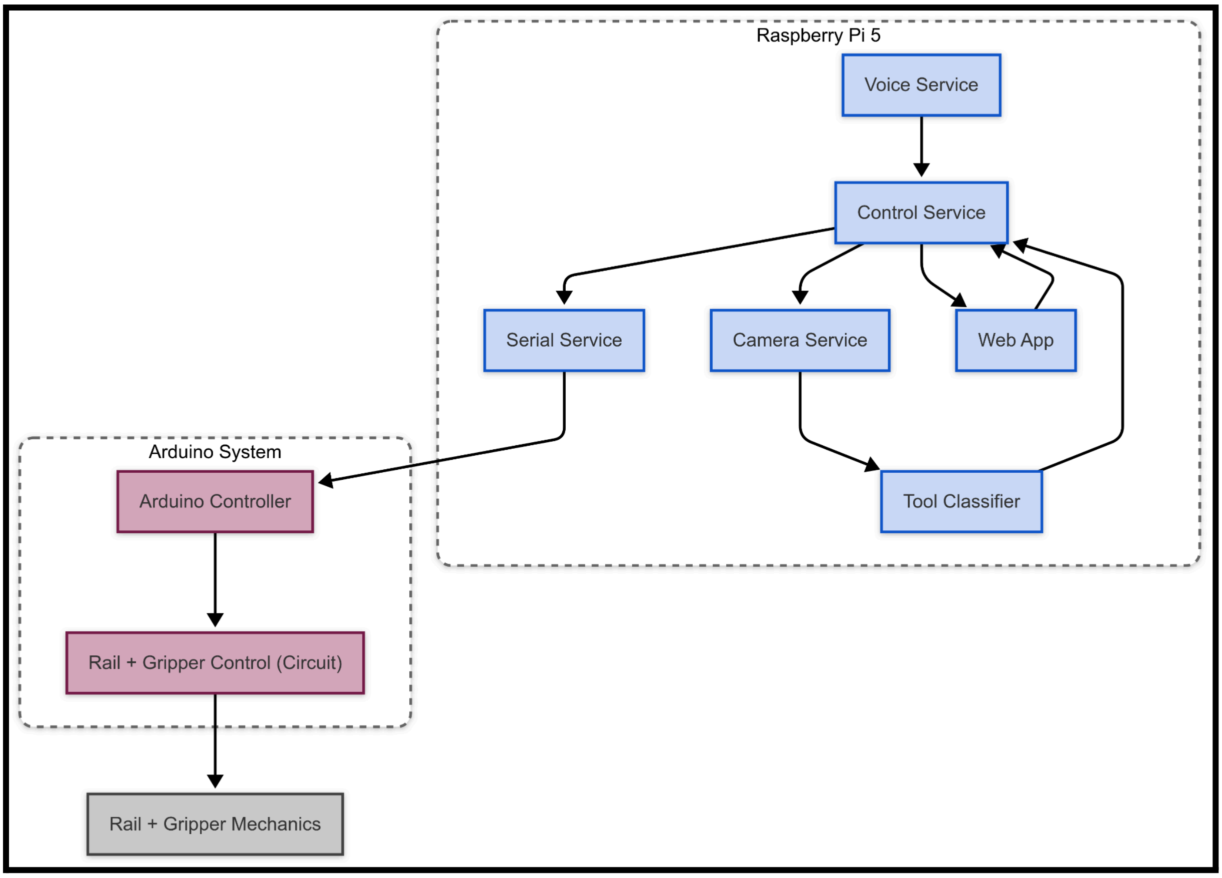

Figure #2: *System Architecture Diagram with Software & Hardware Components.*

3.2 Component Descriptions

3.2.1 Voice Service

Design Overview

The Voice Service provides real-time speech-to-text transcription and classification, allowing surgeons to control the O-Robot through natural language commands.

Diagram

https://github.com/cjohst/orobot/blob/main/assets/documentation/classes/VoiceService.png

Implementation Details

- **Speech-to-Text**: Implements WhisperLive, a real-time adaptation of OpenAI's speech recognition model.
- **Command Classification**: Uses Google Gemini 1.5 Flash 8B for robust natural language understanding.
- **Two Main Classes**:
  - `VoiceTranscriptionService`: Handles audio input and real-time transcription.
  - `CommandClassifier`: Processes transcribed text into structured commands.

Functional Requirements Addressed

- **FR1 (Core Actions)**: Enables hands-free mechanism for tool dispensing and return.
    - Processes commands like “robot, scalpel” and “robot, return tool”.
    - Passes structured commands to the Control Service for execution.

Non-Functional Requirements Addressed

- **NFR1 (Efficiency and Speed)**:
    - Local processing minimizes latency (no network dependency).
    - Cooldown mechanism prevents command flooding.

Constraints Addressed

- **C3 (Response Time)**:
    - Optimized for low latency to help achieve the <10-second response requirement.
    - Efficient text processing with minimal overhead.

Design Decisions

- **WhisperLive vs. Cloud APIs**:
    - Chosen for local deployment without internet dependency.
    - Critical for reliable operation in the surgical room.
- **Adaptive Command Priority**:
    - Implements context-aware prioritization between dispense and return commands.
    - Prevents local minimization or getting "stuck" on one command type.
- **Noise and Variation Handling**:
    - Tolerates speech recognition errors through flexible matching.
    - Maintains a dictionary of tool name variations to accommodate for different phrasings.
- **Filtering Mechanisms**:
    - Cooldown period (3 seconds) prevents accidental repetition of commands.
    - Jaccard similarity detection identifies duplicate commands.

3.2.2 Control Service

Design Overview

The Control Service is the central orchestrator of the ORobot system, integrating all components and coordinating the workflow for tool dispensing and returning.

Diagram

https://github.com/cjohst/orobot/blob/main/assets/documentation/classes/ControlService.png

Implementation Details

- **Multi-Threaded Architecture**: Manages parallel processing for simultaneous voice, web, and hardware control.
- **Service Integration**: Coordinates all system components through a unified interface.
- **State Management**: Tracks tool status and maintains system state.

Functional Requirements Addressed

- **FR1 (Core Actions)**:
    - Directly implements the tool retrieval workflow.
    - Coordinates tool return workflow including image capture and classification.
    - Orchestrates hardware movement via serial commands.
- **FR2 (Tool Tracking and Display)**:
    - Maintains a real-time status of inventory.
    - Updates shared state of tools accessible to a web interface.
    - Tracks which tools are dispensed or returned.
- **FR3 (Manual Tool Access)**:
    - Processes commands from the web interface when voice is unavailable.
    - Provides a backup control path for surgical staff in case of breakdown.

Non-Functional Requirements Addressed

- **NFR1 (Efficiency and Speed)**:
    - Optimizes command processing workflow.
    - Minimizes overhead in the critical tool tracking path.

Constraints Addressed

- **C2 (Supported Tools)**: Fully supports the four standard surgical tools.
- **C3 (Response Time)**: Optimized for minimal processing delay.
- **C4 (Tool Classification)**: Integrates with ML-based classification system.

Design Decisions

- **Centralized Orchestration**:
    - Single control point simplifies workflow management.
    - Clear hierarchy prevents circular dependencies.
- **Fault Tolerance**:
    - Implements fallback mechanisms for tool classification failures.
    - Can return any tool outside the system if specific identification fails.
- **Dual Interface Support**:
    - Processes commands from both voice and web interfaces.
    - Thread-safe command queue prevents conflicts.
- **Real-time State Synchronization**:
    - Uses a shared state pattern to keep the web interface updated.
    - Broadcasts status change via WebSockets.

3.2.3 Serial Service

Design Overview

The Serial Service manages communication between the Raspberry Pi and Arduino controller, enabling hardware control through a simple text-based protocol.

Diagram

https://github.com/cjohst/orobot/blob/main/assets/documentation/classes/SerialCommunicator.png

Implementation Details

- **Serial Protocol**: Implements a simple text-based command format (e.g., "D3" for "dispense tool 3").
- **Connection Management**: Handles initialization, sending of commands, and connection cleanup.
- **Error Handling**: Manages communication failures diligently.

Functional Requirements Addressed

- **FR1 (Core Actions)**:
    - Provides the critical link between software control and hardware execution.
    - Translates high-level commands to hardware-specific instructions.

Constraints Addressed

- **C3 (Response Time)**:
    - Minimalist protocol reduces communication overhead.
    - Simple text-based commands are efficiently processed by the Arduino.

Design Decisions

- **Text-Based Protocol vs. Binary Protocol**:
    - Text commands are human-readable for easier debugging.
    - Simple format reduces parsing complexity on the Arduino side.
- **Synchronous Communication**:
    - Direct command-response model ensures reliable operation.
    - Small delays after commands ensure proper transmission.
- **Error Handling Strategy**:
    - Graceful handling of connection failures.
    - Reports errors back to the Control Service for appropriate action.
- **Configuration Externalization**:
    - Serial port and baud rate defined in `config.py`.
    - Easy adaptation to different hardware setups.

3.2.4 Web Application

Design Overview

The Web Application provides a graphical interface for monitoring system status, viewing tool inventory, and issuing manual commands when voice control is not ideal.

Diagram

https://github.com/cjohst/orobot/blob/main/assets/documentation/classes/WebApp.png

Implementation Details

- **Flask Framework**: Lightweight web server suitable for Raspberry Pi.
- **SocketIO**: Enables real-time status updates without polling.
- **Shared State Pattern**: Maintains consistency between voice and web interfaces.

Functional Requirements Addressed

- **FR2 (Tool Tracking and Display)**:
  - Visual representation of tool inventory status.
  - Shows which tools are in active use and which remain stored.
- **FR3 (Manual Tool Access)**:
  - Button interfaces for tool dispensing and returning.
  - RESTful API endpoints for programmatic control.

Non-Functional Requirements Addressed

- **NFR1 (Efficiency and Speed)**:
  - Lightweight design for responsiveness.
  - Real-time updates via WebSockets that minimize latency.

Design Decisions

- **Flask + SocketIO Architecture**:
  - Flask provides a lightweight HTTP server.
  - SocketIO enables real-time bidirectional updates.
- **Shared State Pattern**:
  - Thread-safe state shared between web app and control service.
  - Ensures consistent view across all interfaces.
- **RESTful API Design**:
  - Simple endpoints for dispensing and returning tools.
  - JSON responses provide clear status information.
- **Command Queue Mechanism**:
  - Thread-safe queue for passing requests to control service.
  - Prevents race conditions between web and voice interfaces.

3.2.5 Camera Service

Design Overview

The Camera Service manages the webcam hardware, capturing high-quality images for tool classification during the return process.

Diagram
https://github.com/cjohst/orobot/blob/main/assets/documentation/classes/CameraService.png

Implementation Details

- **Hardware Optimization**: Uses `v4l2` utilities for camera configuration.
- **OpenCV Integration**: Captures and processes images.
- **Camera Setup**: Optimal settings for tool recognition.

Functional Requirements Addressed

- **FR1 (Core Actions)**: Essential for enabling the tool return workflow.

Constraints Addressed

- **C4 (Tool Classification)**: Provides images to distinguish between different tools.

Design Decisions

- **Low-Level Camera Configuration**:
  - Uses `v4l2-ctl` for fine-grained control.
  - Custom exposure, brightness, and contrast settings.
- **Resource Management**:
  - Initializes the camera only when needed for tool return.
  - Released from yse immediately after pictures have been taken.
- **Fallback Mechanisms**:
  - Falls back to OpenCV settings if `v4l2-ctl` fails.
- **Image Stabilization**:
  - Warmup and frame discarding to ensure stable images.

3.2.6 Tool Classifier

Design Overview

The Tool Classifier identifies surgical tools from images using a trained deep learning model, enabling automatic classification during the return process.

Diagram
https://github.com/cjohst/orobot/blob/main/assets/documentation/classes/ToolClassifier.png

Implementation Details

- **Model Architecture**: ResNet-50 CNN fine-tuned for tool recognition.
- **PyTorch Implementation**: Inference pipeline built on PyTorch.
- **Image Preprocessing**: Normalization and resizing.

Functional Requirements Addressed

- **FR1 (Core Actions)**: Enables return workflow via image classification.
- **FR2 (Tool Tracking and Display)**: Contributes to accuracy of tool inventory.

Constraints Addressed

- **C2 (Supported Tools)**: Recognizes the four required tools.
- **C4 (Tool Classification)**: Core functionality to enable the return workflow.

Design Decisions

- **ResNet-50 Architecture Selection**:
    - Balanced accuracy and speed for edge deployment.
- **PyTorch vs. TensorFlow**:
    - PyTorch selected for better integration.
- **Class Mapping Strategy**:
    - Maps model outputs to `Tool` enum.
- **Error Handling**:
    - Comprehensive exception handling.
    - Detailed error reporting.
- **Device Flexibility**:
    - Auto-selects GPU if available, falls back to CPU otherwise.

3.2.7 Rail & Gripper Control (Arduino)

Design Overview

The Arduino must "home" the stepper motors first, before it can receive commands from the raspberry pi. Once the system is configured, it receives input of D1, D2, D3, D4, R1, R2, R3, or R4 in order to dispense "D" or return "R" the correct tool into or from the appropriate location.

Diagram

https://github.com/cjohst/orobot/blob/main/assets/documentation/mechatronics/MechanicalSequence.png

Implementation Details

- `Void setup()`: "homes" the gripper across the X and Y-axis.
- `Void loop()`: Takes input, and controls the motor strategy.
- `Void moveToPosition()`: Takes the current position and desired location, facilitating movement of the motor to achieve the desired location.
- `Void homeWithLimitSwitch()`: Configures the motor until the limit switch is triggered and then saves its location as '0'.
- `Void MoveStepper()`: Moves the stepper motor by a certain amount of steps.

Functional Requirements Addressed

- **FR1 (Perform Core Actions)**: Controls the mechanical strategy to dispense and retrieve the tools.

Constraints

- **C3 (Response Time)**: Motors were tuned in order to satisfy the time constraint for the automated workflow.

Design Decisions

- **Homing with Limit Switches**
    - Limit switches are required to home the stepper motors and establish a known reference position.
- **Exclusion of Encoders**
    - Encoders were omitted due to time and budget constraints.
- **Microcontroller Selection: Arduino**
    - Arduino was chosen for its ready availability and cost-effectiveness.
    - Prior experience with the microcontroller streamlined development.

3.2.8 Rail & Gripper Control (Circuit)

Design Overview

Power supply is connected to the wall which gives DC current to the motor drivers. The drivers are connected to a single motor, and are controlled by the Arduino, which is also connected to the two limit switches and a servo motor. Each axis (X and Y) has one limit switch. The Arduino also powers and controls the servo motor, which extends the claw.

Diagram

https://github.com/cjohst/orobot/blob/main/assets/documentation/mechatronics/MotorControl.png

Implementation Details

The centralized power supply controls the motor drivers, which supplies the servos. Meanwhile, the Arduino controls the limit switches and regulates the motor drivers.

Functional Requirements Addressed

- **FR1 (Perform Core Actions)**: Controls the mechatronics for tool handling.

Constraints

- **C3 (Response Time)**: Tuned motor drivers to meet the response time constraint.

Design Decisions

- **Motor Selection: NEMA 23**
    - Chosen for its cost-effectiveness.

    - Readily available as off-the-shelf components.
- **Microcontroller: Arduino Uno**
    - Arduino was chosen for its ready availability and low cost.
    - Prior experience with the microcontroller streamlined development.
- **Claw Actuation: Servo Motor**
    - Used due to easy off-the-shelf availability.
- **Y-Axis Configuration**
    - Implemented two rails with synchronized motors.
    - Enhances stability during movement along the Y-direction.

3.2.9 Rail & Gripper Control (Mechanics)

Design Overview

The motors work together in order to control the position of the claw that is attached to an X-Y rail-guided system for tool dispensing and retrieving.

Rail Diagram
https://github.com/cjohst/orobot/blob/main/assets/documentation/mechatronics/Rails.png

Gripper Diagram
https://github.com/cjohst/orobot/blob/main/assets/documentation/mechatronics/Gripper.png

ORobot Assembly
https://github.com/cjohst/orobot/blob/main/assets/documentation/mechatronics/ORobotMechanical.png

Implementation Details

Two synchronized motors and rails to control the "Y" direction, and one motor and rail to control the "X" direction which is mounted on both Y rails. On top of the X rail slider, is a gripper which will be gripping the bins in which the tools will be placed in.

Functional Requirements Addressed

- **FR1 (Perform Core Actions)**: Mechanical rails use the gripper to retrieve and dispense the tools.

Constraints

- **C3 (Response Time)**: Tuned motor drivers to achieve the <10-second response requirement.

Design Decisions

- **Custom 3D Printed Components**
    - Designed and printed multiple custom parts.
    - Used for mounting the rails together and attaching the claw to the rail slider.
- **Off-the-Shelf Mechanical Components**

        - Rails and motors were sourced from Amazon as off-the-shelf parts. The claw was also readily purchased from Amazon.
- **Motor Driver Selection: TB6600**
    - Chosen to ensure compatibility with NEMA 23 stepper motors.

4. Verification and Validation

4.1 Introduction

The following table outlines the planned verification and validation tests for the voice-activated surgical tool management system, ORobot. Each test corresponds to specific functional or non-functional requirements, as well as constraints identified earlier in the design process. The table summarizes the intended purpose of each test, the primary question it seeks to answer regarding system performance or usability, and the requirements it addresses. These tests are designed to ensure that the system not only meets its technical specifications but also aligns with the operational needs of surgical environments (*Table #1*).

Table #1: *Verification & Validation Tests with their Corresponding Requirements.*

Test	Purpose	Primary Question	Requirements
Tool Retrieval Timing	To assess whether the system allows prompt tool retrieval within 10 seconds by measuring time from voice command to dispensing, averaged over 100 trials.	Does the system permit retrieval of tools within the required time limit?	FR1 (Perform Core Actions), C3 (Response Time)
Tool Return Classification	To evaluate system accuracy in identifying returned tools under varied lighting, temperature, humidity and orientation, targeting ≥95% correct recognition.	Does the system accurately recognize and log returned tools under real-world operating environments?	FR1 (Perform Core Actions), C4 (Tool Classification)
Tool Tracking Accuracy	To verify consistency between system-tracked and manual inventory counts across multiple retrieval-return cycles, ensuring 0% error rate across procedures.	Does the system track tools without discrepancies compared to standard, manual methods?	FR2 (Tool Tracking and Display)
Weight Limit Compliance	To confirm whether the system can handle tools weighing less than	Does the system permit tracking of typical, lightweight	C1 (Tool Weight Limit)

	or equal to 30 g.	tools?	
Tool Compatibility Assessment	To test whether a full range of up to 100 diverse tools with varying lengths, widths, and shapes (≤30g) can be housed and supported by the system.	Is the system equipped to support a variety of instruments and function efficiently at maximum capacity?	C2 (Supported Tools)
Voice Command Performance	To assess voice recognition reliability in simulated noisy environments (60–75 dB) by tracking success rate across 50 commands issued by different users.	Can the system correctly respond to voice commands under real-time operating room conditions?	FR1 (Perform Core Actions), NFR1 (Efficiency and Speed)
Power Management	To measure ORobot operation time under standard load matches the longest expected surgery (≥6 hours) and to assess whether alert systems are present.	Can the robot operate for the required duration for complicated surgeries and alert staff before shutdown?	NFR1 (Efficiency and Speed)
Manual Override Efficiency	To assess for timely manual tool retrieval within 30 seconds of system failure.	Can tools still be accessed reliably and safely when automation fails?	FR3 (Manual Tool Access)
Spatial Footprint Compliance	To confirm the device fits within the standard 1m x 0.5m configuration and does not obstruct staff movement or layout requirements.	Is the device's size suitable for installation within the operating room?	NFR2 (Space and Integration)
Workflow Integration	To log the total number of disruptions, delays, or interruptions that occur with the introduction of the ORobot during a standard, two to three hour-long operation.	Does the system integrate effectively in surgical suites without interfering with established workflows for tool handling?	NFR1 (Efficiency and Speed), NFR2 (Space and Integration)
Interface Usability Assessment	To administer a System Usability Scale (SUS) questionnaire post-interaction and analyze results for	Is the user interface intuitive for operability by the surgical team?	FR2 (Tool Tracking and Display)

	ease-of-use and user clarity.		
Patient Safety Assurance	To conduct an external safety review and audit sterility compliance using ATP swabs with material assessments against hospital norms.	Does the system meet clinical safety and sterility regulations?	NFR3 (Sterilizability & Patient Safety)

4.2 Verification Tests

The verification phase will focus on determining whether the system meets the specified functional and non-functional requirements, based on controlled engineering tests. Each test will evaluate a measurable performance metric using predefined protocols, ensuring traceability to system requirements and engineering standards. These assessments are designed to appraise critical aspects of the system, such as tool retrieval speed, tracking accuracy, voice command performance, and compliance with operational requirements like power management and spatial fit. Due to constraints imposed by time and budget, these tests were not conducted on the minimum viable prototype, which is simply a proof of concept. Instead, this section serves as a guideline for robust performance evaluation and iterative refinement of the ORobot prior to pilot testing in clinical settings.

i. Tool Retrieval Timing

To assess the tool retrieval performance of the ORobot, this test will measure the time taken for the system to retrieve and deliver surgical tools upon command. The goal of this evaluation is to ensure that ORobot meets the time-sensitive demands of a surgical procedure, with a maximum retrieval time of 10 seconds. This time limit has been imposed to align with the typical speed required for tool retrieval in fast-paced environments, where delays could compromise patient safety (8).

The retrieval time will be averaged for consistency across 100 multiple trials under controlled conditions, simulating real surgical procedures. This test is essential to ensure that the ORobot can maintain high operational efficiency while minimizing any disruption to the surgical workflow (9). It directly addresses both **FR1 (Perform Core Actions)** and **C3 (Response Time)** requirements, confirming that the robot can execute its core functions within a realistic timeframe, even under varying conditions.

ii. Tool Return Classification

To ensure the reliability of tool management, this test will verify that ORobot can accurately classify and log tools returned after use. The system's ability to correctly identify and register the returned tools under various environmental conditions, such as fluctuating lighting, temperature, and humidity, will be assessed (10). A minimum accuracy threshold of 95% has been set for this test, based on industry standards for tool tracking in healthcare.

This rate makes certain that the system maintains a high level of precision in tool recognition, minimizing the risk of tool mismanagement, which is crucial for safety and operational efficiency. During the test, multiple tools will be returned over several trials, and the results will be compared against manual logging to check for discrepancies. This test is directly linked to **FR1 (Perform Core Actions)** and **C4 (Tool Classification)**, to ensure that ORobot can function effectively in an environment where precision and tracking are vital for maintaining the sterile field and proper tool inventory.

iii. Tool Tracking Accuracy

To assess ORobot's tool tracking capabilities, this test will evaluate how accurately the system tracks tools during multiple retrieval-return cycles. The goal is to achieve an accuracy rate of 100%, with no discrepancy between the system's recorded inventory and the actual tools in use. This accuracy threshold has been chosen to reflect the high standards expected in medical environments where every tool must be accounted for to avoid errors or loss of equipment which can result in retained surgical items (RSIs) (11).

This test will involve simulating a series of tool retrievals and returns, with each instrument being tracked by the system in real time. The results will be validated by comparing the system's log with manual counts performed by the surgical staff. This test is essential to make certain that ORobot can reliably track tools and maintain an accurate inventory. This meets operational and safety requirement **FR2 (Tool Tracking and Display)**.

iv. Weight Limit Compliance

To ensure the ORobot is capable of handling a broad range of surgical tools, a weight limit compliance test will be conducted to evaluate its ability to manage tools up to a maximum weight of 30 grams (12). This threshold was chosen based on the typical weight range of surgical instruments used in many common procedures, ensuring that the system is versatile enough to accommodate a wide variety of tools without strain or malfunction.

The test will involve placing tools of varying sizes and weights up to the 30-gram limit into the robot's tool retrieval system to confirm that it can reliably carry and deliver tools without mechanical failure or performance degradation. This test is important because it ensures the ORobot can perform its task efficiently, without the risk of overload or damage to the system. Thus, this supports the requirement, **C1 (Tool Weight Limit)** and ensures that the device is versatile and dynamic enough for real-world surgical applications.

v. Tool Compatibility Assessment

To evaluate the ORobot's ability to support a diverse array of surgical instruments, a compatibility test will be performed. This assessment will verify that the system can handle up to 100 different tools of various sizes and shapes, while maintaining performance and functionality. This test is designed to address the flexibility of the system in working with different types of surgical tools, ranging from small, delicate instruments to larger, more robust ones, all within the 30-gram weight limit.

The 100-tool requirement reflects the typical variety of instruments used during surgical procedures, which could involve implements with various ergonomics and handling requirements (13). The test will involve running the system through a series of retrieval and return tasks using the various tools, ensuring that it operates seamlessly with a 0% error rate in dispensing and retrieving. This test addresses **C2 (Supported Tools)** and makes certain that ORobot's design is versatile enough to handle the diverse range of instruments found in typical operating room settings.

vi. Voice Command Performance

To ensure that ORobot can reliably respond to voice commands in a noisy operating room environment, this performance test will be conducted. The verification method will evaluate the system's ability to recognize and respond to spoken commands, with background noise ranging from 60 to 75 dB, which is typical in a surgical environment (14).

A total of 50 commands will be issued by different users, and the system's accuracy in recognizing and executing these commands will be measured. A success rate of 95% in command recognition will be considered acceptable for this test, ensuring that ORobot can function in real-world settings where verbal communication is essential. This test is essential for verifying **FR1 (Perform Core Actions)** and **NFR1 (Efficiency and Speed)**, confirming that the system can reliably operate under dynamic and noisy conditions.

vii. Power Management

To evaluate ORobot's power management system, a test will be conducted to ensure that the device can operate continuously for at least 6 hours, which is the typical duration of complex surgeries (15). This test will also verify that the system has a reliable mechanism incorporated within its graphical interface for alerting the surgical team when the power level is low, hence allowing for timely recharging or maintenance without disrupting the procedure.

The 6-hour duration is based on the typical length of most high-risk surgical procedures, ensuring that ORobot's power management system supports extended usage without failure. Additionally, the alert system will be tested for effectiveness, ensuring that it provides adequate notice before power depletion occurs. This test is important to make certain that ORobot can operate without interruption, addressing **NFR1 (Efficiency and Speed)** while providing continuous functionality during critical operations.

viii. Manual Override Efficiency

To assess the ORobot's fail-safe mechanisms, this test will be performed to evaluate how quickly surgical staff can manually retrieve tools in the event of a system failure. The test will simulate a scenario where the system malfunctions, and staff will need to manually access the tools within 30 seconds. This timeframe has been selected to ensure that manual retrieval can occur quickly enough to minimize disruption to the surgical procedure (16).

This test will involve timing how long it takes for the staff to access tools manually, and any delays will be carefully evaluated to ensure that ORobot's manual override system is as efficient as possible. This test is vital for ensuring that in case of a system failure, the surgical team can continue their work without compromising the procedure's timing or patient safety. It directly addresses **FR3 (Manual Tool Access)**, confirming that ORobot includes an effective and efficient manual override feature in case of system errors.

ix. Spatial Footprint Compliance

To make sure that ORobot can be seamlessly integrated into a standard operating room environment, a compliance test will be conducted to verify that the system's size does not interfere with staff movement or obstruct other critical equipment. The test will confirm that ORobot fits within the operating room space allocation of 1m x 0.5m, a size that ensures it does not encroach on essential workspaces (17).

The compact size is crucial to minimize procedural workflow disturbances, particularly in high-density operating rooms where space is limited. This test will evaluate the ORobot's ability to function effectively in these constrained environments, ensuring that it does not obstruct the overall workflow or pose a safety hazard due to its size. This test supports **NFR2 (Space and Integration)**, ensuring that ORobot's design is suitable for clinical implementation.

4.3 Validation Tests

This section outlines the planned validation tests for the ORobot, which are critical to ensure that the device meets the needs of its users, operates reliably within its intended environment, and integrates effectively into existing workflows. As the formal validation process has not yet been conducted with stakeholders, this section describes the anticipated methods and outcomes for each test, which will be executed upon the completion of prototype development and prior to the device's commercial release.

i. Workflow Integration Assessment

To determine the seamless nature of the ORobot's introduction into the operating room environment, its workflow integration will be qualitatively assessed. This test will simulate a real surgical procedure within a controlled setting, where the ORobot will be tasked with performing tool retrieval and return functions as needed while the surgical team proceeds with their tasks (18). The test will assess the degree to which the system's movements interfere with the surgical staff's actions, as well as whether it impedes access to tools or equipment, and disrupts the general flow of the procedure (NFR1, NFR2).

In particular, the integration test will focus on space management and operational efficiency, evaluating whether the robot's compact and ergonomic design allows it to exist in the operating room without encroaching on critical workspace infrastructure or causing delays. Additionally, the test will verify whether the ORobot can operate in close proximity to the surgical team without compromising the safety and efficiency of the procedure.

ii. Interface Usability Review

To assess the effectiveness of the ORobot's interactive user interface, a usability review will be conducted. Surgical staff will engage with the interface during the mock procedure and provide feedback through a standardized questionnaire based on the System Usability Scale (SUS) (19). This review will focus on whether the interface allows for quick and intuitive interaction, such as issuing commands to retrieve tools, monitoring tool status, and ensuring that all necessary information is clearly displayed and easily accessible (FR2).

Key considerations will include whether the interface is intuitive for staff with varying levels of experience and whether it allows for quick decision-making during time-sensitive procedures without compromising patient outcomes. The results will help determine whether any adjustments or improvements are needed to enhance the interface's effectiveness, ensuring that it contributes positively to the efficiency of the surgical team.

iii. Patient Safety Assurance

To ensure that the ORobot adheres to all relevant safety standards for use in the operating room, particularly regarding sterility and hazard compliance (NFR3), an external safety audit will be conducted. The test will evaluate the robot's design to make certain that it does not introduce any contamination risks or compromise the sterile field during operation (20). This involves assessing the materials used for ORobot's construction, its ability to be properly sterilized, and its design features to prevent the accumulation of contaminants.

The safety audit will also examine whether the ORobot poses any physical hazards to the surgical team or patient, such as sharp edges, malfunctioning parts, or exposed electrical parts. Additionally, the assessment will evaluate whether the robot's integration into the operating room complies with relevant regulatory standards for medical devices, ensuring that it is safe to use in high-risk environments.

5. IP Assessment

5.1 Related IP

i. Voice-Controlled Tool Dispensing

We aim to patent ORobot's advanced voice-controlled system, optimized for dynamic surgical environments. Unlike standard voice recognition, ORobot can reliably interpret commands,while accounting for background noise, multiple speakers, and diverse accents.

ii. AI-Powered Tool Recognition and Arbitrary Mapping

ORobot's AI algorithms track tools even when stored arbitrarily, thus overcoming limitations of predefined placements. Our system adapts to real-world conditions, such as poor lighting, occlusions, and variable tool positions, making it a unique, patentable asset.

iii. Rail-Guided Dual-Tool Retrieval Mechanism

We aim to protect ORobot's rail-guided gripper system, which enables simultaneous retrieval and dispensing of instruments. The precision and modularity of the rail-guided system allows ease of integration while maintaining sterility, and thus will be protected.

iv. Real-Time Tool Tracking and Error Detection

Unlike traditional systems that use manual scanning or RFID tags, ORobot combines AI-based image recognition and real-time reconciliation for accurate tool tracking. These proprietary algorithms will be filed as trade secrets to preserve our competitive advantage.

v. System Integration and Workflow Innovation

ORobot's most significant innovation lies in its assimilation of AI-driven voice and image recognition, dual-tool robotic retrieval, and real-time error detection. We aim to patent the system's end-to-end workflow and unique integration for reliable tool management.

5.2 Freedom to Operate

i. Voice-Controlled Systems

Patent: *US20220249178A1* – Voice-Activated Surgical Equipment Control

This patent covers use of voice commands to operate medical equipment. Although ORobot shares similarities in voice control, we can avoid overlap while securing protection by emphasizing our superior noise resilience, reduced latency, and hands-free efficiency.

ii. AI-Powered Tool Recognition and Mapping

Patent: *US20210327567A1* – Machine-Learning Systems for Surgical Tool Recognition

This patent describes use of machine learning for identifying surgical instruments under predefined arrangements. ORobot can recognize tools placed arbitrarily, highlighting its adaptability and robustness under real-world conditions (e.g., poor lighting, occlusions).

iii. Robotic Retrieval Mechanisms

Patent: *US11886772B2* – Robotic Systems for Voice-Controlled Tool Retrieval

This patent involves robotic systems with voice commands for single-tool management. ORobot distinguishes itself with a dual-tool rail-guided mechanism, which enables simultaneous handling of two instruments. To further differentiate ORobot, we would emphasize its modular design, precision-guided mechatronics, and AI-driven recognition.

iv. Tool Tracking and Error Detection

Patent: *RFID-Based Tool Tracking Systems* (Various Existing Patents)

Many existing patents focus on RFID-based tool tracking workflows. ORobot eliminates reliance on RFID tags by combining AI-based image recognition with multi-sensor reconciliation algorithms, establishing a clear differentiation from existing operating systems.

v. System Integration and Workflow Innovation

Patent: *Existing Patents Covering Individual Components (Voice, AI, Robotics)*

While patents exist for voice recognition, AI-based image classification, and robotic retrieval individually, ORobot’s core innovation lies in the integration of these technologies. Protecting this cohesive system as a utility patent will help ORobot develop a competitive IP.

5.3 Patentability

Our IP strategy focuses on securing protection for ORobot’s unique innovations that integrate voice recognition, AI adaptability, robotics, and real-time error detection, while managing risks posed by existing patents. To achieve this, we will pursue utility patents for the system’s core features: (a) voice-controlled system for noisy surgical environments, (b) AI-powered tool recognition for arbitrary placements, and (c) dual-tool rail-mechanism.

Proprietary algorithms for multi-sensor reconciliation and tool counting will be safeguarded as trade secrets through confidentiality agreements and internal security measures. To protect the aesthetic and functional design of our components, we will file design patents, covering elements like user interface and the rail-guided retrieval system.

Given the competitive IP landscape, we will conduct a thorough Freedom to Operate (FTO) analysis to identify existing patents that may pose risks, enabling us to design around potential overlaps and ensure commercial viability. Simultaneously, we will register the "ORobot" name and logo as trademarks to protect our brand identity and market presence.

To secure early priority dates while refining our technology, we will file provisional patent applications for key innovations. This comprehensive strategy, combining patent protection, trade secret management, and trademark registration, will establish ORobot as a leader in surgical automation while creating significant barriers to entry for competitors.

The financial model reflects this strategy through an early allocation of $18,000 CAD under Legal and IP expenses in the first two fiscal years. These funds cover provisional and utility patent filings, trademark registration, and a Freedom to Operate (FTO) analysis, aligning with the timeline for prototyping and pilot deployment. This early investment secures key filing dates and supports valuation growth ahead of commercialization.

Hence, ORobot's IP strategy enhances our valuation by establishing a defensible competitive moat early in the company's lifecycle. Patents, trade secrets, and trademarks create tangible assets that reduce replication risk, signalling long-term value and market growth to potential investors. This strengthens ORobot's position in funding negotiations and increases its attractiveness for future acquisition or licensing deals.

6. Archive Contents

Directory Details https://github.com/cjohst/orobot

`/assets`

Contains design files and documentation:

- `assets/cad/`: 3D CAD models for the robot hardware components, including mounting brackets, Arduino housing, and tool holders
- `assets/documentation/`: Additional documentation including class diagrams, and other figures embedded in this report

`/data`

Stores datasets used for machine learning:

- `data/captured-dataset/`: Images captured using the camera service
- `data/surgical-dataset/`: Dataset of surgical tools for classification training

`/src`

Main source code directory where all the application code resides:

Core Files:

- `config.py`: Configuration settings for the entire system
- `control_service.py`: Main orchestrator of the O-Robot system
- `main.py`: Entry point for the application
- `run_server.py`: Script to run the WhisperLive transcription server

Subdirectories:

- `arduino/`: Code for the Arduino microcontroller
- `libs/`: External libraries including WhisperLive for voice transcription
- `ml/`: Machine learning code for tool classification
- `models/`: Data models and business logic
- `services/`: Core services including voice, camera, and serial communication
- `web/`: Web application for controlling the robot

Key Components

- **Voice Transcription:** Located in `src/services/voice_service.py` utilizing the WhisperLive library in `src/libs/WhisperLive/`
- **Tool Classification:** Implemented in `src/ml/tool_classifier.py` using a ResNet-50 model trained on surgical tool images
- **Serial Communication:** Found in `src/services/serial_service.py` for communication with the physical robot
- **Web Interface:** Located in `src/web/` for a browser-based control panel via Flask
- **Control Service:** The main orchestrator in `src/control_service.py` that integrates all components

7. References

(1). Agnoletti, V., Buccioli, M., Padovani, E., Corso, R. M., Perger, P., Piraccini, E., Orelli, R. L., Maitan, S., Dell'amore, D., Garcea, D., Vicini, C., Montella, T. M., & Gambale, G. (2013). *Operating room data management: improving efficiency and safety in a surgical block.* BMC Surgery, 13, 7. https://doi.org/10.1186/1471-2482-13-7

(2). Addressing Health care's talent emergency. (2022, November 15). Deloitte Insights. https://www2.deloitte.com/us/en/insights/industry/health-care/healthcare-workforce-shortage-solutions.html

(3). E. Collins, S. P. Gilbert, and P. J. Griffin, *"Technological advances in perioperative care,"* PubMed Central, 2021. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10613559/

(4). Crowe, C. (2023, November 16). *Identifying Stakeholders in the Healthcare Sector. Boréalis*. https://www.boreal-is.com/blog/stakeholders-in-healthcare/

(5). Katz, J. D. (2017). *Control of the Environment in the Operating Room*. Anesthesia & Analgesia, 125(4), 1214–1218. https://doi.org/10.1213/ANE.0000000000001626

(6). CDC. (2003). *Guidelines for Environmental Infection Control in Health-Care Facilities Recommendations of CDC and the Healthcare Infection Control Practices Advisory Committee (HICPAC)*. https://www.cdc.gov/infection-control/media/pdfs/Guideline-Environmental-H.pdf

(7). Labeled Surgical Tools and Images. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/dilavado/labeled-surgical-tools

(8). Schouten, A. M., Flipse, S. M., van Nieuwenhuizen, K. E., Jansen, F. W., van der Eijk, A. C., & van den Dobbelsteen, J. J. (2023). *Operating Room Performance Optimization Metrics: a Systematic Review*. Journal of Medical Systems, 47(1). https://doi.org/10.1007/s10916-023-01912-9

(9). Cardoso, S., Jenisha Suyambu, Iqbal, J., Jaimes, D., Amin, A., Jarin Tasnim Sikto, Valderrama, M. J., Simranjit Singh Aulakh, Venkata Ramana, Shaukat, B., & Patel, T. (2023). *Exploring the Role of Simulation Training in Improving Surgical Skills Among Residents: A Narrative Review.* Cureus, 15(9). https://doi.org/10.7759/cureus.44654

(10). Smith, D. (2024, August 28). *Perioperative Environment: Why Temperature and Humidity Matter.* Courtemanche & Associates. https://www.courtemanche-assocs.com/blogs/temperature-and-humidity-variations-in-the-perioperative-environment

(11). K. D. Ward, T. M. Boltz, and D. J. Cook, "*Retained surgical items: Current perspectives,"* Patient Safety in Surgery, vol. 15, no. 5, pp. 1–9, 2021. [Online]. Available: https://pssjournal.biomedcentral.com/articles/10.1186/s13037-021-00297-3

(12). International Committee of the Red Cross, "*Medical items: Weight and volume catalogue*," 2021. [Online]. Available: https://itemscatalogue.redcross.int/medical-weight-volume.aspx

(13). Warwick, V. R., Gillespie, B. M., McMurray, A., & Clark-Burg, K. G. (2021). *Undertaking the surgical count: An observational study.* Journal of Perioperative Nursing, 34(3). https://doi.org/10.26550/2209-1092.1089

(14). Mcleod, R., Myint-Wilks, L., Davies, S., & Elhassan, H. (2021). *The impact of noise in the operating theatre: a review of the evidence*. The Annals of the Royal College of Surgeons of England, 103(2), 83–87. https://doi.org/10.1308/rcsann.2020.7001

(15). List of Average Surgery Operation & Recovery Times. (2020, April 19). Weiss & Paarz. https://www.weisspaarz.com/surgery-recovery-times/

(16). Instrument Malfunctions & Risks - Surgical Solutions. (2024, December 6). Surgical Solutions. https://www.surgical-solutions.com/problems/instrument-malfunctions-risks/

(17). Tyler, D. C., Pasquariello, C. A., & Chen, C.-H. (2003). *Determining Optimum Operating Room Utilization*. Anesthesia & Analgesia, 1114–1121. https://doi.org/10.1213/01.ane.0000050561.41552.a6

(18). Pasquer, A., Ducarroz, S., Lifante, J. C., Skinner, S., Poncet, G., & Duclos, A. (2024). *Operating Room Organization and Surgical performance: a Systematic Review*. Patient Safety in Surgery, 18(1). https://doi.org/10.1186/s13037-023-00388-3

(19). Sauro, J. (2011, February 3). MeasuringU: Measuring Usability with the System Usability Scale (SUS). Measuringu.com. https://measuringu.com/sus/

(20). Guidelines for Best Practices for Establishing the Sterile Field in the Operating Room. (2008). https://www.ast.org/uploadedFiles/Main_Site/Content/About_Us/Guidelines%20Establishing%20the%20Sterile%20Field.pdf

8. Appendices

Appendix #1: Marketing Video

Marketing Video

Appendix #2: Demo Video

ORobot Demo

Appendix #3: Total Expenditure

Table #2: *Total Expenses Incurred for Prototyping of the ORobot.*

Expenses				
Date	**Item**	**Category**	**Amount (CAD)**	**Purchased By**
Dec 15	Rasberry PI, SD Card, Power Supply, Cooling Fan	Software	189.00	Connor
Dec 22	Goose Neck Microphone	Software	22.39	Connor
Jan 23	Motor Driver Board x3 RATTMMOTOR x1 RATTMMOTOR x1	Hardware	489.41	Isaac
Jan 23	Mechanical Claw Robot Gripper Aluminium	Hardware	35.41	Isaac
Jan 30	RATTMMOTOR x1	Hardware	140.00	Isaac
Feb 09	Scalpel Graduated Handle Straight Mayo Scissors Straight Blunt Tweezers (Clamp) Curved Mayo Scissors	Hardware	60.68	Aly
Feb 13	USB Camera Module	Software	16.65	Connor
Feb 13	Bi-Directional Shifter Module	Hardware	12.19	Connor
Feb 13	Multi-colored breadboard ribbon cable	Hardware	14.55	Connor
Mar 04	Extrusion	Hardware	114.28	Isaac
Mar 13	E stop	Hardware	19.03	Isaac
Mar 13	Mechanical Claw Robot Gripper Aluminium	Hardware	76.47	Isaac
Mar 18	High Torque Servo Motors	Hardware	32.99	Isaac
Mar 18	Micro Limit Switch, Extension Cable	Hardware	38.05	Isaac
			Budget	2,000.00
			Total Expenses	1,261.10
			Balance	738.90